

**IMPLEMENTASI KONTROL LOGIKA FUZZY PADA SISTEM  
PENDINGIN DISPENSER AIR MINUM MENGGUNAKAN TEC12715  
TERINTEGRASI DENGAN IOT**

**SKRIPSI**

**TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL**

Diajukan untuk memenuhi persyaratan  
memperoleh gelar Sarjana Teknik



**CANDRA SABDANA NUGROHO**  
**NIM. 165060301111008**

**UNIVERSITAS BRAWIJAYA**  
**FAKULTAS TEKNIK**  
**MALANG**

**2020**



**LEMBAR PENGESAHAN**  
**IMPLEMENTASI KONTROL LOGIKA FUZZY PADA SISTEM PENDINGIN**  
**DISPENSER AIR MINUM MENGGUNAKAN TEC12715 TERINTEGRASI**  
**DENGAN IOT**  
**SKRIPSI**

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Ditujukan untuk memenuhi persyaratan  
memperoleh gelar Sarjana Teknik



**CANDRA SABDANA NUGROHO**

**NIM. 165060301111008**

Skripsi ini telah direvisi dan disetujui oleh dosen pembimbing  
pada tanggal 22 Juni 2020

Dosen Pembimbing I

Goegoes Dwi Nusantoro, S.T., M.T.  
NIP. 19711013 200604 1 001

Dosen Pembimbing II

M. Aziz Muslim, S.T., M.T., Ph.D.  
NIP. 19741203 200012 1 001

Mengetahui,

Ketua Jurusan Teknik Elektro



Prof. I. Hadiyanto, S.T., M.T., Ph.D., IPM.  
NIP. 19730520 200801 1 013

JUDUL SKRIPSI:

IMPLEMENTASI KONTROL LOGIKA *FUZZY* PADA SISTEM PENDINGIN  
DISPENSER AIR MINUM MENGGUNAKAN TEC12715 TERINTEGRASI DENGAN  
IOT

Nama Mahasiswa : Candra Sabdana Nugroho  
NIM : 165060301111008  
Program Studi : S1 Teknik Elektro  
Konsentrasi : Teknik Kontrol

KOMISI PEMBIMBING:

Ketua : Goegoes Dwi Nusantoro, S.T., M.T.



.....

Anggota : M. Aziz Muslim, S.T., M.T., Ph.D.



.....

KOMISI PENGUJI:

Dosen Penguji 1 : Dr. Ir. Mochammad Rusli, Dipl.Ing. disetujui tanggal 19 Juni 2020

Dosen Penguji 2 : Dr. Tri Nurwati, S.T., M.T. disetujui tanggal 19 Juni 2020

Tanggal Ujian : 5 Juni 2020

SK Penguji : 953 Tahun 2020



*Teruntuk Ibu, Ayah, Adik tercinta  
Terima kasih atas segalanya*

## PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Skripsi ini adalah asli dari pemikiran saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Malang, 10 Juni 2020

Mahasiswa,



**CANDRA SABDANA NUGROHO**

**NIM. 165060301111008**



## PENGANTAR

*Bismillahirrohmanirrohim. Alhamdulillah*, puji syukur penulis panjatkan kehadiran Allah `Azza Wa Jalla yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Implementasi Kontrol Logika Fuzzy Pada Sistem Pendingin Dispenser Air Minum Menggunakan TEC12715 Terintegrasi Dengan IoT” dengan baik. Tak lepas shalawat serta salam tercurahkan kepada junjungan kita Nabi Muhammad SAW yang telah menjadi suri tauladan bagi yang mengharapkan rahmat dan hidayah-Nya .

Penulis menyadari bahwa selama masa studi dan penyusunan skripsi ini tidak lepas dari bantuan, bimbingan serta dorongan dari berbagai pihak. Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar – besarnya kepada:

- Allah `Azza Wa Jalla, karena ketika saya meminta kekuatan, Allah memberi saya kesulitan untuk membuat saya kuat. Ketika saya meminta kebijaksanaan, Allah memberi saya masalah untuk diselesaikan. Ketika saya meminta nikmat, Allah memberi saya peluang. Ketika saya tidak meminta apa-apa untuk diri saya, Allah selalu memberi yang saya butuhkan. Allah Maha Baik.
- Bapak dan Ibu tercinta, Bapak Agus Tjatur Satriyo Nugroho dan Srianah yang selalu memberikan dukungan, semangat, kasih sayang dan do’a yang tiada akhir. Tak lupa juga dua adikku tercinta dan tercantik, yaitu Elsa Dela Nugrahani dan Elvina Faiqa Nugrahani yang juga selalu memberikan semangat dan dukungan.
- Bapak Ir. Hadi Suyono, S.T., M.T., Ph.D., IPM. selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya.
- Ibu Ir. Nurussa’adah, M.T. selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya.
- Ibu Rahmadwati, S.T.,M.T., Ph.D selaku Ketua Program Studi S1 Jurusan Teknik Elektro Universitas Brawijaya yang telah banyak memberikan pengarahan dan nasehat kepada saya selama menjalani proses kegiatan akademik dan non-akademik.
- Bapak Goegoes Dwi Nusantoro, S.T., M.T. selaku Dosen Pembimbing I atas segala bimbingan, pengarahan, saran, dan kritik yang telah diberikan selama proses pengerjaan skripsi.
- Bapak Muhammad Aziz Muslim, S.T., M.T., Ph.D., selaku KKDK Teknik Kontrol sekaligus Dosen Pembimbing II yang telah banyak memberikan kesempatan, waktu, nasehat, pengarahan, motivasi, saran dan masukan dalam penyusunan skripsi.

- Bapak Eka Maulana, S.T., M.T., M.Eng. selaku *Counselor of IEEE UB Student Branch* yang telah sabar dalam memberikan wawasan, arahan, serta masukan selama saya mengikuti beberapa kegiatan perlombaan karya ilmiah dalam lingkup Nasional maupun Internasional.
- Bapak Ir. Wijono, M.T., Ph.D. dan bapak Waru Djuriatno, S.T., M.T. selaku dosen yang telah banyak memberikan pengetahuan, pemikiran dan kemampuan mengenai ilmu teknik elektro diluar kegiatan akademik perkuliahan.
- Segenap dosen dan staff administrasi Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.
- Segenap pihak MODENA INDONESIA yang telah mengadakan kompetisi inovasi produk, sehingga penulis dapat mengikuti perlombaan dan mendapatkan judul dari kompetisi tersebut.
- Sahabat-sahabat satu kontrakan (Farid, Mahdi, Najib, Luqman, Aiman, Yusril, Yusuf, dan Alfaridzi) yang telah memberikan semangat, hiburan, dan saran selama pengerjaan skripsi.
- Teman-teman asisten laboratorium Sistem Kontrol, Agung, Adrian, Amell, Ikma, Rizal, mas-mbak angkatan 2015 dan 2014, adik-adik angkatan 2017, adik-adik angkatan 2018 dan Mbak Eka atas pengalaman, bimbingan, perhatian, dan semangat yang telah diberikan.
- Keluarga besar Divisi Mikrokontroller dan Internet of Things (IoT) atas bimbingan, pengalaman, kebersamaan, dan semangat yang telah diberikan. Kehadiran dan usaha yang telah kalian lakukan, sangat berharga untuk kemajuan divisi ini.
- Keluarga besar RKIM-UB (Riset dan Karya Ilmiah Mahasiswa Universitas Brawijaya) atas pengalaman, bimbingan, kebersamaan, dan semangat yang telah diberikan. Banyak sekali ilmu *softskill* dalam karya ilmiah, pengetahuan dan pengalaman dalam berorganisasi yang telah saya dapatkan.
- Arif Reza Dwi Kurniawan, Bayu Abi Pamungkas, Ashar Bagus Satryo, Mohammad Mufti Fajar, Alif Deni Kuncoro yang telah banyak membantu dalam penyusunan skripsi dengan berbagai dukungan, baik dukungan moril maupun materiil.
- Ariq, Taufiqurrahman, Amell, Aidil, Ribut, Zahro, Aulia Muhammad, Faisal Himawan, Mugni, Arya, Didit, dan teman-teman yang tidak mampu saya sebutkan satu per satu yang mau menemani saya disaat saya membutuhkan banyak sekali dukungan, terutama dukungan moril. Terimakasih sebanyak-banyaknya.
- Keluarga besar MOSFET 2016 untuk dukungan dan semangat.



- Teruntuk KAMU, siapapun itu yang namanya telah tertulis di *Lauhul Mahfudz*, yang selalu menjadi motivasi untuk mengejar gelar sarjana, memotivasi untuk selalu belajar, dan mempersiapkan kehidupan yang lebih baik nantinya bersamamu.
- Keluarga Besar Mahasiswa Teknil Elektro Universitas Brawijaya.
- Seluruh teman-teman dan semua pihak, yang tidak mungkin bagi penulis untuk mencantumkan satu-persatu yang telah memberikan bantuan serta dukungan baik secara langsung maupun tidak langsung atas penyusunan skripsi ini.

Dalam penyusunan skripsi ini, penulis menyadari bahwa masih terdapat kekurangan karena keterbatasan dan kendala yang dialami penulis selama pengerjaan skripsi. Oleh, karena itu penulis mengharapkan saran dan kritik yang bersifat membangun penulisan maupun kesempurnaan skripsi ini. Semoga penulisan penelitian ini dapat bermanfaat dikemudian hari dan dapat dikembangkan lebih lanjut.

Malang, 30 Mei 2020

Penulis



## RINGKASAN

**Candra Sabdana Nugroho**, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Mei 2020, Implementasi Kontrol Logika *Fuzzy* Pada Sistem Pendingin Dispenser Air Minum Menggunakan TEC12715 Terintegrasi Dengan IoT, Dosen Pembimbing: Goegoes Dwi Nusantoro dan Muhammad Aziz Muslim.

Kontrol Logika *Fuzzy* menyediakan metodologi yang bersifat formal untuk merepresentasikan, memanipulasi, dan mengimplementasikan pengetahuan heuristik manusia tentang bagaimana mengontrol sebuah sistem berdasarkan sistem basis aturan (*rule based system*) untuk pengambilan keputusan. Aturan (*fuzzy rule*) dibuat dengan menggunakan pernyataan “*IF-THEN*” untuk memperoleh data *crisp*. Pengujian dilakukan dengan cara memberikan variasi nilai masukan atau *setpoint* suhu ke sistem. Variasi nilai *setpoint* yang diberikan yaitu, suhu 21°C, 19°C, dan 17°C. *Output* dari Kontrol Logika *Fuzzy* adalah mengatur level PWM pada *driver* BTS7960 yang akan memberikan tegangan *input* pada peltier TEC12715 untuk mendinginkan air pada tangki air dispenser. Dilakukan pengujian sistem tanpa gangguan dan dengan gangguan terhadap ketiga variasi *setpoint* tersebut. Ketika diberikan *setpoint* suhu 21°C, diperoleh nilai *settling time* sekitar 56,76 menit, *error steady state* sekitar 0,47%. Namun, saat diberikan *setpoint* 19°C, diperoleh nilai *settling time* sekitar 123,25 menit, *error steady state* sekitar 0,57%. Kemudian saat diberikan *setpoint* 17°C, diperoleh nilai *settling time* sekitar 403,46 menit, *error steady state* sekitar 0,29%. Pada pengujian dengan diberikan gangguan pada *setpoint* suhu 21°C, 19°C, dan 17°C, diperoleh nilai *recovery time* dari masing-masing *setpoint*, yaitu sekitar 37,73 menit, 83,13 menit, dan 509,83 menit. Sehingga dapat disimpulkan bahwa implementasi Kontrol Logika *Fuzzy* (KLF) pada dispenser air minum dan pembuatan sistem bekerja dengan spesifikasi yang diinginkan.

**Kata Kunci:** *driver* BTS7960, Kontrol Logika *Fuzzy*, peltier TEC12715, tangki air dingin,

## SUMMARY

**Candra Sabdana Nugroho**, Department of Electrical Engineering, Faculty of Engineering University of Brawijaya, May 2020, *Implementation of Fuzzy Logic Control in Drinking Water Dispenser Cooling System Using TEC12715 Integrated with IoT, Academic Supervisor: Goegoes Dwi Nusantoro dan Muhammad Aziz Muslim.*

*Fuzzy Logic Control provides a formal methodology for representing, manipulating, and implementing human heuristic knowledge about how to control a system based on a rule-based system for decision making. Fuzzy rules are made by using the statement "IF-THEN" to obtain crisp data. System testing is done by providing variations in input values or temperature setpoint to the system. Variations in temperature setpoint values are 21°C, 19°C and 17°C. The output of the Fuzzy Logic Control is to regulate the PWM level of the BTS7960 motor driver which will provide the input voltage to the TEC12715 peltier to cool the water in the water dispenser tank. System testing is carried out without disturbance and with disturbance to the three variations of the setpoint. When given a temperature setpoint of 21°C, a settling time value of around 56.76 minutes was obtained, the error steady state was around 0.47%. However, when given a setpoint of 19°C, a settling time value of around 123.25 minutes was obtained, the error steady state was around 0.57%. Then when given a 17°C setpoint, a settling time value of around 403.46 minutes was obtained, the error steady state was around 0.29%. In testing with a given setpoint of temperature 21°C, 19°C, and 17°C, the recovery time values of each setpoint were obtained, which were around 37.73 minutes, 83.13 minutes, and 509.83 minutes. So it can be concluded that the implementation of Fuzzy Logic Control (KLF) on drinking water dispensers and manufacturing systems works to the desired specifications.*

**Keywords:** *BTS7960 driver motor, cool water tank, Fuzzy Logic Control, peltier TEC12715*

DAFTAR ISI

**PENGANTAR**..... i

**DAFTAR ISI** ..... vi

**DAFTAR GAMBAR** ..... ix

**DAFTAR TABEL** ..... xi

**BAB I PENDAHULUAN** ..... 1

    1.1. Latar Belakang ..... 1

    1.2. Rumusan Masalah ..... 2

    1.3. Batasan Masalah ..... 3

    1.4. Tujuan ..... 3

    1.5. Manfaat ..... 3

**BAB II TINJAUAN PUSTAKA** ..... 5

    2.1. Dispenser Air ..... 5

    2.2. Komponen Dispenser Air ..... 6

        2.2.1. Casing Dispenser ..... 6

        2.2.2. Selang ..... 7

        2.2.3. Tangki ..... 7

        2.2.4. Sensor Suhu ..... 8

        2.2.5. Elemen Peltier ..... 9

        2.2.6. Keran ..... 10

        2.2.7. Kabel ..... 10

        2.2.8. Saklar ..... 11

    2.3. Mikrokontroler ESP 32 ..... 11

    2.4. *Driver Direct Current* (DC) BTS7960 ..... 12

    2.5. *Analog to Digital Converter* ..... 13

    2.6. *Pulse Width Modulation* ..... 14

    2.7. Kontroler ..... 15

    2.8. Kontrol Logika Fuzzy ..... 16



|   |  |           |
|---|--|-----------|
| 2.8.1.                                  | Himpunan <i>Fuzzy</i> .....                            | 16        |
| 2.8.2.                                  | Fuzzifikasi.....                                       | 18        |
| 2.8.3.                                  | Defuzzifikasi.....                                     | 19        |
| 2.9                                     | <i>Firebase Realtime Database</i> .....                | 22        |
| <b>BAB III METODE PENELITIAN.....</b>   |  | <b>23</b> |
| 3.1.                                    | Perancangan Blok Diagram.....                          | 23        |
| 3.2.                                    | Spesifikasi Sistem.....                                | 24        |
| 3.3.                                    | Studi Literatur.....                                   | 24        |
| 3.4.                                    | Karakterisasi Blok.....                                | 25        |
| 3.4.1.                                  | Karakterisasi <i>Driver</i> BTS7960.....               | 25        |
| 3.4.2.                                  | Karakterisasi Sensor <i>Thermistor</i> NTC.....        | 27        |
| 3.5.                                    | Pembuatan <i>Hardware</i> .....                        | 29        |
| 3.6.                                    | Perancangan Kontrol Logika Fuzzy.....                  | 30        |
| 3.6.1.                                  | Variabel Masukan dan Keluaran.....                     | 30        |
| 3.6.2.                                  | Fungsi Keanggotaan Masukan dan Keluaran.....           | 30        |
| 3.6.3.                                  | Perancangan Aturan Fuzzy.....                          | 35        |
| 3.6.4.                                  | Defuzzyfikasi.....                                     | 35        |
| 3.6.5.                                  | <i>Surface Fuzzy</i> .....                             | 35        |
| 3.7.                                    | Perancangan Antarmuka <i>Internet of Things</i> .....  | 37        |
| 3.7.1                                   | Tujuan Perancangan.....                                | 37        |
| 3.7.2                                   | Blok Diagram Komunikasi Antarmuka.....                 | 37        |
| 3.7.3                                   | Langkah Perancangan Antarmuka.....                     | 37        |
| 3.8.                                    | Pengujian Alat.....                                    | 40        |
| 3.8.1                                   | Pengujian Tiap Blok.....                               | 40        |
| 3.8.2                                   | Pengujian Keseluruhan Sistem.....                      | 40        |
| 3.8.3                                   | Rancangan Pengujian.....                               | 40        |
| <b>BAB IV HASIL DAN PEMBAHASAN.....</b> |  | <b>42</b> |
| 4.1                                     | Pengujian Keseluruhan Sistem Tanpa Gangguan.....       | 42        |
| 4.1.1                                   | Pengujian Sistem dengan Suhu <i>Setpoint</i> 21°C..... | 42        |



|   |  |           |
|---|--|-----------|
| 4.1.2                                   | Pengujian Sistem dengan Suhu <i>Setpoint</i> 19°C.....                         | 43        |
| 4.1.3                                   | Pengujian Sistem dengan Suhu <i>Setpoint</i> 17°C.....                         | 44        |
| 4.2                                     | Pengujian Keseluruhan Sistem dengan Gangguan.....                              | 45        |
| 4.2.1                                   | Pengujian Sistem dengan Suhu <i>Setpoint</i> 21°C yang Diberikan Gangguan..... | 45        |
| 4.2.2                                   | Pengujian Sistem dengan Suhu <i>Setpoint</i> 19°C yang Diberikan Gangguan..... | 46        |
| 4.2.3                                   | Pengujian Sistem dengan Suhu <i>Setpoint</i> 17°C yang Diberikan Gangguan..... | 46        |
| 4.3                                     | Antarmuka Aplikasi ( <i>Software</i> ) <i>Internet of Things</i> .....         | 47        |
| <b>BAB V KESIMPULAN DAN SARAN .....</b> |  | <b>50</b> |
| 5.1                                     | Kesimpulan.....  | 50        |
| 5.2                                     | Saran.....   | 50        |
| <b>DAFTAR PUSTAKA .....</b>             |  | <b>52</b> |
| <b>LAMPIRAN</b>                         |  |           |



DAFTAR GAMBAR

Gambar 2.1 Skematik saluran air pada tabung air panas dispenser air galon bawah .....6

Gambar 2.2 Casing dispenser .....7

Gambar 2.3 Selang pada dispenser .....7

Gambar 2.4 Tangki air dingin .....8

Gambar 2.5 Sensor NTC .....9

Gambar 2.6 Elemen pendingin elektrik .....10

Gambar 2.7 Solenoid valve .....10

Gambar 2.8 Contoh kabel pada dispenser .....11

Gambar 2.9 Saklar yang digunakan pada dispenser .....11

Gambar 2.10 Mikrokontroler ESP32 .....12

Gambar 2.11 BTS7960 Driver 43 A H-Bridge Driver PWM .....12

Gambar 2.12 Sinyal PWM .....14

Gambar 2.13 Kontrol loop terbuka .....15

Gambar 2.14 Kontrol loop tertutup .....15

Gambar 2.15 Arsitektur Kontroler Fuzzy .....16

Gambar 2.16 Keanggotaan fungsi yang menunjukkan ketidakpastian pada pembacaan voltage (crisp) .....18

Gambar 2.17 Perbandingan himpunan Fuzzy atau pembacaan Fuzzy: (a) himpunan Fuzzy dan pembacaan (crisp); (b) himpunan Fuzzy dan pembacaan Fuzzy .....18

Gambar 2.18 Bentuk Diskrit dari Himpunan Fuzzy A .....19

Gambar 2.19 Himpunan Lambda-cut untuk  $\lambda = 1, 0,9, 0,6, 0,3, 0^+, 0$  .....20

Gambar 2.20 Karakteristik keluaran dari kontrol Fuzzy: (a) keluaran pertama Fuzzy; (b) keluaran kedua Fuzzy; (c) gabungan keduanya .....21

Gambar 2.21 Metode “Weighted average” .....22

Gambar 3.1 Blok diagram sistem .....24

Gambar 3.2 Hubungan nilai PWM terhadap tegangan keluaran driver BTS7960 .....27

Gambar 3.3 Perbandingan suhu antara infrared thermometer dengan serial monitor pembacaan sensor suhu thermistor NTC .....29

Gambar 3.4 Skema pembuatan perangkat keras (hardware) .....30

Gambar 3.5 Fungsi keanggotaan masukan fuzzy error .....31

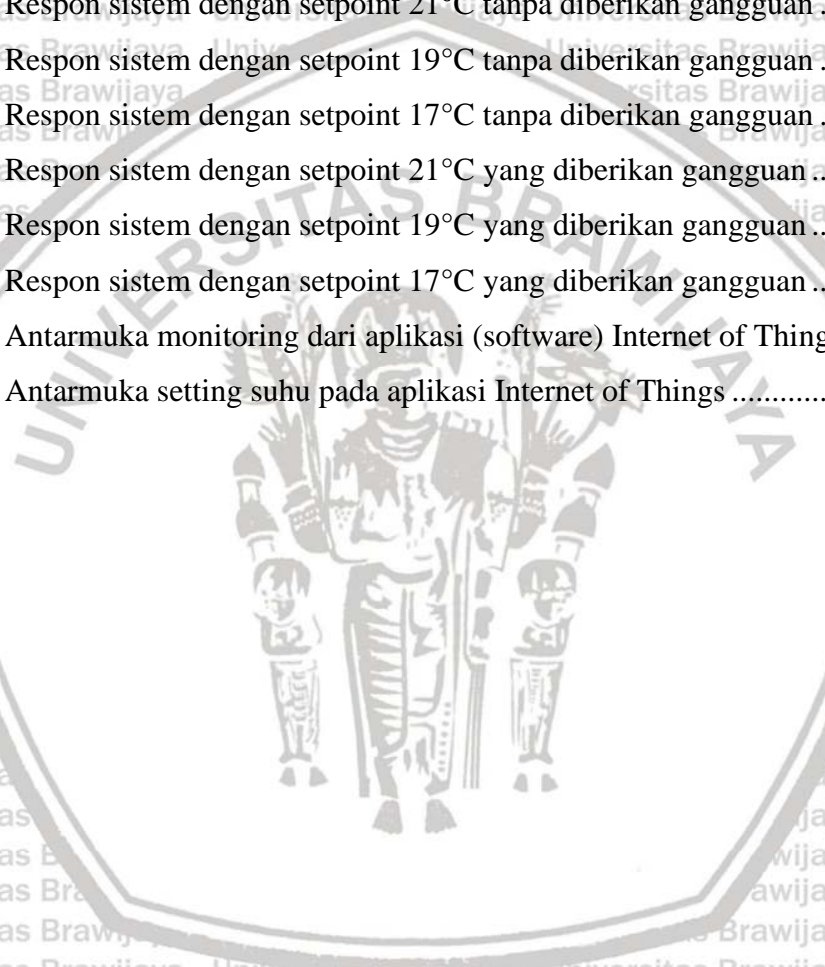
Gambar 3.6 Fungsi keanggotaan masukan fuzzy delta error .....32

Gambar 3.7 Fungsi keanggotaan fuzzy keluaran .....34

Gambar 3.8 Tampilan Surface fuzzy awal dari sistem .....36



|  |    |
|--|----|
| Gambar 3.9 Tampilan surface fuzzy yang terbaik dari sistem .....                 | 36 |
| Gambar 3.10 Blok diagram komunikasi antarmuka.....                               | 37 |
| Gambar 3.11 Halaman awal MIT app inventor.....                                   | 38 |
| Gambar 3.12 Halaman untuk mengatur desain tampilan aplikasi android.....         | 38 |
| Gambar 3.13 Tampilan block untuk mengatur mekanisme kerja aplikasi android.....  | 39 |
| Gambar 3.14 Tampilan menu untuk menyelesaikan pembuatan aplikasi android.....    | 39 |
| Gambar 4.1 Respon sistem dengan setpoint 21°C tanpa diberikan gangguan.....      | 42 |
| Gambar 4.2 Respon sistem dengan setpoint 19°C tanpa diberikan gangguan.....      | 43 |
| Gambar 4.3 Respon sistem dengan setpoint 17°C tanpa diberikan gangguan.....      | 44 |
| Gambar 4.4 Respon sistem dengan setpoint 21°C yang diberikan gangguan.....       | 45 |
| Gambar 4.5 Respon sistem dengan setpoint 19°C yang diberikan gangguan.....       | 46 |
| Gambar 4.6 Respon sistem dengan setpoint 17°C yang diberikan gangguan.....       | 46 |
| Gambar 4.7 Antarmuka monitoring dari aplikasi (software) Internet of Things..... | 47 |
| Gambar 4.8 Antarmuka setting suhu pada aplikasi Internet of Things.....          | 48 |





DAFTAR TABEL

|              |    |
|--------------|----|
| Tabel 1..... | 26 |
| Tabel 2..... | 28 |
| Tabel 3..... | 35 |
| Tabel 4..... | 45 |
| Tabel 5..... | 47 |





## BAB I PENDAHULUAN

### 1.1. Latar Belakang

Air merupakan unsur atau elemen yang terpenting di dalam tubuh manusia. Dalam kehidupan sehari-hari pada saat ini semua orang membutuhkan air sebagai salah satu kebutuhan mereka. Kebutuhan air ini diperlukan untuk menunjang kehidupan, antara lain dalam kondisi yang layak diminum tanpa mengganggu kesehatan (Depkes RI, 2006). Selain itu, meningkatnya aktivitas pembangunan dan jumlah penduduk berakibat pada peningkatan kebutuhan akan air bersih, baik yang diperoleh secara tak langsung melalui proses pengolahan maupun secara langsung. Kualitas dan kuantitas air menjadi faktor utama bagi masyarakat dalam beraktifitas sehari-hari.

Salah satu bentuk menjalankan pola hidup sehat adalah dengan cara meminum air putih dan hal itu harus dilakukan sedini mungkin. Air putih adalah salah satu minuman yang memiliki banyak manfaat bagi tubuh. *Institute of Medicine* mengungkapkan bahwa kebutuhan asupan air orang dewasa pria kurang lebih 3 liter atau setara dengan 13 gelas, sedangkan untuk wanita dewasa antara 2,2 liter atau setara dengan 9 gelas setiap harinya. Salah satu bentuk pentingnya air putih adalah manusia dapat bertahan berminggu-minggu tidak makan, akan tetapi manusia tidak bisa hidup apabila tidak minum dalam waktu beberapa hari saja. Terkadang mengkonsumsi air minum sering disepelekan begitu saja, apalagi seorang pasien yang sedang menderita penyakit karena kekurangan air minum, mereka kurang memperhatikan konsumsi air minum dengan baik (Mar'atus Arifah, 2016).

Pemenuhan kebutuhan air minum masyarakat saat ini sangat bervariasi. Mulai dari mengambil air minum dari berbagai sumber air yaitu air sungai, air tanah baik dari sumur dangkal ataupun dalam dan juga air yang diproduksi Perusahaan Daerah Air Minum (PDAM) setempat yang dimasak dahulu sebelum dikonsumsi maupun air minum kemasan (AMDK), karena tergolong lebih praktis dan dianggap lebih higienis (Putra dan Prawiroedjo, 2017). Hal ini dapat diketahui dari bertumbuhnya konsumsi masyarakat terhadap AMDK yang meningkat dari tahun 2014 ke tahun 2015 yaitu sebesar 11,3% (Tribunnews, 2015). AMDK ini sendiri diproduksi oleh industri air minum tertentu melalui proses otomasi dan disertai dengan pengujian kualitas sebelum sampai ketangan masyarakat. Dalam proses tersebut, industri air minum menggunakan teknologi canggih demi menjaga mutu kualitas produk mereka.

Seiring berjalannya waktu, perkembangan teknologi saat ini sangat pesat yang dipengaruhi oleh kebutuhan dan gaya hidup manusia. Ketergantungan manusia terhadap suatu perangkat elektronik dalam melakukan sesuatu menjadikan sebuah alat elektronik harus dapat memenuhi keinginan manusia. Hal tersebut juga berlaku pada peralatan rumah tangga seperti dispenser air minum.

Dispenser air minum merupakan alat penyedia air minum dengan menggunakan galon berkapasitas 19 liter yang fungsinya menyediakan air panas dan dingin siap diminum. Pada umumnya dispenser mempunyai fungsi yang sama yaitu memanaskan dan mendinginkan air minum sampai suhu tertentu saja tanpa bisa diatur sendiri suhunya (Benny, *et.al*, 2015). Namun beberapa dispenser saat ini sudah terdapat pengaturan suhu namun masih berupa pilihan suhu tetap yang disediakan dari produsen dispenser tersebut. Tetapi ada kendala dalam pengaturan suhu dari pabrikan ini, yaitu suhu yang keluar kurang sesuai dengan suhu yang telah diatur. Padahal dalam kenyataannya manusia memerlukan dispenser melakukan lebih dari itu. Manusia menginginkan pengaturan suhu yang fleksibel agar dapat mengatur sesuai keinginan mereka.

Dari beberapa uraian diatas, dibutuhkan suatu dispenser yang memiliki sistem pengendalian yang mampu menjaga performa dan kestabilan suhu dingin yang fleksibel. Selain itu terdapat preferensi nilai suhu yang dapat diatur oleh pengguna sesuai kebutuhan. Perancangan dispenser ini menggunakan kontrol logika *fuzzy*, dimana kontroler ini diharapkan dapat menjaga suhu sesuai dengan preferensi maupun suhu yang diatur oleh pengguna secara tepat dan minim dari offset. Kontrol logika *fuzzy* ditentukan oleh persamaan logika bukan dari persamaan diferensial kompleks dan berasal dari pemikiran yang mengidentifikasi serta mengambil keuntungan dari Grayness antara dua ekstrem (Ilyas, 2013). Pengaturan suhu dan preferensi dispenser ini nantinya akan diintegrasikan dengan sistem IoT (*Internet of Things*). Sehingga sistem dapat melakukan *monitoring* kondisi dispenser dan memberikan *setpoint* suhu sesuai kebutuhan secara *realtime* dan dapat diakses dimana saja. Kemudahan ini diharapkan mampu mewujudkan *Smart Home* yang digadangkan mendukung program Revolusi Industri 4.0.

## 1.2. Rumusan Masalah

Berdasarkan masalah yang telah dijelaskan pada latar belakang, maka dapat disimpulkan rumusan masalah sebagai berikut:

1. Bagaimana perancangan sistem pengendali suhu pada tangki air dingin dispenser menggunakan kontrol logika *fuzzy* sehingga mampu memberikan variasi nilai suhu air dingin bagi pengguna?
2. Bagaimana pengaruh kontrol logika *fuzzy* terhadap keluaran suhu dispenser?
3. Bagaimana penerapan sistem *Internet of Things* pada dispenser?

### 1.3. Batasan Masalah

Dengan mengacu pada permasalahan yang telah dirumuskan, maka hal-hal yang berkaitan dengan penelitian akan diberi batasan sebagai berikut:

1. Alat dirancang dengan mikrokontroler ESP 32.
2. Alat dirancang agar dapat mengetahui dan sesuai dengan prinsip kerja dispenser dengan pengaturan suhu dengan kontrol logika *fuzzy*.
3. Alat dirancang untuk mengetahui respon pencapaian suhu air pada dispenser.
4. Perancangan ini hanya sebatas kontroler logika *fuzzy* untuk mendinginkan air pada dispenser saja dan tidak memanaskan air saat *setpoint* dinaikkan.
5. Penerapan *Internet of Things* pada perancangan ini hanya sampai pada *monitoring* dan mengontrol *setpoint* suhu air menggunakan aplikasi *android*.
6. Perancangan ini hanya sebatas kontroler fuzzy untuk pendingin pada dispenser bukan pada pemanas dispenser.
7. Proses penurunan suhu air dipengaruhi oleh kondisi suhu diluar tangki air dingin, dengan demikian titik minimal suhu yang dicapai berbeda-beda.

### 1.4. Tujuan

Adapun tujuan dari penelitian ini adalah:

1. Penelitian ini bertujuan untuk mengimplementasikan kontrol logika fuzzy untuk diterapkan pada dispenser.
2. Penelitian ini bertujuan untuk memberi suhu air pada dispenser yang sesuai dengan keinginan pengguna.
3. Penelitian ini bertujuan untuk memberikan kemudahan dalam *monitoring* dan mengontrol kondisi dispenser menggunakan sistem *Internet of Things*.

### 1.5. Manfaat

Penelitian ini turut berkontribusi dalam pengembangan teknologi di bidang elektronika yaitu sebagai pengembangan dari dispenser konvensional saat ini yang hanya menggunakan kontrol on/off. Selain itu berkontribusi dalam program Revolusi Industri 4.0 yang menerapkan sistem *Internet of Things*.



## BAB II

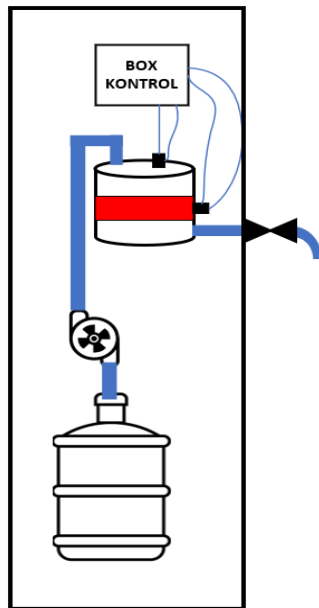
### TINJAUAN PUSTAKA

#### 2.1. Dispenser Air

Menurut Fery Ariyanto (2018), dispenser adalah sebuah alat yang dipergunakan untuk menyimpan air minum, menggantikan fungsi daripada alat rumah tangga sejenis yang sebelumnya sudah ada yaitu teko, ceret, termos, tetapi sebagai pengembangan dari alat penyimpan air biasa dispenser memiliki banyak kelebihan diantaranya daya tampung yang besar, hingga bisa menyimpan persediaan air dalam kapasitas yang banyak, selain itu dispenser juga ada yang memiliki fitur bisa membuat air menjadi panas atau dingin, sehingga di saat membutuhkan air panas atau air dingin tidak lagi harus memasak atau memasukkannya kedalam kulkas. Umumnya air panas yang dihasilkan mencapai suhu 85 derajat Celsius dan air dingin yang dihasilkan bisa mencapai suhu 15 derajat Celsius. Karena kelebihan-kelebihannya tersebut dispenser saat ini menjadi alat yang disukai dan banyak dipergunakan, baik di perumahan – perumahan, perkantoran, hotel, bahkan ditempat-tempat umum yang strategis dan layak untuk penempatannya, kita bisa temukan dispenser di sekolah, di rumah sakit, di Bank dan lain-lain.

Beberapa jenis dispenser

1. Dispenser biasa yaitu dispenser yang hanya berfungsi sebagai tempat menyimpan air saja, dalam pemakaiannya, sebuah gallon berisi air minum harus disimpan dalam posisi terbalik diatas dispenser, kemudian sebuah keran yang didesain khusus untuk dispenser akan bekerja mengeluarkan air jika keran tersebut di tekan.
2. Dispenser Hot and Normal yaitu dispenser yang memiliki elemen pemanas dan tidak memiliki mesin pendingin. Dispenser ini hanya dapat digunakan untuk memasak atau memanaskan air dan mengambil air biasa dari galon.
3. Dispenser Hot and Cold, yaitu dispenser yang dapat digunakan untuk memanaskan dan mendinginkan air. Apabila ingin memanaskan air, dispenser ini menggunakan pemanas (heater). Apabila ingin mendinginkan air, dispenser ini menggunakan prinsip ini menggunakan prinsip kerja elemen pendingin (fan atau refrigen).



Gambar 2.1 Skematik saluran air pada tabung air panas dispenser air galon bawah

Proses air mengalir dari galon yang bersuhu normal hingga sampai kedalam cangkir atau gelas yang bersuhu panas yaitu melalui beberapa komponen, mulai dari galon air dipompa menuju tabung pemanas dan kemudian air mengalir dalam keadaan panas melalui keran. Tabung pemanas merupakan tabung yang terbuat dari logam yang diselimuti oleh elemen pemanas, sehingga ketika air pada tabung pemanas sudah penuh, maka *thermostat* yang ada pada tabung pemanas akan bekerja. Temperatur tinggi yang dihasilkan *heater* diserap oleh air, setelah suhu air dalam tabung meningkat dan mencapai temperatur maksimum maka *thermostat* yang ada pada tabung akan memutuskan arus listrik sehingga *heater* akan mati.

## 2.2. Komponen Dispenser Air

Dikutip dari Safrudin(2017), komponen-komponen yang digunakan pada. Dispenser air antara lain: *Casing*, Selang, Tangki, *Thermostat*, *Electric Heater*, *Isolator*, Keran, Kabel, dan Saklar (*Switch*).

### 2.2.1. Casing Dispenser

*Casing* adalah komponen yang berfungsi membungkus komponen-komponen lain dalam satu *unit* dispenser air. *Casing* dispenser air terbuat dari plastik dan aluminium. Didalam *casing* ini terdapat banyak komponen pendukung dispenser. Gambar *casing* dapat dilihat pada Gambar 2.2.





Gambar 2.2 Casing dispenser

### 2.2.2. Selang

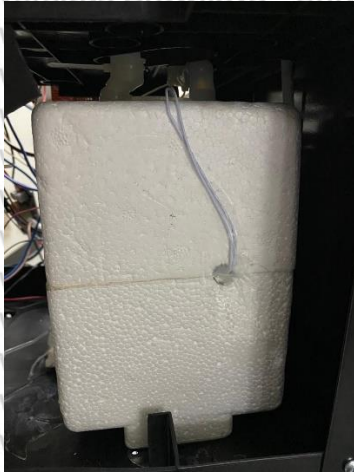
Selang adalah saluran atau komponen yang berfungsi membawa air dari tangki penampung ke tabung dan keran. Selang air terbuat dari plastik dan sebagian ada yang terbuat dari *stainless steel*. Selang dalam dispenser harus memiliki sertifikasi *food grade* agar air yang lewat didalamnya aman dari zat berbahaya yang terkandung dalam selang.



Gambar 2.3 Selang pada dispenser

### 2.2.3. Tangki

Tangki adalah komponen yang berfungsi menyimpan air dan sebagai wadah proses pemanasan atau pendinginan air. Tangki air Dispenser air terbuat dari *stainless steel* atau sering disebut baja tahan karat. *Stainless steel* merupakan baja paduan dengan unsur paduan dengan unsur utamanya *Cr* yang memiliki kandungan *Cr* minimal 12%. Dalam tangki ini terdapat elemen pemanas yang akan memanaskan air. Gambar tangki dapat dilihat pada Gambar 2.4.



Gambar 2.4 Tangki air dingin

### 2.2.4. Sensor Suhu

Sensor suhu ini berfungsi untuk membaca suhu air pada tangki. Dispenser umumnya menggunakan sensor suhu berupa *thermostat* ataupun NTC. *Thermostat* adalah komponen yang berfungsi mengatur kerja *heater*. *Thermostat* merupakan sensor temperatur yang bekerja berdasarkan prinsip bahwa sifat tahanan listrik berbagai material akan berubah terhadap perubahan temperatur. Terdapat perhitungan khusus agar mikrokontroler dapat mengkonversi dari pembacaan resistansi sensor ke nilai suhu. Perhitungan pertama yang digunakan adalah dengan rumus sederhana yang disebut persamaan dengan parameter B.

$$RT = R0 \times e^{B(\frac{1}{T} - \frac{1}{T0})} \quad (2-1)$$

$$T = \frac{1}{\frac{\ln(\frac{RT}{R0})}{B} + \frac{1}{T0}} \quad (2-2)$$

Keterangan:

e = logaritma natural

Ro = resistansi *thermistor* yang diukur pada suhu T0

B = koefisien konstanta yang tergantung pada karakteristik material

T = *temperature* yang dibaca

Material yang digunakan untuk sensor ini umumnya berupa semi konduktor.

Sedangkan NTC adalah *Negative Temperature Coefficient*. Prinsip kerja dari NTC ini adalah resistansinya berubah seiring perubahan suhu. Jika suhu semakin panas maka nilai resistansi sensor NTC akan semakin turun. Bentuk sensor NTC dapat dilihat pada

Gambar 2.5.



Gambar 2.5 Sensor NTC

### 2.2.5. Elemen Peltier

Elemen peltier atau pendingin termoelektrik (*thermoelectric cooler*) adalah alat yang dapat menimbulkan perbedaan suhu antara kedua sisinya jika dialiri arus listrik searah pada kedua kutub materialnya, dalam hal ini semikonduktor. Keramik Peltier atau lebih dikenal dengan lempengan Peltier adalah lempengan berbahan dasar keramik yang memiliki fungsi sangat unik. Peltier ini adalah modul *Thermo-Electric*, umumnya dibungkus oleh keramik tipis yang berisikan batang-batang *Bismuth Telluride* di dalamnya. Ketika disupply tegangan DC 12 volt-15 volt salah satu sisi akan menjadi panas dan sisi lainnya menjadi dingin. Mengapa bisa begitu? karena Peltier memiliki 2 bagian yang berbeda, yakni :

1. *Cool Side (Heat Absorbed)* yang bekerja menyerap kalor (panas) sehingga bagian ini merupakan lempengan yang dingin
2. *Hot Side (Heat Released)* yang bekerja melepas kalor (panas) sehingga bagian ini merupakan lempengan panas

Pendingin termoelektrik merupakan *solid state technology* yang bisa menjadi alternatif teknologi pendingin selain sistem *vapor compression* yang masih memanfaatkan refrigeran. Dibandingkan dengan teknologi kompresi uap yang menggunakan uap sebagai media penyerap kalor, teknologi pendingin termoelektrik relatif lebih ramah lingkungan, tahan lama dan dapat digunakan dalam skala besar dan kecil.

Perbedaan suhu antara kedua bagian adalah sekitar 30 derajat celcius. Sehingga apabila bagian Hot Side bersuhu 45 derajat C maka Cool Side akan bersuhu sekitar 15 derajat jadi semakin dingin Hot Side maka Cold Side akan semakin dingin pula dan bisa sampai dibawah 0 derajat celcius (Jenny, *et.al*, 2016).



Gambar 2.6 Elemen pendingin elektrik

### 2.2.6. Keran

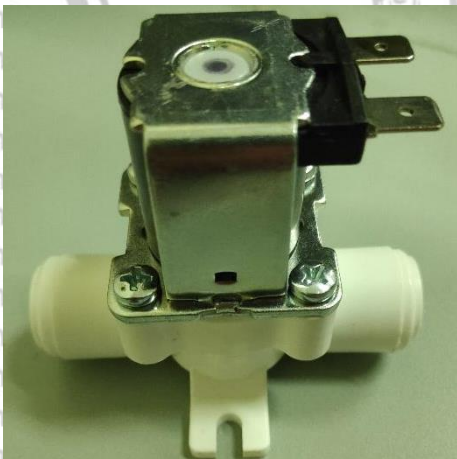
Keran adalah suatu alat untuk mengalirkan fluida yang dapat dibuka dan ditutup.

Berdasarkan jenis pengoperasiannya keran ini terdiri dari keran mekanik dan elektrik.

Keran mekanik akan mengalirkan air jika tuas pembuka diberikan gaya tekan atau tarik.

Sedangkan untuk keran elektrik (*solenoid valve*) prinsip kerjanya adalah ketika koil mendapat *supply* tegangan, katup listrik yang mempunyai koil sebagai penggerakannya akan berubah menjadi medan magnet sehingga menggerakkan piston pada bagian dalamnya dan menyebabkan membuka atau menutupnya aliran air (Kitoma Indonesia,

2019). Bentuk *solenoid valve* dapat dilihat pada Gambar 2.7.

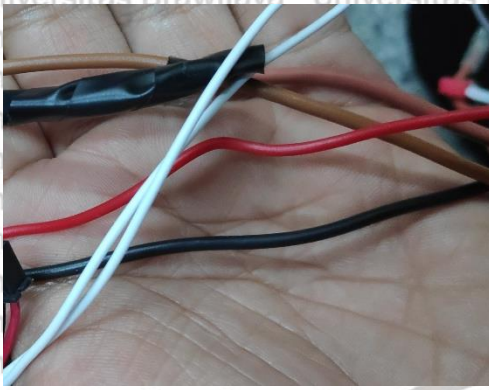


Gambar 2.7 Solenoid valve

### 2.2.7. Kabel

Kabel adalah konduktor yang bersifat sangat baik dialiri arus listrik. Kabel diklasifikasikan kedalam dua jenis yaitu kabel tunggal dan kabel jamak/serabut. Kabel ini berfungsi mengalirkan arus listrik kepada komponen-komponen elektrik yang ada dalam Dispenser agar komponen tersebut dapat bekerja sebagaimana mestinya. Bentuk

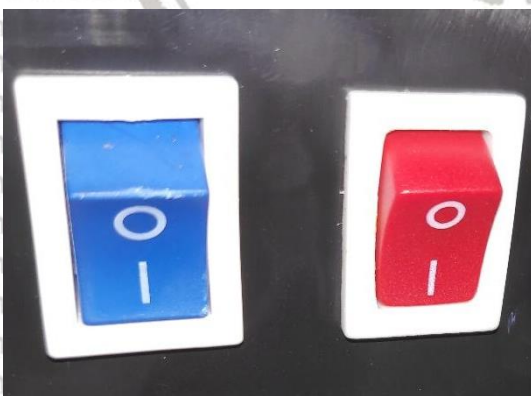
kabel dapat dilihat pada Gambar 2.8.



Gambar 2.8 Contoh kabel pada dispenser

### 2.2.8. Saklar

Dikutip dari penelitian sebelumnya (Handayani, 2015), *Switch*/saklar adalah komponen elektikal yang berfungsi untuk memberikan sinyal atau untuk memutuskan atau menyambungkan suatu sistem kontrol. *Switch* berupa komponen kontaktor mekanik yang digerakan karena suatu kondisi tertentu. *Switch* merupakan komponen yang mendasar dalam sebuah rangkaian listrik maupun rangkaian kontrol sistem. Komponen ini sederhana namun memiliki fungsi yang paling vital di antara komponen listrik yang lain. Jadi *switch*/saklar pada dasarnya adalah suatu alat yang dapat atau berfungsi menghubungkan atau memutuskan aliran listrik (arus listrik) baik itu pada jaringan arus listrik kuat maupun pada jaringan arus listrik lemah. Yang memebedakan saklar arus listrik kuat dan saklar arus listrik lemah adalah bentuknya kecil jika dipakai untuk peralatan elektronika arus lemah, demikian pula sebaliknya semakin besar saklar yang digunakan jika aliran arus listrik semakin besar.



Gambar 2.9 Saklar yang digunakan pada dispenser

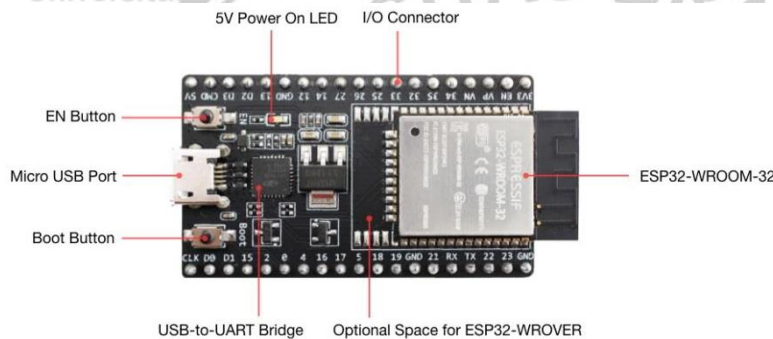
### 2.3. Mikrokontroler ESP 32

ESP adalah board berbasis mikrokontroler atau papan rangkaian elektronik yang dibuat oleh perusahaan *Espressif System* yang di dalamnya terdapat komponen utama yaitu sebuah *chip* mikrokontroler yang telah diintegrasikan dengan modul wifi. Mikrokontroler itu sendiri

adalah *chip* atau IC (*integrated circuit*) yang bisa diprogram menggunakan komputer.

Tujuan menanamkan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca *input*, memproses *input* tersebut dan kemudian menghasilkan *output* sesuai yang diinginkan. Jadi mikrokontroler bertugas sebagai otak yang mengendalikan proses *input*, dan *output* sebuah rangkaian elektronik.

Pada gambar 2.10 merupakan jenis ESP32, ESP32 adalah mikrokontroler yang dibuat sebagai penerus dari mikrokontroler ESP8266. Pada mikrokontroler ini sudah tersedia modul wifi dan bluetooth sehingga sangat mendukung untuk membuat sistem aplikasi Internet of Things. Modul ESP32 memiliki 18 ADC (Analog Digital Converter), 2 DAC, 16 PWM, 10 Sensor sentuh, 2 jalur antarmuka UART, pin antarmuka I2C, I2S, dan SPI.

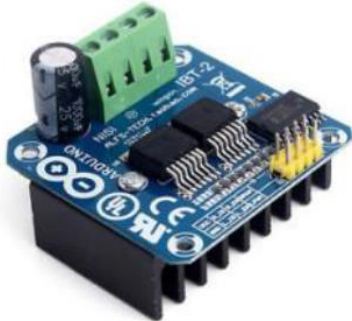


Gambar 2.10 Mikrokontroler ESP32

Sumber: esp\_32\_datasheet\_en.pdf

#### 2.4. Driver Direct Current (DC) BTS7960

Pada *driver* DC ini dapat mengeluarkan arus hingga 43A, dengan memiliki fungsi PWM. Tegangan sumber DC yang dapat diberikan antara 5.5V-27V, sedangkan tegangan *input* level antara 3.3V-5VDC, *driver* ini menggunakan rangkaian *full H-bridge* dengan IC BTS7960 dengan perlindungan saat terjadi panas dan arus berlebihan.



Gambar 2.11 BTS7960 Driver 43 A H-Bridge Driver PWM

Sumber: Abidatul A. (2017)

Pin konfigurasi dari penggunaan driver 43A H-Brige Drive PWM ini dapat dilihat berikut :

#### Detail Pin Input

1. RPWM = Input PWM Forward Level ,Aktif High
2. LPWM = Input PWM Reverse Level ,Aktif High
3. R\_EN = Input Enable Forward Driver, Aktif High
4. L\_EN = Input Enable Reverse Driver, Aktif High
5. R\_IS = Forward Drive ,Side current alarm output
6. L\_IS = Reverse Drive ,Side current alarm output
7. Vcc = +5 V Power Supply Mikrokontroler
8. Gnd = Gnd Power Supply Mikrokontroler

#### Detail Pin Output

1. W- = Di hubungkan ke Motor DC (V-)
2. W+= Di hubungkan ke Motor DC (V+)
3. B+ = Tegangan Input V+ Motor
4. B- = Tegangan Input V- Motor

### 2.5. Analog to Digital Converter

Dikutip dari silabus perkuliahan (Hariyanto, 2015), *Analog to Digital Converter* (ADC) adalah pengubah input analog menjadi kode – kode digital. ADC banyak digunakan sebagai pengatur proses industri, komunikasi digital dan rangkaian pengukuran/pengujian. Umumnya ADC digunakan sebagai perantara antara sensor yang kebanyakan analog dengan sistim komputer seperti sensor suhu, cahaya, tekanan/ berat, aliran dan sebagainya kemudian diukur dengan menggunakan sistim digital (komputer).

Secara umum, ADC (*Analog to Digital Converter*) memiliki 2 karakter prinsip, yaitu kecepatan *sampling* dan resolusi. Kecepatan *sampling* suatu ADC menyatakan seberapa sering sinyal analog dikonversikan ke bentuk sinyal digital pada selang waktu tertentu.

Kecepatan *sampling* biasanya dinyatakan dalam *sample per second* (SPS). Sementara resolusi ADC menentukan ketelitian nilai hasil konversi ADC.

Pada dunia mikrokontroler, ADC sudah terpasang atau *build in* di dalam *chip* sehingga jika sebuah mikrokontroler memiliki resolusi ADC sebesar 10 bit, ini berarti nilai ADC memiliki rentang nilai  $2^{10} = 1024$ . Hal tersebut artinya ADC akan memiliki rentang nilai antara 0 hingga 1024. Dalam kata lain, ini berarti sinyal *input* dapat dinyatakan dalam 1024

nilai diskrit. Prinsip kerja ADC adalah mengkonversi sinyal analog ke dalam bentuk besaran yang merupakan rasio perbandingan sinyal *input* dan tegangan referensi. Berikut merupakan persamaan yang digunakan untuk mengidentifikasi besar nilai ADC yang ditunjukkan pada persamaan (2-3) sebagai berikut:

$$ADC_{reading} = \frac{V_{measured}}{V_{system}} \times ADC_{resolution} \quad (2-3)$$

Keterangan:

$ADC_{reading}$  = nilai ADC pada yang terbaca

$ADC_{resolution}$  = nilai resolusi ADC

$V_{measured}$  = Tegangan yang terukur

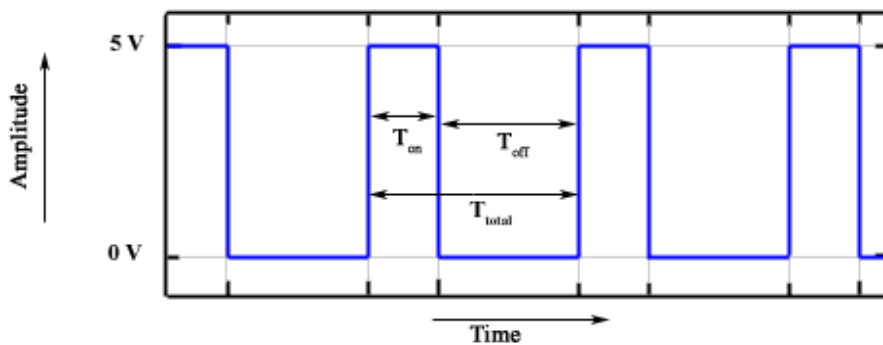
$V_{system}$  = Tegangan referensi

## 2.6. Pulse Width Modulation

*Pulse Width Modulation* (PWM) adalah salah satu jenis modulasi. Modulasi PWM dilakukan dengan cara mengubah lebar pulsa dari suatu pulsa data. Total 1 periode (T) pulsa dalam PWM adalah tetap, dan data PWM pada umumnya menggunakan perbandingan pulsa positif terhadap total pulsa. Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. *Duty cycle* adalah besarnya sinyal kontrol yang diberikan pada plant. *Duty cycle* dapat didefinisikan pada persamaan (2-4) sebagai berikut:

$$D = \frac{T_{on}}{T} \times 100\% \quad (2-4)$$

Dengan mengatur *duty cycle* akan diperoleh *output* yang diinginkan. Misal tegangan, maka tegangan keluaran dapat bervariasi dengan *duty-cycle* (*mengikuti besar  $T_{on}$* ). Untuk penjelasan lebih lanjut, berikut merupakan bentuk sinyal PWM secara umum yang ditunjukkan pada Gambar 2.12.



Gambar 2.12 Sinyal PWM

Sumber: diambil dari website (<http://www.arisulistiono.com/2010/02/pulse-width-modulation-pwm-pengenalan.html>)



## 2.7. Kontroler

Kontroler merupakan salah satu komponen sistem pengaturan yang berfungsi mengolah sinyal umpan balik dan sinyal masukan acuan (*setpoint*) atau sinyal *error* mejadi sinyal kontrol. Sinyal *error* disini adalah selisih antara sinyal umpan balik yang dapat berupa sinyal keluaran plant sebenarnya atau sinyal keluaran terukur dengan sinyal masukan acuan (*setpoint*). Selain itu analisis juga dilakukan untuk mengetahui bagaimana kecepatan sistem dalam merespon input, dan bagaimana peredaman terhadap adanya lonjakan (*overshoot*).

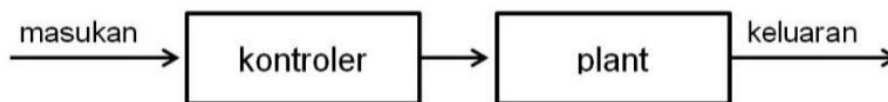
Suatu sistem dikatakan stabil apabila diberi gangguan maka sistem tersebut akan kembali ke keadaan *steady state* dimana *output* berada dalam keadaan tetap seperti tidak ada gangguan.

Sistem dikatakan tidak stabil jika *output* beresilasi terus-menerus ketika dikenai suatu gangguan.

Dalam sistem pengendalian terdapat 2 macam *loop*:

### 1. Pengendalian dengan *Loop* Terbuka

Sistem kontrol *loop* terbuka adalah sistem kontrol yang keluarannya tidak berpengaruh pada aksi pengontrolan. Jadi pada sistem kontrol *loop* terbuka, keluaran tidak diukur atau diumpkan balikan untuk dibandingkan dengan masukan. Tampilan sistem *loop* terbuka dapat dilihat pada Gambar 2.13.

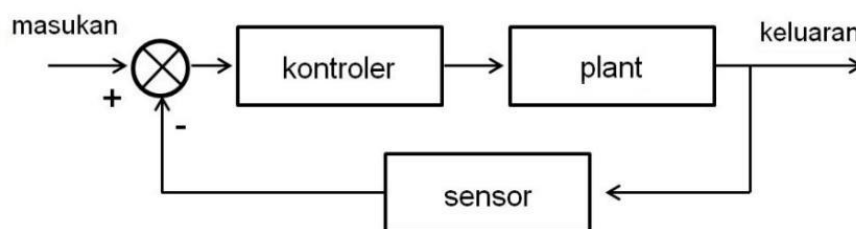


Gambar 2.13 Kontrol loop terbuka

Sumber: diambil dari website (<https://serbatelekomunikasi.wordpress.com>)

### 2. Pengendalian dengan *Loop* Tertutup

Sistem kontrol *loop* tertutup adalah sistem kontrol yang keluarannya mempunyai pengaruh langsung pada aksi pengontrolan. Sistem ini disebut juga sistem kontrol yang menggunakan umpan balik untuk memperkecil kesalahan sistem. Tampilan sistem *loop* tertutup dapat dilihat pada Gambar 2.14.



Gambar 2.14 Kontrol loop tertutup

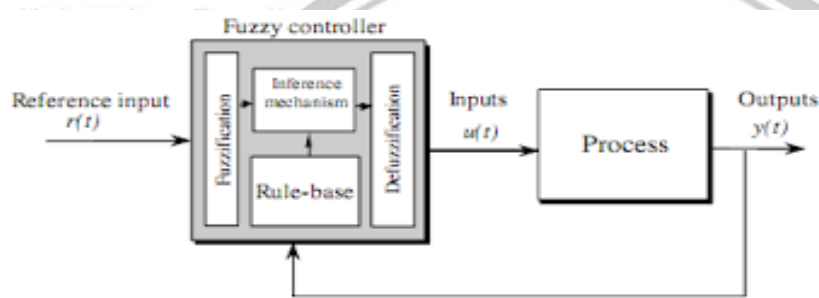
Sumber: diambil dari website (<https://serbatelekomunikasi.wordpress.com>)

## 2.8. Kontrol Logika Fuzzy

Sistem *Fuzzy* merupakan pemetaan nonlinear antara *masukan* dan *keluaran*-nya.

*Masukan* dan *keluaran* dari *Fuzzy* adalah nilai real bukan merupakan himpunan *Fuzzy*.

Kontrol *Fuzzy* menyediakan metodologi yang bersifat formal untuk merepresentasikan, memanipulasi, dan mengimplementasikan pengetahuan heuristik manusia tentang bagaimana mengontrol sebuah sistem. Blok kontrol *Fuzzy* ditunjukkan oleh Gambar 2.15 yang disajikan dalam sebuah sistem kontrol *closed-loop*.



Gambar 2.15 Arsitektur Kontroler *Fuzzy*

Sumber: Shaumi Arif dan Josaphat (2013)

Kontroler *Fuzzy* mempunyai empat komponen utama:

1. Basis aturan, berisi satu set aturan tentang bagaimana cara mengontrol sebuah sistem
2. Mekanisme inferensi, yang mengevaluasi aturan mana yang paling baik untuk digunakan pada saat itu dan kemudian memutuskan *masukan* apa yang akan diberikan pada *plant*
3. Fuzzifikasi, menerjemahkan *masukan* kontroler agar sesuai dengan basis aturan dan kemudian membandingkannya
4. Defuzzifikasi, merubah kesimpulan yang didapat pada mekanisme inferensi menjadi *masukan* yang sesuai untuk *plant*

Untuk mendesaian sebuah kontroler *Fuzzy*, *engineer* harus mempunyai pengetahuan tentang bagaimana kontroler *Fuzzy* bertindak dalam sebuah sistem *closed-loop*. Informasi ini bisa didapat dari orang yang paham dengan proses kontrol yang terjadi atau dengan memahami sendiri dinamika dari *plant* dan kemudian menuliskannya dalam sebuah basis aturan berbentuk “*if-then*”. (Shaumi dan Josaphat, 2013).

### 2.8.1. Himpunan *Fuzzy*

Himpunan *Fuzzy* Himpunan didefinisikan sebagai suatu kolerasi obyek-obyek yang terdefinisi secara tegas apakah suatu obyek adalah anggota himpunan itu atau tidak.

Dengan perkataan lain terdapat batas yang tegas antara unsur-unsur yang tidak merupakan anggota dari suatu himpunan. Tetapi dalam kenyataannya tidak semua himpunan yang kita jumpai dalam kehidupan sehari-hari terdefinisi secara demikian. Untuk mengatasi permasalahan himpunan dengan batas yang tidak tegas itu, Zadeh mengaitkan himpunan semacam itu dengan suatu fungsi yang menyatakan derajat kesesuaian unsur-unsur dalam semestanya dengan konsep yang merupakan syarat keanggotaan himpunan tersebut. Fungsi itu disebut fungsi keanggotaan suatu unsur dalam himpunan itu yang selanjutnya disebut himpunan *Fuzzy*. Dengan demikian setiap unsur dalam semesta wacananya mempunyai derajat keanggotaan tertentu dalam himpunan tersebut. Derajat keanggotaannya dinyatakan dengan suatu bilangan real dalam selang tertutup  $[0,1]$ . Dengan perkataan lain, fungsi keanggotaan dari suatu himpunan *Fuzzy A* dalam semesta  $X$  adalah memetakan  $X$  dari  $\mu_A$  keselang  $[0,1]$ . Nilai fungsi  $\mu_A(x)$  menyatakan derajat keanggotaan unsur  $x$  dalam himpunan kabur  $A$ . Nilai fungsi sama dengan 1 menyatakan keanggotaan penuh, dan nilai fungsi sama dengan 0 menyatakan sama sekali bukan anggota himpunan kabur tersebut. Maka himpunan tegas juga dapat dipandang sebagai kejadian khusus dari himpunan *Fuzzy*, yaitu himpunan *Fuzzy* yang fungsi keanggotaannya hanya bernilai 0 atau 1 saja. Jadi fungsi keanggotaan dari himpunan tegas  $A$  dalam semesta  $X$  adalah pemetaan dari  $X$  ke himpunan  $\{0,1\}$ .

Ada beberapa hal yang perlu diketahui dalam memahami sistem *Fuzzy*, yaitu:

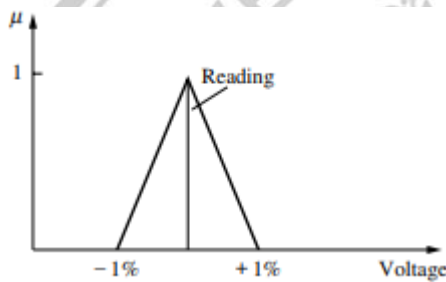
- Variabel *Fuzzy*, merupakan variabel yang hendak dibahas dalam suatu sistem *Fuzzy*. Contoh: umur, temperatur, perMINTaan, dsb.
- Himpunan *Fuzzy*, merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *Fuzzy*.

Setiap himpunan *Fuzzy* dapat dinyatakan dengan fungsi keanggotaan, ada beberapa cara untuk menyatakan himpunan *Fuzzy* dengan fungsi keanggotaannya. Untuk semesta hingga diskrit biasanya dipakai cara mendatar, yaitu daftar anggota-anggota semesta bersama dengan derajat keanggotaannya. Kebanyakan himpunan *Fuzzy* berada dalam semesta himpunan semua bilangan riil dengan fungsi keanggotaannya yang dinyatakan dalam bentuk suatu formula matematis. Ada beberapa fungsi keanggotaan yang dapat digunakan untuk mendefinisikan himpunan *Fuzzy*, antara lain: fungsi keanggotaan liner, fungsi keanggotaan Segitiga, fungsi keanggotaan trapezium, fungsi keanggotaan bentuk lonceng. Pemetaan dalam penelitian ini akan digunakan fungsi segitiga dan fungsi trapezium sebagai fungsi keanggotaannya.

### 2.8.2. Fuzzifikasi

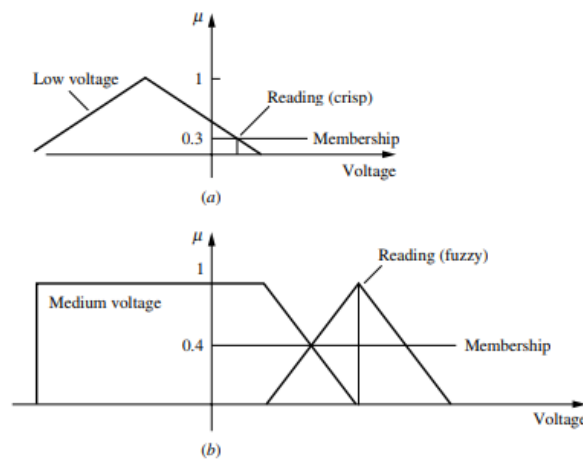
Fuzzifikasi adalah proses pembuatan kuantitas *Fuzzy*(*crisp*). Kami melakukan ini dengan hanya mengakui bahwa banyak dari jumlah yang kami anggap *crisp* dan *deterMINistik* sebenarnya tidak *deterMINistik* sama sekali. Mereka membawa ketidakpastian yang cukup besar. Jika bentuk ketidakpastian muncul karena ketidaktepatan, ambiguitas, atau ketidakjelasan, maka variabel tersebut mungkin semu dan dapat diwakili oleh fungsi keanggotaan.

Pada dunia nyata, perangkat keras seperti voltmeter digital menghasilkan data yang tajam, tetapi data ini dapat mengalami kesalahan eksperimental. Informasi yang ditunjukkan pada Gambar 2.16 menunjukkan satu kemungkinan kisaran kesalahan untuk tipikal pembacaan tegangan dan fungsi keanggotaan terkait yang mungkin mewakili ketidaktepatan tersebut.



Gambar 2.16 Keanggotaan fungsi yang menunjukkan ketidakpastian pada pembacaan *voltage* (*crisp*)

Sumber: Timothy J. Ross, (2004)



Gambar 2.17 Perbandingan himpunan *Fuzzy* atau pembacaan *Fuzzy*: (a) himpunan *Fuzzy* dan pembacaan (*crisp*); (b) himpunan *Fuzzy* dan pembacaan *Fuzzy*

Sumber: Timothy J. Ross, (2004)

Representasi data semu sebagai himpunan *Fuzzy* adalah langkah yang berguna tetapi tidak wajib ketika data tersebut digunakan dalam sistem *Fuzzy*. Gagasan ini ditunjukkan pada Gambar 2.17, di mana kami menganggap data sebagai pembacaan



yang tajam (Gambar 2.17a) atau sebagai pembacaan *Fuzzy*, seperti yang ditunjukkan pada Gambar 2.17b. Pada Gambar 2.17a, kita mungkin ingin membandingkan pembacaan tegangan *crisp* dengan himpunan *Fuzzy* “tegangan rendah”, pada keanggotaan 0,3, himpunan *Fuzzy* dan pembacaan berpotongan pada nilai keanggotaan 0,3. Pada Gambar 2.17b, perpotongan himpunan *Fuzzy* “tegangan menengah” dan pembacaan tegangan *fuzzified* terjadi pada keanggotaan 0,4. Kita dapat melihat pada Gambar 2.17b bahwa titik potong dari dua himpunan *Fuzzy* adalah segitiga kecil, yang keanggotaan terbesarnya terjadi pada nilai keanggotaan 0,4.

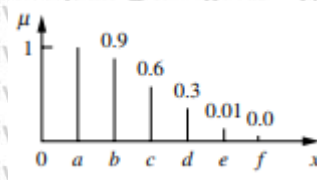
**2.8.3. Defuzzifikasi**

Secara matematis, defuzzifikasi himpunan *Fuzzy* adalah proses 'pembulatan' dari lokasi di unit *hypercube* ke verteks terdekat (dalam arti geometris). Jika seseorang menganggap himpunan *Fuzzy* sebagai kumpulan nilai keanggotaan, atau vektor nilai pada interval unit, defuzzifikasi mengurangi vektor ini menjadi kuantitas skalar tunggal, mungkin ke nilai yang paling khas (prototipe) atau representatif. Berbagai bentuk populer dari konversi himpunan *Fuzzy* ke himpunan *crisp* atau nilai skalar tunggal.

a. Defuzzifikasi ke Himpunan *Crips*

Setiap elemen  $x \in A_\lambda$  milik  $\underline{A}$  dengan tingkat keanggotaan yang lebih besar atau sama dengan nilai  $\lambda$ . Mari kita perhatikan himpunan *Fuzzy* diskrit, menggunakan notasi Zadeh, yang didefinisikan pada semesta  $X = \{a, b, c, d, e, f\}$ , himpunan *Fuzzy* ini ditunjukkan secara skematis pada Gambar 2.18. Kita dapat mengurangi himpunan *Fuzzy* ini menjadi beberapa himpunan  $\lambda$ -cut, yang semuanya *crisp*. Sebagai contoh, kita dapat mendefinisikan himpunan  $\lambda$ -cut untuk nilai  $\lambda = 1, 0,9, 0,6, 0,3, 0^+, \text{ dan } 0$ .

$$A = \left\{ \frac{1}{a} + \frac{0,9}{b} + \frac{0,6}{c} + \frac{0,3}{d} + \frac{0,01}{e} + \frac{0}{f} \right\} \dots\dots\dots(2-5)$$



Gambar 2. 18 Bentuk Diskrit dari Himpunan *Fuzzy* A

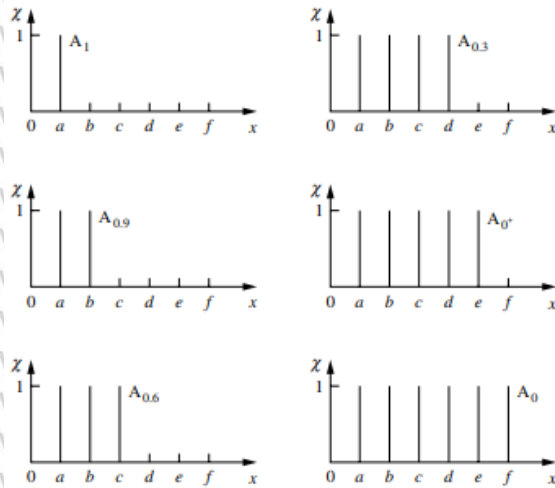
Sumber: Timothy J. Ross, (2004)

$$A_1 = (a), \quad A_{0,9} = (a,b) \dots\dots\dots(2-6)$$

$$A_{0,6} = (a,b,c), \quad A_{0,3} = (a,b,c,d) \dots\dots\dots(2-7)$$

$$A_{0^+} = (a,b,c,d,e), \quad A_0 = X \dots\dots\dots(2-8)$$





Gambar 2.19 Himpunan Lambda-cut untuk  $\lambda = 1, 0,9, 0,6, 0,3, 0^+, 0$ .  
 Sumber: Timothy J. Ross, (2004)

Kita dapat menuliskan set  $\lambda$ -cut menggunakan notasi Zadeh. Sebagai contoh, himpunan  $\lambda$ -cut untuk nilai  $\lambda = 0,9$  dan  $0,25$  diberikan di sini:

$$A_{0,9} = \left\{ \frac{1}{a} + \frac{1}{b} + \frac{0}{c} + \frac{0}{d} + \frac{0}{e} + \frac{0}{f} \right\} \dots\dots\dots(2-9)$$

$$A_{0,25} = \left\{ \frac{1}{a} + \frac{1}{b} + \frac{1}{c} + \frac{1}{d} + \frac{0}{e} + \frac{0}{f} \right\} \dots\dots\dots(2-10)$$

Empat sifat istimewa berikut berlaku untuk Himpunan  $\lambda$ -cut:

1.  $(A \cup B)_\lambda = A_\lambda \cup B_\lambda$
2.  $(A \cap B)_\lambda = A_\lambda \cap B_\lambda$
3.  $(\bar{A})_\lambda \neq \bar{A}_\lambda$  Kecuali untuk nilai  $\lambda = 0,5$
4. untuk nilai  $\lambda \leq \alpha$ , dimana  $0 \leq \alpha \leq 1$ , benar bahwa  $A_\lambda \subseteq A_\alpha$ , dimana  $A_0 = X$

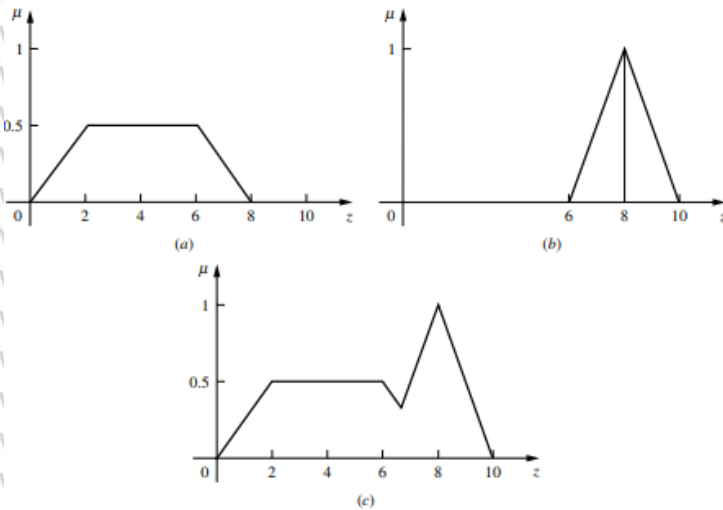
b. Defuzzifikasi ke Bentuk Skalar

Defuzzifikasi adalah konversi kuantitas *Fuzzy* ke kuantitas yang tepat, seperti halnya fuzzifikasi adalah konversi dari jumlah yang tepat ke jumlah yang tidak jelas.

Keluaran dari proses *Fuzzy* bisa menjadi logis penyatuan dua atau lebih fungsi keanggotaan *Fuzzy* didefinisikan pada semesta sementara variabel *keluaran*. Secara umum kita dapat menuliskan notasi sebagai berikut:

$$C_k = \bigcup_{i=1}^k C_j = C \dots\dots\dots(2-11)$$





Gambar 2.20 Karakteristik keluaran dari kontrol Fuzzy: (a) keluaran pertama Fuzzy; (b) keluaran kedua Fuzzy; (c) gabungan keduanya.

Sumber: Timothy J. Ross, (2004)

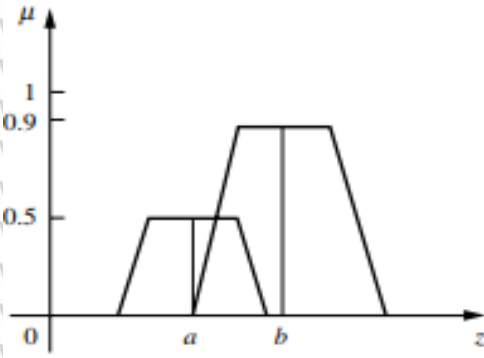
Di antara banyak metode yang telah diusulkan dalam literatur dalam beberapa tahun terakhir, dalam penelitian ini menggunakan salah satu metode defuzzifikasi yaitu “Weighted average”. Metode “Weighted average” adalah yang paling sering digunakan dalam aplikasi Fuzzy karena merupakan salah satu metode yang lebih efisien secara komputasi. Sayangnya itu biasanya terbatas pada fungsi keanggotaan keluaran simetris. Ini diberikan oleh ekspresi aljabar.

$$z^* = \frac{\sum \mu c(\bar{z}) \cdot \bar{z}}{\sum \mu c(\bar{z})} \dots\dots\dots(2-12)$$

Di mana  $\sum$  menunjukkan jumlah aljabar dan di mana  $z$  adalah pusat massa dari masing-masing fungsi keanggotaan simetris. Metode ini ditunjukkan pada Gambar 2.21 Metode rata-rata tertimbang dibentuk dengan menimbang setiap fungsi keanggotaan dalam keluaran dengan nilai keanggotaan maksimum masing-masing.

Sebagai contoh, dua fungsi yang ditunjukkan pada Gambar 2.21 akan menghasilkan bentuk umum berikut untuk nilai defuzzifikasi:

$$z^* = \frac{a(0.5)+b(0.9)}{0.5+0.9} \dots\dots\dots(2-13)$$



Gambar 2.21 Metode “Weighted average”

Sumber: Timothy J. Ross, (2004)

Karena metode ini terbatas pada fungsi keanggotaan simetris, nilai  $a$  dan  $b$  adalah sarana (centroid) dari bentuk masing-masing.

## 2.9 *Firestore Realtime Database*

Firestore Realtime Database merupakan *cloud database*. Data disimpan dalam format JSON dan disinkronkan secara *realtime* ke setiap klien yang terhubung. Ketika membangun aplikasi *hybrid* lintas platform, seperti Android dan iOS maka semua klien berbagi satu *instance* Realtime Database dan secara otomatis menerima pembaruan dengan data tertentu.

Firestore Realtime Database adalah basis data NoSQL dan karena itu memiliki optimalisasi dan fungsionalitas yang berbeda dibandingkan dengan basis data relasional. Membuat database Firestore bisa melalui import file JSON ke konsol Firestore, atau dapat juga dibuat langsung melalui halaman konsol Realtime Database secara manual.



### BAB III METODE PENELITIAN

Metode penelitian pada dasarnya merupakan suatu cara ilmiah untuk mendapatkan informasi dari suatu tujuan dan manfaat tertentu. Dalam menyelesaikan rumusan masalah dan merealisasikan penelitian yang terdapat di bab pendahuluan maka diperlukan langkah-langkah untuk menyelesaikan masalah tersebut. Metode penelitian pada skripsi ini meliputi:

1. Perancangan blok diagram
2. Spesifikasi sistem
3. Studi literatur
4. Karakteristik blok
5. Perancangan *hardware*
6. Perancangan Kontrol Logika Fuzzy
7. Perancangan antarmuka *Internet of Things*
8. Pengujian Alat
9. Pengambilan Kesimpulan
10. Rencana Kegiatan

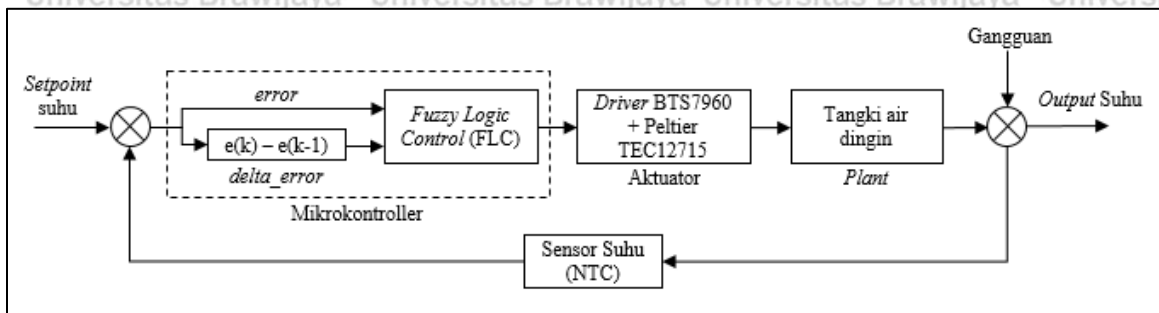
#### 3.1. Perancangan Blok Diagram

Pada perancangan alat, diperlukan perencanaan diagram blok yang akan digunakan oleh sistem sehingga sistem dapat diketahui garis besar prinsip kerja dan diharapkan dapat bekerja sesuai dengan desain. Spesifikasi alat dan diagram blok sistem dijelaskan sebagai berikut:

- a. Sistem yang akan dibuat merupakan *Single Input Single Output (SISO)* yaitu mempunyai satu masukan dan keluaran yaitu berupa suhu.
- b. *Setpoint*: Nilai masukan sistem yang diinginkan pada keluaran sistem. Pada pengujian penelitian ini, adapun *setpoint* suhu ditentukan oleh pengguna dispenser.
- c. Kontroler yang digunakan adalah Kontrol Logika Fuzzy.
- d. *Plant* yang digunakan adalah tangki pendingin pada dispenser.
- e. Pusat pengendalian sistem adalah board mikrokontroler ESP32 untuk memberikan sinyal kontrol berupa PWM.
- f. Gangguan pada plant berupa suhu luar tangki dan juga jumlah air dalam tangki.
- g. Sensor suhu digunakan sebagai *feedback* (umpan balik) sistem. Adapun sensor yang digunakan adalah sensor NTC.

h. Aktuator pendingin yang digunakan yaitu sebuah peltier TEC12715 (*Thermo Electric Cooling*) dengan daya 93 W.

Diagram blok sistem dijelaskan pada Gambar 3.1 sebagai berikut:



Gambar 3.1 Blok diagram sistem

### 3.2. Spesifikasi Sistem

Penentuan spesifikasi yang diinginkan merupakan tahap yang penting pada sebuah perancangan, khususnya pada perancangan *dispenser* menggunakan Kontrol Logika Fuzzy (KLF). Parameter-parameter ini akan menjadi dasar untuk melihat apakah respon keluaran sistem sudah sesuai dengan spesifikasi atau belum. Adapun spesifikasi kontrol yang diinginkan adalah sebagai berikut:

a. *Error Steady State*  $< 5\%$

*Error Steady State* (ESS) merupakan ukuran dari akurasi sistem kontrol dalam pelacakan masukan perintah atau menyaring gangguan dalam bentuk perubahan beban.

Kesalahan atau *error* yang terjadi bergantung pada jenis sistem dan input, meliputi sinyal perintah dan gangguan.

b. *Maximum Overshoot*  $< 5\%$

*Maximum overshoot* merupakan besaran maksimum dimana respon sistem melebihi nilai keadaan tunaknya. Dalam kata lain, merupakan amplitudo puncak pertama.

Semakin besar nilai *overshoot* yang dihasilkan, akan berpengaruh pada *settling time* yang lebih lama.

### 3.3. Studi Literatur

Studi literatur dilakukan untuk mendapatkan pengetahuan dasar tentang segala sesuatu yang mendukung perancangan serta pembuatan alat ini. Data yang digunakan sebagai penunjang referensi keputusan dan berbagai teori pendukung didapatkan dari berbagai sumber pustaka yang terdiri dari jurnal ilmiah, buku, dan sumber lain yang berkaitan dengan rumusan masalah. Selain itu, sumber pustaka digunakan sebagai referensi guna mengetahui

prinsip kerja, karakteristik komponen, serta teori-teori yang menunjang. Adapun teori-teori yang dikaji, yaitu:

- a. *Dispenser*.
- b. Komponen *Dispenser*.
- c. Pemrograman Bahasa C++ pada Arduino Ide.
- d. Karakteristik sensor NTC.
- e. Karakteristik modul *Driver* BTS7690
- f. Karakteristik aktuator Peltier TEC (*Thermo Electric Cooling*).
- g. Kontrol Logika Fuzzy.
- h. Metode Sistem Inferensi Mamdani.
- i. Metode Sistem IoT (*Internet of Things*).

### 3.4. Karakterisasi Blok

#### 3.4.1. Karakterisasi *Driver* BTS7960

##### a. Tujuan

Karakterisasi ini dilakukan untuk mengetahui apakah kemampuan komponen ini dapat digunakan sebagai *driver* atau pengatur level tegangan (PWM) untuk pendingin dengan membandingkan perubahan nilai suhu dari peltier TEC12715.

##### b. Peralatan yang Digunakan

- ESP32
- Rangkaian *Driver* BTS7960
- Sumber Tegangan DC
- *Infrared Thermometer*
- Peltier TEC12715
- Laptop
- *Smartphone*

##### c. Langkah Karakterisasi

- Menghubungkan rangkaian *driver* BTS7960 dengan ESP32.
- Menghubungkan rangkaian *driver* BTS7960 dengan sumber DC.
- Menghubungkan ESP32 dengan laptop.
- Memberikan beban (Peltier TEC12715) pada rangkaian *driver* BTS7960.

- Memberikan variasi nilai level tegangan PWM (*Pulse Width Modulation*) pada rangkaian *driver* BTS7960 melalui *software* Arduino.

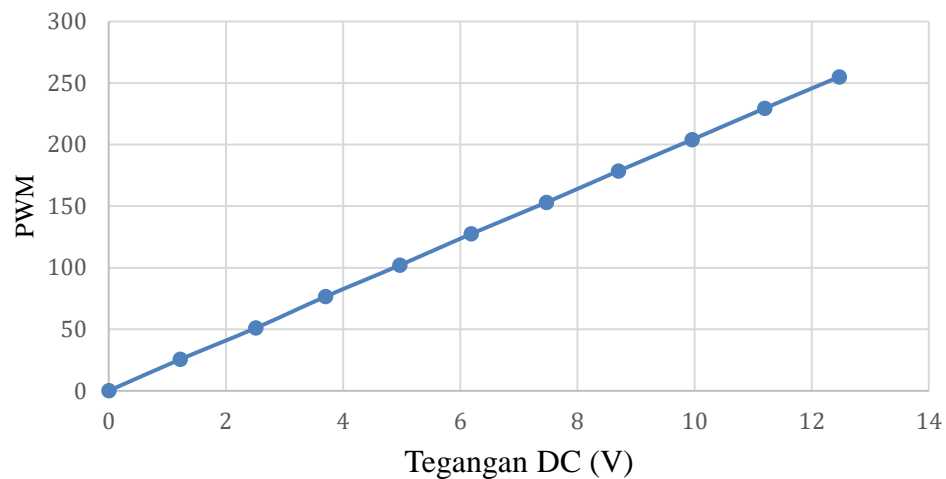
#### d. Karakterisasi dan Analisis

Dari hasil karakterisasi rangkaian *driver* BTS7960 didapatkan hasil berupa perubahan nilai tegangan DC yang berbeda-beda sesuai dengan perubahan nilai *PWM* yang diberikan. Hasil karakterisasi dapat dilihat dalam Tabel 1 dan Gambar 3.2 sebagai berikut:

Tabel 1  
Perbandingan nilai PWM terhadap tegangan keluaran *driver* BTS7960

| No. | <i>Duty Cycle</i> | PWM | Tegangan DC (V) |
|-----|-------------------|-----|-----------------|
| 1.  | 0                 | 0   | 0               |
| 2.  | 10                | 25  | 1,22            |
| 3.  | 20                | 51  | 2,51            |
| 4.  | 30                | 76  | 3,7             |
| 5.  | 40                | 102 | 4,97            |
| 6.  | 50                | 127 | 6,19            |
| 7.  | 60                | 153 | 7,47            |
| 8.  | 70                | 178 | 8,7             |
| 9.  | 80                | 204 | 9,96            |
| 10. | 90                | 229 | 11,2            |
| 11. | 100               | 255 | 12,47           |

Berdasarkan Tabel 1 dapat disimpulkan bahwa rangkaian *driver* BTS7960 dapat berfungsi dan dapat digunakan sebagai *driver* aktuator elemen pendingin peltier TEC12715, ditandai dengan adanya perubahan tegangan DC yang tentunya berpengaruh terhadap kecepatan pendinginan peltier TEC12715 akibat perubahan nilai PWM yang diberikan. Adapun korelasi perubahan PWM dan tegangan keluaran *driver* BTS7960 dapat ditunjukkan pada Gambar 3.2 sebagai berikut:



Gambar 3.2 Hubungan nilai PWM terhadap tegangan keluaran *driver* BTS7960

### 3.4.2. Karakterisasi Sensor *Thermistor* NTC

#### a. Tujuan

Karakterisasi ini dilakukan untuk mengetahui performansi atau kemampuan (keberhasilan dan keakuratan) pembacaan sensor NTC 10k terhadap perubahan suhu dengan melihat hasil pembacaan sensor yang ditampilkan.

#### b. Peralatan yang Digunakan

- Sensor *Thermistor* NTC
- ESP32
- *Infrared Thermometer*
- Tangki Pendingin *Dispenser*
- Peltier TEC12715 93 W
- *Driver* BTS7960

- Laptop

#### c. Langkah Karakterisasi

- Menghubungkan sensor *Thermistor* NTC dengan ESP32, dilanjutkan dengan menghubungkan ESP32 dengan laptop.
- Menghubungkan *Driver* BTS7960 dengan Peltier TEC12715 93W serta memberi catu daya tegangan DC.
- Mengunggah program dan menjalankan *serial monitor* pada software Arduino 1.8.7.
- Mengamati dan mencatat setiap perubahan suhu berdasarkan suhu *infrared thermometer* sebagai acuan.

d. Karakterisasi dan Analisis

Hasil karakterisasi sensor *thermistor* NTC dapat dilihat dalam Tabel 2 sebagai berikut:

Tabel 2  
Hasil karakterisasi sensor *thermistor* NTC 10K

| No.          | Suhu dibaca <i>infrared thermometer</i> (°C) | Suhu dibaca sensor <i>thermistor</i> NTC (°C) | Nilai ADC |
|--------------|--|---|-----------|
| 1.           | 11   | 10.72   | 676       |
| 2.           | 13   | 12.95   | 651       |
| 3.           | 15   | 14.66   | 631       |
| 4.           | 17   | 17.61   | 597       |
| 5.           | 19   | 18,69   | 584       |
| 6.           | 21   | 21.13   | 556       |
| 7.           | 23   | 23.16   | 532       |
| 8.           | 25   | 24.76   | 514       |
| 9.           | 27   | 27.71   | 481       |
| <b>Total</b> |  |   |           |
|              | 171  | 171.39  |           |

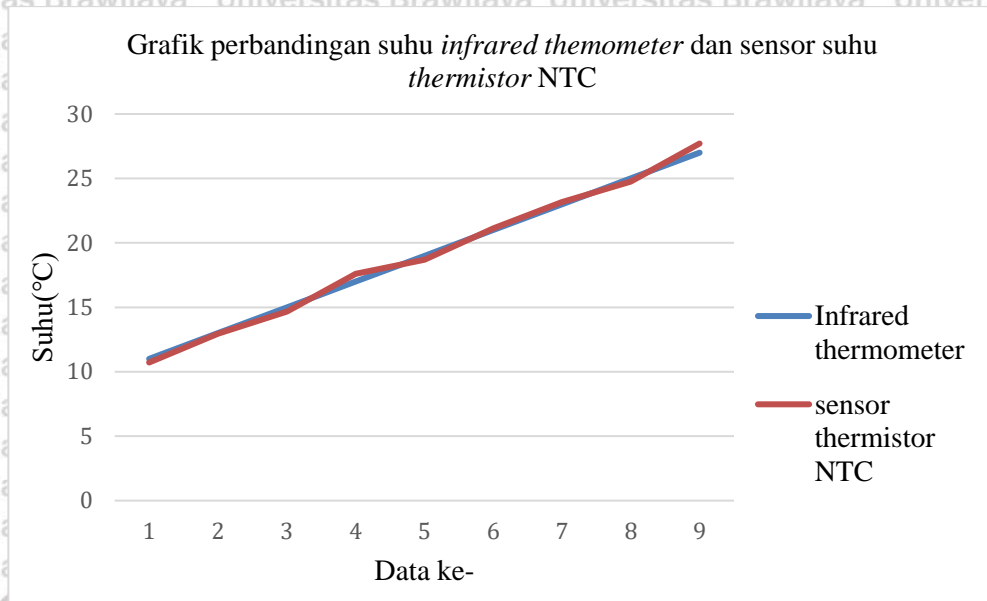
Perhitungan kesalahan pembacaan sensor:

$$\% \text{ error} = \frac{|\sum \text{Hasil pengukuran} - \sum \text{Referensi}|}{\sum \text{Referensi}} \times 100\%$$

$$\% \text{ error suhu} = \frac{171,39 - 171}{171} \times 100\% = 0.228\%$$

Berdasarkan hasil karakterisasi Tabel 2 terkait % *error* antara pembacaan suhu *infrared thermometer* dengan pembacaan pada *serial monitor* adalah sangat kecil, yakni 0,228% untuk suhu air yang dibaca. Adapun grafik perbandingan suhu *infrared thermometer* sebagai acuan dan *serial monitor* dapat dilihat pada gambar 3.3 sebagai berikut:





Gambar 3.3 Perbandingan suhu antara *infrared thermometer* dengan serial monitor pembacaan sensor suhu *thermistor NTC*

### 3.5. Pembuatan *Hardware*

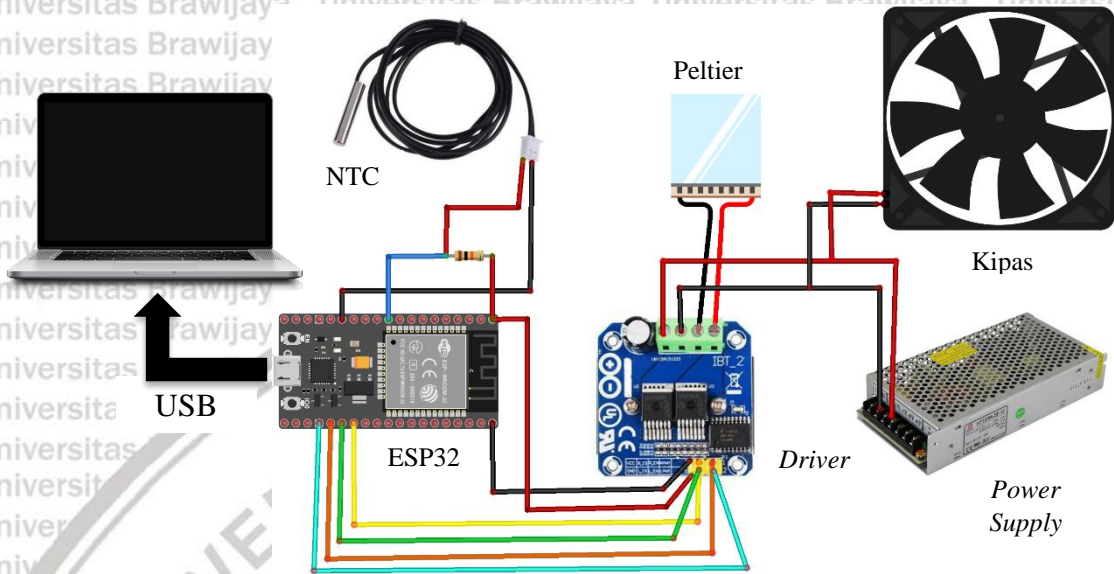
Pembuatan *hardware* atau perangkat keras atau lebih tepatnya memodifikasi dispenser konvensional adalah sebagai langkah awal sebelum terbentuknya kesatuan sistem yang utuh beserta pemrogramannya. Hal ini bertujuan agar pemodelan pendingin dispenser menggunakan Kontrol Logika Fuzzy (KLF) berjalan sesuai deskripsi awal atau rencana yang diinginkan.

1. Skema pembuatan *hardware*
2. Penentuan komponen elektronik yang digunakan meliputi:
  - a. Mikrokontroler ESP32 sebagai perangkat kontroler.
  - b. Sensor *thermistor NTC* sebagai *feedback* suhu.
  - c. Peltier TEC12715 93 W.
  - d. *Driver* BTS7960.

Perangkat keras diatas nantinya akan membentuk sebuah sistem yang saling terhubung satu sama lain. Proses skema perancangan alat adalah sebagai berikut :

- a. Dari komputer, program ditransfer ke ESP32.
- b. ESP32 memberikan sinyal PWM (*Pulse Width Modulation*) pada Peltier TEC12715 yang terhubung dengan *Driver* BTS7960..
- c. Peltier TEC12715 diberi catu daya 12 V yang telah dihubungkan dengan *Driver* BTS7960.

Adapun tampilan rangkaian dari beberapa komponen utama yang sudah disebutkan yang ditunjukkan pada gambar 3.4 sebagai berikut:



Gambar 3.4 Skema pembuatan perangkat keras (*hardware*)

### 3.6. Perancangan Kontrol Logika Fuzzy

#### 3.6.1. Variabel Masukan dan Keluaran

Kontrol Logika Fuzzy yang digunakan pada penelitian ini mempunyai dua *crisp* masukan, yaitu *error* dan *delta error* suhu serta satu keluaran yaitu PWM untuk *driver* BTS7960. *Error* suhu didapatkan dari selisih antara *setpoint* suhu dengan umpan balik keluaran *plant* berupa suhu yang dibaca oleh sensor *thermistor* NTC. Sementara *delta error* merupakan selisih antara *error* yang didapat dengan *error* sebelumnya. Keluaran berupa PWM digunakan untuk memberikan perubahan tegangan yang selanjutnya digunakan untuk merubah kemampuan Peltier TEC12715 untuk mendinginkan tangka. Penentuan nilai dan bentuk dari variabel masukan dan keluaran berdasarkan beberapa percobaan *trial and error* pada sistem sampai menemukan yang sudah sesuai dengan sistem yang diinginkan. Selain itu juga memperhatikan spesifikasi atau kemampuan komponen yang digunakan pada sistem.

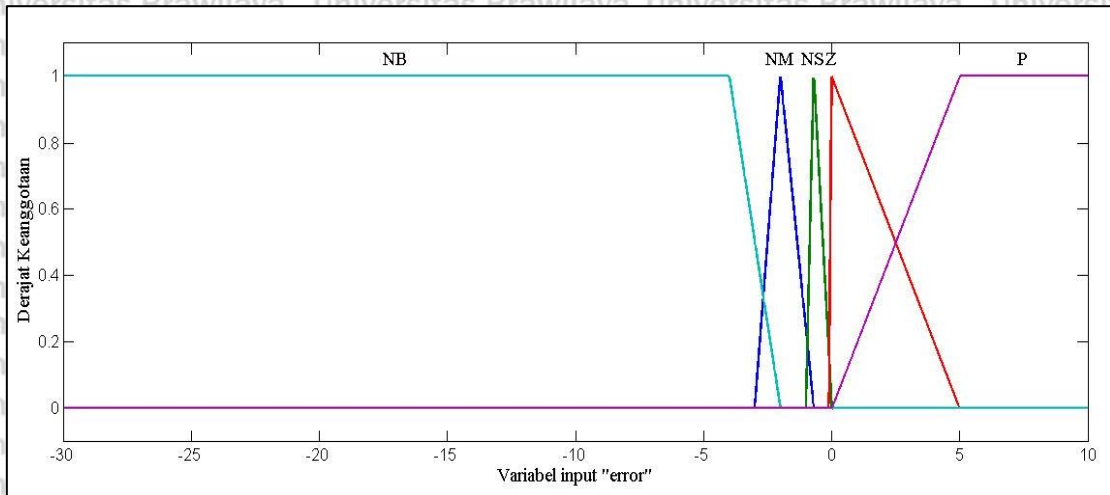
#### 3.6.2. Fungsi Keanggotaan Masukan dan Keluaran

Banyaknya fungsi keanggotaan tergantung dari bagaimana respon dari sistem yang kita inginkan serta berdasarkan observasi terhadap variabel-variabel yang ingin dijadikan masukan. Semakin banyak aksi kontrol yang akan terjadi maka semakin banyak pula label fungsi keanggotaanya.



Perancangan fuzzy yang dilakukan merupakan fuzzy error dan delta error.

Perancangan fuzzy error ini terbagi menjadi 5 yaitu NB (*Negative Big*), NM (*Negative Medium*), NS (*Negative Small*), Z (*Zero*), P (*Positive*). Perancangan fuzzy error ini adalah representasi dari error variabel suhu yang dideteksi oleh sensor suhu *thermistor* NTC. Gambar 3.5 adalah bentuk perancangan fungsi keanggotaan fuzzy error.



Gambar 3.5 Fungsi keanggotaan masukan fuzzy error

$$\mu[x1]NB = \begin{cases} 1, & x \leq -4 \\ \frac{-x-2}{2}, & -4 < x < -2 \\ 0, & x \geq -2 \end{cases} \dots\dots\dots (3-1)$$

$$\mu[x1]NM = \begin{cases} 0, & x \leq -3 \\ \frac{x+3}{1}, & -3 < x < -2 \\ 1, & x = -2 \\ \frac{-x-0,7}{1,3}, & -2 < x < -0,7 \\ 0, & x \geq -0,7 \end{cases} \dots\dots\dots (3-2)$$

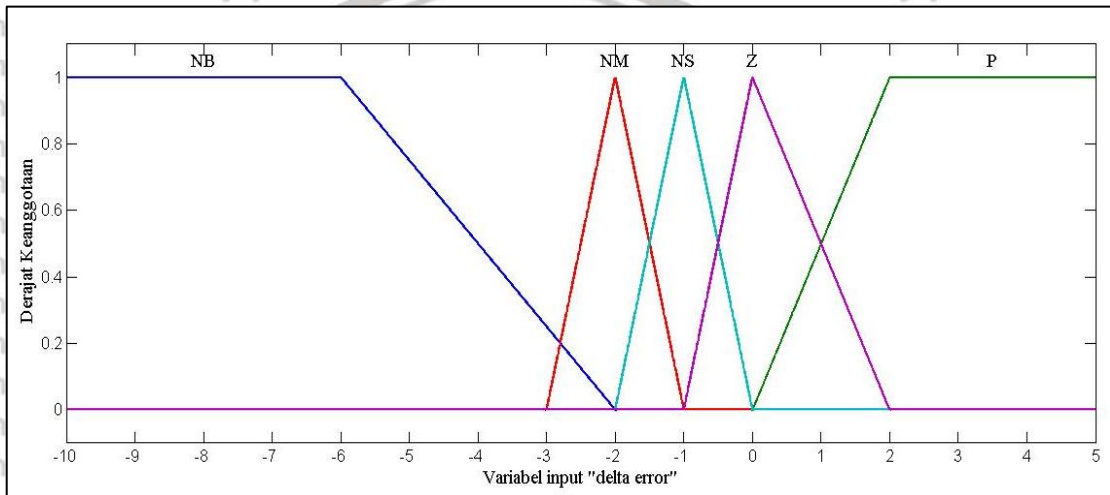
$$\mu[x1]NS = \begin{cases} 0, & x \leq -1 \\ \frac{x+1}{0,3}, & -1 < x < -0,7 \\ 1, & x = -0,7 \\ \frac{-x}{0,7}, & -0,7 < x < 0 \\ 0, & x \geq 0 \end{cases} \dots\dots\dots (3-3)$$

$$\mu[x1]Z = \begin{cases} 0, & x \leq -0,1 \\ \frac{x+0,1}{0,1}, & -0,1 < x < 0 \\ 1, & x = 0 \\ \frac{-x+5}{5}, & 0 < x < 5 \\ 0, & x \geq 5 \end{cases} \dots\dots\dots (3-4)$$



$$\mu[x1] P = \begin{cases} 0, & x \leq 0 \\ \frac{x}{5}, & 0 < x < 5 \\ 1, & x \geq 5 \end{cases} \dots\dots\dots (3-5)$$

Sedangkan perancangan *fuzzy error* ini terbagi juga menjadi 5 yaitu NB (*Negative Big*), NM (*Negative Medium*), NS (*Negative Small*), Z (*Zero*), P (*Positive*). Perancangan *fuzzy error* ini adalah representasi dari perubahan *error* variabel suhu yang dikomputasikan dalam ESP 32. Gambar 3.6 adalah bentuk perancangan fungsi keanggotaan *fuzzy delta error*.



Gambar 3.6 Fungsi keanggotaan masukan *fuzzy delta error*

$$\mu[x2]NB = \begin{cases} 1, & x \leq -6 \\ \frac{-x-4}{2}, & -6 < x < -2 \\ 0, & x \geq -2 \end{cases} \dots\dots\dots (3-6)$$

$$\mu[x2]NM = \begin{cases} 0, & x \leq -3 \\ \frac{x+3}{1}, & -3 < x < -2 \\ 1, & x = -2 \\ \frac{-x-1}{1}, & -2 < x < -1 \\ 0, & x \geq -1 \end{cases} \dots\dots\dots (3-7)$$

$$\mu[x2]NS = \begin{cases} 0, & x \leq -2 \\ \frac{x+2}{1}, & -2 < x < -1 \\ 1, & x = -1 \\ \frac{-x}{1}, & -1 < x < 0 \\ 0, & x \geq 0 \end{cases} \dots\dots\dots (3-8)$$

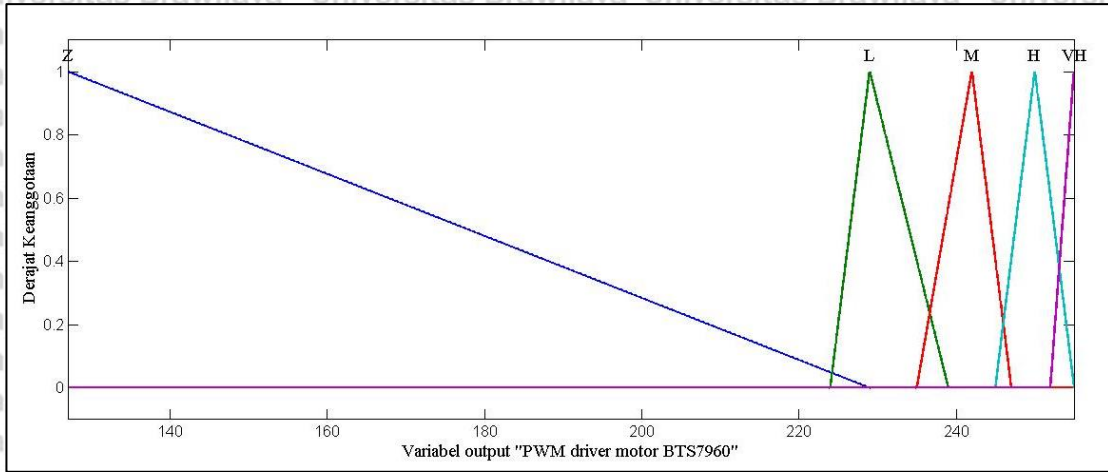
$$\mu[x2] Z = \begin{cases} 0, & x \leq -1 \\ \frac{x+1}{1}, & -1 < x < 0 \\ 1, & x = 0 \\ \frac{-x+2}{2}, & 0 < x < 2 \\ 0, & x \geq 2 \end{cases} \dots\dots\dots (3-9)$$



$$\mu[x2] P = \begin{cases} 0, & x \leq 0 \\ \frac{x}{2}, & 0 < x < 2 \\ 1, & x \geq 2 \end{cases} \dots\dots\dots (3-10)$$

Sehingga dari dua input *error* dan *derror* dapat ditentukan bentuk *fuzzy* dari keluaran yaitu PWM untuk input *driver* BTS7960. Terdapat 5 variabel yang akan diproses dalam *fuzzifikasi* ini, yaitu Z (Zero), L (Low), M (Medium), H (High), dan VH (Very High). Gambar perancangan fuzzy keluaran PWM adalah pada gambar 3.7.





Gambar 3.7 Fungsi keanggotaan fuzzy keluaran

$$\mu[x3]Z = \begin{cases} 0 & , x < 127 \\ 1 & , x = 127 \\ \frac{-x+229}{127} & , 127 < x < 229 \end{cases} \dots\dots\dots (3-11)$$

$$\mu[x3]L = \begin{cases} 0 & , x \leq 224 \\ \frac{x-224}{5} & , 224 < x < 229 \\ 1 & , x = 229 \\ \frac{-x+239}{10} & , 229 < x < 239 \\ 0 & , x \geq 239 \end{cases} \dots\dots\dots (3-12)$$

$$\mu[x3]M = \begin{cases} 0 & , x \leq 235 \\ \frac{x-235}{8} & , 235 < x < 242 \\ 1 & , x = 242 \\ \frac{-x+247}{5} & , 242 < x < 247 \\ 0 & , x \geq 247 \end{cases} \dots\dots\dots (3-13)$$

$$\mu[x3]H = \begin{cases} 0 & , x \leq 245 \\ \frac{x-245}{5} & , 245 < x < 250 \\ 1 & , x = 250 \\ \frac{-x+255}{5} & , 250 < x < 255 \\ 0 & , x = 255 \end{cases} \dots\dots\dots (3-14)$$

$$\mu[x3]VH = \begin{cases} 0 & , x \leq 252 \\ \frac{x-252}{2} & , 252 < x < 255 \\ 1 & , x = 255 \end{cases} \dots\dots\dots (3-15)$$



### 3.6.3. Perancangan Aturan Fuzzy

Proses inferensi mengaplikasikan aturan-aturan *Fuzzy* pada masukan *Fuzzy* kemudian mengevaluasi setiap aturan. Prinsip logika *Fuzzy* digunakan untuk mengkombinasi aturan-aturan *if-then* yang terdapat dalam basis aturan suatu pemetaan dari suatu himpunan *Fuzzy* masukan ke himpunan *Fuzzy* keluaran. Dalam penentuan variabel nilai pada aturan *fuzzy* ini, telah dilakukan beberapa kali percobaan *trial and error* sampai menemukan sesuai dengan sistem yang diinginkan. Adapun basis aturan yang dirancang berdasarkan masukan dan keluaran yang ditetapkan dapat dilihat pada Tabel 3.

Tabel 3  
Aturan Fuzzy Keluaran *Driver* BTS7960

| Aturan Fuzzy |    | Delta Error Suhu |    |    |    |   |
|--------------|----|------------------|----|----|----|---|
|              |    | NB               | NM | NS | Z  | P |
| Error Suhu   | NB | VH               | VH | VH | VH | H |
|              | NM | VH               | VH | VH | H  | H |
|              | NS | VH               | VH | H  | H  | M |
|              | Z  | L                | L  | Z  | Z  | Z |
|              | P  | Z                | Z  | Z  | Z  | Z |

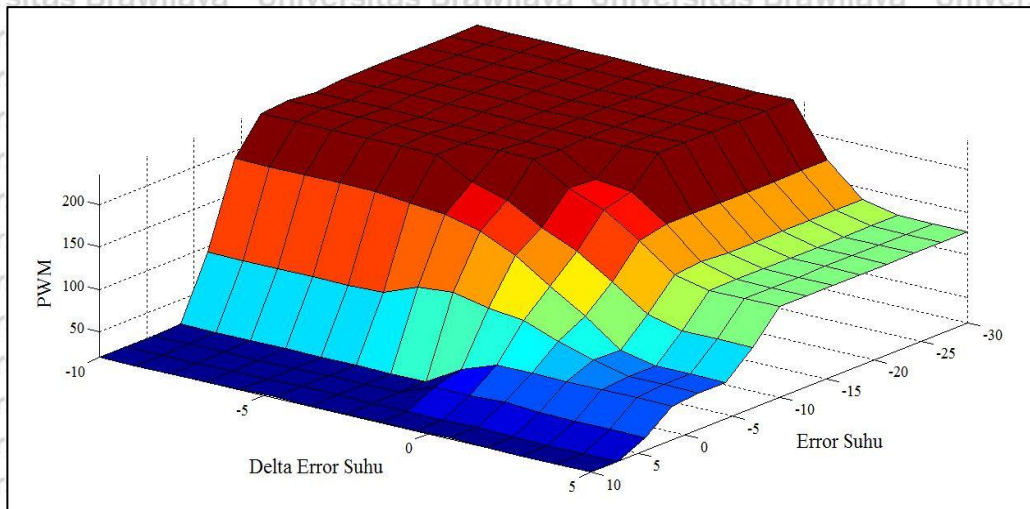
### 3.6.4. Defuzzyfikasi

Metode Defuzzyfikasi menggunakan metode *Mean of Max* dimana pada metode ini akan dipilih nilai *crisp* untuk mewakili hasil dari inferensi *min-max*. Nilai yang dipilih adalah nilai rata-rata dari nilai *crisp* yang memiliki fungsi keanggotaan tertinggi. Metode ini dirasa cukup baik untuk plant dispenser dan juga memiliki komputasi yang cukup mudah.

### 3.6.5. Surface Fuzzy

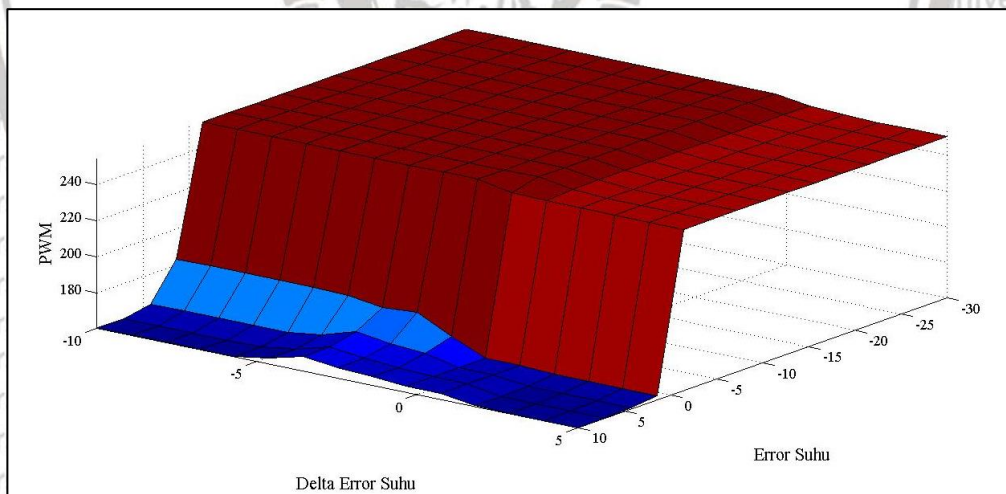
Penggunaan *surface fuzzy* ini adalah untuk melihat gambar pemetaan antara variabel-variabel masukan dan keluaran. Dari perancangan *fuzzy* pada sistem ini, terdapat proses pengamatan masukan dan keluaran yang diharapkan yang akan mempengaruhi bentuk *surface fuzzy*. Berikut merupakan gambar *surface fuzzy* yang pertama kali digunakan pada sistem dan fungsi keanggotaannya masih kongruen, ditunjukkan pada gambar 3.8.





Gambar 3.8 Tampilan *Surface fuzzy* awal dari sistem

Dalam penelitian ini, penulis melakukan beberapa kali *manual tuning* pada nilai keanggotaan *fuzzy input* dan *output* untuk menemukan bentuk *fuzzy* yang dapat bekerja baik pada sistem yang diinginkan. Berikut merupakan hasil *surface fuzzy* yang terbaik menurut penulis yang bekerja sesuai sistem yang diinginkan setelah beberapa kali percobaan, ditunjukkan pada gambar 3.9.



Gambar 3.9 Tampilan *surface fuzzy* yang terbaik dari sistem

Dari kedua gambar *surface fuzzy* diatas, dapat dilihat keduanya cukup berbeda. Yang membedakan keduanya adalah dari pemilihan nilai keanggotaan *fuzzy* saja. Walaupun yang berbeda hanya fungsi keanggotaannya saja, namun saat diterapkan pada alat hasil yang didapatkan jauh berbeda.

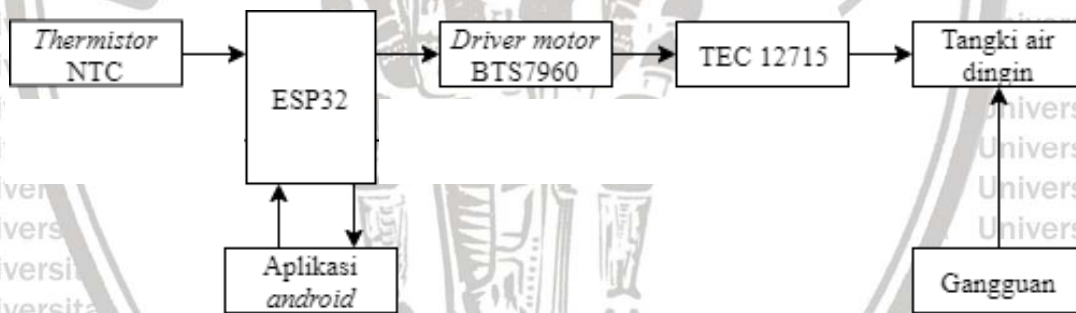
### 3.7. Perancangan Antarmuka *Internet of Things*

#### 3.7.1 Tujuan Perancangan

Perancangan antarmuka ini bertujuan untuk mengetahui proses terbentuknya tampilan aplikasi *android* yang menerapkan konsep *Internet of Things* yang nantinya mampu berinteraksi dan terhubung secara *realtime* dengan kondisi dalam dispenser air minum melalui perantara ESP32. Adapun data yang ditampilkan dalam aplikasi adalah nilai suhu air tangki dingin yang nantinya mengalami perubahan terhadap waktu. Selain itu, dari aplikasi *android* ini diberikan sebuah menu *setting* untuk mengatur nilai *setpoint* suhu air yang diinginkan. Sehingga dapat mempermudah pengguna nanti dalam *monitoring* dan mengontrol kondisi dispenser.

#### 3.7.2 Blok Diagram Komunikasi Antarmuka

Pada perancangan antarmuka, diperlukan perencanaan diagram blok yang akan digunakan oleh sistem sehingga sistem dapat diketahui secara garis besar dan diharapkan dapat bekerja sesuai dengan yang didesain. Adapun diagram blok sistem dijelaskan Gambar 3.10 sebagai berikut:



Gambar 3.10 Blok diagram komunikasi antarmuka

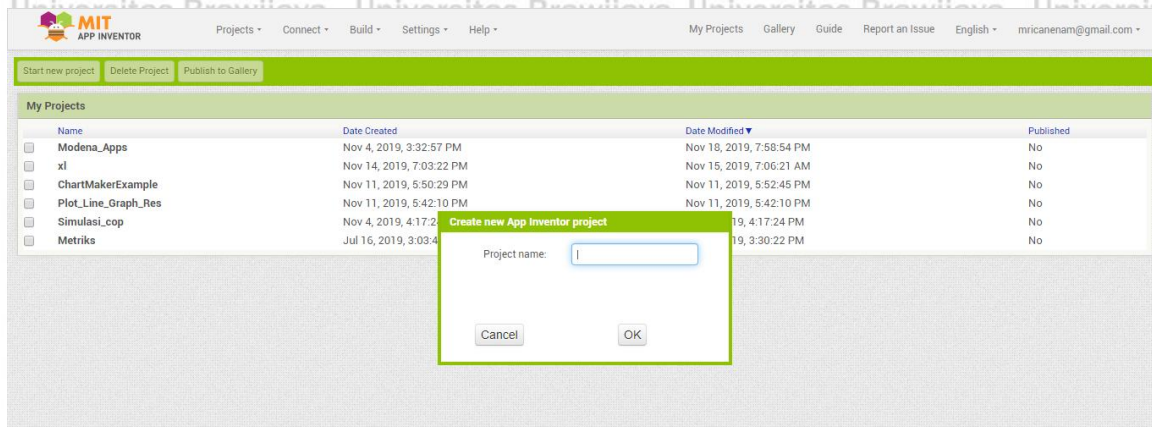
*Setpoint* yang telah dikuantisasi dalam laptop akan dikirim atau di-*upload* menuju ESP32 sebagai program dasar bagaimana perancangan bekerja. Kemudian, ESP32 akan mengondisikan *plant* tangki air dingin melalui mekanisme kerja aktuator (peltier TEC12715) sehingga didapat respon *plant* yang terbaca oleh sensor *thermistor* NTC (*Negative Temperature Coefficient*). Data yang terbaca oleh sensor *thermistor* NTC akan diolah mikrokontroler ESP32 untuk diunggah menuju aplikasi *android* yang terhubung dengan internet atau biasa digunakan untuk sistem *Internet of Things*. Sehingga dengan kemampuan ini, pengguna dapat dengan mudah mengetahui kondisi dispenser dengan parameter-parameter tertentu melalui aplikasi *android*.

#### 3.7.3 Langkah Perancangan Antarmuka

Adapun perancangan antarmuka memuat mengenai perancangan rangkaian mikrokontroler ESP32 dengan sensor, konfigurasi program aplikasi *android*, serta

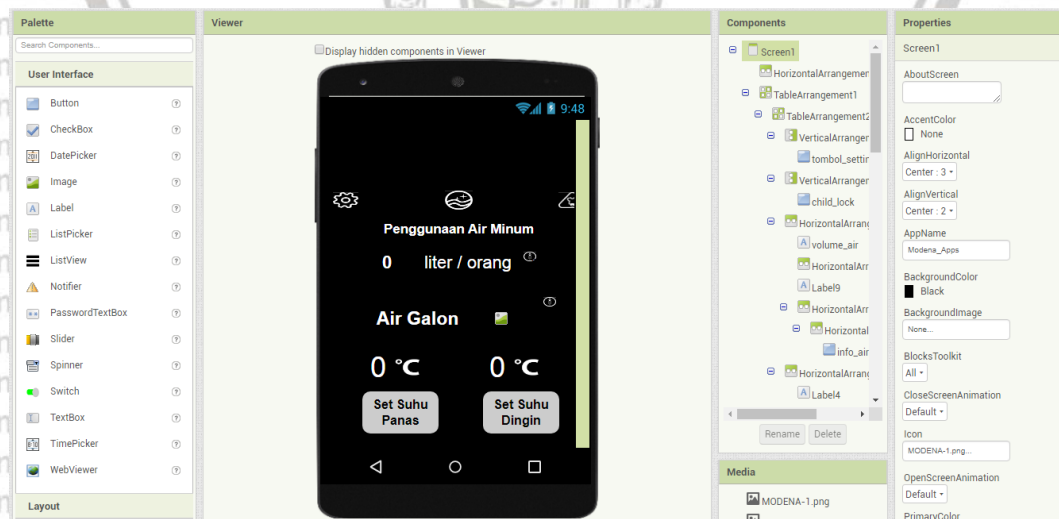
listing program sensor untuk pembacaan *Internet of Things*. Adapun konfigurasi aplikasi *android* dapat ditunjukkan pada langkah-langkah berikut:

- Buka link <https://appinventor.mit.edu/>
- Pilih menu *Create Apps* dan register seperti biasa
- Buat *project* baru di MIT *app inventor*



Gambar 3.11 Halaman awal MIT *app inventor*

- Setelah pemberian nama *project* klik OK untuk menuju halaman pembuatan *apps*.
- Di halaman selanjutnya merupakan proses pembuatan *mock up* atau desain tampilan *interface* aplikasi *android* yang akan dibuat.

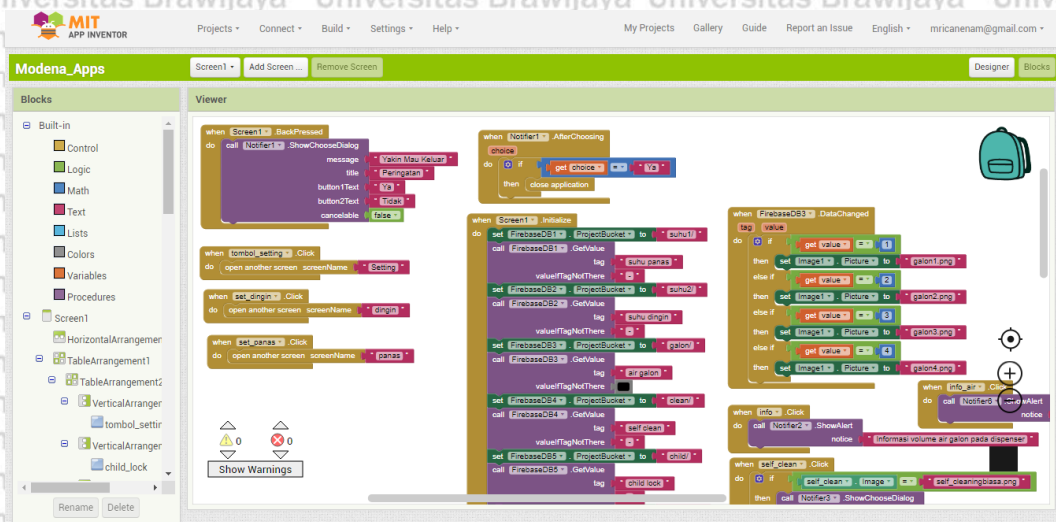


Gambar 3.12 Halaman untuk mengatur desain tampilan aplikasi *android*

- Pada halaman yang ditunjukkan pada gambar 3.12 terdapat beberapa menu dibagian kiri untuk memberikan konfigurasi-konfigurasi pada aplikasi *android* yang akan dibuat seperti tombol, notifikasi, penampil data dan lain-lain.

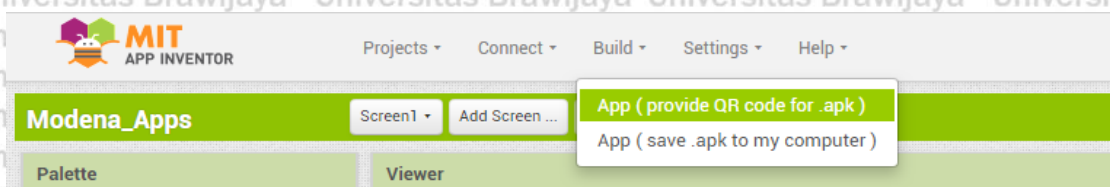


- Kemudian pada bagian kanan terdapat menu *setting* dari konfigurasi-konfigurasi yang dipilih. Terdapat pilihan untuk mengatur warna, ukuran, tulisan dan lain-lain. Semua hal itu disesuaikan dengan keinginan pembuat aplikasi *android*.



Gambar 3.13 Tampilan *block* untuk mengatur mekanisme kerja aplikasi *android*

- Setelah tampilan *mock up* aplikasi *android* sudah dibuat, selanjutnya merupakan menu *block* untuk mengatur segala sistem kerja aplikasi *android*. Sistem kerja yang dimaksud adalah konfigurasi seperti tombol, penampil data, notifikasi yang sudah dibuat seperti gambar 3.13, akan diatur *input* dan *output* sesuai keinginan pembuat.
- Dengan mengambil logika *program* pada bagian kiri gambar 3.13, pembuat hanya perlu melakukan *drag and drop* untuk membuat logika *program* untuk aplikasi *android*.
- Setelah *mock up* dan logika *program* aplikasi *android* sudah selesai, selanjutnya mengekspor ke format file *.apk*. Hal itu didapatkan dengan memilih menu *Build* untuk menghasilkan file aplikasi *android*. Pembuat dapat memilih salah satu cara dari QR code atau file format *.apk* yang disimpan pada *personal computer* nanti.



Gambar 3.14 Tampilan menu untuk menyelesaikan pembuatan aplikasi *android*

Adapun detail *listing program* yang digunakan mikrokontroler ESP32 untuk mengoperasikan sistem *Internet of Things* pada aplikasi *android* dari MIT *app inventor* telah terdapat pada lampiran. Aplikasi *android* ini juga terintegrasi dengan *database*

*Firestore* sebagai penyimpanan data dan penghubung antara aplikasi *android* dengan mikrokontroler ESP32. Adapun deklarasi yang perlu dijelaskan adalah letak pin data, nama SSID dan password Wi-Fi yang digunakan untuk mengunggah data ke *database Firestore*, serta *authentication code* yang telah terdapat pada *Firestore account*.

Hasil pembacaan sensor *thermistor* NTC akan dikirim ke *database Firestore* terlebih dahulu melalui koneksi internet. Hal ini terjadi sebab pada program telah dimasukkan *authentication code* serta SSID dan password internet yang digunakan.

Kemudian aplikasi *android* yang telah dibuat akan menerima data dari *database Firestore* dan ditampilkan pada *interface* aplikasi *android*. Sehingga dari sistem pengiriman data tersebut, pengguna dapat melakukan *monitoring* dan mengontrol kondisi dispenser secara *realtime*.

### 3.8. Pengujian Alat

Untuk memastikan bahwa sistem ini berjalan sesuai yang direncanakan maka perlu dilakukan pengujian alat meliputi perangkat keras (*hardware*) yang dilakukan baik per blok maupun keseluruhan sistem.

#### 3.8.1 Pengujian Tiap Blok

Pengujian per blok dilakukan dengan tujuan untuk menyesuaikan nilai masukan dan nilai keluaran tiap-tiap blok sesuai dengan perancangan yang dilakukan sebelumnya.

#### 3.8.2 Pengujian Keseluruhan Sistem

Pengujian sistem secara keseluruhan dilakukan dengan tujuan untuk mengetahui unjuk kerja alat setelah perangkat keras dan perangkat lunak diintegrasikan bersama.

#### 3.8.3 Rancangan Pengujian

Rancangan pengujian ini dibuat agar dapat mempermudah dalam proses pengujian alat. Rancangan percobaan akan dibuat sampai tercapai spesifikasi sistem yang telah ditetapkan diawal. Adapun rancangan pengujian yang dibuat adalah sebagai berikut:

1. Menguji pencapaian panas dispenser konvensional.
2. Dilanjutkan dengan menguji tiap blok dimana tiap blok akan dilakukan karakterisasi yaitu bagaimana masukan dan keluaran *fuzzy*, *driver* BTS7960, dan peltier TEC12715.
3. Setelah keseluruhan sistem selesai dibuat dan disatukan akan dilakukan pengujian sistem dengan rancangan *fuzzy* yang ada dengan masukan sistem atau *setpoint* 17°C, 19°C, dan 21°C dan akan dilihat suhu respon keluaran sistem.

4. Setelah selesai satu kali pengujian, air dalam tangki akan dikeluarkan kurang lebih 360 ml dan akan dilakukan pengujian seperti langkah sebelumnya. Pengujian akan diulang 3 kali. Setelah mendapat data hasil pengujian dilakukan pengolahan data dan dibuat grafik. Dalam pengolahan data ini akan dihitung rata-rata suhu stabil dan dihitung berapa besar *error* suhu yang didapat dan juga berapa lama suhu tersebut bisa tercapai dan juga akan dilihat apakah ada *overshoot*.
5. Jika nilai *error*, *settling time*, dan *maximum overshoot* masih belum sesuai dengan spesifikasi sistem yang dibuat, maka dilakukan penalaan ulang terhadap kontroler logika *fuzzy* dan dilakukan pengulangan pengujian nomer 3 dan 4 sampai diperoleh nilai *error*, *settling time*, dan *maximum overshoot* yang sesuai dengan spesifikasi sistem



## BAB IV HASIL DAN PEMBAHASAN

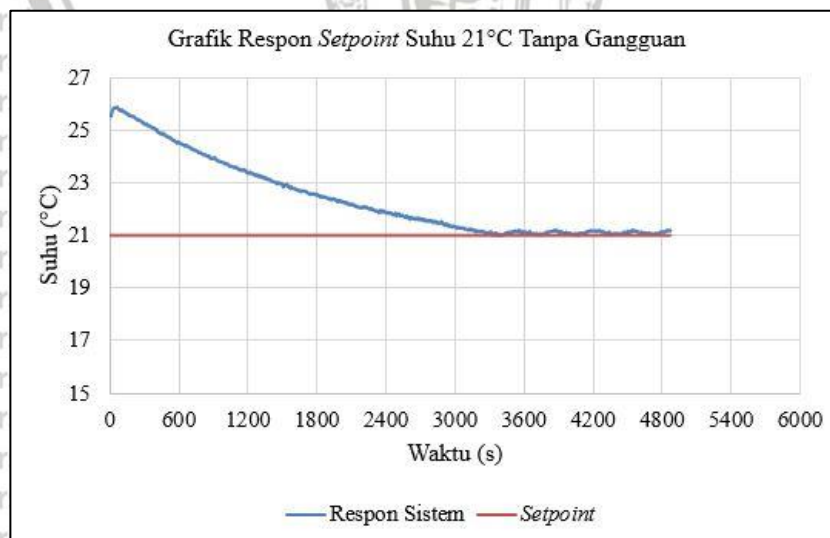
Pada bab ini hasil dan pembahasan dilakukan dengan melakukan pengujian sistem secara keseluruhan. Pengujian yang dilakukan meliputi dua pengujian dengan tiga variasi *setpoint* suhu, yakni pengujian dengan kontroler tanpa gangguan dan dengan kondisi diberikan gangguan. Pengujian yang dilakukan bertujuan untuk menguji apakah sistem hasil perancangan sudah sesuai dengan spesifikasi desain yang diinginkan. Selain itu, juga terdapat penjelasan tentang antarmuka IoT (*Internet of Things*) yang diterapkan pada sistem ini.

### 4.1 Pengujian Keseluruhan Sistem Tanpa Gangguan

Pengujian sistem ini dilakukan untuk mengetahui performansi sistem dengan nilai *setpoint* suhu 21°C, 19°C, dan 17°C. Prosedur pada pengujian ini terdapat pada bab 3.8. Penentuan nilai *setpoint* berdasarkan faktor titik minimal suhu yang dapat dicapai dengan peltier dan suhu luar disekitar tangki air dingin.

#### 4.1.1 Pengujian Sistem dengan Suhu *Setpoint* 21°C

Grafik respon sistem pendingin air dispenser pada tangki air dingin dengan *setpoint* 21°C ditunjukkan pada gambar 4.1.



Gambar 4.1 Respon sistem dengan *setpoint* 21°C tanpa diberikan gangguan

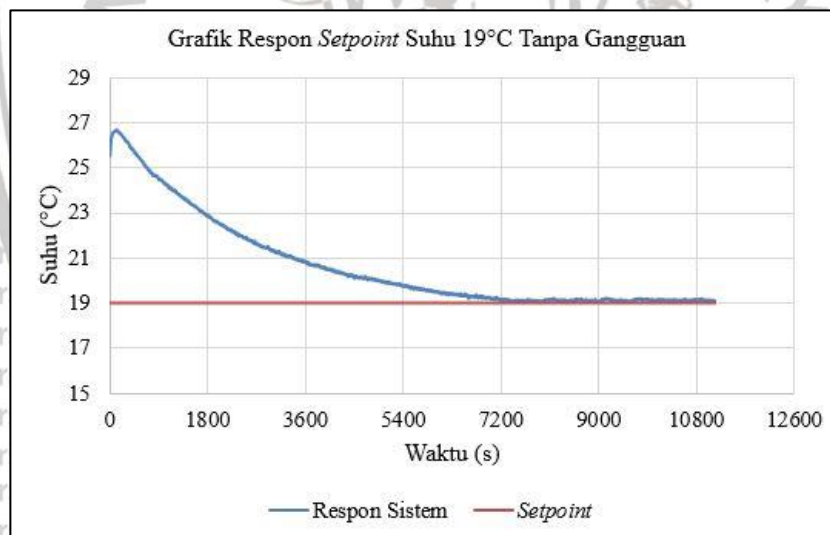
Berdasarkan respon sistem pada gambar 4.1. Didapatkan nilai *settling time* ( $t_s$ ) sebesar kurang lebih 3.406 detik atau sekitar 56,76 menit, *error steady state* 0,47%. dan *overshoot* tidak ada. Tidak adanya *overshoot* disebabkan ketika respon sistem

mendekati nilai *setpoint*, sistem akan mulai menstabilkan nilai yang dikeluarkan oleh *driver* BTS7960 sebagai aktuator untuk mempertahankan sesuai nilai *setpoint* yang ditentukan. Selain itu, ketika sistem mulai dinyalakan sampai dengan beberapa saat, terjadi sedikit lonjakan keatas nilai suhu. Hal itu dipengaruhi oleh sisi panas peltier TEC12715 yang cepat memanaskan ketika diberikan tegangan *input*, sehingga akan mempengaruhi suhu pada sisi dingin peltier.

$$e_{ss}(\%) = \frac{|21,1 - 21|}{21} \times 100\% \\ = 0,47\%$$

#### 4.1.2 Pengujian Sistem dengan Suhu *Setpoint* 19°C

Grafik respon sistem pendingin air dispenser pada tangki air dingin dengan *setpoint* 19°C ditunjukkan pada gambar 4.2.



Gambar 4.2 Respon sistem dengan *setpoint* 19°C tanpa diberikan gangguan

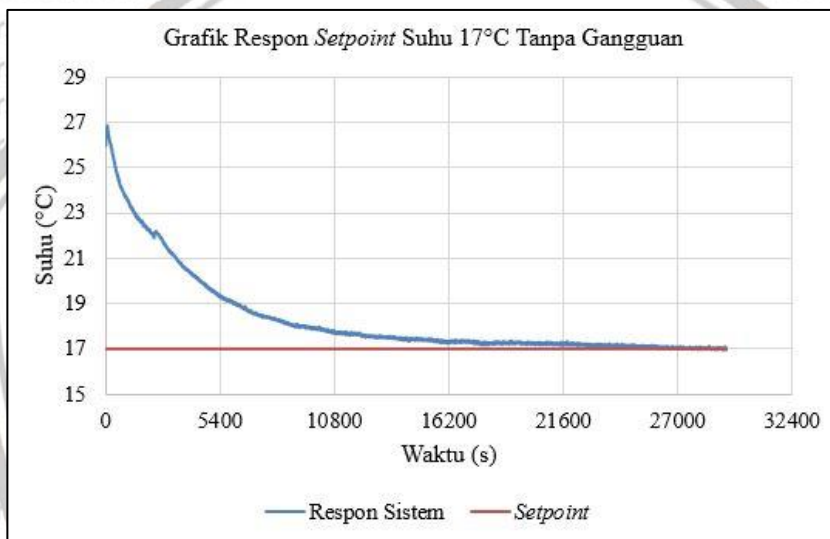
Berdasarkan respon sistem pada gambar 4.1. Didapatkan nilai *settling time* (ts) sebesar kurang lebih 7.395 detik atau sekitar 123,25 menit, *error steady state* 0,57%. dan *overshoot* tidak ada. Tidak adanya *overshoot* disebabkan ketika respon sistem mendekati nilai *setpoint*, sistem akan mulai menstabilkan nilai yang dikeluarkan oleh *driver* BTS7960 sebagai aktuator untuk mempertahankan sesuai nilai *setpoint* yang ditentukan. Selain itu, ketika sistem mulai dinyalakan sampai dengan beberapa saat, terjadi sedikit lonjakan keatas nilai suhu. Hal itu dipengaruhi oleh sisi panas peltier

TEC12715 yang cepat memanaskan ketika diberikan tegangan *input*, sehingga akan mempengaruhi suhu pada sisi dingin peltier.

$$e_{ss}(\%) = \frac{|19,11 - 19|}{19} \times 100\% = 0,57\%$$

### 4.1.3 Pengujian Sistem dengan Suhu *Setpoint* 17°C

Grafik respon sistem pendingin air dispenser pada tangki air dingin dengan *setpoint* 17°C ditunjukkan pada gambar 4.3.



Gambar 4.3 Respon sistem dengan *setpoint* 17°C tanpa diberikan gangguan

Berdasarkan respon sistem pada gambar 4.3. Didapatkan nilai *settling time* (ts) sebesar kurang lebih 24.208 detik atau sekitar 403,46 menit, *error steady state* 0,29%.

dan *overshoot* tidak ada. Tidak adanya *overshoot* disebabkan ketika respon sistem mendekati nilai *setpoint*, sistem akan mulai menstabilkan nilai yang dikeluarkan oleh *driver* BTS7960 sebagai aktuator untuk mempertahankan sesuai nilai *setpoint* yang ditentukan. Selain itu, ketika sistem mulai dinyalakan sampai dengan beberapa saat, terjadi sedikit lonjakan keatas nilai suhu. Hal itu dipengaruhi oleh sisi panas peltier

TEC12715 yang cepat memanaskan ketika diberikan tegangan *input*, sehingga akan mempengaruhi suhu pada sisi dingin peltier.

$$e_{ss}(\%) = \frac{|17,05 - 17|}{17} \times 100\% = 0,29\%$$

Dari hasil pengujian yang telah dilakukan secara *close loop*, dapat ditampilkan dalam bentuk data seperti Tabel 4 untuk mempermudah dalam mengamati performansi respon sistem yang telah didapatkan. Berikut data yang dimaksud:

Tabel 4  
Data Hasil Pengujian Sistem Secara Keseluruhan

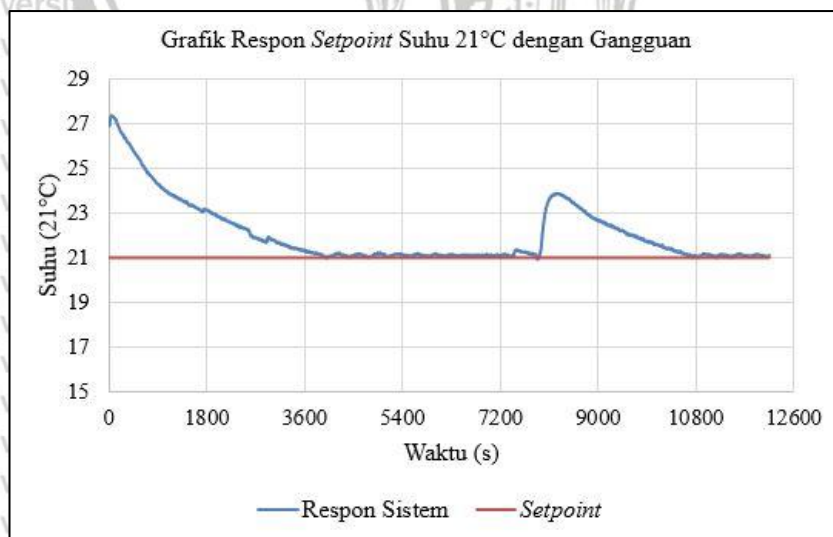
| Setpoint | Settling Time (menit) | Error Steady State (%) | Overshoot (%) |
|----------|-----------------------|------------------------|---------------|
| 21°C     | 56,76                 | 0,47                   | -             |
| 19°C     | 123,25                | 0,57                   | -             |
| 17°C     | 403,46                | 0,29                   | -             |

#### 4.2 Pengujian Keseluruhan Sistem dengan Gangguan

Pengujian sistem ini dilakukan untuk mengetahui performansi sistem dengan nilai *setpoint* suhu 21°C, 19°C, dan 17°C. Prosedur pada pengujian ini terdapat pada bab 3.8. Penentuan nilai *setpoint* berdasarkan faktor titik minimal suhu yang dapat dicapai dengan peltier dan suhu luar disekitar tangki air dingin. Pemberian gangguan bertujuan untuk mengetahui nilai *recovery time* pada sistem ini.

##### 4.2.1 Pengujian Sistem dengan Suhu *Setpoint* 21°C yang Diberikan Gangguan

Grafik respon sistem pendingin air dispenser pada tangki air dingin dengan *setpoint* 21°C yang diberikan gangguan ditunjukkan pada gambar 4.4.



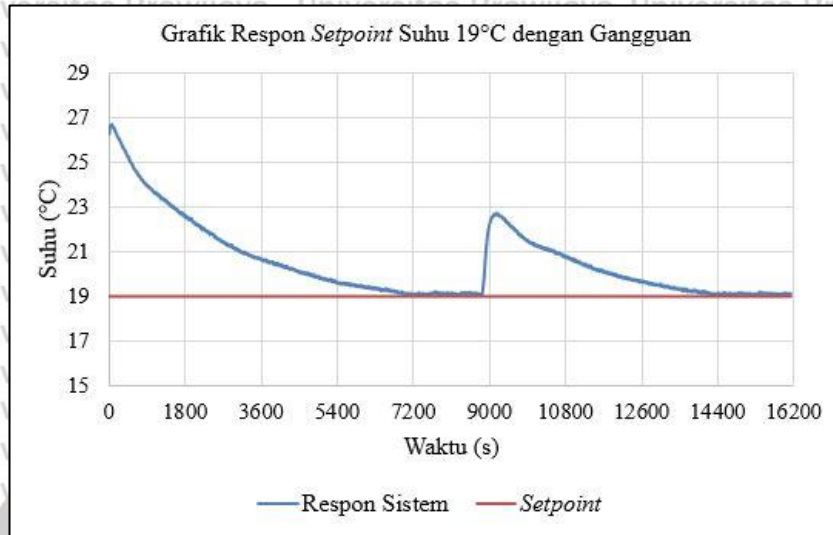
Gambar 4.4 Respon sistem dengan *setpoint* 21°C yang diberikan gangguan

Berdasarkan hasil dari respon sistem pada gambar 4.4, didapatkan nilai *recovery time* yaitu sebesar 2.264 detik atau sebesar 37,73 menit.



#### 4.2.2 Pengujian Sistem dengan Suhu *Setpoint* 19°C yang Diberikan Gangguan

Grafik respon sistem pendingin air dispenser pada tangki air dingin dengan *setpoint* 19°C yang diberikan gangguan ditunjukkan pada gambar 4.5.

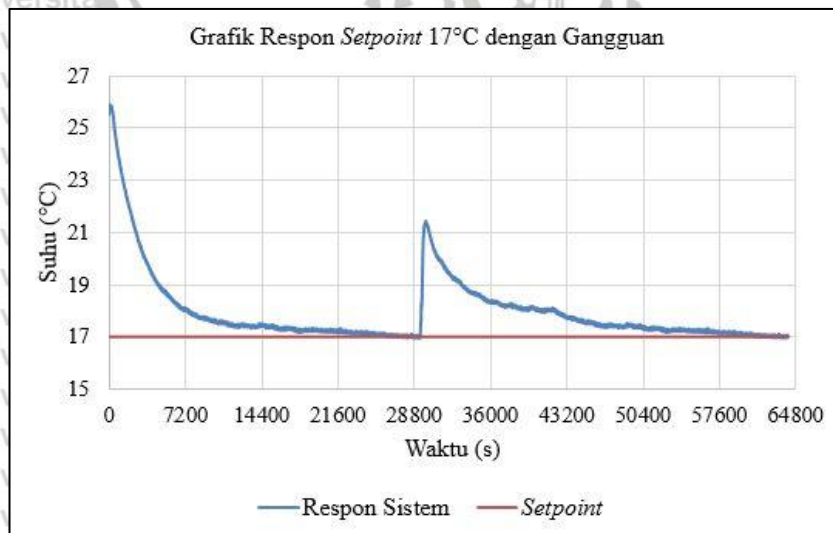


Gambar 4.5 Respon sistem dengan *setpoint* 19°C yang diberikan gangguan

Berdasarkan hasil dari respon sistem pada gambar 4.5, didapatkan nilai *recovery time* yaitu sebesar 4.988 detik atau sebesar 83,13 menit.

#### 4.2.3 Pengujian Sistem dengan Suhu *Setpoint* 17°C yang Diberikan Gangguan

Grafik respon sistem pendingin air dispenser pada tangki air dingin dengan *setpoint* 17°C yang diberikan gangguan ditunjukkan pada gambar 4.6.



Gambar 4.6 Respon sistem dengan *setpoint* 17°C yang diberikan gangguan



Berdasarkan hasil dari respon sistem pada gambar 4.5, didapatkan nilai *recovery time* yaitu sebesar 30.590 detik atau sebesar 509,83 menit.

Dari hasil pengujian yang telah dilakukan secara *close loop* dan diberikannya gangguan pada sistem, dapat ditampilkan dalam bentuk data seperti Tabel 5 untuk mempermudah dalam mengamati performansi respon sistem yang telah didapatkan.

Berikut data yang dimaksud:

Tabel 5

Data Hasil Pengujian Sistem Secara Keseluruhan dengan Diberikan Gangguan

| Setpoint | Recovery Time (menit) |
|----------|-----------------------|
| 21°C     | 37,73                 |
| 19°C     | 83.13                 |
| 17°C     | 509,83                |

#### 4.3 Antarmuka Aplikasi (*Software*) *Internet of Things*

Pada penelitian ini memberikan sebuah sistem yang memanfaatkan teknologi *Internet of Things* (IoT) untuk *monitoring* dan mengontrol kondisi dispenser. Dengan memanfaatkan media sebuah aplikasi, pengguna dapat *me-monitoring* dan mengatur suhu air dingin pada dispenser secara *realtime*. Berikut merupakan antarmuka yang dimaksud seperti pada gambar 4.7.



Gambar 4.7 Antarmuka *monitoring* dari aplikasi (*software*) *Internet of Things*

Pembuatan aplikasi ini menggunakan *platform open source* yaitu MIT App Inventor, sebagai media pembuatan *front end* dan *back end* aplikasi. Sistem *coding* yang digunakan adalah sistem *drag and drop*, sehingga cukup mudah bagi pemula untuk membuat *logic* pada aplikasinya. Dalam proses penampungan data sensor suhu, pada penelitian ini menggunakan *Realtime Database Firebase* yang merupakan lisensi *database* dari Google. *Firebase* ini juga *platform yang open source*, sehingga mudah dalam penggunaannya. Adapun merupakan tampilan antarmuka untuk mengatur besar nilai suhu air dingin pada dispenser yang ditunjukkan pada gambar 4.8 sebagai berikut:



Gambar 4.8 Antarmuka *setting* suhu pada aplikasi *Internet of Things*

Pada gambar 4.8 suhu air dingin dapat diatur per-nilai angka kelipatan satu, sehingga pengguna dapat menyesuaikan sesuai kebutuhan dengan keakuratan nilai suhu yang cukup baik. Pengguna dapat mengatur nilai suhu air dingin secara *wireless*, sehingga mempermudah dalam pengontrolan dispenser.

Selain itu, terkait pengembangan aplikasi ini, dapat berupa adanya pemanfaatan *machine learning* dalam menentukan pola minum air pada pengguna yang menggunakan dispenser, sehingga dispenser dapat secara otomatis mengatur suhu air yang sering digunakan pengguna. Hal itu dapat mengefisienkan waktu, agar pengguna tidak menunggu proses pendinginan air yang cukup lama. Selain itu, dengan adanya teknologi *Internet of Things* dapat dikembangkan dengan penambahan sensor-sensor pada dispenser air minum. Mengingat konsep *Internet of Things* ini adalah menghubungkan sesuatu yang berpotensi dikoneksikan secara *wireless* guna mempermudah suatu pekerjaan manusia





## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan hasil pengujian tiap bagian dan keseluruhan system yang telah dilakukan, maka kesimpulan yang dapat diperoleh dari penelitian ini adalah sebagai berikut:

1. Implementasi Kontrol Logika *Fuzzy* (KLF) pada tangki air dingin dispenser air minum yang telah dibuat dapat berfungsi dengan baik. Dengan adanya sistem pengontrolan ini, dispenser mampu mengatur dan menjaga suhu air sesuai dengan *setpoint* dan memiliki *error steady state* yang kecil.
2. Kontrol Logika *Fuzzy* yang dirancang memiliki 2 fungsi keanggotaan masukan, yaitu *error* suhu dan *delta error* suhu, serta 1 fungsi keanggotaan keluaran untuk parameter PWM pada *driver* BTS7960. Langkah berikutnya dalam merancang KLF adalah membuat aturan *fuzzy* (*Fuzzy Rule*) yang menggunakan metode inferensi *min-max* untuk visualisasi dari masukan ke keluaran. Langkah terakhir yaitu dengan proses defuzzifikasi menggunakan metode *Mean of Max* dimana pada metode ini akan dipilih nilai *crisp* untuk mewakili hasil dari inferensi *min-max*.
3. Pada pengujian tanpa gangguan, didapatkan performansi respon sistem yang sesuai dengan spesifikasi sistem yang diinginkan. Selain itu, sistem juga dapat bertahan pada gangguan yang diberikan dan mampu mengembalikan nilai suhu air dingin sesuai dengan nilai *setpoint*. Dari hasil pengujian didapatkan bahwa nilai *settling time* dari respon cukup lama, hal itu dikarenakan bukan dari Kontrol Logika *Fuzzy* melainkan dari pemilihan aktuator dan sensor.
4. Penerapan konsep teknologi *Internet of Things* pada dispenser ini, dapat bekerja dengan baik. *Monitoring* dan pemilihan nilai *setpoint* suhu air dingin dapat dilakukan secara *wireless* dan *realtime* melalui aplikasi android yang telah dibuat.

#### 5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, saran yang diberikan untuk penelitian selanjutnya adalah dengan memperhatikan kondisi suhu ruang atau suhu sekitar tangki air dingin dispenser selama proses pengendalian. Karena kemampuan sistem pendinginan air menggunakan peltier sangat berpengaruh dengan suhu sekitar. Penulis berharap antarmuka aplikasi yang telah dibuat, dapat dikembangkan lanjutkan guna mewujudkan *Smart Home Appliance* pada peralatan rumah tangga terutama dispenser air minum.



## DAFTAR PUSTAKA

- Mar'atus Arifah. (2016). Dispenser Pemantau Konsumsi Air Minum Berbasis Mikrokontroler Atmega328P. Proyek Akhir UNY.
- Departemen Kesehatan Republik Indonesia. (2006). Pengendalian Penyakit dan Penyehatan Lingkungan Tahun 2005. Dirjen PP&PL, Jakarta.
- \_\_\_\_\_, ESP32 Series Datasheet: *ESP32-D2WD*
- Putra, Y. A., dan Prawioredjo, K. (2017). Pengatur Temperatur Pada Dispenser Berbasis Arduino Mega 2560. *JETri*.
- Safrudin, M. (2017). Peningkatan Efisiensi Dispenser Air Melalui Penambahan Isolasi pada Tabung Air Panas. Bandung: Universitas Pasundan.
- Ross, T. J. (2004). *Fuzzy Logic with Engineering Applications Second Edition*. England: John Wiley & Sons Ltd.
- Hariyanto, D. (2015). Teknik Antarmuka *Analog to Digital Converter*. Yogyakarta: Universitas Negeri Yogyakarta.
- Abidatul A. (2017). Sistem Tekanan Mekanik Berbasis Mikrokontroler AT-Mega 16 Untuk Pembuat Kerupuk Pelompong Guna Menunjang Produksi *Home Industry* Barokah di Tuban Jawa Timur. Proyek Akhir UNY.
- Al Faizin, S. A., dan Pramudijanto, J. (2013). Desain dan Implementasi Kontroler *Fuzzy-Supervised PID* Berbasis PLC pada Sistem Kontrol Level Cairan *Coupled-Tank*. *Jurnal Teknik ITS*, 2, F318-F323.
- Benny, *et.al*. (2015). "*Smart Dispenser*" Dispenser Pintar Dengan Pengontrol Suhu dan Penghematan Energi. *Jurnal Poli-Teknologi PNJ*. Vol.14, Nomor 2.
- Ilyas, Afshan., Jahan, Shagufta., Ayyub, Mohammad, (2013). *Tunning Of Conventional PID and Fuzzy Logic Control Using Different Defuzzification Technique*. *International Journal Of Siencetific & Technology Reaserch* Volume 2.
- Fery Ariyanto, (2018). Rancang Bangun Dispenser Dengan Pengaturan Suhu Berbasis Arduino. Proyek Tugas Akhir, Universitas Teknologi Yogyakarta.



Jenny, *et.al.* (2016). Studi Penggunaan Modul Thermoelektrik Sebagai Sistem Pendingin Portable. *Jurnal Ilmiah Mahasiswa Teknik Mesin*. Vol.1, Nomor 1.

Handayani, T. P. (2015). Rancang Bangun Sistem Keamanan Pintu Rumah Menggunakan *Switch* Magnetik dengan Monitoring *Web Bootsrap* Berbasis *Raspberry PI*. Palembang: Politeknik Negeri Sriwijaya.

Kitoma Indonesia. (2019, Oktober 9). *Solenoid valve pneumatic*, prinsip kerja. Retrieved from [www.kitomaindonesia.com](http://www.kitomaindonesia.com/article/9/solenoid-valve-pneumatic-prinsip-kerja):<http://www.kitomaindonesia.com/article/9/solenoid-valve-pneumatic-prinsip-kerja>

UNIVERSITAS BRAWIJAYA



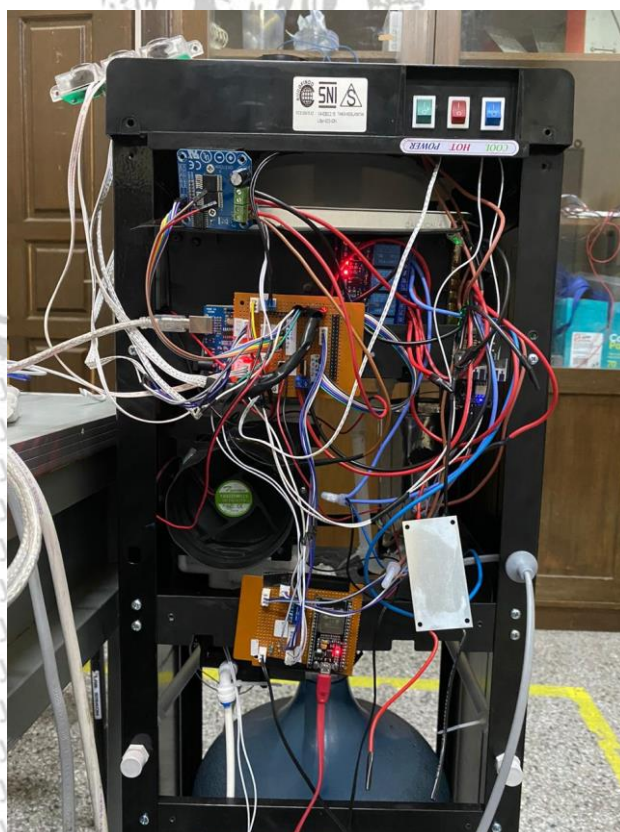
# LAMPIRAN 1 DOKUMENTASI ALAT



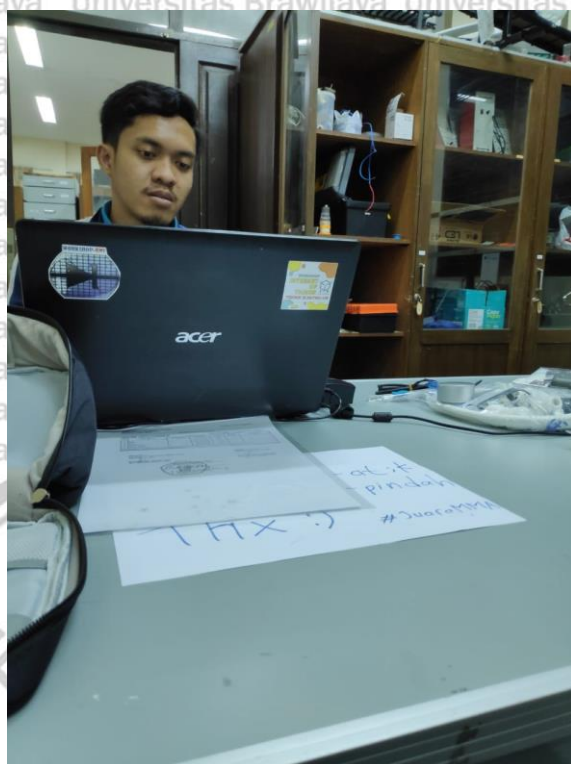




Gambar 1. Tampak depan dispenser pada kondisi awal (belum dibongkar)



Gambar 2. Tampak belakang kondisi dispenser yang sudah dibongkar dan dimodifikasi



Gambar 3. Dokumentasi pada saat pengambilan data karakterisasi



Gambar 4. Dokumentasi pada saat pengambilan data pengujian sistem

UNIVERSITAS BRAWIJAYA





# LAMPIRAN 2 DATASHEET



Datasheet driver BTS7960 :

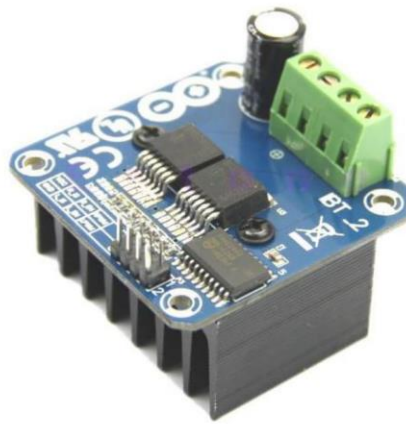


# Handson Technology

[User Guide](#)

## BTS7960 High Current 43A H-Bridge Motor Driver

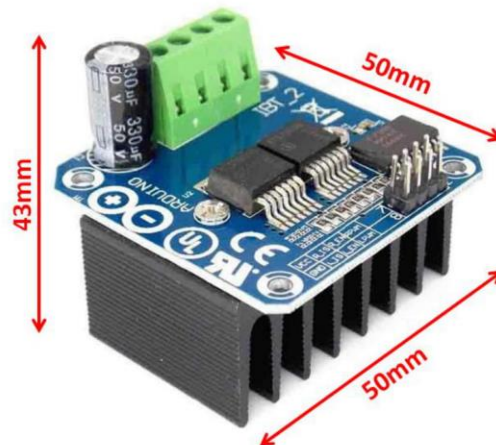
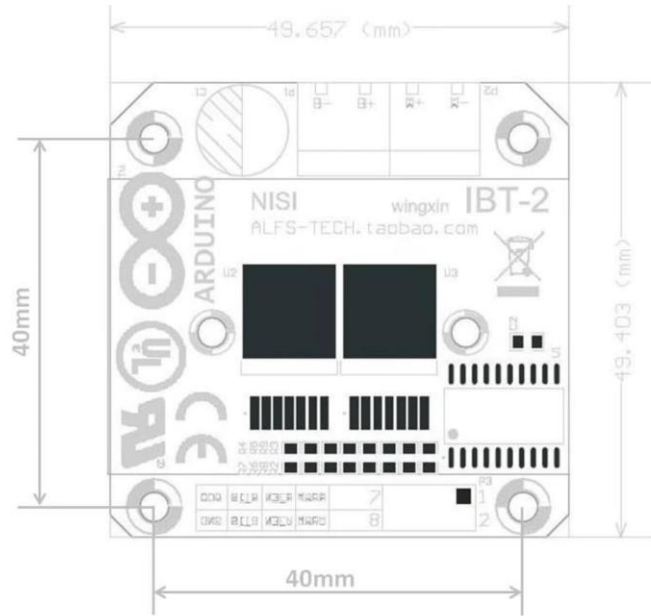
The BTS7960 is a fully integrated high current H bridge module for motor drive applications. Interfacing to a microcontroller is made easy by the integrated driver IC which features logic level inputs, diagnosis with current sense, slew rate adjustment, dead time generation and protection against overtemperature, overvoltage, undervoltage, overcurrent and short circuit. The BTS7960 provides a cost optimized solution for protected high current PWM motor drives with very low board space consumption.

**SKU: [DRV-1012](#)****Brief Data:**

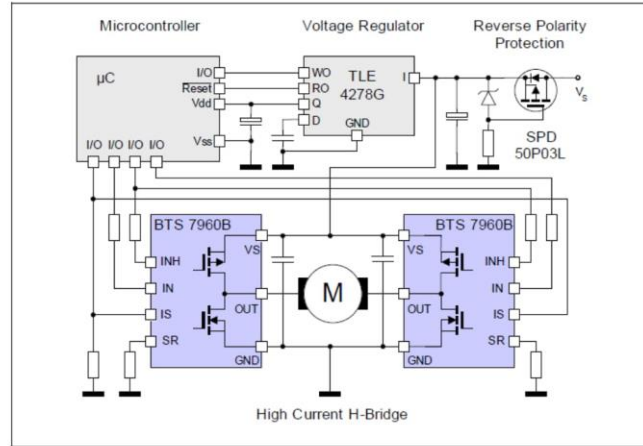
- Input Voltage: 6 ~ 27Vdc.
- Driver: Dual BTS7960 H Bridge Configuration.
- Peak current: 43-Amp.
- PWM capability of up to 25 kHz.
- Control Input Level: 3.3~5V.
- Control Mode: PWM or level
- Working Duty Cycle: 0 ~100%.
- Over-voltage Lock Out.
- Under-voltage Shut Down.
- Board Size (LxWxH): 50mm x 50mm x 43mm.
- Weight: ~66g.



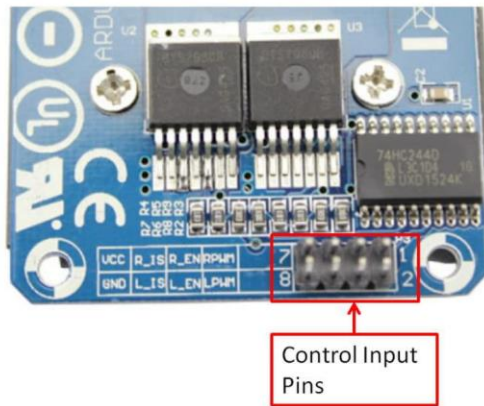
**Mechanical Dimension:**



**Schematic Diagram:**



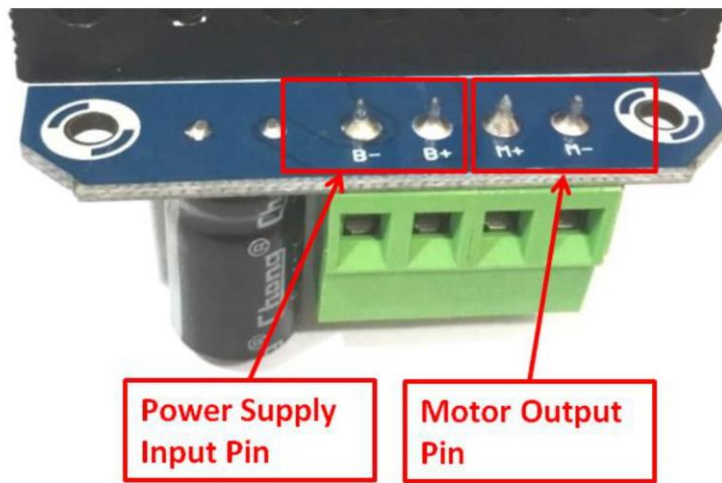
**Control Input Pin Function:**



| Pin No | Function | Description  |
|--------|----------|--|
| 1      | RPWM     | Forward Level or PWM signal, Active High             |
| 2      | LPWM     | Reverse Level or PWM signal, Active High             |
| 3      | R_EN     | Forward Drive Enable Input, Active High/ Low Disable |
| 4      | L_EN     | Reverse Drive Enable Input, Active High/Low Disable  |
| 5      | R_IS     | Forward Drive, Side current alarm output             |
| 6      | L_IS     | Reverse Drive, Side current alarm output             |
| 7      | Vcc      | +5V Power Supply microcontroller                     |
| 8      | Gnd      | Ground Power Supply microcontroller                  |



**Motor Power Supply & Output Pin Assignment:**



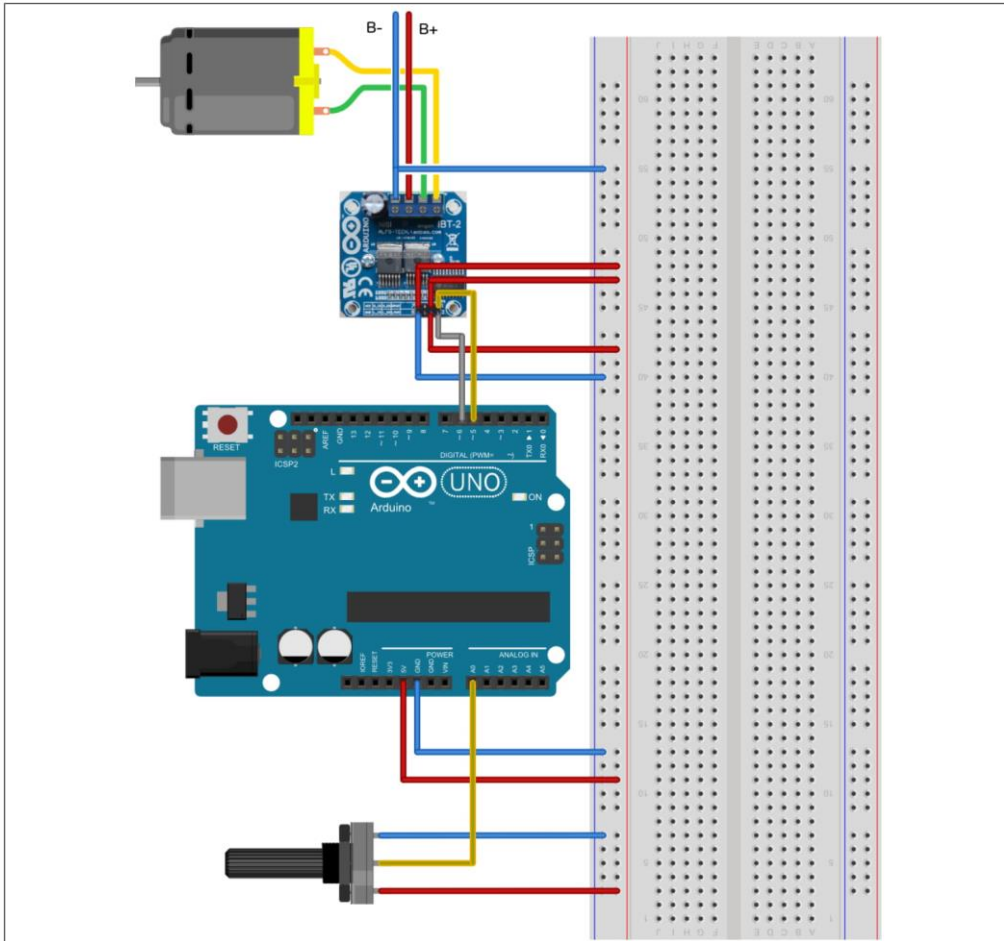
| Pin No | Function | Description                            |
|--------|----------|--|
| 1      | B+       | Positive Motor Power Supply. 6 ~ 27VDC |
| 2      | B-       | Negative Motor Power Supply. Ground    |
| 3      | M+       | Motor Output +                         |
| 4      | M-       | Motor Output -                         |

**Controlling DC Motor with BTS7960 Using Arduino:**

Below is the circuit connection using BTS7960 high power driver to control one DC motor with Arduino board. The potentiometer allows the control of motor speed and rotation direction of the motor.







**Arduino to BST7960 Connecting Table:**

| BST 7960 Pin | Arduino Pin  |
|--------------|--------------|
| 1 (RPWM)     | D5           |
| 2 (LPWM)     | D6           |
| 3 (R_EN)     | Arduino 5V   |
| 4 (L_EN)     | Arduino 5V   |
| 5 (R_IS)     | Un-connected |
| 6 (L_IS)     | Un-connected |
| 7 (VCC)      | Arduino 5V   |
| 8 (GND)      | Arduino GND  |



**Sketch Listing:**

Upload the following sketch to Arduino board. Try to turn the potentiometer clock-wise and anti-clock-wise and observe how the motor turn.

```
/*=====
// Author      : Handson Technology
// Project     : BTS7960 Motor Control Board driven by Arduino.
// Description  : Speed and direction controlled by a potentiometer attached
//               to analog input A0. One side pin of the potentiometer (either one) to
//               ground; the other side pin to +5V
// Source-Code : BTS7960.ino
// Program: Control DC motors using BTS7960 H Bridge Driver.
//=====
// Connection to the BTS7960 board:
// BTS7960 Pin 1 (RPWM) to Arduino pin 5(PWM)
// BTS7960 Pin 2 (LPWM) to Arduino pin 6(PWM)
// BTS7960 Pin 3 (R_EN), 4 (L_EN), 7 (VCC) to Arduino 5V pin
// BTS7960 Pin 8 (GND) to Arduino GND
// BTS7960 Pin 5 (R_IS) and 6 (L_IS) not connected
*/

int SENSOR_PIN = 0; // center pin of the potentiometer

int RPWM_Output = 5; // Arduino PWM output pin 5; connect to IBT-2 pin 1 (RPWM)
int LPWM_Output = 6; // Arduino PWM output pin 6; connect to IBT-2 pin 2 (LPWM)

void setup()
{
  pinMode(RPWM_Output, OUTPUT);
  pinMode(LPWM_Output, OUTPUT);
}

void loop()
{
  int sensorValue = analogRead(SENSOR_PIN);

  // sensor value is in the range 0 to 1023
  // the lower half of it we use for reverse rotation; the upper half for forward
  rotation
  if (sensorValue < 512)
  {
    // reverse rotation
    int reversePWM = -(sensorValue - 511) / 2;
    analogWrite(LPWM_Output, 0);
    analogWrite(RPWM_Output, reversePWM);
  }
  else
  {
    // forward rotation
    int forwardPWM = (sensorValue - 512) / 2;
    analogWrite(LPWM_Output, forwardPWM);
    analogWrite(RPWM_Output, 0);
  }
}
}
```





**Web Resources:**

- <https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-1298n-pwm-h-bridge/>
- <http://www.labelektronika.com/2016/09/high-current-motor-driver-1bt-2-arduino.html>



# Handsontec.com

We have the parts for your ideas

HandsOn Technology provides a multimedia and interactive platform for everyone interested in electronics. From beginner to diehard, from student to lecturer. Information, education, inspiration and entertainment. Analog and digital, practical and theoretical; software and hardware.



HandsOn Technology support Open Source Hardware (OSHW) Development Platform.

## Learn : Design : Share

[www.handsontec.com](http://www.handsontec.com)



The Face behind our product quality...

In a world of constant change and continuous technological development, a new or replacement product is never far away – and they all need to be tested.

Many vendors simply import and sell without checks and this cannot be the ultimate interests of anyone, particularly the customer. Every part sell on Handsontec is fully tested. So when buying from Handsontec products range, you can be confident you're getting outstanding quality and value.

We keep adding the new parts so that you can get rolling on your next project.



www.handsontec.com

Breakout Boards & Modules



Connectors



www.handsontec.com

Electro-Mechanical Parts



Engineering Material



www.handsontec.com

Mechanical Hardware



Electronics Components



www.handsontec.com

Power Supply



Arduino Board & Shield

Tools & Accessory

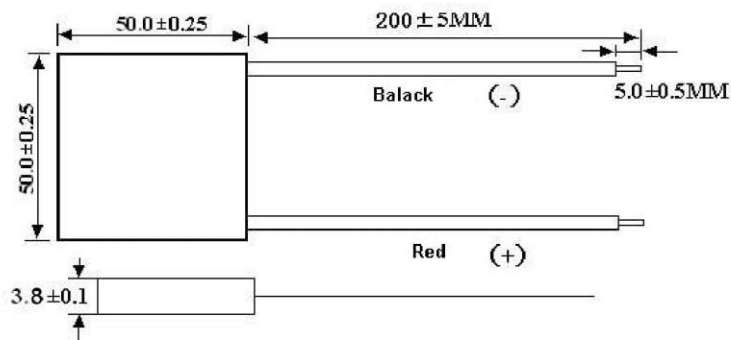


Tools & Accessory



Hebei I.T. (Shanghai) Co., Ltd

**TEC1-12715 Datasheet**



| No. | Items               | Symbol | Parameter | Condition                     |
|-----|---------------------|--------|-----------|-------------------------------|
| 1   | Max. Operating Tem. | T      | <90°C     |                               |
| 2   | Max. Cooling Power  | Qmax   | 136       | Vacuum testing Tem<br>Th=30°C |
| 3   | Tem. Difference Max | ΔTmax  | 70        | Vacuum testing Tem<br>Th=30°C |
| 4   | Input Voltage Max   | Vmax   | 15.4      | Vacuum testing Tem<br>Th=30°C |
| 5   | Max. Current        | Imax   | 15.0      | Vacuum testing Tem<br>Th=30°C |
| 6   | Resistance          | R      | 0.75±0.05 | Ambient Tem<br>Th=25°C        |
| 7   | Parallel            |        | ≤0.05 mm  |                               |
| 8   | Lines               |        | 20AWG     |                               |

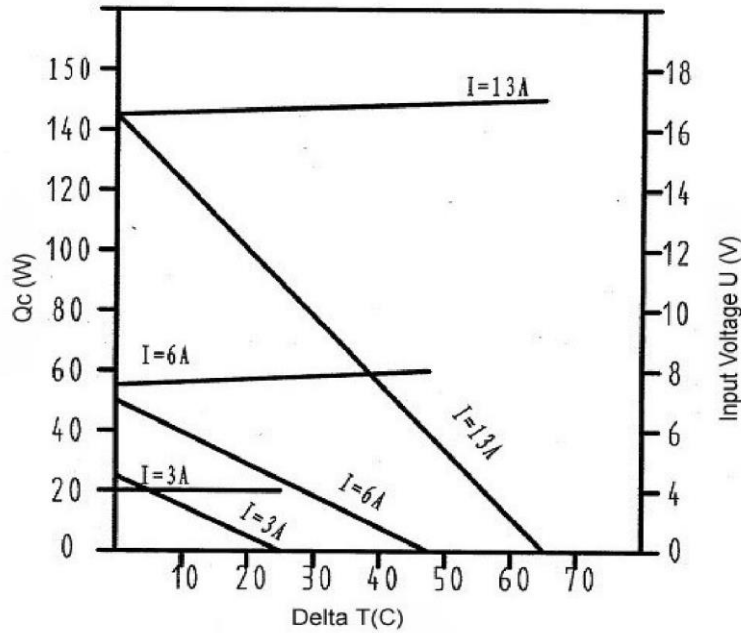
When the ambient temperature raise or fall 1°C , the module resistance will raise or fall 0.015 accordingly.





Hebei I.T. (Shanghai) Co., Ltd

TEC1-12715 Datasheet



40X40 12715 Input Output Characteristic





# LAMPIRAN 3 LISTING PROGRAM





Listing program esp32 :

```
#define FIS_TYPE float
#define FIS_RESOLUTION 101
#define FIS_MIN -3.4028235E+38
#define FIS_MAX 3.4028235E+38
typedef FIS_TYPE(* FIS_MF)(FIS_TYPE, FIS_TYPE*);
typedef FIS_TYPE(* FIS_ARR_OP)(FIS_TYPE, FIS_TYPE);
typedef FIS_TYPE(* FIS_ARR)(FIS_TYPE*, int, FIS_ARR_OP);
#include <RBDdimmer.h>
#include <firFilter.h>
firFilter Filter;
#include <SimpleKalmanFilter.h>
SimpleKalmanFilter simpleKalmanFilter(-1, -1, -0.01);
#define outputPin 26
#define zerocross 5
float ERROR_PANAS=0;
float ERROR_PANAS1=0;
float DELTA_ERROR_PANAS=0;
float ERROR_COOL=0;
float ERROR_COOL1=0;
float DELTA_ERROR_COOL=0;
// Number of inputs to the fuzzy inference system
const int fis_gcI = 4;
// Number of outputs to the fuzzy inference system
const int fis_gcO = 2;
// Number of rules to the fuzzy inference system
const int fis_gcR = 50;
FIS_TYPE g_fisInput[fis_gcI];
FIS_TYPE g_fisOutput[fis_gcO];
float suhu_panas=0;
float suhu_dingin=0;
float sp_cool=0;
float sp_hot=0;
byte fuzzy_pns;
byte fuzzy_dng;
int B = 3950;
float VCC = 3.30;
float TO = 25 + 273.15;
float RT_HOT,RT_COOL, VR_HOT, VR_COOL, VRT_HOT, VRT_COOL;
float ln_hot;
float ln_cool;
byte fuzzyHOT, fuzzyCOOL, TC_COOL_BYTE, TC_HOT_BYTE;
float TC_HOT, TC_HOT_KALMAN, TC_COOL, TC_COOL_KALMAN;
const int COOL_NTC_PIN=12;
const int HOT_NTC_PIN=14;
#include "HardwareSerial.h"
HardwareSerial mega(2);
dimmerLamp dimmer(outputPin, zerocross);
int period = 100;
unsigned long time_now = 0;
byte hot;
byte cool;
const long SERIAL_REFRESH_TIME = 500;
long refresh_time;

void setup() {
  Serial.begin(9600);
  mega.begin(115200, SERIAL_8N1, 16, 17);
  Filter.begin();
  pinMode(HOT_NTC_PIN,INPUT);
  pinMode(COOL_NTC_PIN,INPUT);
  dimmer.begin(NORMAL_MODE, ON);}

void loop() {
  int a=0;
  baca_NTC();
```

```

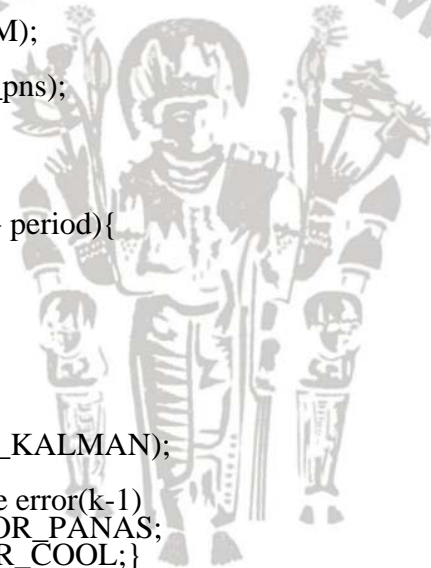
    terima_data_mega();
//program fuzzy
    ERROR_PANAS=sp_hot-TC_HOT_KALMAN; //iki nilai set point tak gawe
    variabel tekan genamu sp_hot karo sp_cool
    DELTA_ERROR_PANAS=ERROR_PANAS-ERROR_PANAS1;
    ERROR_COOL=sp_cool-TC_COOL_KALMAN;
    DELTA_ERROR_COOL=ERROR_COOL-ERROR_COOL1;
//Input fuzzy panas
    g_fisInput[0] = ERROR_PANAS;
    g_fisInput[1] = DELTA_ERROR_PANAS;
    g_fisInput[2] = ERROR_COOL;
    g_fisInput[3] = DELTA_ERROR_COOL;
//Inisialisasi output
    g_fisOutput[0] = 0;
    g_fisOutput[1] = 0;
    fis_evaluate();
    fuzzy_pns=g_fisOutput[0];
    fuzzy_dng=g_fisOutput[1];
    int fuzzyADEM=fuzzy_dng;
    mega.print(TC_HOT_BYTE);
    mega.print(',');
    mega.print(TC_COOL_BYTE);
    mega.print(',');
    mega.println(fuzzyADEM);
    if(fuzzy_pns>=15){
        dimmer.setPower(fuzzy_pns);
    }else{
        dimmer.setPower(0);
    }

    if(millis() >= time_now + period){
        time_now += period;
        Serial.print(fuzzy_pns);
        Serial.print("\t");
        Serial.print(fuzzy_dng);
        Serial.print("\t");
        Serial.print(TC_HOT);
        Serial.print("\t");
        Serial.println(TC_COOL_KALMAN);
    }
//memasukan nilai error ke error(k-1)
    ERROR_PANAS1=ERROR_PANAS;
    ERROR_COOL1=ERROR_COOL;
}

void terima_data_mega(){
    String terima;
    String command;
    char receiveByte;
    while(mega.available()){
        receiveByte = mega.read();
        if(receiveByte != '\n'){
            command = command + receiveByte;
        }
        else{
            terima = command;
            command = "";
        }
    }
    int indexKoma = terima.indexOf(',');
    String setPoint_pns = terima.substring(0, indexKoma);
    String setPoint_dng = terima.substring(indexKoma+1);
    if(setPoint_pns.toInt() > 0 || setPoint_dng.toInt() > 0){
        sp_hot = setPoint_pns.toInt();
        sp_cool = setPoint_dng.toInt();}
}

void baca_NTC(){
    int R;

```



```

R = 10000;
VRT_HOT = 4095 - analogRead(HOT_NTC_PIN);
VRT_COOL = 4095 - analogRead(COOL_NTC_PIN);
// VRT = 1023-analogRead(A6);
VRT_HOT = (3.30 / 4096.00) * VRT_HOT;
VRT_COOL = (3.30 / 4096.00) * VRT_COOL;
VR_HOT = VCC - VRT_HOT;
VR_COOL = VCC - VRT_COOL;
RT_HOT = VRT_HOT / (VR_HOT / R);
RT_COOL = VRT_COOL / (VR_COOL / R);
double lnNum_hot = RT_HOT/(double)10000;
double lnNum_cool = RT_COOL/(double)10000;
ln_hot = log(lnNum_hot);
ln_cool = log(lnNum_cool);
TC_HOT = (1 / ((ln_hot / B) + (1 / TO)));
TC_COOL = (1 / ((ln_cool / B) + (1 / TO)));
TC_HOT = TC_HOT - 273.15;
TC_COOL = TC_COOL - 273.15;
TC_COOL_KALMAN = simpleKalmanFilter.updateEstimate(TC_COOL);
TC_HOT_BYTE= TC_HOT;
TC_COOL_BYTE= TC_COOL_KALMAN;}

//*****
// Inferensi fuzzy
//*****
// fungsi trapesium
FIS_TYPE fis_trapmf(FIS_TYPE x, FIS_TYPE* p)
{
    FIS_TYPE a = p[0], b = p[1], c = p[2], d = p[3];
    FIS_TYPE t1 = ((x <= c) ? 1 : ((d < x) ? 0 : ((c != d) ? ((d - x) / (d - c)) : 0)));
    FIS_TYPE t2 = ((b <= x) ? 1 : ((x < a) ? 0 : ((a != b) ? ((x - a) / (b - a)) : 0)));
    return (FIS_TYPE) _min(t1, t2);
}
// fungsi segitiga
FIS_TYPE fis_trimf(FIS_TYPE x, FIS_TYPE* p)
{
    FIS_TYPE a = p[0], b = p[1], c = p[2];
    FIS_TYPE t1 = (x - a) / (b - a);
    FIS_TYPE t2 = (c - x) / (c - b);
    if ((a == b) && (b == c)) return (FIS_TYPE) (x == a);
    if (a == b) return (FIS_TYPE) (t2*(b <= x)*(x <= c));
    if (b == c) return (FIS_TYPE) (t1*(a <= x)*(x <= b));
    t1 = _min(t1, t2);
    return (FIS_TYPE) _max(t1, 0);
}
FIS_TYPE fis_min(FIS_TYPE a, FIS_TYPE b)
{
    return _min(a, b);
}
FIS_TYPE fis_max(FIS_TYPE a, FIS_TYPE b)
{
    return _max(a, b);
}
FIS_TYPE fis_array_operation(FIS_TYPE *array, int size, _FIS_ARR_OP pfnOp)
{
    int i;
    FIS_TYPE ret = 0;
    if (size == 0) return ret;
    if (size == 1) return array[0];
    ret = array[0];
    for (i = 1; i < size; i++)
    {
        ret = (*pfnOp)(ret, array[i]);
    }
    return ret;
}

```

```

//*****
// Data for Fuzzy Inference System
//*****
// Pointers to the implementations of member functions
_FIS_MF fis_gMF[] =
{
    fis_trapmf, fis_trimf
};
// Count of member function for each Input
int fis_gMFCount[] = { 5, 5, 5, 5 };

// Count of member function for each Output
int fis_gOMFCount[] = { 5, 5 };
// Coefficients for the Input Member Functions
_FIS_TYPE fis_gMFI0Coeff1[] = { -100, -100, -5, 0 };
_FIS_TYPE fis_gMFI0Coeff2[] = { -10, 0, 0.5 };
_FIS_TYPE fis_gMFI0Coeff3[] = { 0.3, 1, 3 };
_FIS_TYPE fis_gMFI0Coeff4[] = { 2.8, 15, 30 };
_FIS_TYPE fis_gMFI0Coeff5[] = { 20, 30, 80, 80 };
_FIS_TYPE* fis_gMFI0Coeff[] = { fis_gMFI0Coeff1, fis_gMFI0Coeff2, fis_gMFI0Coeff3,
    fis_gMFI0Coeff4, fis_gMFI0Coeff5 };
_FIS_TYPE fis_gMFI1Coeff1[] = { -4, -4, -2, 0 };
_FIS_TYPE fis_gMFI1Coeff2[] = { -2, 0, 2 };
_FIS_TYPE fis_gMFI1Coeff3[] = { 0, 2, 4 };
_FIS_TYPE fis_gMFI1Coeff4[] = { 2, 4, 6 };
_FIS_TYPE fis_gMFI1Coeff5[] = { 4, 6, 16, 16 };
_FIS_TYPE* fis_gMFI1Coeff[] = { fis_gMFI1Coeff1, fis_gMFI1Coeff2, fis_gMFI1Coeff3,
    fis_gMFI1Coeff4, fis_gMFI1Coeff5 };
_FIS_TYPE fis_gMFI2Coeff1[] = { -30, -30, -4, -2 };
_FIS_TYPE fis_gMFI2Coeff2[] = { -3, -2, -0.7 };
_FIS_TYPE fis_gMFI2Coeff3[] = { -1, -0.7, 0 };
_FIS_TYPE fis_gMFI2Coeff4[] = { -0.1, 0, 5 };
_FIS_TYPE fis_gMFI2Coeff5[] = { 0, 5, 10, 10 };
_FIS_TYPE* fis_gMFI2Coeff[] = { fis_gMFI2Coeff1, fis_gMFI2Coeff2, fis_gMFI2Coeff3,
    fis_gMFI2Coeff4, fis_gMFI2Coeff5 };
_FIS_TYPE fis_gMFI3Coeff1[] = { -10, -10, -6, -2 };
_FIS_TYPE fis_gMFI3Coeff2[] = { -3, -2, -1 };
_FIS_TYPE fis_gMFI3Coeff3[] = { -2, -1, 0 };
_FIS_TYPE fis_gMFI3Coeff4[] = { -1, 0, 2 };
_FIS_TYPE fis_gMFI3Coeff5[] = { 0, 2, 5, 5 };
_FIS_TYPE* fis_gMFI3Coeff[] = { fis_gMFI3Coeff1, fis_gMFI3Coeff2, fis_gMFI3Coeff3,
    fis_gMFI3Coeff4, fis_gMFI3Coeff5 };
_FIS_TYPE** fis_gMFICoeff[] = { fis_gMFI0Coeff, fis_gMFI1Coeff, fis_gMFI2Coeff,
    fis_gMFI3Coeff };
// Coefficients for the Output Member Functions
_FIS_TYPE fis_gMFO0Coeff1[] = { 0, 0, 5 };
_FIS_TYPE fis_gMFO0Coeff2[] = { 2, 6, 8 };
_FIS_TYPE fis_gMFO0Coeff3[] = { 6, 20, 35 };
_FIS_TYPE fis_gMFO0Coeff4[] = { 30, 50, 78 };
_FIS_TYPE fis_gMFO0Coeff5[] = { 70, 78, 78 };
_FIS_TYPE* fis_gMFO0Coeff[] = { fis_gMFO0Coeff1, fis_gMFO0Coeff2,
    fis_gMFO0Coeff3, fis_gMFO0Coeff4, fis_gMFO0Coeff5 };
_FIS_TYPE fis_gMFO1Coeff1[] = { 50, 50, 90 };
_FIS_TYPE fis_gMFO1Coeff2[] = { 88, 90, 94 };
_FIS_TYPE fis_gMFO1Coeff3[] = { 92, 95, 97 };
_FIS_TYPE fis_gMFO1Coeff4[] = { 96, 98, 100 };
_FIS_TYPE fis_gMFO1Coeff5[] = { 99, 100, 100 };
_FIS_TYPE* fis_gMFO1Coeff[] = { fis_gMFO1Coeff1, fis_gMFO1Coeff2,
    fis_gMFO1Coeff3, fis_gMFO1Coeff4, fis_gMFO1Coeff5 };
_FIS_TYPE** fis_gMFOCoeff[] = { fis_gMFO0Coeff, fis_gMFO1Coeff };
// Input membership function set
int fis_gMFI0[] = { 0, 1, 1, 1, 0 };
int fis_gMFI1[] = { 0, 1, 1, 1, 0 };
int fis_gMFI2[] = { 0, 1, 1, 1, 0 };
int fis_gMFI3[] = { 0, 1, 1, 1, 0 };
int* fis_gMFI[] = { fis_gMFI0, fis_gMFI1, fis_gMFI2, fis_gMFI3 };
// Output membership function set

```





// Rule Outputs

```

int fis_gRO0[] = { 1, 0 };
int fis_gRO1[] = { 1, 0 };
int fis_gRO2[] = { 1, 0 };
int fis_gRO3[] = { 1, 0 };
int fis_gRO4[] = { 1, 0 };
int fis_gRO5[] = { 1, 0 };
int fis_gRO6[] = { 1, 0 };
int fis_gRO7[] = { 2, 0 };
int fis_gRO8[] = { 2, 0 };
int fis_gRO9[] = { 2, 0 };
int fis_gRO10[] = { 2, 0 };
int fis_gRO11[] = { 2, 0 };
int fis_gRO12[] = { 2, 0 };
int fis_gRO13[] = { 3, 0 };
int fis_gRO14[] = { 3, 0 };
int fis_gRO15[] = { 2, 0 };
int fis_gRO16[] = { 4, 0 };
int fis_gRO17[] = { 4, 0 };
int fis_gRO18[] = { 4, 0 };
int fis_gRO19[] = { 4, 0 };
int fis_gRO20[] = { 3, 0 };
int fis_gRO21[] = { 5, 0 };
int fis_gRO22[] = { 4, 0 };
int fis_gRO23[] = { 4, 0 };
int fis_gRO24[] = { 4, 0 };
int fis_gRO25[] = { 0, 5 };
int fis_gRO26[] = { 0, 5 };
int fis_gRO27[] = { 0, 5 };
int fis_gRO28[] = { 0, 5 };
int fis_gRO29[] = { 0, 4 };
int fis_gRO30[] = { 0, 5 };
int fis_gRO31[] = { 0, 5 };
int fis_gRO32[] = { 0, 5 };
int fis_gRO33[] = { 0, 4 };
int fis_gRO34[] = { 0, 4 };
int fis_gRO35[] = { 0, 5 };
int fis_gRO36[] = { 0, 5 };
int fis_gRO37[] = { 0, 4 };
int fis_gRO38[] = { 0, 4 };
int fis_gRO39[] = { 0, 3 };
int fis_gRO40[] = { 0, 2 };
int fis_gRO41[] = { 0, 2 };
int fis_gRO42[] = { 0, 1 };
int fis_gRO43[] = { 0, 1 };
int fis_gRO44[] = { 0, 1 };
int fis_gRO45[] = { 0, 1 };
int fis_gRO46[] = { 0, 1 };
int fis_gRO47[] = { 0, 1 };
int fis_gRO48[] = { 0, 1 };
int fis_gRO49[] = { 0, 1 };
int* fis_gRO[] = { fis_gRO0, fis_gRO1, fis_gRO2, fis_gRO3, fis_gRO4, fis_gRO5,
fis_gRO6, fis_gRO7, fis_gRO8, fis_gRO9, fis_gRO10, fis_gRO11, fis_gRO12,
fis_gRO13, fis_gRO14, fis_gRO15, fis_gRO16, fis_gRO17, fis_gRO18, fis_gRO19,
fis_gRO20, fis_gRO21, fis_gRO22, fis_gRO23, fis_gRO24, fis_gRO25, fis_gRO26,
fis_gRO27, fis_gRO28, fis_gRO29, fis_gRO30, fis_gRO31, fis_gRO32, fis_gRO33,
fis_gRO34, fis_gRO35, fis_gRO36, fis_gRO37, fis_gRO38, fis_gRO39, fis_gRO40,
fis_gRO41, fis_gRO42, fis_gRO43, fis_gRO44, fis_gRO45, fis_gRO46, fis_gRO47,
fis_gRO48, fis_gRO49 };
// Input range Min
FIS_TYPE fis_gIMin[] = { -100, -4, -30, -10 };
// Input range Max
FIS_TYPE fis_gIMax[] = { 80, 16, 10, 5 };
// Output range Min
FIS_TYPE fis_gOMin[] = { 0, 0 };
// Output range Max
FIS_TYPE fis_gOMax[] = { 80, 100 };

```



```

//*****
// Data dependent support functions for Fuzzy Inference System
//*****
FIS_TYPE fis_MF_out(FIS_TYPE** fuzzyRuleSet, FIS_TYPE x, int o)
{
    FIS_TYPE mfOut;
    int r;
    for (r = 0; r < fis_gcR; ++r)
    {
        int index = fis_gRO[r][o];
        if (index > 0)
        {
            index = index - 1;
            mfOut = (fis_gMF[fis_gMFO[o][index]])(x, fis_gMFOCoeff[o][index]);
        }
        else if (index < 0)
        {
            index = -index - 1;
            mfOut = 1 - (fis_gMF[fis_gMFO[o][index]])(x, fis_gMFOCoeff[o][index]);
        }
        else
        {
            mfOut = 0;
        }
        fuzzyRuleSet[0][r] = fis_min(mfOut, fuzzyRuleSet[1][r]);
    }
    return fis_array_operation(fuzzyRuleSet[0], fis_gcR, fis_max);
}

FIS_TYPE fis_defuzz_mom(FIS_TYPE** fuzzyRuleSet, int o)
{
    FIS_TYPE step = (fis_gOMax[o] - fis_gOMin[o]) / (FIS_RESOLUTION - 1);
    FIS_TYPE max = 0, dist, value, sum;
    int count;
    int i;
    for (i = 0; i < FIS_RESOLUTION; ++i)
    {
        dist = fis_gOMin[o] + (step * i);
        value = fis_MF_out(fuzzyRuleSet, dist, o);
        max = _max(max, value);
    }
    sum = 0;
    count = 0;
    for (i = 0; i < FIS_RESOLUTION; i++)
    {
        dist = fis_gOMin[o] + (step * i);
        value = fis_MF_out(fuzzyRuleSet, dist, o);
        if (max == value)
        {
            ++count;
            sum += i;
        }
    }
    return (fis_gOMin[o] + ((step * sum) / count));
}

//*****
// Fuzzy Inference System
//*****
void fis_evaluate()
{
    FIS_TYPE fuzzyInput0[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE fuzzyInput1[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE fuzzyInput2[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE fuzzyInput3[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE* fuzzyInput[fis_gcI] = { fuzzyInput0, fuzzyInput1, fuzzyInput2,
    fuzzyInput3, };
    FIS_TYPE fuzzyOutput0[] = { 0, 0, 0, 0, 0 };
}

```

```

FIS_TYPE fuzzyOutput1[] = { 0, 0, 0, 0, 0 };
FIS_TYPE* fuzzyOutput[fis_gcO] = { fuzzyOutput0, fuzzyOutput1, };
FIS_TYPE fuzzyRules[fis_gcR] = { 0 };
FIS_TYPE fuzzyFires[fis_gcR] = { 0 };
FIS_TYPE* fuzzyRuleSet[] = { fuzzyRules, fuzzyFires };
FIS_TYPE sW = 0;
// Transforming input to fuzzy Input
int i, j, r, o;
for (i = 0; i < fis_gcI; ++i)
{
    for (j = 0; j < fis_gIMFCount[i]; ++j)
    {
        fuzzyInput[i][j] =
            (fis_gMF[fis_gMFI[i][j]])(g_fisInput[i], fis_gMFCoeff[i][j]);
    }
}
int index = 0;
for (r = 0; r < fis_gcR; ++r)
{
    if (fis_gRType[r] == 1)
    {
        fuzzyFires[r] = FIS_MAX;
        for (i = 0; i < fis_gcI; ++i)
        {
            index = fis_gRI[r][i];
            if (index > 0)
                fuzzyFires[r] = fis_min(fuzzyFires[r], fuzzyInput[i][index - 1]);
            else if (index < 0)
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1 - fuzzyInput[i][-index - 1]);
            else
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1);
        }
    }
    else
    {
        fuzzyFires[r] = FIS_MIN;
        for (i = 0; i < fis_gcI; ++i)
        {
            index = fis_gRI[r][i];
            if (index > 0)
                fuzzyFires[r] = fis_max(fuzzyFires[r], fuzzyInput[i][index - 1]);
            else if (index < 0)
                fuzzyFires[r] = fis_max(fuzzyFires[r], 1 - fuzzyInput[i][-index - 1]);
            else
                fuzzyFires[r] = fis_max(fuzzyFires[r], 0);
        }
    }
    fuzzyFires[r] = fis_gRWeight[r] * fuzzyFires[r];
    sW += fuzzyFires[r];
}
if (sW == 0)
{
    for (o = 0; o < fis_gcO; ++o)
    {
        g_fisOutput[o] = ((fis_gOMax[o] + fis_gOMin[o]) / 2);
    }
}
else
{
    for (o = 0; o < fis_gcO; ++o)
    {
        g_fisOutput[o] = fis_defuzz_mom(fuzzyRuleSet, o);
    }
}
}

```



Listing program *arduino* Mega :

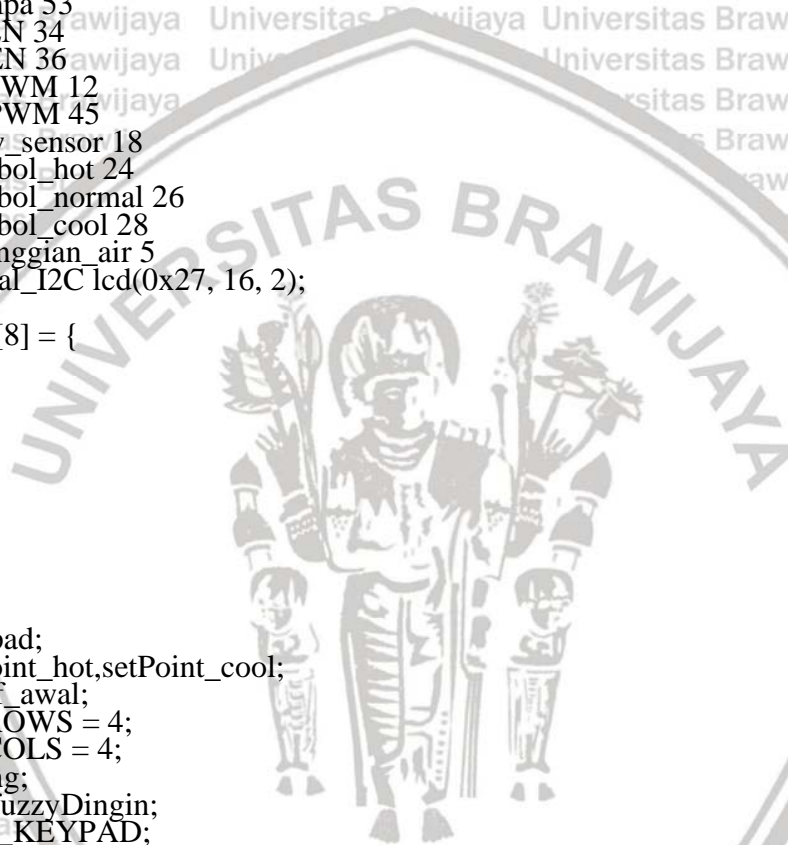
```
#include <Keypad.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#define esp32 Serial3
#define saklar_hot 30
#define saklar_cool 32
#define relay_hot 46
#define relay_cool 48
#define relay_valve_hot 50
#define relay_valve_cool 52
#define pompa 53
#define L_EN 34
#define R_EN 36
#define L_PWM 12
#define R_PWM 45
#define flow_sensor 18
#define tombol_hot 24
#define tombol_normal 26
#define tombol_cool 28
#define ketinggian_air 5
LiquidCrystal_I2C lcd(0x27, 16, 2);

byte derajat[8] = {
  0b01100,
  0b10010,
  0b10010,
  0b01100,
  0b00000,
  0b00000,
  0b00000,
  0b00000
};
char key;
byte SPkeypad;
byte T,setPoint_hot,setPoint_cool;
byte setSP, f_awal;
const byte ROWS = 4;
const byte COLS = 4;
int fuzzy_dng;
int Output_fuzzyDingin;
/*int NILAI_KEYPAD;
int SET_POINT_HOT=0;
int SET_POINT_COOL=0;
char customKey;*/

//define the symbols on the buttons of the keypad
char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {3, 4, 27, 25}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {23, 29, 31, 22}; //connect to the column pinouts of the keypad

Keypad keypad = Keypad(makeKeymap(hexaKeys),rowPins, colPins, ROWS, COLS);
byte ntc_hot=0;
byte ntc_cool=0;
int period = 2500;
unsigned long time_now = 0;

void setup(){
  Serial.begin(9600);
  esp32.begin(115200);
  lcd.begin();
```



80

```

pinMode(saklar_hot, INPUT);
pinMode(saklar_cool, INPUT);
pinMode(relay_hot, OUTPUT);
pinMode(relay_cool, OUTPUT);
pinMode(relay_valve_hot, OUTPUT);
pinMode(relay_valve_cool, OUTPUT);
pinMode(pompa, OUTPUT);
pinMode(L_EN, OUTPUT);
pinMode(R_EN, OUTPUT);
pinMode(L_PWM, OUTPUT);
pinMode(R_PWM, OUTPUT);
pinMode(flow_sensor, INPUT);
pinMode(tombol_hot, INPUT);
pinMode(tombol_normal, INPUT);
pinMode(tombol_cool, INPUT);
pinMode(ketinggian_air, INPUT);

```

```

digitalWrite(relay_hot, HIGH);
digitalWrite(relay_cool, HIGH);
digitalWrite(relay_valve_hot, HIGH);
digitalWrite(relay_valve_cool, HIGH);
digitalWrite(pompa, LOW);
}

```

```

void loop(){
key = keypad.getKey();
utama();
isi_air();
tombol_air_hot();
tombol_air_cool();
terima_esp();
pendingin_fuzzy();
pemanas_fuzzy();

```

```

if(key != NO_KEY){
if(key == 'A'){
iniPanas();
}
else if(key == 'B'){
iniDingin();
}
else if(key == '#'){
cpt_Panas();
}
else if(key == '*'){
cpt_Dingin();}}
}

```

```

void isi_air(){
int tinggi_air = digitalRead(ketinggian_air);
//Serial.println(tinggi_air);
if(tinggi_air == 1){
digitalWrite(pompa, HIGH);
}
else if(tinggi_air == 0){
digitalWrite(pompa, LOW);}}

```

```

void kirim_esp(){
esp32.print(setPoint_hot);
esp32.print(',');
esp32.println(setPoint_cool);}

```

```

void terima_esp(){
String terima;
String command;
char receiveByte;
while(esp32.available()){

```



```

receiveByte = esp32.read();
if(receiveByte != '\n'){
  command = command + receiveByte;
}
else{
  terima = command;
  command = "";
}

int indexKoma = terima.indexOf(',');
String tanki_hot = terima.substring(0, indexKoma);
int indexKoma1 = terima.indexOf('.', indexKoma+1);
String tanki_cool = terima.substring(indexKoma+1, indexKoma1+1);
int indexKoma2 = terima.indexOf(';', indexKoma1+1);
String fuzzyDingin = terima.substring(indexKoma1+1);
//Serial.println(terima);
if(fuzzyDingin.toInt() > 0){
  fuzzy_dng = fuzzyDingin.toInt();
  //Serial.println(fuzzy_dng);
}

if(millis() >= time_now + period){
  time_now += period;

  if(tanki_hot.toInt() > 0 || tanki_cool.toInt() > 0){
    ntc_hot = tanki_hot.toInt();
    ntc_cool = tanki_cool.toInt();
    // Serial.print(ntc_hot);
    // Serial.print("\t");
    // Serial.println(ntc_cool);
    // Serial.print("\t");
    // Serial.println(fuzzy_dng);
  }
}

void tombol_air_normal(){
  int tombol2 = digitalRead(tombol_normal);

  if (tombol2 == 1){
    //digitalWrite(relay_valve_cool, LOW);
  }
  else if(tombol2 == 0){
    //digitalWrite(relay_valve_cool, HIGH);
  }
}

void tombol_air_cool(){
  int tombol3 = digitalRead(tombol_cool);

  if (tombol3 == 1){
    digitalWrite(relay_valve_cool, LOW);
  }
  else if(tombol3 == 0){
    digitalWrite(relay_valve_cool, HIGH);
  }
}

void tombol_air_hot(){
  int tombol1 = digitalRead(tombol_hot);

  if (tombol1 == 1){
    digitalWrite(relay_valve_hot, LOW);
  }
  else if(tombol1 == 0){
    digitalWrite(relay_valve_hot, HIGH);
  }
}

```

82

```

void pendingin_fuzzy(){
  Output_fuzzyDingin = map(fuzzy_dng, 0, 100, 0, 255);
  //Serial.println(Output_fuzzyDingin);
  int saklar2 = digitalRead(saklar_cool);

```

```

  digitalWrite(L_EN, HIGH);
  digitalWrite(R_EN, HIGH);

```

```

  if(saklar2 == 1){
    digitalWrite(relay_cool, LOW);
    analogWrite(R_PWM, Output_fuzzyDingin);
    analogWrite(L_PWM, 0);
  }

```

```

  else if(saklar2 == 0){
    digitalWrite(relay_cool, HIGH);
    analogWrite(R_PWM, 0);
    analogWrite(L_PWM, 0);
  }
}

```

```

void pemanas_fuzzy(){
  int saklar1 = digitalRead(saklar_hot);
  //Serial.println(saklar1);
  if(saklar1 == 1){
    digitalWrite(relay_hot, LOW);
  }
  else if(saklar1 == 0){
    digitalWrite(relay_hot, HIGH);
  }
}

```

```

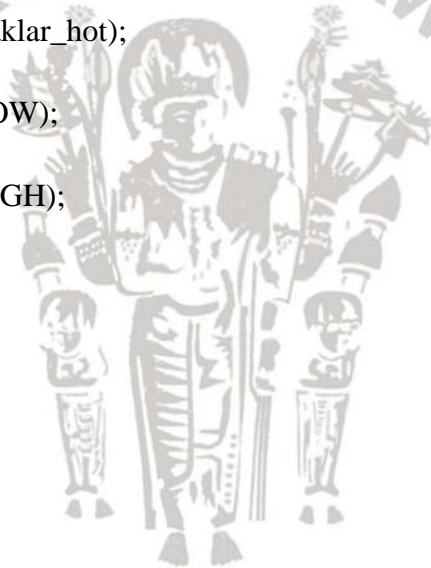
void utama(){
  lcd.setCursor(0,0);
  lcd.print("T.HOT=");
  lcd.setCursor(6,0);
  lcd.print(ntc_hot);
  //delay(500);
  lcd.setCursor(11,0);
  lcd.print("SP=");
  lcd.print(setPoint_hot);
  lcd.setCursor(0,1);
  lcd.print("T.COOL=");
  lcd.setCursor(7,1);
  lcd.print(ntc_cool);
  //delay(500);
  lcd.setCursor(11,1);
  lcd.print("SP=");
  lcd.print(setPoint_cool);
}

```

```

void iniPanas(){
  lcd.clear();
  lcd.print("Setting Setpoint");
  delay(1000);
  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("SP HOT=");
  lcd.print(" ");
  lcd.write(223);
  lcd.print("C");
  lcd.blink();
  lcd.setCursor(10,0);
  byte kursor=10;
  byte selesai=1;

```



```

do{
key=keypad.getKey();
if(key != NO_KEY){
if (key!='*' && key!='#' && key!='A' && key!='B' && key!='C' && key!='D' &&
kursor<12){
lcd.print(key);
key=key-48;
SPkeypad=SPkeypad*10;
SPkeypad=SPkeypad+key;
kursor++;
lcd.setCursor(kursor,0);
}
else if (key=='D' && kursor==12){
selesai=0;
}
}
if(kursor==12){
lcd.noBlink();
lcd.setCursor(4,1);
lcd.print("tekan OK");
}
while(selesai);
setPoint_hot = SPkeypad;
SPkeypad = 0;
f_awal=0;
lcd.clear();
lcd.print("Setpoint OK!");
delay(1000);
lcd.clear();
}

void iniDingin(){
lcd.clear();
lcd.print("Setting Setpoint");
delay(1000);
lcd.clear();
lcd.setCursor(2,0);
lcd.print("SP COOL= ");
lcd.print(" ");
lcd.write(223);
lcd.print("C");
lcd.blink();
lcd.setCursor(10,0);
byte kursor=10;
byte selesai=1;

do{
key=keypad.getKey();
if(key != NO_KEY){
if (key!='*' && key!='#' && key!='A' && key!='B' && key!='C' && key!='D' &&
kursor<12){
lcd.print(key);
key=key-48;
SPkeypad=SPkeypad*10;
SPkeypad=SPkeypad+key;
kursor++;
lcd.setCursor(kursor,0);
}
else if (key=='D' && kursor==12){
selesai=0;
}
}
if(kursor==12){
lcd.noBlink();
lcd.setCursor(4,1);

```



84

```

    lcd.print("tekan OK");
  }
  while(selesai);
  setPoint_cool = SPkeypad;
  SPkeypad = 0;
  f_awal=0;
  lcd.clear();
  lcd.print("Setpoint OK!");
  delay(1000);
  lcd.clear();
}

void cpt_Panas(){
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("1.Kopi");
  lcd.setCursor(11,0);
  lcd.print("3.Teh");
  lcd.setCursor(0,1);
  lcd.print("2.Susu");
  byte selesai=1;
  //byte state=1;
  do{
    key=keypad.getKey();
    if(key != NO_KEY){
      if(key=='1'){
        SPkeypad = 85;
        selesai=0;
      }
      else if(key=='2'){
        SPkeypad = 70;
        selesai=0;
      }
      else if(key=='3'){
        SPkeypad = 60;
        selesai=0;
      }
    }
  }
  while(selesai);
  setPoint_hot = SPkeypad;
  SPkeypad = 0;
  f_awal=0;
  lcd.clear();
  lcd.print("Setpoint OK!");
  delay(1000);
  lcd.clear();
}

void cpt_Dingin(){
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("1.Cool 2.Cooler");
  lcd.setCursor(0,1);
  lcd.print("3.Cooler");
  byte selesai=1;
  //byte state=1;
  do{
    key=keypad.getKey();
    if(key != NO_KEY){
      if(key=='1'){
        SPkeypad = 18;
        selesai=0;
      }
      else if(key=='2'){
        SPkeypad = 14;

```



```
selesai=0;
}
else if(key=='3'){
  SPkeypad = 10;
  selesai=0;
}
}
while(selesai);
setPoint_cool = SPkeypad;
SPkeypad = 0;
f_awal=0;
lcd.clear();
lcd.print("Setpoint OK!");
delay(1000);
lcd.clear();
}
```

