**PROPOSAL**

## RANCANG BANGUN APLIKASI KANOPI OTOMATIS BERBASIS *ANDROID* MENGGUNAKAN LOGIKA *FUZZY* TSUKAMOTO

**Diajukan untuk memenuhi salah satu syarat memperoleh derajat Sarjana Teknik**



**ELSA MEILAN**

**E1E1 15 014**

## JURUSAN TEKNIK INFORMATIKA

## FAKULTAS TEKNIK

## UNIVERSITAS HALU OLEO

## KENDARI

**2020**

**BAB I**

**PENDAHULUAN**

### 1.1 Latar Belakang

Kanopi adalah tirai atau langit- langit dari terpal, kain, logam, dan sebagainya pada teras terdapat bertiang sebagai pemisah halaman dengan bagian dalam rumah. Kanopi merupakan sejenis atap untuk melindungi bagian luar rumah dari panas matahari dan terpaan hujan. Kanopi dalam perkembangannya kini marak digunakan pada bagian garasi mobil dan balkon rumah, kantor dan bangunan lainnya. Kanopi juga merupakan suatu media penghalang cuaca demi melindungi barang-barang berharga yang berada di luar ruangan atau di halaman rumah. Kanopi juga dapat memberikan nuansa yang teduh dan tidak terasa panas yang juga berimbas pada ruangan lainnya baik didalam rumah, kantor, rumah sakit, cafe dan sebagainya (Kania, 2019).

Panasnya sinar matahari dalam pengaplikasian kanopi tentunya membuat radiasi sinar matahari tidak akan dapat langsung menyentuh barang-barang seperti halnya pada barang-barang yang dicat, seperti mobil, motor, meja, dan kursi , dan lain-lain, Semua cat akan aman dan tidak mudah rusak bila tidak melalui kontak langsung dengan guyuran air hujan ataupun sinar matahari yang akan membuat cat melepuh secara perlahan. Kanopi juga berfungsi melindungi tanaman yang berada di halaman rumah terhindar dari sinar matahari dan guyuran hujan yang berlebih sehingga tanaman pun tetap aman dan tetap tumbuh subur, atau jika jemuran berada di halaman rumah juga tetap terjaga. Fungsi kanopi disamping melindungi juga silau dan panas yang masuk ke halaman rumah sudah dikurangi dan difilter oleh kanopi (Kania, 2019).

Pesatnya perkembangan teknologi akhir-akhir ini, memicu berkembangnya teknologi baru yang memanfaatkan sistem kontrol sebagai media, untuk mewujudkan impian manusia akan sebuah aplikasi pengontrolan peralatan daritempat lain yang tanpa harus berada di tempat tersebut. Pemanfaatan dibidang teknologi tersebut beragam, salah satunya adalah alat pengontrol otomatis.

Otomatis adalah proses untuk mengontrol operasi dari suatu alat secara otomatis yang dapat mengganti peran manusia untuk mengamati dan mengambil keputusan. Sistem kontrol yang ada saat ini mulai bergeser pada otomatis sistem kontrol, sehingga campur tangan manusia dalam pengontrolan sangat kecil. Sistem peralatan yang dikendalikan secara otomatis sangat memudahkan apabila dibandingkan dengan sistem manual, karena lebih efisien, aman, dan teliti. Mikrokontroler diciptakan untuk mengontrol suatu sistem yang dibuat untuk mempermudah proses pengontrolan atau pengendalian suatu alat yang dibuat. Mikrokontroler adalah sebuah *chip* yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program di dalamnya. Mikrokontroler umumnya terdiri dari *Central Processing Unit* (CPU), memori, I/O tertentu dan unit pendukung seperti *Analog-to-Digital Converter* (ADC) yang sudah terintegrasi di dalamnya (Ihsanto, 2016).

Logika *fuzzy* digunakan sebagai dasar pemikiran untuk menentukan intensitas suhu dan intensitas cahaya yang sebagai *input* pada aplikasi *Accsys* dan *outputnya* adalahkanopi yang akan mengeluarkan apakah kanopi terbuka ataupun menutup. Logika *fuzzy* yang digunakan yaitu metode tsukamoto merupakan perluasan dari penalaran yang monoton. Pada metode tsukamoto, Setiap konsekuen pada aturan yang berbentuk *IF-THEN* harus dipresentasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, *output* hasil inferensi dari tiap-tiap aturan diberikan secara tegas *(crisp)* berdasarkan α-predikat (*fire strength*). sehingga lebih memudahkan saat melakukan proses perhitungan.

Berdasarkan latar belakang yang telah diuraikan tersebut, maka dibuat penelitian dengan judul **“*Rancang Bangun Aplikasi Kanopi Otomatis Berbasis Android Menggunakan Logika Fuzzy Tsukamoto*”.** Alat kanopi otomatis dibuat untuk melindungi kendaraan yang berada di halaman rumah ketika kondisi cuaca yang buruk. Kanopi otomatis ini menggunakan jaringan *Internet of Things* (IoT) sehingga bisa diakses dari manapun dan kapanpun pada saat memonitoring maupun mengontrol kanopi tersebut melalui aplikasi *Android*.

### 1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana cara merancang kanopi otomatis berbasis *android* menggunakan logika fuzzy tsukamoto*?*
2. Bagaimana cara kerja alat kanopi otomatis berbasis *android*?

### 1.3 Batasan Masalah

Yang menjadi batasan masalah dalam penulisan tugas akhir ini adalah :

1. Kanopi di kendalikan oleh sensor cahaya, sensor suhu, sensor hujan dan *android*.
2. Pengguna harus terhubung internet untuk mengendalikan sistem.
3. Komunikasi antara alat dan monitoring menggunakan modul *Wifi ESP8266* yang tertanam di *NodeMCU*.

### 1.4 Tujuan Penelitian

Tujuan dari penelitian ini sebagai berikut:

1. Merancang dan membangun sistem kanopi otomatis berbasis *android* dengan menggunakan intensitas cahaya, suhu, dan hujan sebagai parameter sistem.
2. Dapat menerapkan logika *fuzzy* pada sistem kanopi otomatis berbasis *android.*
3. Mengetahui hasil kerja dari alat kanopi secara otomatis berbasis *android* menggunakan logika *fuzzy*.

### 1.5 Manfaat Penelitian

Manfaat dari pembuatan proyek akhir ini antara lain :

1. Mendapat pemahaman bagaimana cara memanfaatkan teknologi internet untuk mengontrol kanopi.
2. Memberi kemudahan kepada masyarakat yang menggunakan atap kanopi otomatis yang dapat melindungi kendaraannya dari kondisi cuaca buruk.

### 1.6 Sistematika Penulisan

Sistematika penulisan tugas akhir ini terdiri atas beberapa bagian utama sebagai berikut:

### BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan dan tinjauan pustaka.

### BAB II LANDASAN TEORI

Bab ini berisi teori-teori yang menjadi landasan dan mendasari penelitian yang mendukung penelitian ini yaitu *NodeMCU ESP8266, Motor Stepper, Jumper Wire,* Sensor *LDR,* Sensor Suhu *DHT11*, Sensor Hujan, *Internet Of Things* (IoT),

*Android, Android Studio, Pemrograman Arduino,* Bahasa Pemrograman C, *Software Arduino IDE,* Logika *Fuzzy* Tsukamoto.

### BAB III METODOLOGI PENELITIAN

Bab ini membahas mengenai metode pengumpulan data, metode

pengembangan sistem, pengujian sistem, waktu dan tempat penelitian,

### BAB IV ANALISIS DAN PERANCANGAN SISTEM

Bab ini Bab ini berisi tentang analisis sistem, analisis kebutuhan sistem, analisis kebutuhan perangkat keras, analisis kebutuhan perangkat lunak, proses perancangan sistem, perancangan mekanik, perancangan elektronik, perancangan *software,* perancangan antarmuka (*Interface*), *rule* *fuzzy* tsukamoto, hitungan manual *fuzzy* tsukamoto.

### BAB V IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini membahas mengenai implementasi dan pengujian sitem terhadap sistem yang telah dibuat.

### BAB VI PENUTUP

Bab ini berisi kesimpulan dan saran dari hasil pelaksanaan tugas akhir.

#### 1.7 Tinjauan Pustaka

Penelitian ini dibuat berdasarkan penelitian yang pernah dilakukan antara lain penelitian yang dilakukan oleh Eko Ihsanto (2015) dengan judul penelitian “*Rancang Bangun Kendali Gordeng Dengan Saklar Lampu Otomatis Berbasis Smartphone Android”***.** Kesimpulan dari penelitian ini adalah sistem ini dapat meringankan aktivitas penggunanya agar lebih efektif dan efisien.

Muhammad Dawud (2016) dengan judul penelitian “*Sistem Monitoring*

*Lampu Penerangan Jalan Umum Menggunakan Mikrokontroler Arduino Dan Sensor LDR Dengan Notifikasi SMS”.* Kesimpulan dari penelitian ini adalah sistem dengan pemantauan Lampu Penerang Jalan. Selain harus dirancang *ON* dan *OFF* secara otomatis untuk menghemat energi listrik dan tenaga manusia, dapat menghemat waktu dalam memonitoring kondisi lampu penerang jalan umum. Aplikasi menggunakan SMS sebagai media penghubung dari lampu penerang jalan dengan koordinator lapangan, sehingga dapat menghemat pengeluaran biaya perbulan untuk monitoring lampu penerang jalan. dapat memberikan laporan pengawasan lampu, sehingga koordinator lapangan dapat memonitoring dan mengevaluasi kondisi lampu penerang jalan umum.

Suryanto (2017) dengan judul penelitian “*Atap Otomatis Tanaman Hidroponik Berbasis Mikrokontroler Atmega 89s52”.* Kesimpulan dari penelitian ini adalah alat ini memudahkan para petani hidroponik dalam mengatur kondisi cuaca dalam pengembang biakan suatu tanaman tanpa perlu khawatir tanaman akan rusak maupun bibit tanaman akan membusuk apabila terkena hujan secara berlebihan, karena atap akan menutup secara otomatis saat terjadi hujan serta cahaya lampu akan menyala jika atap tertutup sehingga dapat memberikan pencahayaan yang cukup dan para petani hidroponik dapat bekerja dengan baik dalam mengembang biakan bibit tanam di waktu hujan dan atap tertutup.

Araffi Alief Handaru (2015) dengan judul penelitian “*Rancang Bangun Alat Pendeteksi Hujan Otomatis Menggunakan Modul GSM Berbasis*

*Mikrokontroler Atmega 328p”.*Kesimpulan dari penelitian ini adalah perancangan dan pembuatan alat yang mampu mendeteksi terjadinya hujan serta cuaca cerah maupun mendung dikawasan lokasi pemasangan alat tersebut yang dilengkapi dengan sensor hujan dan sensor cahaya serta *arduino uno* sebagai pengendali.

La Ode Ichsan (2018) dengan judul penelitian *“Rancang Bangun Dan*

*Monitoring Alat Jemur Pakaian Berbasis Web Menggunakan Metode Naive Bayes”.* Kesimpulan dari penelitian ini adalah alat jemur otomatis , alat penjemur pakaian yang dapat bergerak ke arah luar dan ke arah dalam dalam miniatur rumah secara otomatis berdasarkan kondisi cuaca dan cahaya lingkungan sekitar. Prinsip kerja alat jemur otomatis yaitu menggunakan saklar yang dikendalikan oleh mikrokontroler yan menerima *input-an* dari tiga sensor. Nilai dari hasil perhitungan *Naive Bayes* akan diklasifikasikan menjadi dua yaitu tarik dan jemur.

**BAB II**

**LANDASAN TEORI**

### 2.1 *NodeMCU*

*NodeMCU* pada dasarnya adalah pengembangan dari *ESP8266* dengan *firmware* berbasis *e-Lua.* Pada *NodeMCU* dilengkapi dengan *micro usb port* berfungsi untuk pemograman maupun *power supply*. Selain itu juga pada *NodeMCU*  dilengkapi dengan tombol *push button* yaitu tombol *reset* dan *flash. NodeMCU*  menggunakan bahasa pemograman *Lua* merupakan *package* dari *ESP8266.* Bahasa *Lua* memiliki logika dan susunan pemograman yang sama dengan bahasa c hanya berbeda dengan syntax. Jika menggunakan bahasa *Lua* maka dapat menggunakan *tool Lua loader* maupun *Lua uploder.*

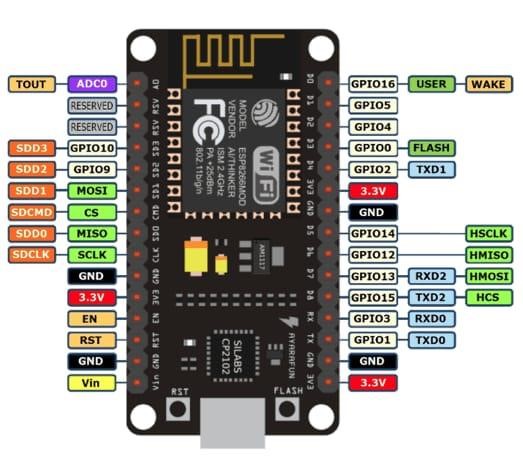
*NodeMCU*  merupakan sebuah *opensource platform* IoT dan *pengembangan* *Kit* yang menggunakan bahasa pemograman *Lua* untuk membantu *programmer* dalam membuat *prototype* produk IoT atau bisa memakai *sketch* dengan *Arduino* IDE. Pengembangan *Kit* ini didasarkan pada *Modul ESP8266*, yang mengintegrasikan GPIO, PWM (*Pulse Width Modulation*), 1-Wire dan ADC (*Analog to Digital Converter*) semua dalam satu *board*.

*NodeMCU* adalah *platform Internet Of Things (IoT) open source. NodeMCU*  mencakup *firmware* yang berjalan di *ESP8266* *Wi-fi SoC* dari Espressif Systems, dan perangkat keras yang berbasis pada modul ESP-12. Istilah *“NodeMCU”*  secara *default* mengacu pada *fimware* menggunakan bahasa scipting *Lua.* Hal ini didasarkan pada proyek *e-Lua,* dan dibangun diatas *Espressif Non-OS SDK*  untuk *ESP8266.*

7

# Tabel 2.1 Spesifikasi *NodeMCU ESP8266*

|  |  |
| --- | --- |
| Spesifikasi | *NODEMCU* |
| Mikrokontroller | *ESP8266* |
| Ukuran *board* | 57 mmx 30 mm |
| Tegangan Input | 3.3 – 5V |
| GPIO | 13 PIN |
| Kanal PWM | 10 Kanal |
| 10 bit ADC Pin | 1 Pin |
| *Flash Memory* | 4 MB |
| *Clock Speed* | 40/26/24 MHz |
| *Wifi* | IEEE 802.11 b/g/n |
| *Frekuensi* | 2.4 GHz-22.5 Ghz |
| *USB Port* | Micro USB |

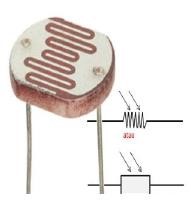


**Gambar 2.1 *NodeMCU*  (Wicaksono, 2017)**

### 2.2 *Light Dependent Resistor* (LDR)

*Light Dependent Resistor* atau yang biasa di sebut LDR adalah jenis resistor yang nilainya berubah seiring intensitas cahaya yang diterima oleh komponen tersebut. Biasa digunakan sebagai detektor cahaya atau pengukur besaran konversi cahaya. *Light Dependent Resistor*, terdiri dari sebuah cakram semikonduktor yang mempunyai dua buah elektroda pada permukaannya. Pada saat gelap atau cahaya redup, bahan dari cakram tersebut menghasilkan elektron bebas dengan jumlah yang relatif kecil. Sehingga hanya ada sedikit elektron untuk mengangkut muatan elektrik. Artinya pada saat cahaya redup LDR menjadi konduktor yang buruk, atau bisa disebut juga LDR memiliki resistansi yang besar pada saat gelapatau cahaya redup. Pada saat cahaya terang, ada lebih banyak elektron yang lepas dari atom bahan semi konduktor tersebut. Sehingga akan ada lebih banyak elektron unntuk mengangkut muatan elektrik. Artinya pada saat cahaya terang LDR menjadi konduktor yang baik , atau bisa disebut juga LDR memiliki resistansi yang kecil pada saat cahaya terang.

LDR merupakan salah satu jenis resistor yang disebut sebagai fotoresistor, nilai hambatan LDR di pengaruhi oleh cahaya yang diterima dari lingkungan sekitar. Resistansi LDR dapat berubah-ubah tergantung pada intensitas cahaya yang diterima oleh LDR itu sendiri (Supatmi, 2015).



**Gambar 2.2 Sensor LDR (Widodo, 2015)**

# Tabel 2.2 Spefikasi Sensor LDR

|  |  |
| --- | --- |
| Tegangan maksimum | 150 V DC |
| Dark resistence | 1.0 M Ohm |
| Daya maksimum | 100Mw |
| Spectrum peak | 540 mm |

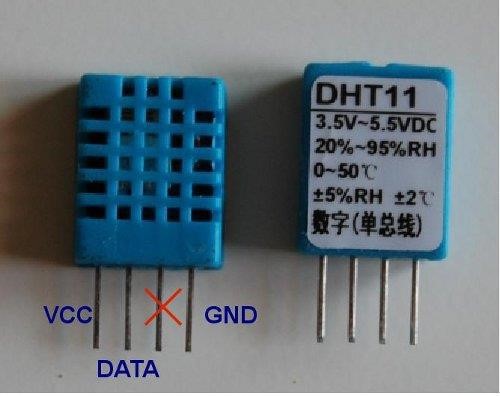
#### 2.3 Sensor *DHT11*

*Sensor DHT11* merupakan sensor yang dengan kalibrasi sinyal digital yang memberikan informasi suhu dan kelembaban. Sensor ini tergolong komponen yang memiliki tingkat stabilitas yang sangat baik, apalagi digandeng dengan kemampuan mikrokontroler Atmega8. Produk dengan kualitas terbaik, respon pembacaan yang cepat, dengan harga yang terjangkau.

*DHT11* memiliki fitur kalibrasi yang sangat akurat. Koefisien kalibrasi ini disimpan dalam OTP program *memory*, sehingga ketika internal sensor mendeteksi sesuatu suhu atau kelembaban, maka modul ini membaca koefisien sensor tersebut. Ukurannya yang kecil, dengan transmisi sinyal hingga 20 meter, membuat produk ini digunakan untuk banyak aplikasi–aplikasi pengukuran suhu (Wibawanto, 2015).

# Tabel 2.3 Spesifikasi Sensor *DHT11*

|  |  |
| --- | --- |
| Supply Voltage | +5V |
| Temperature Range | 0-50 error of ± 2 |
| Humididty | 20-90 % RH ± 5% RH error |
| Interface | Digital |



### Gambar 2.3 Sensor *DHT11* (Wibawanto, 2017)

#### 2.4 Sensor Hujan

Sensor hujan adalah sensor yang di fungsikan untuk mendeteksi ada tidaknya kondisi rintik hujan, yang dimana dapat dimanfaatkan dalam berbagai aplikasi mulai dari yang yang sederhana sehingga aplikasi kompleks. Di pasaran sensor ini dijual dalam bentuk *module* sehingga hanya perlu menyediakan kabel *jumper* untuk dihubungkan ke *Arduino*.

Prinsip kerja dari modul ini adalah dimana pada saat air hujan mengenai panel sensor, maka akan terjadi proses elektrolisasi oleh air hujan. Dan kerena air hujan termasuk dalam golongan cairan elektrolit, cairan tersebuat dapat menghantarkan arus listrik. Dan pada modul terdapat ic komparator yang dimana keluaran dari sensor ini dapat berupa logika atau kondisi *on* atau *off* . Serta pada modul sensor ini terdapat keluaran yang berupa tegangan pula. Sehingga dapat di koneksikan ke pin khusus *Arduino* yaitu analog Digital *Converter* (Pratiwi, 2017).

# Tabel 2.4 Spesifikasi Sensor Hujan

|  |  |
| --- | --- |
| Tegangan | 3.3 V -5 V |
| Output | 0 dan 1 |
| Size | 3.2 cm X 1.4 |



### Gambar 2.4 Sensor Hujan (Pratiwi, 2017)

#### 2.5 Motor *Stepper*

Motor *stepper* adalah perangkat yang bersifat *electromagnetic, rotary* dan *incremental* yang akan merubah pulsa listrik kedalam bentuk gerakan mekanik. Dimana jumlah putaran atau rotasi yang dibuat berbanding lurus dengan jumlah pulsa digital yang di-*input*-kan dan kecepatan rotasinya relatif terhadap besar frekuensi pulsa yang di-*input*-kan.

Motor *stepper* memiliki beberapa keunggulan dibandingkan dengan penggunaan motor DC biasa. Keunggulannya antara lain adalah :

1. Sudut rotasi motor proporsional dengan pulsa masukan sehingga lebih mudah diatur.
2. Motor dapat langsung memberikan torsi penuh pada saat mulai bergerak
3. Posisi dan pergerakan repetisinya dapat ditentukan secara presisi
4. Memiliki respon yang sangat baik terhadap mulai, stop dan berbalik (perputaran)
5. Sangat realibel karena tidak adanya sikat yang bersentuhan dengan rotor seperti pada motor DC
6. Dapat menghasilkan perputaran yang lambat sehingga beban dapat dikopel langsung ke porosnya
7. Frekuensi perputaran dapat ditentukan secara bebas dan mudah pada range yang luas.



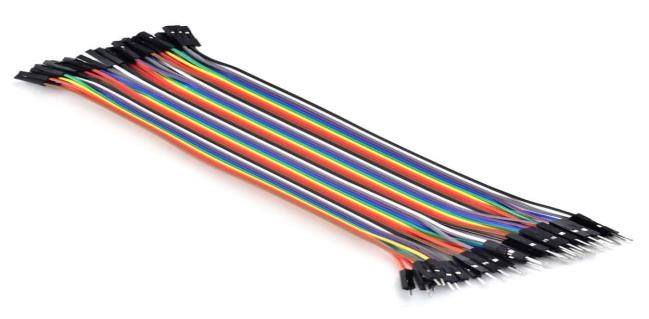
### Gambar 2.5 Motor *Stepper* (Herawan, 2015)

#### 2.6 Kabel *Jumper Wire*

Kabel *Jumper Wire* adalah kabel yang biasanya digunakan untuk penghubung antara perangkat sensor*, breadboard*, *mikrokontroler* dan media transmisi penghantar listrik maupun signal-signal dari sensor (Sunaryo, 2017).

Secara umum kabel *jumper wire* terdiri dari 3 jenis yaitu:

1. *Male – Male*
2. *Male – Female*
3. *Female – Female*



### Gambar 2.6 Kabel *Jumper Wire (*Sunaryo, 2017*)*

#### 2.7 Catu Daya

Catu daya memegang peranan yang sangat penting dalam hal perancangan sebuah robot. Tanpa bagian ini robot tidak akan berfungsi. Begitu juga bila pemilihan catu daya tidak tepat, maka robot tidak akan bekerja dengan baik.

Penentuan sistem catu daya yang akan digunakan ditentukan oleh banyak faktor, diantaranya :

1. Tegangan

Setiap aktuator tidak memiliki tegangan yang sama. Hal ini akan berpengaruh terhadap disain catu daya. Tegangan tertinggi dari salah satu aktuator akan menentukan nilai tegangan catu daya.

1. Arus

Arus memiliki satuan Ah (*Ampere-hour*). Semakin besar Ah, semakin lama daya tahan baterai bila digunakan pada beban yang sama.

1. Teknologi Baterai

Baterai isi ulang ada yang dapat diisi hanya apabila benar-benar kosong, dan ada pula yang dapat diisi ulang kapan saja tanpa harus menunggu baterai tersebut benar-benar kosong.

Banyak jenis baterai yang bisa dipakai untuk mensuplai tegangan untuk robot antara lain (*Alkaline, Fuel Cell, Lead Acid,Lithium, NiCad, NiMH*, baterai *Lithium Polymer* (LIPO)).Namun catu daya yang digunakan dalam robot ini menggunakan baterai *Lithium Polymer* (LIPO). Daya yang dimiliki baterai lipo adalah 3,7 V adalah tegangan nominal 1 *cell* dari kimia LIPO. Tegangan sebenarnya dapat dinaikan sampai pada pengisian ( *charging* ) maksimum pada 4,25 V dan dapat diturunkan sampai minimum 3,0 V. Serta berat beban baterai lipo cukup ringan dibandingkan jika menggunakan baterai yang jenis lain. Rangkaian seri baterai berdasarkan jumlah *cell*nya adalah sebagai berikut :

1. Untuk baterai 2 *cell volt* yang disusun seri adalah 2 x 3,7 V = 7,4 V
2. Untuk baterai 3 *cell volt* yang disusun seri adalah 3 x 3,7 V = 11,1 V
3. Untuk baterai 4 *cell volt* yang disusun seri adalah 4 x 3,7 V = 14,8 V

Untuk baterai 6 *cell* tegangan yang disusun seri hanya maksimum adalah 3 *cell* yang kemudian disusun paralel agar hanya memilik maksimum tegangan11,1 V. Total *Ampere* akan mengikuti pembuatan seri dan paralel dari baterai tersebut. *Range* dengan minimum 3,7 V sampai pada maksimum 4,2 V adalah *range* yang paling baik digunakan untuk menjaga kemampuan kimiawi dari *cell*  baterai LIPO tersebut agar tidak mudah rusak. Oleh karena itu normalnya alat-alat *charger* LIPO akan otomatis melakukan *cut off charging* pada 4,2 V per cellnya. Kelemahannya adalah *charger* LIPO tidak dapat melakukan pembagian pengisian arus setiap *cell* dengan rata.



### Gambar 2.7 Baterai *Lithium Polymer* (LIPO)

#### 2.8 *Android*

*Android* adalah sistem operasi berbasis *Linux* yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer *tablet*. *Android* awalnya dikembangkan oleh *android,* Inc. Dukungan finansial dari *google* yang kemudian membelinya pada tahun 2005. *Android* disebut sebagai *platform mobile* pertama yang lengkap, terbuka dan bebas (Juansyah, 2015).

Para *desainer* dapat melakukan pendekatan *komprehensif,* ketika mengembangkan *platform android* merupakan sistem operasi yang aman dan banyak menyediakan tools dalam membangun *software* (Juansyah, 2015).

1. Terbuka (*Open Source*)

*Platform* *android* disediakan melalui lisensi *open source*. Pengembang dapat dengan bebas mengembangkan aplikasi *Android* sendiri menggunakan *Linux* Kernel 2.6.

1. *Free* (*Free Platform*)

*Android* adalah *platform* atau aplikasi yang bebas atau gratis untuk pengembang. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform android*. Tidak ada biaya keanggotaan, biaya pengujian dan kontrak yang diperlukan. *Android* dapat didistribusikan dan dikembangkan dalam bentuk apapun. Pengembang memiliki beberapa pilihan ketika membuat aplikasi berbasis *android*. Kebanyakan pengembang menggunakan *Eclipse* yang tersedia secara bebas untuk merancang dan mengembangkan aplikasi *android.*



### Gambar 2.8 Logo *Android* (Juansyah, 2015)

#### 2.9 *Android Studio*

*Android Studio* merupakan sebuah *Integrated Development Environment* (IDE) khusus untuk membangun aplikasi yang berjalan pada *platform android*. *Android studio* ini berbasis pada IntelliJ IDEA, sebuah IDE untuk bahasa pemrograman Java. Bahasa pemrograman utama yang digunakan adalah Java, sedangkan untuk membuat tampilan atau layout, digunakan bahasa XML. *Android studio* juga terintegrasi dengan *Android* *Software Development Kit* (SDK) untuk *deploy* ke perangkat a*ndroid*. *Android Studio* juga merupakan pengembangan dari *eclipse*, dikembangkan menjadi lebih kompleks dan professional yang telah tersedia didalamnya *Android Studio IDE*, *Android SDK tools*.

Pengembangan aplikasi *Android* dan bersifat *open source* atau gratis. Peluncuran *Android Studio* ini diumumkan oleh Google pada 16 mei 2013 pada *event* Google I*/O Conference* untuk tahun 2013. Sejak saat itu, *Android Studio* mengantikan *Eclipse* sebagai IDE resmi untuk mengembangkan aplikasi *Android*.

*Android studio* sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan *Eclipse* disertai dengan ADT plugin (*Android Development Tools*). *Android studio* memiliki fitur :

1. Projek berbasis pada *Gradle Build*
2. Refactory dan pembenahan *bug* yang cepat
3. Tools baru yang bernama “*Lint*” dikalim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
4. Mendukung *Proguard And App-signing* untuk keamanan.
5. Memiliki GUI aplikasi *android* lebih mudah
6. Didukung oleh Google *Cloud Platfrom* untuk setiap aplikasi yang dikembangkan.

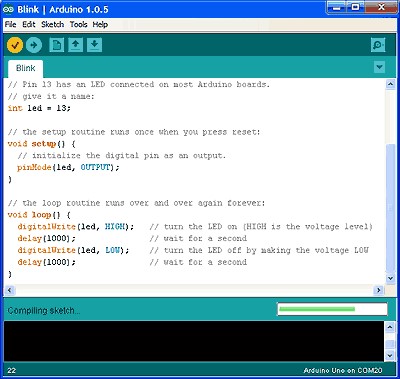


### Gambar 2.9 Logo *Android Studio* (Juansyah, 2015)

#### 2.10 *Software Arduino* IDE

IDE *arduino* adalah *software* yang sangat canggih ditulis dengan menggunakan java. IDE *Arduino* terdiri dari :

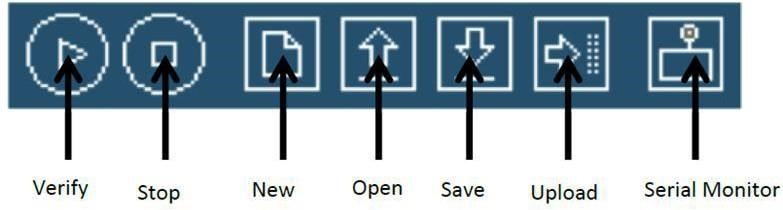
1. Editor program, sebuah *windows* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *Processing.*
2. *Compiler,* sebuah modul yang mengubah kode program (bahasa *processing* menjadi kode *biner*. Bagaimanapun sebuah *microcontroller* tidak akan bisa memahami bahasa *processing.*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



### Gambar 2.10 Tampilan *Arduino* IDE (Dawud, 2016)

Pada Gambar 2.11 dapat melihat *toolbar* IDE yang memberian akses instan ke fungsi-fungsi yang penting :

1. Dengan tombol *Verify* dapat mengkompilasi program yang saat ini di *editor*.
2. Tombol *New* menciptakan program baru dengan mengosongkan isi dari jendela *editor* saat ini. Sebelum hal itu terjadi, IDE memberikan kesempatan untuk menyimpan semua perubahan belum disimpan.
3. Dengan *Open,* dapat membuka program yang ada dari sistem *file*.
4. Tombol *Save,* menyimpan program saat ini.
5. Ketika mengklik tombol *Upload*, IDE mengkompilasi saat ini program dan *upload* ke papan *arduino* yang telah dipilih di IDE menu *Tools* > *Serial port*.
6. *Arduino* dapat berkomunikasi dengan komputer melalui koneksi serial. Mengklik tombol serial monitor membuka jendela serial monutor yang memungkinkan dapat melihat yang dikirimkan oleh *arduino* dan juga untuk mengirim data kembali.
7. Tombol *stop* menghentikan *serial monitor*.



### Gambar 2.11 *Toolbar Arduino* IDE (Dawud, 2016)

#### 2.11 *Pemrograman Arduino*

Bahasa pemrograman *arduino* pada dasarnya secara konsep sama dengan bahasa pemrograman C/C++. Tetapi dalam penulisannya terdapat sedikit perbedaan dengan bahasa C/C++. Perbedaan yang paling terlihat adalah dari struktur bahasa yang digunakan pada bahasa *arduino* yaitu pada fungsi utama dipisahkan menjadi dua bagian yaitu fungsi *setup*() dan fungsi *loop*(), untuk memberikan kemudahan ketika membaca *sketch* (Dawud, 2016).

Fungsi *setup*() berguna sebagai *default* yang dibuat bersama dengan fungsi *loop*() saat proses pembuatan *sketch*, fungsi *setup*() digunakan untuk inisialisasi variable mode pin yang digunakan pengguna fungsi *library* dan lain-lain, fungsi setup() hanya berjalan satu kali yaitu saat *board arduino* pertama kali dinyalakan atau saat tombol reset ditekan (Dawud, 2016).

Sedangkan fungsi *loop*() digunakan untuk menuliskan kode program yang berjalan terus-menerus. Setelah fungsi *setup*() selesai dijalankan fungsi ini otomatis dijalankan dan akan terus berjalan sampai *board arduino* mati atau tombol reset ditekan (Dawud, 2016).

*Arduino* mempunyai *software* IDE tersendiri untuk keperluan pemrograman yaitu IDE *Arduino*. *Software* IDE *Arduino* bersifat *multiplatform* yang dapat dijalankan pada sistem operasi *windows*, *linux* maupun *mac*. Untuk mendukung kebutuhan tertentu banyak tersedia *library* di *internet* yang dapat diunduh secara gratis. (Dawud, 2016)

#### 2.12 Bahasa Pemrograman C

Bahasa C adalah bahasa yang dikembangkan oleh *Martin Richards* pada tahun 1967. Bahasa C adalah bahasa standar, artinya suatu program yang ditulis dengan versi bahasa C tertentu akan dapat dikompilasi dengan versi bahasa C yang lain dengan sedikit modifikasi. Beberapa alasan mengapa bahasa C banyak digunakan. (Suprapto, 2008)

Diantaranya adalah sebagai berikut :

1. Bahasa C tersedia hampir di semua jenis komputer.
2. Kode bahasa C sifatnya portabel.
3. Bahasa C hanya menyediakan sedikit kata – kata kunci.
4. Proses executable program bahasa C lebih cepat.
5. Dukungan Pustaka yang banyak.
6. C adalah bahasa yg terstruktur.
7. Selain bahasa tingkat tinggi, C juga dianggap bahasa tingkat menengah.
8. Bahasa C adalah Kompiler (;)

Bahasa C mempunyai kemampuan lebih dibanding dengan bahasa pemrograman yang lain, bahasa C merupakan bahasa pemrograman yang bersifat portable yaitu suatu program yang dibuat dengan bahasa C, pada suatu komputer akan dapat dijalankan pada komputer lain dengan sedikit (atau tanpa) ada perubahan. Bahasa C merupakan bahasa yang biasa digunakan untuk keperluan pemograman sistem antara lain untuk membuat: (Suprapto, 2008).

1. Assembler
2. Interpreter
3. Compiler
4. Sistem Operasi
5. Program bantu (*utility*)
6. Editor
7. Paket program aplikasi

Bahasa C juga menunjukkan karateristik sebagai berikut :

* 1. Ada sejumlah kecil tetap kata kunci, termasuk set lengkap aliran kontrol primitif seperti *for , if/else , while , switch , dan do/while .* Ada dasarnya satu *name space*, dan *user-defined* nama tidak dibedakan dari kata kunci oleh jenis sigil.
  2. Ada sejumlah besar operator aritmatika dan logis, seperti + , += , ++ , & , ~ , dan lain-lain.
  3. Lebih dari satu tugas dapat dilakukan dalam sebuah pernyataan tunggal.
  4. Fungsi kembali nilai-nilai dapat diabaikan ketika tidak diperlukan.
  5. Mengetik statis, tetapi lemah ditegakkan semua data memiliki tipe, tetapi konversi implisit dapat dilakukan, misalnya, karakter dapat digunakan sebagai bilangan bulat.
  6. Deklarasi sintaks meniru konteks penggunaan. C tidak mendefinisikan kata kunci, melainkan sebuah pernyataan dimulai dengan nama tipe diambil sebagai deklarasi. Tidak ada fungsi kata kunci, melainkan fungsi ditunjukkan dengan tanda kurung dari daftar argumen.
  7. *User-defined* (*typedef*) dan jenis senyawa yang mungkin.
  8. Jenis data *heterogen agregat* (*struct*) memungkinkan data terkait elemen yang dapat diakses dan ditugaskan sebagai satu unit.
  9. Array pengindeksan adalah gagasan sekunder, didefinisikan dalam istilah *aritmetik pointer*. Tidak seperti *struct*, array tidak kelas benda, tidak dapat ditugaskan atau dibandingkan menggunakan satu *built-in operator*. Tidak ada "array" kata kunci, digunakan atau definisi, melainkan, tanda kurung siku mengindikasikan array sintaks, misalnya :

{ month[11] ;

}.

* 1. Tipe *enumerated* yang mungkin dengan *enum* kata kunci. Tidak ditandai dan secara bebas dengan bilangan bulat.
  2. *String* bukan merupakan tipe data yang terpisah, tetapi secara konvensional diimplementasikan sebagai diakhiri *null-array* dari karakter.
  3. Tingkat rendah akses ke memori komputer dimungkinkan dengan mengubah alamat mesin untuk diketik *pointer*.
  4. Prosedur (subrutin tidak kembali nilai-nilai) adalah kasus khusus dari fungsi, dengan tipe kembali *untyped voi*d.
  5. Fungsi tidak dapat didefinisikan dalam lingkup leksikal fungsi lainnya.
  6. Fungsi dan data *pointer* memungkinkan ad hoc polimorfisme *run-time*.
  7. Sebuah *processor* melakukan makro definisi, kode sumber inklusi file, dan kompilasi bersyarat .
  8. Ada bentuk dasar dari modularitas, file dapat dikompilasi secara terpisah dan dihubungkan bersama-sama, dengan kontrol atas fungsi dan objek data yang terlihat ke file lain melalui *static* dan *extern* atribut.
  9. Fungsi kompleks seperti I / O , tali manipulasi, dan fungsi matematika secara konsisten didelegasikan kepada rutinitas perpustakaan.

#### 2.13 Pemrograman *Java Development Kit*

*Java Development Kit* (JDK) adalah sebuah produk yang dikembangkan oleh *Oracle* produk yang dikembangkan oleh *Oracle* yang ditunjukan untuk para *Developer java*. JDK dilengkapi dengan banyak komponen untuk melakukan pemrograman, JDK juga berisi paket *Java Runtime Environment* (JRE) yang *komplet*, biasanya disebut *private runtime* dari *JREreguler* dan dilengkapi dengan konten tambahan, yaitu terdiri atas java *Virtual Machine* dan semua *class library* yang ada di *environment* produk dan juga *library* tambahan yang berguna untuk *developer*.

#### 2.14 *Internet of Things* (IoT)

*Internet of Things* pertama kali dimunculkan oleh Kevin Ashton pada tahun 1999 di salah satu presentasinya. Kini banyak perusahaan besar mulai mendalami *Internet of Things* sebut saja *Intel, Microsoft, Oracle*, dan banyak lainnya.

Menurut analisa *McKinsey Global Institute*, *internet of things* adalah sebuah teknologi yang memungkinkan kita untuk menghubungkan mesin, peralatan, dan benda fisik lainnya dengan sensor jaringan dan aktuator untuk memperoleh data dan mengelola kinerjanya sendiri, sehingga memungkinkan mesin untuk berkolaborasi dan bahkan bertindak berdasarkan informasi baru yang diperoleh secara independen. Sedangkan menurut Wikipedia, *internet of things* adalah interkoneksi yang unik antara *embedded computing devices* dalam infrastruktur internet yang ada. Sebuah publikasi mengenai *Internet of things* in 2020 menjelaskan bahwa *internet of things* adalah suatu keadaan ketika benda memiliki identitas, bisa beroperasi secara intelijen, dan bisa berkomunikasi dengan sosial, lingkungan, dan pengguna.

Dengan demikian, dapat kita simpulkan bahwa *internet of things* membuat kita membuat suatu koneksi antara mesin dengan mesin, sehingga mesin-mesin tersebut dapat berinteraksi dan bekerja secara independen sesuai dengan data yang diperoleh dan diolahnya secara mandiri. Tujuannya adalah untuk membuat manusia berinteraksi dengan benda dengan lebih mudah, bahkan supaya benda juga bisa berkomunikasi dengan benda lainnya.

Cara kerja dari *internet of things* cukup mudah. Setiap benda harus memiliki sebuah IP Address. IP Address adalah sebuah identitas dalam jaringan yang membuat benda tersebut bisa diperintahkan dari benda lain dalam jaringan yang sama. Selanjutnya, IP address dalam benda-benda tersebut akan dikoneksikan ke jaringan internet. Saat ini, koneksi internet sudah sangat mudah kita dapatkan. Dengan demikian, kita dapat memantau benda tersebut bahkan memberi perintah kepada benda tersebut.(Dendy, 2015).

#### 2.15 Cahaya Matahari

Sinar matahari atau radiasi matahari adalah sinar yang berasal dari Matahari. Tanaman menggunakan cahaya matahari untuk berfotosintesis dan membuat makanan. Tanpa cahaya matahari, takkan ada kehidupan di bumi. Sinar matahari bisa berakibat baik maupun buruk kepada kesehatan seseorang. Dalam terang, tubuh manusia memproduksi vitamin D sendiri. Terlalu lama terpapar sinar matahari bisa menyebabkan kulit terbakar. Tanaman memerlukan cahaya matahari agar tumbuh hijau. Dengan air tanpa cahaya matahari, tanaman akan tumbuh tinggi dengan cepat, tetapi akan terlihat kuning dan kekurangan air, meskipun saat disentuh, daunnya teraba amat basah.

#### 2.16 Suhu

Suhu adalah suatu besaran yang menunjukan derajat panas khususnya pada benda. Benda yang mempunyai panas dapat menunjukan suhu yang tinggi dibandingkan pada benda yang dingin.

#### 2.17 Hujan

Hujan adalah sebuah [presipitasi](https://id.wikipedia.org/wiki/Presipitasi_(meteorologi)) berwujud cairan, berbeda dengan presipitasi non-cair seperti [salju,](https://id.wikipedia.org/wiki/Salju) [batu es](https://id.wikipedia.org/wiki/Hujan_es) dan [slit.](https://id.wikipedia.org/w/index.php?title=Campuran_hujan_dan_salju&action=edit&redlink=1) Hujan memerlukan keberadaan lapisan atmosfer tebal agar dapat menemui suhu diatas titik leleh es di dekat dan diatas permukaan Bumi. Di Bumi, hujan adalah proses [kondensasi](https://id.wikipedia.org/wiki/Kondensasi) [uap air](https://id.wikipedia.org/wiki/Uap_air) di atmosfer menjadi [butir](https://id.wikipedia.org/w/index.php?title=Butir_air&action=edit&redlink=1) [air](https://id.wikipedia.org/wiki/Air) yang cukup berat untuk jatuh dan biasanya tiba di daratan. Dua proses yang mungkin terjadi bersamaan dapat mendorong udara semakin jenuh menjelang hujan, yaitu pendinginan udara atau penambahan uap air ke udara.

#### 2.18 *Flowchart*

*Flowchart* atau Bagan alir adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir (*flowchart*) digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi (Ardiansyah, 2016).

# Tabel 2.5 Daftar Simbol - Simbol *Flowchart*

|  |  |  |  |
| --- | --- | --- | --- |
| NO. | GAMBAR | NAMA | KETERANGAN |
| 1 |  | Proses | Mempresentasikan operasi. |
| 2 |  | *Input / Output* | Mempresentasikan *Input* data atau *Output* data yang diproses atau informasi. |
| 3 |  | Keputusan | Keputusan dalam program. |

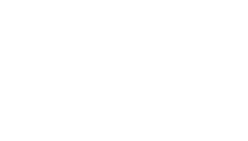
**Tabel 2.5 Daftar Simbol - Simbol *Flowchart* (Lanjutan)**

|  |  |  |  |
| --- | --- | --- | --- |
| 4 |  | Dokumen | Dokument I / O dalam format cetak. |
| 5 |  | Terminal points | Awal / akhir flowchart. |
| 6 |  | Preparation | Pemberian harga awal. |
| 7 |  | Manual input | Input yang dimasukkan secara manual dari keyboard. |
| 8 |  | Penghubung | Keluar atau masuk dari bagian lain flowchart khususnya |
| 9 |  | Penghubung | Keluar atau masuknya dari bagian lain flowchart khususnya halaman lain. |
| 10 |  | Display | Output yang ditampilkan pada terminal |
| 11 |  | Anak panah | Mempresentasikan alur kerja. |

#### 2.19 Logika *Fuzzy*

Logika *fuzzy* pertama kali di kenalkan oleh Prof. Lotfi A.Zadeh Pada tahun 1965. Logika *fuzzy* merupakan suatu metode pengambilan keputusan berbasis aturan yang digunakan ntuk memecahkan keabu-abuan masalah pada sistem yang sulit dimodelkan atau memiliki ambiguitas. Dasar logika *Fuzzy* adalah teori himpunan *fuzzy*.

*INPUT*



*BLACK BOX*

*BL*

*B*

*ACK*

*BOX*

*OUTPUT*

### Gambar 2.12 Diagram Blok Logika Fuzzy sebagai *Black Box*

Pada gambar 2.12 logika *fuzzy* dapat dianggap sebagai kotak hitam yang berhubungan antara ruang input menuju ruang output (Kusumadewi, 2015). Kotak hitam yang dimaksudkan adalah metode yang dapat digunakan untuk mengolah data input menjadi output dalam bentuk informasi yang baik. Adapun beberapa alasan mengapa digunakannya logika fuzzy adalah :

1. Konsep logika *fuzzy* mudah dimengerti.
2. Penggunaan logika *fuzzy* yang fleksibel.
3. Logika fuzzy mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks
4. Tidak perlu adanya proses pelatihan untuk memodelkan pengetahuan yang dimiliki oleh pakar.
5. Logika *fuzzy* didasari pada bahasa sehari-hari sehingga mudah dimengerti. Himpunan *fuzzy* disebut himpunan tegas (crisp), nilai keanggotaan suatu item x dalam suatu himpunan A yang dituliskan dengan [x], dimana memiliki dua buah kemungkinan nilai yaitu :
   1. Satu (1), yang memiliki arti bahwa suatu item tidak menjadi anggota dalam suatu himpunan tertentu.
   2. Nol (0), yang memiliki arti bahwa suatu item tidak menjadi anggota dalam suatu himpunan tertentu.

Himpunan fuzzy memiliki dua atribut :

* + 1. Lingustik, merupakan penamaan grub yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami/ sehari-hari.

Contohnya : PENDEK, SEDANG, TINGGI

* + 1. Numeris, merupakan suatu nilai angka yang menunjukkan ukuran dari suatu variabel.

Contohnya : 140, 160, 180

#### 2.20 Fuzzifikasi

Fuzzifikasi adalah proses mengubah suatu satuan masukan dari bentuk tegas (*crisp*) menjadi *Fuzzy* (variabel linguistik). Proses ini secara umum dijalankan dalam bentuk himpunan-himpunan *Fuzzy* dengan suatu fungsi keanggotaan yang berbeda. Fungsi keanggotaan yang digunakan dalam perancangan *Fuzzy* menggunakan fungsi keanggotaan segitiga dan fungsi keanggotaan trapesium.

#### 2.21 Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data, ke dalam nilai keanggotaannya sering juga disebut dengan derajat keanggotaan yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang bisa digunakan yaitu:

1. Representasi Linear Naik

Pada representasi linear, pemetaan input ke derajat keanggotannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas. Ada 2 keadaan himpunan *Fuzzy* yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi Gambar 2.1.

1

Derajat

Kenggotaan

𝜇[x]

0

a domain b

### Gambar 2.13 Representasi Linear Naik

Fungsi keanggotaan:

0; 𝑥 ≤ 𝑎

#### 𝜇[𝑥] = {(𝑥 − 𝑎)/(𝑏 − 𝑎); 𝑎 ≤ 𝑥 ≤ 𝑏 (2.1)

1; 𝑥 ≥ 𝑏

2. Representasi Linear Turun

Kedua, merupakan kebalikan yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah Gambar 2.2.

1

Derajat

Kenggotaan

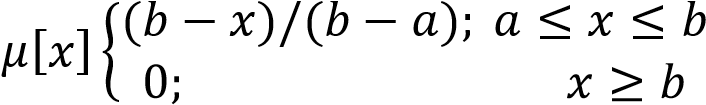


0

a domain b

### Gambar 2.14 Representasi Linear Turun

Fungsi keanggotaan:

 (2.2)

3. Representasi Kurva Segitiga

Fungsi keanggotaan segitiga berasal dari representasi kurva segitiga. Kurva segitiga merupakan gabungan antara dua garis (linear) dan bentuk kurva tersebut dapat dilihat pada Gambar 2.3. Fungsi keanggotaan segitiga diperoleh dari persamaan :

1

Derajat

Kenggotaan

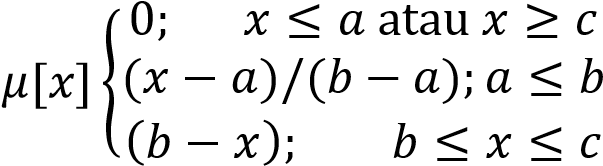


0

a b c

### Gambar 2.15 Representasi Kurva Segitiga

Fungsi keanggotaan :

 (2.3)

4. Representasi kurva Trapesium

Fungsi keanggotaan trapesium berasal dari representasi kurva trapesium. Kurva trapesium merupakan bentuk segitiga yang pada beberapa titiknya memiliki nilai keanggotaan 1. Bentuk kurva tersebut dapat dilihat pada Gambar 2.4 Fungsi keanggotaan trapesium diperoleh dari persamaan.

1

Derajat

Kenggotaan

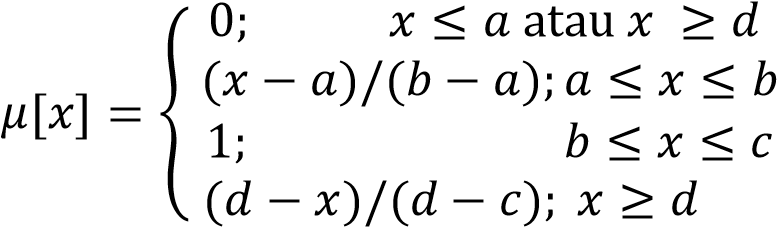


0

a b c d domain

### Gambar 2.16 Representasi kurva trapezium

Fungsi keanggotaan :

 (2.3)

#### 2.22 Aplikasi Fungsi Implikasi (Basis Aturan)

Aplikasi fungsi implikasi (basis aturan) berisi aturan-aturan *fuzzy* yang digunakan untuk pengendalian sistem. Aturan-aturan ini dibuat berdasarkan logika manusia, serta berkaitan erat dengan jalan pikiran dan pengalaman pribadi. Sehingga aturan ini bersifat subjektif, tergantung dari pembuatnya. Tiap-tiap aturan (proposisi) pada basis pengetahuan *fuzzy* akan berhubungan dengan suatu relasi *fuzzy*.

Bentuk umum dari aturan yang digunakan dalam fungsi implikasi adalah (Kusumadewi, 2015) :

IF x is A THEN y is B

Dengan x dan y adalah saklar, A dan B adalah himpunan fuzzy . Proposisi yang mengikuti IF disebut sebagai anteseden, sedangkan proposisi yang mengikuti THEN disebut konsekuen. Secara umum ada 2 fungsi implikasi yang dapat digunakan, yaitu Min (*minimum*) dimana fungsi ini akan memotong *output* himpunan *fuzzy* dan Dot (*product*) fungsi ini akan menskala output himpunan *fuzzy* (Kusumadewi, 2015).

#### 2.23 Operator Dasar Zadeh Untuk Operasi Hitungan *Fuzzy*

Seperti halnya himpunan konvensional, ada beberapa operasi yang disefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan *fuzzy*. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan sering dikenal dengan nama *fire strength* atau 𝛼 − 𝑝𝑟𝑒𝑑𝑖𝑘𝑎𝑡. Ada 3 operator dasar yang diciptakan oleh Zadeh, yaitu (Kusumadewi, 2015) :

1. Operator AND

Operator ini berhubungan dengan operasi interseksi pada himpunan. 𝛼 − 𝑝𝑟𝑒𝑑𝑖𝑘𝑎𝑡 sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan.

#### 𝜇𝐴 ∩ 𝐵 = min (𝜇𝐴[𝑋], 𝜇𝐵[𝑦]) (2.4)

1. Operator OR

Operator ini berhubungan dengan operasi union pada himpunan. 𝑎 − 𝑝𝑟𝑒𝑑𝑖𝑘𝑎𝑡 sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan.

𝜇𝐴 ∪ 𝐵 = max(𝜇𝐴[𝑥], 𝜇𝐵[𝑦]) (2.5)

1. Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan 𝑎 − 𝑝𝑟𝑒𝑑𝑖𝑘𝑎𝑡 sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangkan nilai keanggotaan elemen pada himpunan yang bersangkutan dari

1.

*µA‟=* 1-*µA*[x] (2.6)

##### 2.24 Tsukamoto

Pada Metode Tsukamoto, setiap konsekuen pada aturan yang berbentuk *IF THEN* harus direpresentasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, *output* hasil inferensi dari tiap-tiap aturan diberikan secara tegas (*crisp*) berdasarkan *α-predikat* (*fire strength*). Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot (Kusumadewi, 2015 ).

Misalkan ada variabel *input*, yaitu x dan y, serta satu variabel *output* yaitu z. Variabel x terbagi atas 2 himpunan yaitu A1 dan A2, variabel y terbagi atas 2 himpunan juga, yaitu B1 dan B2, sedangkan variabel *output* Z terbagi atas 2 himpunan yaitu C1 dan C2. Tentu saja himpunan C1 dan C2 harus merupakan himpunan yang bersifat monoton. Diberikan 2 aturan sebagai berikut:

[R1] IF x is A1 and y is B2 THEN z is C1

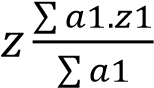
[R2] IF x is A2 and y is B2 THEN z is C1s

#### *Fuzzy Inference System* Tsukamoto

1. Saat proses evaluasi aturan dalam mesin inferensi, metode *fuzzy* Tsukamoto menggunakan fungsi implikasi MIN untuk mendapatkan nilai *α-predikat* tiap-tiap *rule* (α1, α2, α3,.... αn). Masing-masing nilai *α-predikat* digunakan untuk menghitung hasil inferensi secara tegas (*crisp*) masing-masing *rule*

(z1, z2, z3,.... zn).

1. Proses defuzzyfikasi pada metode Tsukamoto menggunakan metode ratarata (*Average*) dengan rumus berikut:



**BAB III**

**METODOLOGI PENELITIAN**

### 3.1 Metode Pengumpulan Data

Untuk pengumpulan sejumlah data yang diperlukan dalam penyusunan tugas akhir ini penulis menggunakan studi literatur. Pengumpulan data, gambaran dan keterangan yang lebih lengkap peneliti menggunakan studi literatur dengan cara mengumpulkan dan mempelajari literatur yang berkaitan perangkat lunak dan perangkat keras yang dibuat dan membantu memperjelas teori-teori yang ada. Sumber literatur berupa buku teks, jurnal*,* karya ilmiah dan situs-situs penunjang.

### 3.2 Metode Pengembangan Sistem

*Network Development Life Cycle* (NDLC) adalah metode yang dapat digunakan untuk mengembangkan suatu jaringan komputer dan kunci dibalik proses perancangan jaringan komputer (Goldman, 2004).

Dalam merancang dan membangun sebuah sistem dibutuhkan suatu permodelan dari keseluruhan proses-proses yang akan dilakukan selama pembuatan perangkat sistem tersebut. Model yang digunakan dalam sistem ini yaitu *Network Development Life Cycle* (NDLC).

# Tabel 3.1 *Fase* pengembangan sistem

|  |  |
| --- | --- |
| ***Fase* NDLC** | **Proses yang dikerjakan** |
| *Analysis* | Melakukan analisis kebutuhan, analisis permasalahan dan analisis cara kanopi otomatis menggunakan logika *fuzzy* tsukamoto. |
| *Design* | Tahap *design* yaitu membuat gambar rancangan aplikasi dan sistem yang akan dibangun. |

35

### Tabel 3.1 *Fase* pengembangan sistem (lanjutan)

|  |  |
| --- | --- |
| ***Fase* NDLC** | **Proses yang dikerjakan** |
| *Simulation Prototype* | Pada tahap ini biasanya pekerjaan jaringan akan dibuat dalam bentuk simulasi menggunakan *tools* khusus di bidang *network.* |
| *Implementation* | Pada tahap ini akan menerapkan semua yang telah direncanakan dan didesain sebelumnya. |
| *Analysis system* | Tahap ini yaitu melakukan pengujian serta analisis terhadap sistem yang telah dibuat. |
| *Management* | Pada tahap ini merupakan pembuatan atau mengatur agar sistem yang dibuat berjalan dengan baik. |

#### 3.3 Perancangan Sistem

Desain sistem menentukan bagaimana sistem akan memenuhi tujuan yang diinginkan. Desain sistem ini dibagi dalam 2 tahapan yaitu :

1. Desain sistem alat kanopi otomatis

Pada tahapan ini ditetapkan alat-alat yang digunakan dalam sistem. Ala-alat pendukung sistem yang diperlukan yaitu : a. *NodeMCU ESP8266*

1. Sensor LDR (*Light Dependent Resistor*)
2. Sensor *DHT11*
3. Sensor Air
4. Motor *Stepper*
5. Catu daya

37

2. Desain Program Pengendali

Pada tahapan ini yang harus dilakukan yaitu :

1. Implementasi program pada alat kanopi menggunakan logika *fuzzy* tsukamoto.
2. Implementasi pengendalian alat menggunakan *android*.

Adapun *software* pendukung yang digunakan dalam perancangan sistem ini adalah *arduino* IDE yang digunakan sebagai *tools* perancangan program yang menggunakan bahasa pemrograman C.

#### 3.4 Pengujian Sistem

Sistem *prototype* diuji, dievaluasi dan dimodifikasi berulang-ulang hingga dapat diterima pemakainya (O'Brien, 2005). Pengujian sistem bertujuan menemukan kesalahan-kesalahan yang terjadi pada sistem dan melakukan revisi sistem. Tahap ini penting untuk memastikan bahwa sistem bebas dari kesalahan (Mulyanto, 2009).

Dalam sistem ini dilakukan dengan 3 pengujian berdasarkan logika *fuzzy* metode tsukamoto yang digunakan antara lain *fuzzyfikasi*, *defuzzyfikasi* dan analisis efisiensi sistem.

1. *Fuzzifikasi*

Pada proses *fuzzyfikasi* dilakukan perhitungan derajat keanggotaan dari *input*an yang ada akan diproses kedalam aturan-aturan *fuzzy* yang dirancang.

1. *Defuzzyfikasi*

Pada tahapan ini dilakukan penarikan himpunan tegas (*crips*) yang berfungsi sebagai *output* dari sistem.

1. Analisis Efisiensi Sistem

Pada tahap ini akan dihitung pengeluaran dari nilai cahaya, nilai suhu, nilai hujan. Setelah menggunakan sistem ini untuk diketahui seberapa besar tingkat efisiensitas dari sistem yang akan diterapkan.

#### 3.5 Waktu dan Tempat Penelitian

##### 3.5.1 Waktu Penelitian

Waktu pelaksanaan penelitian tugas akhir ini dilaksanakan mulai dari bulan Februari 2020. Rincian dapat dilihat pada Tabel 3.2 sebagai berikut.

# Tabel 3.2 Waktu Penelitian

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Uraian | Waktu (2020) | | | | | | | | | | | | | | | | | | | | | | | |
| Februari | | | | Maret | | | | April Mei | | | | | | | | Juni | | | | Juli | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | *Analysis* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | *Design* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | *Simulation*  *Prototype* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | *Analysis system* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | *Implementasi* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | *Management* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

#### 3.5.2 Tempat Penelitian

Adapun tempat penelitian tugas akhir yang akan dilakukan tidak mengacu pada tempat tertentu.

**BAB IV** **ANALISIS DAN PERANCANGAN SISTEM**

### 4.1 Analisis Sistem

Alat kanopi otomatis ini merupakan atap teras rumah yang dapat bergerak ke arah luar dan ke arah dalam miniatur secara otomatis berdasarkan kondisi cuaca dan cahaya lingkungan sekitar. Alat ini dikendalikan oleh tiga sensor yaitu sensor *DHT11,* sensor LDR (*Light Dependent Resistor*) dan sensor hujan serta mempunyai *output* motor *stepper.* Alat ini menggunakan mikrokontroler *NodeMCU ESP8266* di gunakan untuk menggerakan motor *stepper.* Saklar yang terpasang pada alat tersebut digunakan untuk menghentikan pergerakan kanopi atau motor *stepper* yang sedang bekerja dengan mengirimkan logika *input* ke mikrokontroler ketika ujung dari kanopi tersebut telah mengenai saklar.

Prinsip kerja alat kanopi otomatis ini yaitu menggunakan saklar yang dikendalikan oleh mikrokontroler yang menerima *inputan* nilai dari tiga sensor. Nilai dari hasil perhitungan *fuzzy* tsukamoto akan diklasifikasikan menjadi dua kategori yakni buka dan menutup. Pada saat nilai probabilitas dari suatu kategori lebih besar dari kategori yang lain maka kanopi akan bergerak secara otomatis ke luar untuk proses menutup. Sebaliknya, pada saat nilai probabilitas dari suatu kategori lebih kecil maka kanopi akan bergerak masuk dalam miniatur untuk proses membuka.

### 4.2 Analisis Kebutuhan Sistem

Pada analisis sistem yang telah diuraikan terdapat beberapa kebutuhan pada sistem. Kebutuhan tersebut terbagi beberapa bagian yaitu kebutuhan pada perangkat keras dan perangkat lunak.

Kebutuhan perangkat keras yaitu kebutuhan perangkat atau komponen yang dibutuhkan pada sistem dan kebutuhan perangkat lunak untuk membantu agar komponen perangkat keras dapat berfungsi dan dapat dijalankan pada sistem.

39

#### 4.2.1 Analisis Kebutuhan Perangkat Keras

Untuk membangun alat kanopi secara otomatis dibutuhkan beberapa perangkat keras yang menjadi penyusun dan fisik utama agar dapat berfungsi dengan baik. Berikut spesifikasi perangkat keras yang dibutuhkan.

# Tabel 4.1 Spesifikasi Perangkat Keras

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Komponen** | **Jenis**  **Komponen** | **Jumlah** | **Deskripsi** |
| **1** | *NodeMCU*  *ESP8266* | Proses | 1 unit | Digunakan untuk menghubungkan sistem kanopi secara otomatis. |
| **2** | *Handphone* | *Input* dan *output* | 1 unit | Digunakan untuk mengontrol aplikasi  *ACCSYS* |
| **3** | *Motor Stepper* | *Output* | 1 unit | Digunakan untuk membuka dan menutup atap kanopi. |
| **4** | Sensor LDR  (*Light*  *Dependent*  *Resistor)* | *Input* | 1 unit | Digunakan untuk mendeteksi nilai cahaya yang masuk. |
| **5** | Sensor DHT11 | *Input* | 1 unit | Digunakan untuk mengukur suhu. |
| **6** | Sensor Hujan | *Input* | 1 unit | Digunakan untuk mendeteksi adanya air atau tidak. |

#### 4.2.2 Analisis Kebutuhan Perangkat Lunak

Untuk membangunRancang Bangun Aplikasi Alat kanopi Secara Otomatis Berbasis *Android* Menggunakan Logika *Fuzzy* Tsukamoto sehingga dapat berfungsi dengan baik dan benar dibutuhkan beberapa perangkat lunak yang dapat mengubah perangkat keras untuk bekerja sesuai kebutuhan. Perangkat lunak ini juga dibutuhkan untuk mengkofigurasi alat kanopi. Berikut spesifikasi perangkat lunak yang digunakan:

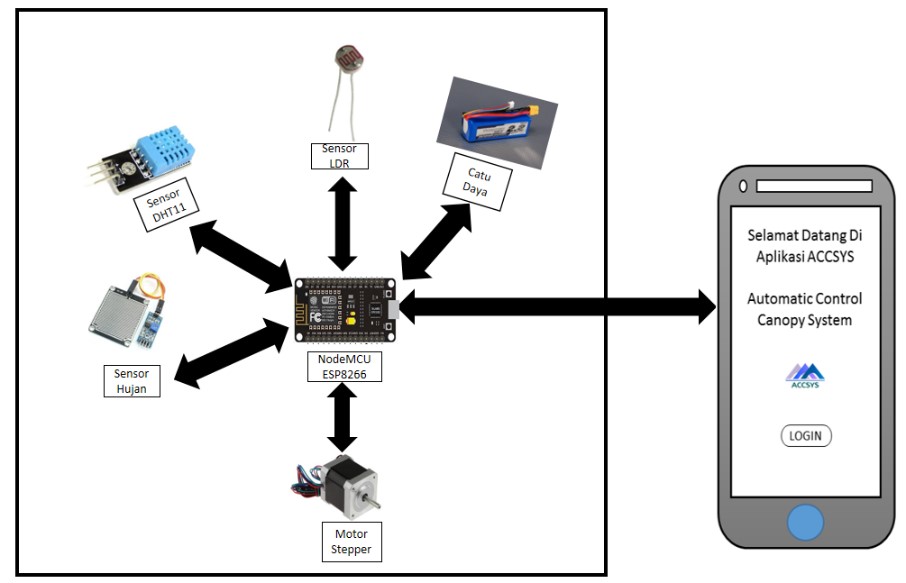
# Tabel 4.2 Spesifikasi perangkat lunak

|  |  |  |
| --- | --- | --- |
| **No** | **Nama alat** | **Deskripsi** |
| **1** | *Software*  *Arduino* IDE | *Arduino* *Integrated Development Environment* (IDE) digunakan untuk melakukan *coding.*  Penulisan *code* dilakukan pada *arduino IDE* lalu dikirim ke *arduino uno.* |
| **2** | Aplikasi  *ACCSYS* | Digunakan untuk monitoring dan mengendalikan alat kanopi otomatis. |
| **3** | Pemrograman *arduino* ( Bahasa pemograman C) | Digunakan untuk pembuat kode sistem kanopi secara otomatis. |

#### 4.3 Proses Perancangan Sistem

Perancangan sistem kanopi terbagi menjadi tiga bagian yaitu perancangan *hardware*, elektronik dan perancangan *software*. Perancangan *hardware* berupa desain *body* sistem, perancangan elektronik adalah rangkaian-rangkaian umum elektronika yang digunakan dan rangkaian *software* adalah rancangan program alat kanopi yang dibuat.

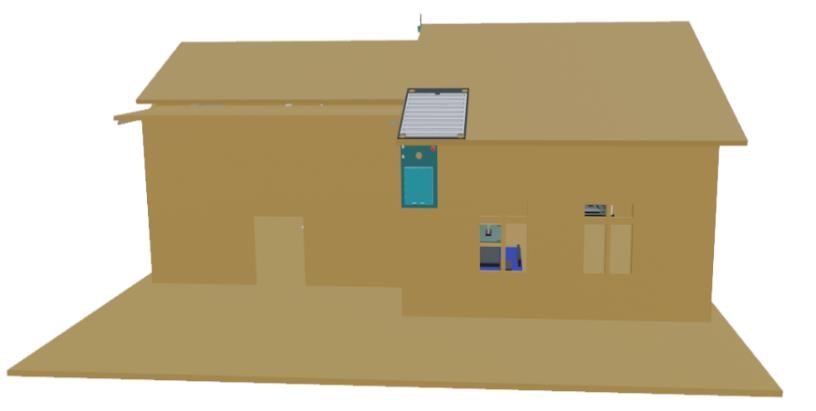
Secara umum sistem dibuat agar dapat memproses perintah yang telah dimasukkan ke dalam *NodeMCU ESP8266* sebagai mikrokontroler dari kanopi kemudian melakukan kontrol yang ada pada *android*.



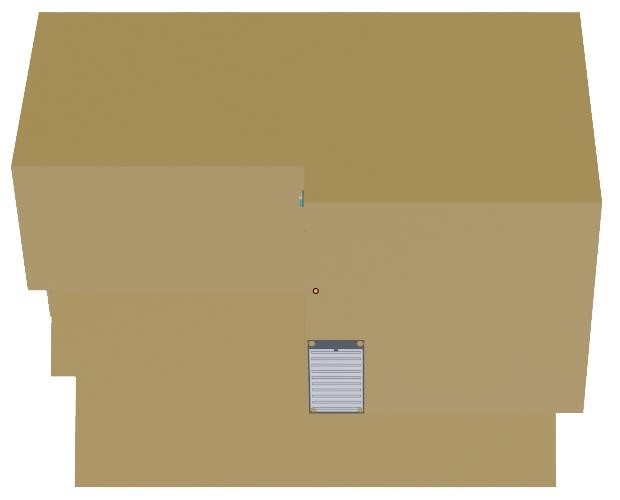
### Gambar 4.1 Gambaran Umum Sistem

#### 4.3.1 Perancangan Mekanik

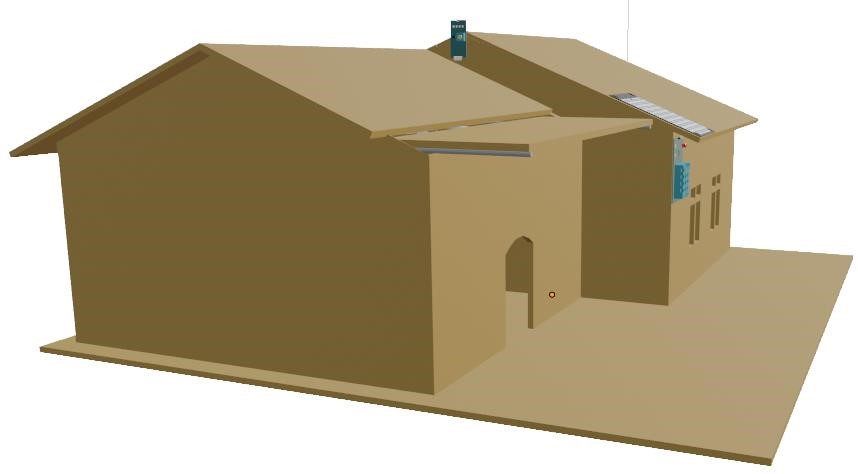
Perancangan alat kanopi otomatis sebagai berikut.



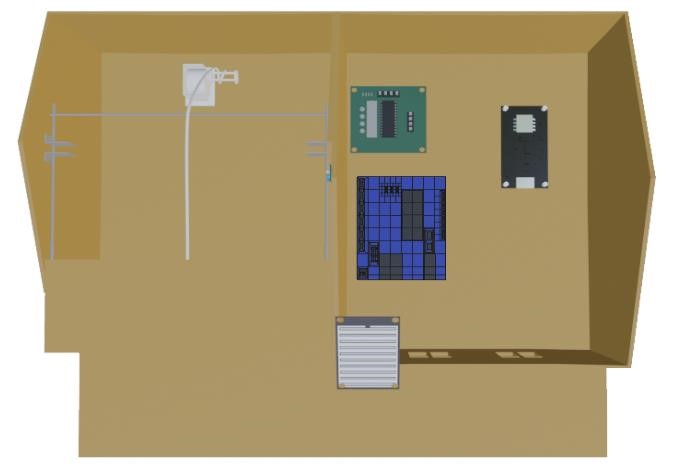
### Gambar 4.2 Alat Tampak Depan



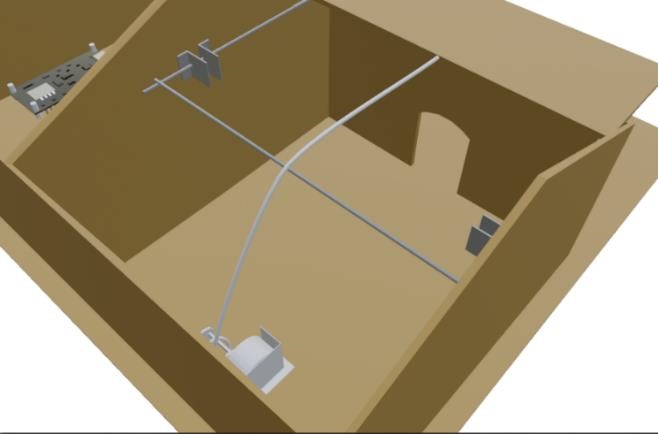
### Gambar 4.3 Alat Tampak Atas



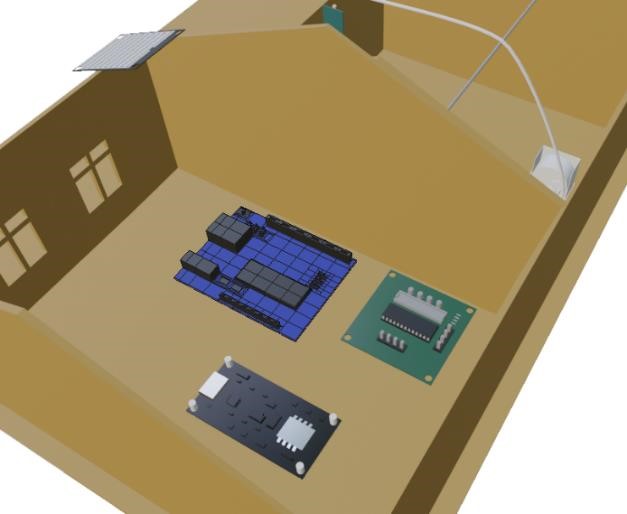
### Gambar 4.4 Alat Tampak Samping



### Gambar 4.5 Alat Tampak Dalam (1)



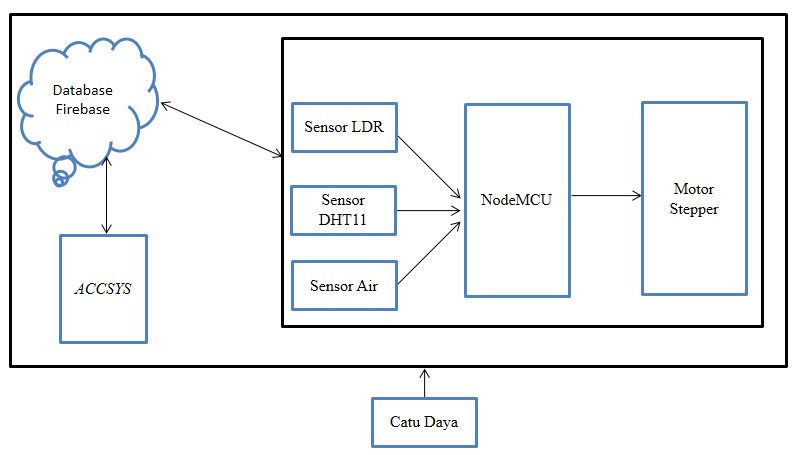
### Gambar 4.6 Alat Tampak Dalam (2)



### Gambar 4.7 Alat Tampak Dalam (3)

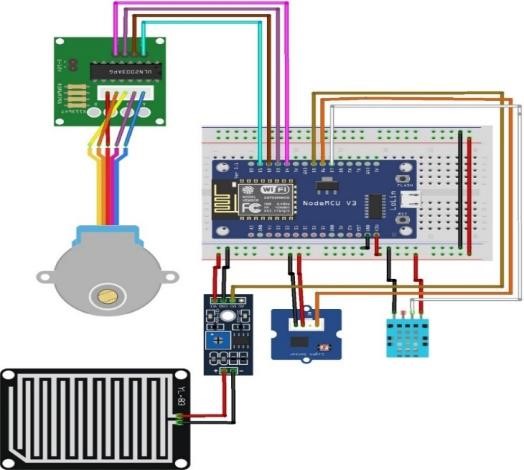
#### 4.3.2 Perancangan Elektronik

Pada perancangan elektronik alat kanopidisediakan rancangan perangkat keras dan rancangan sistem perkabelan. Adapun komponen-komponen utama dari alat kanopi otomatis ini dapat dilihat pada Gambar 4.8 dan Gambar 4.9.



### Gambar 4.8 Rancangan Perangkat Keras

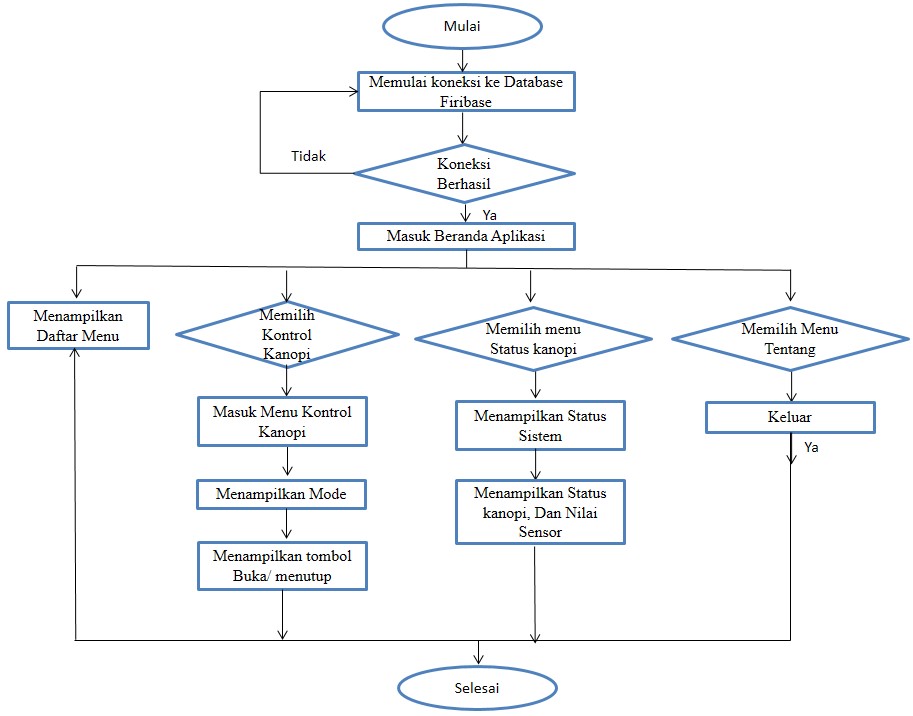
*NodeMCU ESP8266* sebagai pengendali sensor yang digunakan. Sensor LDR (*Light Dependent Resistor*) digunakan untuk mendeteksi nilai cahaya yang masuk. Sensor *DHT11* digunakan untuk mengukur suhu. Sensor Hujan berfungsi untuk mendeteksi adanya air atau tidak. *Motor stepper* berfungsi untuk membuka dan menutup kanopi. *Accsys* merupakan aplikasi yang berfungsi untuk mengontrol dan mengendalikan kanopi. *Power bank* berfungsi sebagai sumber daya.



### Gambar 4.9 Perancangan SistemPerkabelan

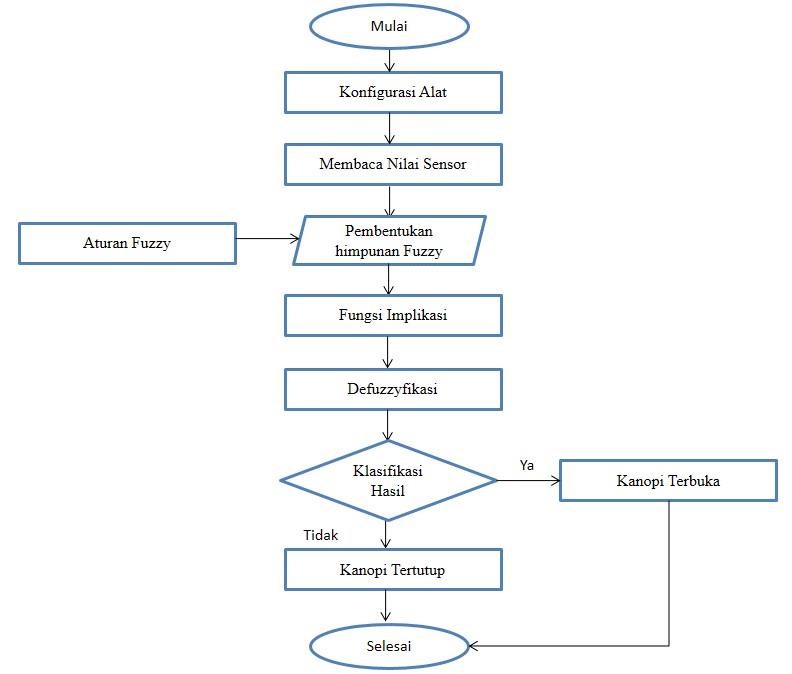
#### 4.3.3 Perancangan *Sofware*

Gambar 4.10 menjelaskan *flowchart* sistem *mobile* yang dimulai dari user menghubungkan koneksi ke *firebase*, setelah berhasil maka *user* akan masuk ke beranda aplikasi. Pada beranda akan menampilkan daftar yang berada pada pojok kiri aplikasi. pada tampilan daftar menu terdapat pilihan yaitu kontrol kanopi, status kanopi, menu tentang dan keluar. Pada saat *user* memilih kontrol kanopi akan menampilkan *mode* apakah user ingin menggunakan *mode* manual atau otomatis, di bawah menu *mode* menampilkan tombol buka dan menutup apakah *user* ingin membuka atau menutup atap kanopi. Pada saat *user* memilih menu status kanopi akan menampilkan status sistem yaitu menampilkan status kanopi, dan nilai dari sensor mengirim ke aplikasi. Selanjutnya menu tentang terdapat informasi yang mengembangkan aplikasi dan cara mengoperasikan aplikasi.



### Gambar 4.10 *Flowchart* Sistem *Mobile*

Gambar 4.11 menjelaskan *flowchart* alat kanopi otomatisyang dimulai dari alat mengkonfigurasi lalu membaca nilai sensor cahaya LDR (*Light Dependent Resistor*) dan sensor suhu *DHT11.* Pada aturan *fuzzy* menunjukkan bagaimana suatu sistem beroperasi. Pada pembentukan himpunan *fuzzy* yaitu *fuzzifikasi* adalah proses mengubah suatu satuan masukan dari bentuk tegas (*crisp*) menjadi *fuzzy* (variabel linguistik). Pada fungsi implikasi mencari nilai minimum dari setiap himpunan *fuzzy.* Pada *Deffuzyfikasi* tujuannya untuk mengklasifikasikan hasil dari pembacaan sensor apakah suhu yang didapat suhu dingin dan cahaya normal maka kanopi akan terbuka, namun jika klasifikasi hasil yang didapat panas atau hujan maka kanopi tidak terbuka tetapi kanopi akan tertutup.



### Gambar 4.11 *Flowchart* Alat Kanopi Otomatis

#### 4.4 Perancangan Antarmuka (*Interface*)

Perancangan *user interface* adalah tahapan pembuatan antarmuka yang akan digunakan pada pembangunan aplikasi *Accsys*. Ada lima bagian yaitu *form* beranda*, form* daftar menu,  *form* kontrol kanopi,*form* status kanopi dan  *form* tentang.

##### 4.4.1 *Form* beranda

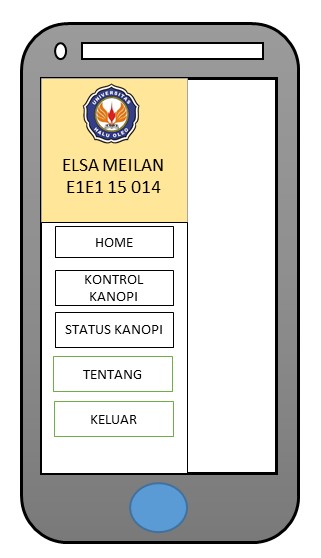
*Form* beranda adalah tampilan pertama yang akan digunakan pada perangkat *mobile android*. Pada *form* beranda berisi selamat datang, logo aplikasi dan *form* daftar menu.



### Gambar 4.12 *Form* Beranda

#### 4.4.2 Form Daftar Menu

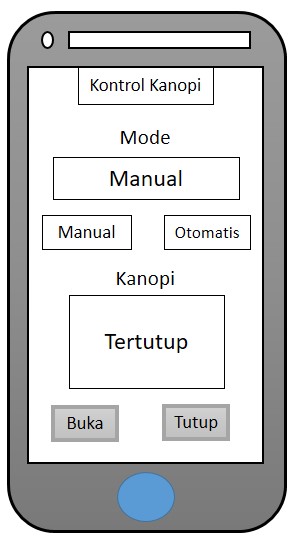
Pada *form* daftar menu berisi beberapa menu yaitu menu beranda, menu kontrol kanopi, menu status kanopi, menu tentang dan menu keluar. Pada *form* ini *user* dapat memilih daftar menu yang diinginkan.



### Gambar 4.13 *Form* Daftar Menu

#### 4.4.3 *Form* Kontrol Kanopi

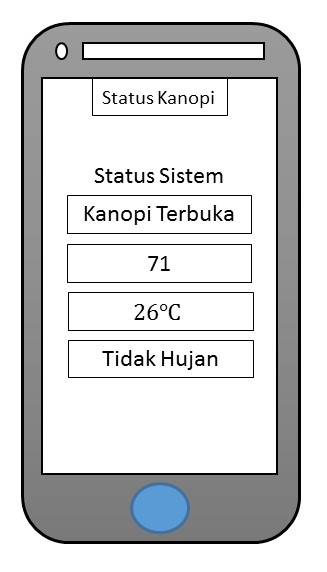
Pada *form* kontrol kanopi berisi beberapa menu yaitu menu mode status, manual dan otomatis dan juga tombol buka untuk membuka dan tombol tutup untuk menutup.



### Gambar 4.14 *Form* Kontrol Kanopi

#### 4.4.4 *Form* Status Kanopi

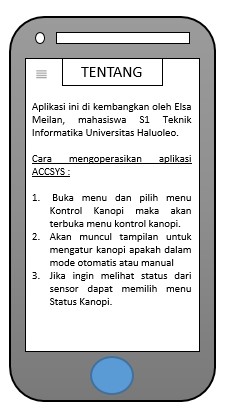
Pada *form* status kanopi berisi beberapa menu status sistem yaitu status kanopi terbuka, nilai cahaya, nilai suhu, dan status hujan dan tidak hujan.



### Gambar 4.15 *Form* Status Kanopi

#### 4.4.5 *Form* tentang

Pada *form* tentang berisi nama yang mengembangkan aplikasi dan cara mengoperasikan aplikasi.



### Gambar 4.16 *Form* Menu Tentang

**4.5 Hitungan Manual *Fuzzy* Tsukamoto**

**KASUS :**

1. Suatu rumah memiliki alat membantu pekerjaan manusia yang canggih yaitu kanopi otomatis. Alat ini akan terbuka dan menutup secara otomatis berdasarkan sensor suhu dan cahaya. Suhu terbesar mencapai 35℃, dan suhu terkecil sampai 20℃. Cahaya matahari yang mempengaruhi terbesar mencapai 100 dan terkecil 1.

Apakah kanopi terbuka jika suhu 32℃ dan cahaya 50.

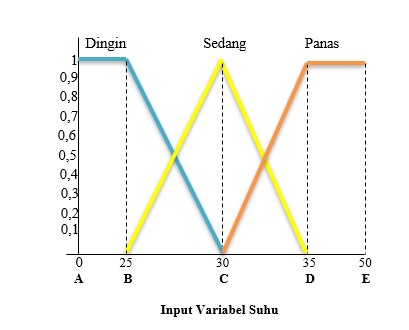
|  |  |
| --- | --- |
|  |  |
| [R1] | *IF* suhu Dingin *AND* cahaya Terang *THEN* kanopi Terbuka |
| [R2] | *IF* suhu Dingin *AND* cahaya Normal *THEN* kanopi Terbuka |
| [R3] | *IF* suhu Dingin *AND* cahaya Gelap *THEN*  kanopi Tertutup |
| [R4] | *IF* suhu Sedang *AND* cahaya Terang *THEN* kanopi Terbuka |
| [R5] | *IF* suhu Sedang *AND* cahaya Normal *THEN*  kanopi Terbuka |
| [R6] | *IF* suhu Sedang *AND* cahaya Gelap *THEN* kanopi Tertutup |
| [R7] | *IF* suhu Panas *AND* cahaya Terang *THEN* kanopi Tertutup |
| [R8] | *IF* suhu Panas *AND* cahaya Normal *THEN*  kanopi Tertutup |
| [R9] | *IF* suhu Panas *AND* cahaya Gelap *THEN* kanopi Tertutup |

Penyelesaian:

# Tabel 4.3 Pembentukan Himpunan *Fuzzy*

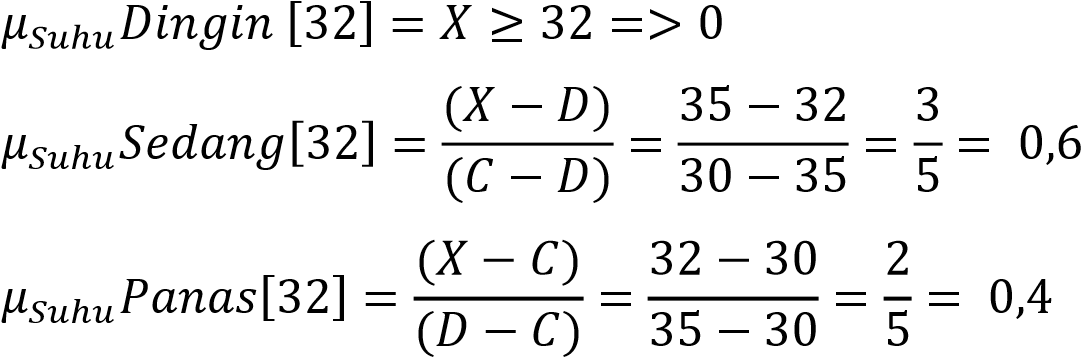
|  |  |  |  |
| --- | --- | --- | --- |
| NO | Variabel | Himpunan | Parameter |
| 1 | Suhu | Dingin  Sedang  Panas | 25℃ − 30℃  25℃ − 35℃  30℃ − 35℃ |
| 2 | Cahaya | Gelap  Normal  Terang | 20 − 40  30 − 70  60 − 80 |
| 3 | Kanopi | Tertutup  Terbuka | 0  1 |

1. Himpunan *fuzzy* Variabel Suhu, pada variabel suhu didefinisikan tiga himpunan *fuzzy* yaitu dingin, sedang, panas. Variabel suhu menggunakan bentuk kurva trapesium dan kurva segitiga, Gambar himpunan *fuzzy* untuk variabel suhu ditunjukkan pada Gambar 4.17.

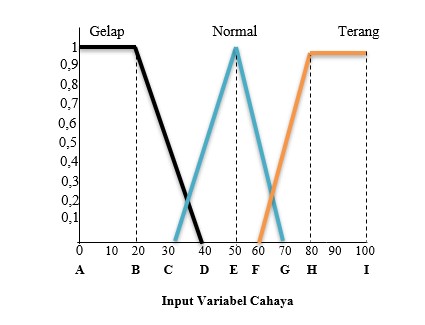


### Gambar 4.17 Himpunan *Fuzzy* Variabel Suhu

Kita bisa mencari nilai keanggotaannya:

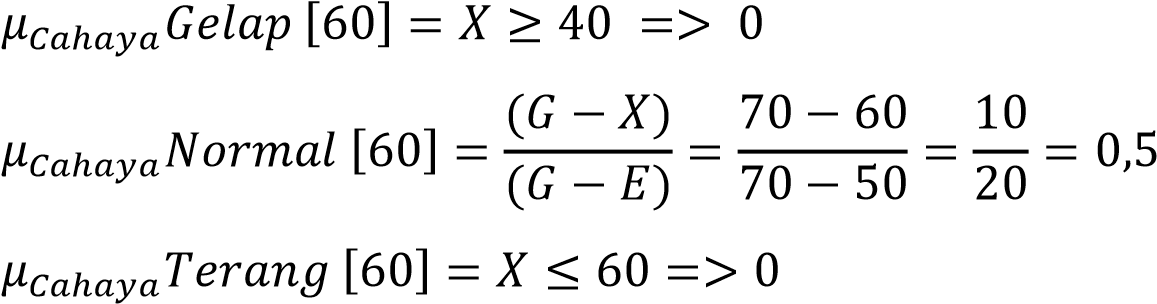


2. Himpunan *fuzzy* Variabel Cahaya, pada variabel cahaya didefinisikan tiga himpunan *fuzzy* yaitu gelap, normal, terang. Variabel cahaya menggunakan bentuk bentuk kurva trapesium dan kurva segitiga, Gambar himpunan *fuzzy* untuk variabel cahaya ditunjukkan pada Gambar 4.18.

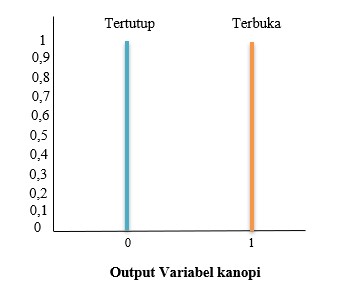


### Gambar 4.18 Himpunan *Fuzzy* Variabel Cahaya

Kita bisa mencari nilai keanggotaannya:



3. Himpunan *fuzzy* Variabel Kanopi, pada variabel kanopi didefinisikan dua himpunan *fuzzy* yaitu terbuka dan tertutup. Gambar himpunan *fuzzy* untuk variabel kanopi ditunjukkan pada Gambar 4.19.



### Gambar 4.19 Himpunan *Fuzzy* Variabel Kanopi

[R1] IF suhu Dingin AND cahaya Terang THEN kanopi Terbuka

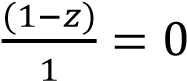
𝛼 − 𝑝𝑟𝑒𝑑𝑖𝑘𝑎𝑡1 = 𝜇𝑆𝑢ℎ𝑢𝐷𝑖𝑛𝑔𝑖𝑛 ∩ 𝜇𝐶𝑎ℎ𝑎𝑦𝑎𝑇𝑒𝑟𝑎𝑛𝑔

#### = min (𝜇𝑆𝑢ℎ𝑢𝐷𝑖𝑛𝑔𝑖𝑛[32],𝜇𝐶𝑎ℎ𝑎𝑦𝑎𝑇𝑒𝑟𝑎𝑛𝑔[50])

= min (0 , 0)

= 0

Lihat himpunan kanopi terbuka:

,  𝑧1 = 1

[R2] IF suhu Dingin AND cahaya Normal THEN kanopi Terbuka

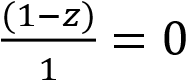
𝛼 − 𝑝𝑟𝑒𝑑𝑖𝑘𝑎𝑡2 = 𝜇𝑆𝑢ℎ𝑢𝐷𝑖𝑛𝑔𝑖𝑛 ∩ 𝜇𝐶𝑎ℎ𝑎𝑦𝑎𝑁𝑜𝑟𝑚𝑎𝑙

= min (𝜇𝑆𝑢ℎ𝑢𝐷𝑖𝑛𝑔𝑖𝑛[32],𝜇𝐶𝑎ℎ𝑎𝑦𝑎𝑁𝑜𝑟𝑚𝑎𝑙[50])

= min (0; 0,5)

= 0

Lihat himpunan kanopi terbuka:

  𝑧2 = 1

[R3] IF suhu Dingin AND cahaya Gelap THEN kanopi Tertutup

𝛼 − 𝑝𝑟𝑒𝑑𝑖𝑘𝑎𝑡3

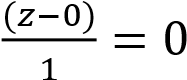
#### = 𝜇𝑆𝑢ℎ𝑢𝐷𝑖𝑛𝑔𝑖𝑛 ∩ 𝜇𝐶𝑎ℎ𝑎𝑦𝑎𝐺𝑒𝑙𝑎𝑝

= min (𝜇𝑆𝑢ℎ𝑢𝐷𝑖𝑛𝑔𝑖𝑛[32],𝜇𝐶𝑎ℎ𝑎𝑦𝑎𝐺𝑒𝑙𝑎𝑝[50])

= min (0; 0)

= 0

Lihat himpunan kanopi tertutup:

  𝑧3 = 0

[R4] IF suhu Sedang AND cahaya Terang THEN kanopi Terbuka

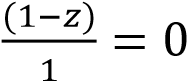
𝛼 − 𝑝𝑟𝑒𝑑𝑖𝑘𝑎𝑡4= 𝜇𝑆𝑢ℎ𝑢𝑆𝑒𝑑𝑎𝑛𝑔 ∩ 𝜇𝐶𝑎ℎ𝑎𝑦𝑎𝑇𝑒𝑟𝑎𝑛𝑔

= min (𝜇𝑆𝑢ℎ𝑢𝑆𝑒𝑑𝑎𝑛𝑔[32], 𝜇𝐶𝑎ℎ𝑎𝑦𝑎𝑇𝑒𝑟𝑎𝑛𝑔[50])

= min (0,6; 0)

= 0

Lihat himpunan kanopi terbuka:

  𝑧4 = 1

[R5] IF suhu Sedang AND cahaya Normal THEN kanopi Terbuka

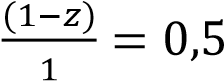
𝛼 − 𝑝𝑟𝑒𝑑𝑖𝑘𝑎𝑡5 = 𝜇𝑆𝑢ℎ𝑢𝑆𝑒𝑑𝑎𝑛𝑔 ∩ 𝜇𝐶𝑎ℎ𝑎𝑦𝑎𝑁𝑜𝑟𝑚𝑎𝑙

= min (𝜇𝑆𝑢ℎ𝑢𝑆𝑒𝑑𝑎𝑛𝑔[32], 𝜇𝐶𝑎ℎ𝑎𝑦𝑎𝑁𝑜𝑟𝑚𝑎𝑙[50])

= min (0,6; 0,5)

= 0,5

Lihat himpunan kanopi terbuka:

  𝑧5 = 0,5

[R6] IF suhu Sedang AND cahaya Gelap THEN kanopi Tertutup

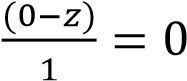
𝛼 − 𝑝𝑟𝑒𝑑𝑖𝑘𝑎𝑡6 = 𝜇𝑆𝑢ℎ𝑢𝑆𝑒𝑑𝑎𝑛𝑔 ∩ 𝜇𝐶𝑎ℎ𝑎𝑦𝑎𝐺𝑒𝑙𝑎𝑝

= min (𝜇𝑆𝑢ℎ𝑢𝑆𝑒𝑑𝑎𝑛𝑔[32], 𝜇𝐶𝑎ℎ𝑎𝑦𝑎𝐺𝑒𝑙𝑎𝑝[50])

= min (0,6; 0)

= 0

Lihat himpunan kenopi tertutup :

  𝑧6 = 0

[R7] IF suhu Panas AND cahaya Terang THEN kanopi Tertutup

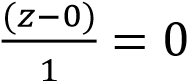
𝛼 − 𝑝𝑟𝑒𝑑𝑖𝑘𝑎𝑡7 = 𝜇𝑆𝑢ℎ𝑢𝑃𝑎𝑛𝑎𝑠 ∩ 𝜇𝐶𝑎ℎ𝑎𝑦𝑎𝑇𝑒𝑟𝑎𝑛𝑔

= min (𝜇𝑆𝑢ℎ𝑢𝑃𝑎𝑛𝑎𝑠[32], 𝜇𝐶𝑎ℎ𝑎𝑦𝑎𝑇𝑒𝑟𝑎𝑛𝑔[50])

= min (0,4; 0)

= 0

Lihat himpunan kanopi tertutup:

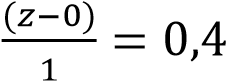
  𝑧7 = 0

[R8] IF suhu Panas AND cahaya Normal THEN kanopi Tertutup

𝛼 − 𝑝𝑟𝑒𝑑𝑖𝑘𝑎𝑡8 = 𝜇𝑆𝑢ℎ𝑢𝑃𝑎𝑛𝑎𝑠 ∩ 𝜇𝐶𝑎ℎ𝑎𝑦𝑎𝑁𝑜𝑟𝑚𝑎𝑙

= min (𝜇𝑆𝑢ℎ𝑢𝑃𝑎𝑛𝑎𝑠[32], 𝜇𝐶𝑎ℎ𝑎𝑦𝑎𝑁𝑜𝑟𝑚𝑎𝑙[50])

Lihat himpunan kanopi tertutup:= min (0,4; 0, 5)

= 0,4

 𝑧8 = 0,4

[R9] IF suhu Panas AND cahaya Gelap THEN kanopi Tertutup

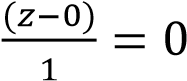
𝛼 − 𝑝𝑟𝑒𝑑𝑖𝑘𝑎𝑡9 = 𝜇𝑆𝑢ℎ𝑢𝑃𝑎𝑛𝑎𝑠 ∩ 𝜇𝐶𝑎ℎ𝑎𝑦𝑎𝐺𝑒𝑙𝑎𝑝

= min (𝜇𝑆𝑢ℎ𝑢𝑃𝑎𝑛𝑎𝑠[32], 𝜇𝐶𝑎ℎ𝑎𝑦𝑎𝐺𝑒𝑙𝑎𝑝[50])

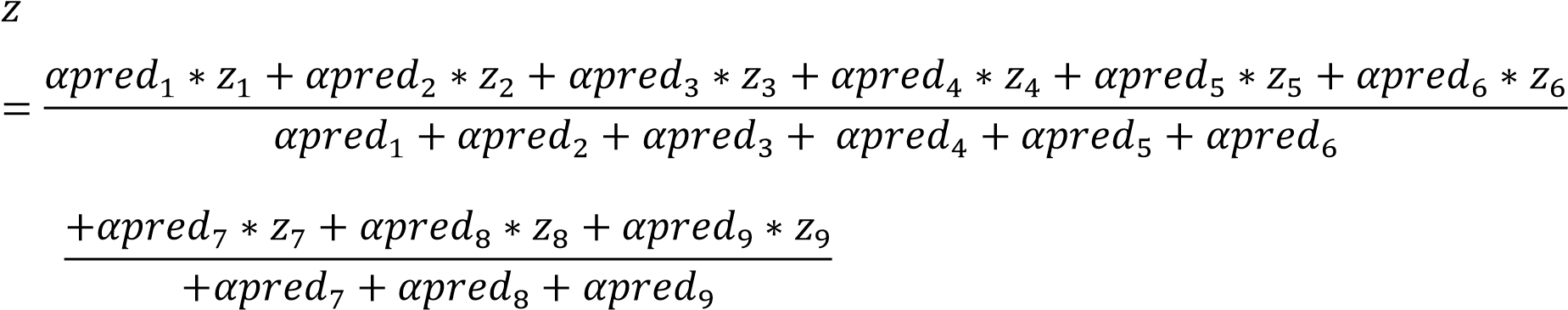
= min (0,4; 0)

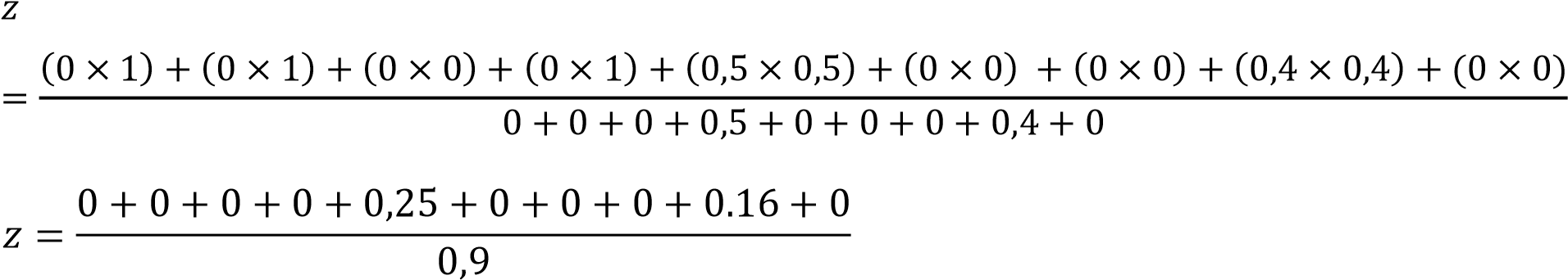
= 0

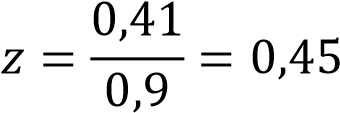
Lihat himpunan kanopi tertutup :

  𝑧9 = 0

Dari sini kita dapat mencari berapakah nilai z :







Karena nilai z = 0,45 mendekati 0 maka kanopi tertutup.

### DAFTAR PUSTAKA

Ambarita, J., P, R. A., & Wibowo, A. S. (2019). *RANCANG BANGUN PROTOTIPE SMARTHOME BERBASIS INTERNET OF THINGS ( IoT ) MENGGUNAKAN APLIKASI BLYNK DENGAN MODUL ESP 8266 DESIGN SMARTHOME PROTOTYPE BASED ON IOT USING BLYNK*

*APPLICATION WITH THE ESP MODULE 8266 internet technology and other communication media , the growing human need for technology , so more room access, house User + Smartphone Server 1 . Motor DC*. *6*(2), 3006–3013.

Andi, J. (2015). Pembangunan Aplikasi Child Tracker Berbasis Assisted – Global Positioning System ( a-Gps ) Dengan Platform Android. *Jurnal Ilmiah Komputer Dan Informatika (KOMPUTA)*, *1*(1), 1–8. Retrieved from elib.unikom.ac.id/download.php?id=300375.

Dawud, E. I. (2016). Sistem Monitoring Lampu Penerangan Jalan Umum Menggunakan Mikrokontroler Arduino Dan Sensor Ldr Dengan Notifikasi Sms. *Jurnal Teknik Elektro, Universitas Mercu Buana*, *7*(2), 101–105.

Eko Ihsanto, M. (2016). Rancang Bangun Kendali Gordeng Dengan Saklar Lampu Otomatis Berbasis Smartphone Android. *Jurnal Teknik Elektro*, *6*(1), 28–37.

Handaru, A. A., Afroni, M. J., Basuki, B. M., Elektro, M. T., Program, D., Teknik, S., & Malang, U. I. (n.d.). *Rancang Bangun Alat Pendeteksi Hujan Otomatis Menggunakan Modul Gsm Berbasis Mikrokontroler Atmega 328P*. 25–30.

Microcontroller, A. (2018). *Jurnal Keteknikan dan Sains (JUTEKS) – LPPM UNHAS Vol. 1, No.1, Juni 2018 23*. *1*(1), 23–31.

Puspita, E. S., & Yulianti, L. (2016). Perancangan Sistem Peramalam Cuaca Berbasis Logika Fuzzy. *Media Infotama*, *12*(1), 1–10.

Putro, M. D., Elektro, J. T., Teknik, F., Sam, U., & Manado, R. (2016). *ANDROID PADA RUMAH PINTAR*. (3).

Ratna, A., Yendri, D., Kasoep, W., Amelia, K., & Arifnur, A. A. (2017). Prototipe Sistem Prakiraan Cuaca Berdasarkan Suhu Dan Kelembapan Dengan Metode Logika Fuzzy Dan Backpropagation Berbasis Mikrokontroler. *Jurnal Teknik Komputer Amik*, *3*(November), 29–42.

Roihan, A., Asep Damyati, M., Sistem Komputer, J., Raharja, S., & Jenderal Sudirman, J. (2018). Prototipe Automatic Air Filtration Memanfaatkan Mikrokontroler ATmega328 Sebagai Air Quality Control. *Jurnal Informatika:*

*Jurnal Pengembangan IT Poltek Tegal*, *3*(1), 15–19.

Sains, J. T., Gita, Y. T., Wiguna, A. S., Harianto, W., Informatika, P. T., & Malang, U. K. (2019). *Implementasi iot pada penanganan listrik padam menggunakan mikrokontroller*. *1*(2), 1–7.

Seminar, P., & Penelitian, N. (n.d.). *BERBASIS ARDUINO DAN APLIKASI BLYNK*.

SRI SUPATMI. (2010). Pengaruh Sensor Ldr Terhadap Pengontrolan Lampu. *Majalah Ilmiah UNIKOM*, *8*(2), 175–180. Retrieved from

http://jurnal.unikom.ac.id/\_s/data/jurnal/v08-n02/volume-82-artikel5.pdf/pdf/volume-82-artikel-5.pdf

Sunaryo, E., & Atmaja, R. (2017). *ATAP OTOMATIS TANAMAN HIDROPONIK BERBASIS MIKROKONTROLER ATMEGA 89s52*. *3*(1), 95–104.

Ode, L., Chumaidi, I., & Muchlis, N. F. (2018). *Rancang bangun dan monitoring alat jemur pakaian berbasis web menggunakan metode naive bayes*. *4*(1), 87– 96.

Waspada, I., Studi, P., Informatika, T., & Diponegoro, U. (n.d.). *center of area ),*. *2*, 27–38.

Wicaksono, H. (2015). *Analisa Performansi dan Robustness Beberapa Metode Tuning Kontroler PID pada Motor DC*. *4*(2), 70–78.

Wicaksono, M. F. (2017). Implementasi Modul Wifi Nodemcu Esp8266 Untuk Smart Home. *Jurnal Teknik Komputer Unikom*, *6*(1), 1–6.

Yan, M., Adiptya, E., & Wibawanto, H. (2013). Sistem Pengamatan Suhu Dan Kelembaban Pada Rumah Berbasis Mikrokontroller ATmega8. *Jurnal Teknik Elektro Unnes*, *5*(1), 15–17.

Kontrol, D. A. N. R. (n.d.). *Vol.6, No. 2*. *6*(2), 187–202. Dapat, D. C. (n.d.).  *Prinsip Kerja Motor Stepper*.

Pembelajaran, H., Stepper, M., Bagian, T., Stepper, M., Aktuator, P., Stepper, M., & Stepper, M. (n.d.). *Mekatronika Modul 9 Motor Stepper*.

Rozikin, K., Kom, S., Kom, M., Sasmoko, D., Eng, M., Suhartono, D. B., … Kom, M. (n.d.). *Data Penulis : Penanggung Jawab : Ketua Sekolah Tinggi Elektronika dan Komputer Pemimpin Redaksi : Unang Achlison , S . T , M . Kom Mitra Bestari : Prof . YL Sukestiyarno M . S , Ph . D ( Universitas Negeri Semarang ) Sekretaris Redaksi : Purwanto , S . Kom Dewan Redaksi :*

*Sulartopo , S . Pd . M . Kom Desain Grafis : Setyo Adi Nugroho , S . E , M . Kom Alamat Redaksi : Lembaga Penelitian dan Pengabdian Masyarakat Sekolah Tinggi Elektronika dan Komputer Jl . Majapahit No . 605 Semarang Telp . 024-6723456*.

Herawan, A., & Judianto, C. T. (n.d.). *OPTIMALISASI AKURASI ANTENA PENJEJAK SATELIT ORBIT RENDAH MENGGUNAKAN MOTOR STEPPER HYBRID 2 FASA ( OPTIMIZATION OF LOW EARTH ORBIT*

*TRACKING ANTENNA USING STEPPER MOTOR HYBRID 2-PHASE )*. 1– 12.

Ilmiah, J., Komputa, I., Volume, E., Issn, A., & Juansyah, A. (2015). *PEMBANGUNAN APLIKASI CHILD TRACKER BERBASIS ASSISTED – GLOBAL POSITIONING SYSTEM ( A-GPS ) DENGAN PLATFORM ANDROID Jurnal Ilmiah Komputer dan Informatika ( KOMPUTA )*.

Najib, M., Sunaryono, D., Yuhana, L., Penuntun, A. A., & Bergerak, P. (2013). *Analisis dan Perancangan Aplikasi Penuntun Jalan dengan Perintah Suara Berbahasa Indonesia pada Perangkat Bergerak Berbasis Android*. *2*(2).