

**PENERAPAN LOGIKA *FUZZY* PADA PERANCANGAN
COOLING KOMPUTER OTOMATIS MENGGUNAKAN
PELTIER BERBASIS MIKROKONTROLER
ATMEGA 328**

SKRIPSI

RIZKY PRATAMA PUTRA

0701172060



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2021**

**PENERAPAN LOGIKA FUZZY PADA PERANCANGAN
COOLING KOMPUTER OTOMATIS MENGGUNAKAN
PELTIER BERBASIS MIKROKONTROLER
ATMEGA 328**

SKRIPSI

Diajukan Untuk Memenuhi Syarat Mencapai Gelar Sarjana Komputer

RIZKY PRATAMA PUTRA

0701172060



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2021**

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi

Lamp : -

Kepada Yth.,

Dekan Fakultas Sains dan Teknologi

Universitas Islam Negeri Sumatera Utara Medan

Assalamu'alaikum Wr. Wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengatakan perbaikan, maka kami selaku pembimbing berpendapat bahwa skripsi saudara,

Nama : Rizky Pratama Putra
Nomor Induk Mahasiswa : 0701172060
Program Studi : Ilmu Komputer
Judul : Penerapan Logika *Fuzzy* Pada Perancangan
Cooling Komputer Otomatis Menggunakan
Peltier Berbasis Mikrokontroler Atmega 328

Dapat disetujui untuk segera dimunaqasyahkan. Atas perhatiannya kami ucapkan terimakasih.

Medan, 25 Oktober 2021 M
18 Rabi'ul-Awwal 1443 H

Komisi Pembimbing,

Pembimbing I,



Rakhmat Kurniawan, R., M.Kom
NIP. 198503162015031003

Pembimbing II,



Abdul Halim Hasugian, M.Kom
NIB. 1100000113

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini,

Nama : Rizky Pratama Putra
Nomor Induk Mahasiswa : 0701172060
Program Studi : Ilmu Komputer
Judul : Penerapan Logika *Fuzzy* Pada Perancangan *Cooling* Komputer Otomatis Menggunakan Peltier Berbasis Mikrokontroler Atmega 328

Menyatakan bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya. Apabila di kemudian hari ditemukan plagiat dalam skripsi ini maka saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi lainnya sesuai dengan peraturan yang berlaku.

Medan, 25 Oktober 2021



Rizky Pratama Putra
NIM. 0701172060



KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA MEDAN
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Lap. Golf, Durin Jangak, Kec. Pancur Batu, Kab. Deli Serdang K.Pos 20353
Url: www.saintek.uinsu.ac.id, E-mail: saintek@uinsu.ac.id

PENGESAHAN SKRIPSI

Nomor : B.003/ST/ST.V.2/PP.01.1/01/2022

Judul : Penerapan Logika Fuzzy Pada Perancangan Cooling Komputer Otomatis Menggunakan Peltier Berbasis Mikrokontroler Atmega 328
Nama : Rizky Pratama Putra
Nomor Induk Mahasiswa : 0701172060
Program Studi : Ilmu Komputer
Fakultas : Sains dan Teknologi

Telah dipertahankan di hadapan Dewan Penguji Skripsi Program Studi Ilmu Komputer Fakultas Sains dan Teknologi UIN Sumatera Utara Medan dan dinyatakan **LULUS**.

Pada hari/tanggal : Senin, 08 November 2021

Tempat : Ruang Sidang Fakultas Sains dan Teknologi

Tim Ujian Munaqasyah
Ketua

Ilka ZulFria, S.Kom., M.Kom
NIP. 198506042015031006

Dewan Penguji,

Penguji I,

Rakhmat Kurniawan, R., M.Kom
NIP. 198503162015031003

Penguji III,

Yusuf Ramadhan Nasution, M.Kom
NIB. 1100000075

Penguji II,

Abdul Halim Hasugian, M.Kom
NIB. 1100000113

Penguji IV,

Muhammad Siddik Hasibuan, M.Kom
NIP. 198611152019031008

Mengesahkan,

Dekan Fakultas Sains dan Teknologi
UIN Sumatera Utara Medan



Dr. Mhd. Wahnan, MA
NIP. 196609051991031002

ABSTRAK

Permasalahan komputer lambat atau lag bisa disebabkan karena terjadi *overheat* pada *hardware* komputer seperti pada prosesornya, apabila terjadi *overheat* komputer akan menurunkan kinerja dari performa prosesor. Pada umumnya komputer sudah dilengkapi dengan pendingin seperti *air cooling* untuk menghindari terjadinya *overheat*, tetapi untuk pekerjaan berat dan intensif *air cooling* masih belum efektif dalam mendinginkan prosesor komputer sehingga pada penelitian ini dibuat sebuah alat *cooling* komputer otomatis yang efektif dalam mendinginkan komputer. Alat ini menggunakan peltier untuk mendinginkan cairan coolant yang mengalir disisi dingin peltier. Untuk mengotrol peltier diterapkan logika *fuzzy* sebagai pengendali suhu dari prosesor komputer. Logika *fuzzy* ini akan disematkan kedalam mikrokontroler. Dingin yang dihasilkan berdasarkan informasi yang diberikan komputer ke mikrokontroler lewat komunikasi serial. Berdasarkan dari pengujian uji *benchmark* rata-rata SuhuCore1 26,75°C, SuhuCore2 25,25°C dan SuhuPackage 29°C dengan menggunakan *water cooling* peltier dengan logika *fuzzy*, sedangkan dengan *air cooling* rata-rata SuhuCore1 33,5°C, SuhuCore2 33°C dan SuhuPackage 35,25°C serta dengan *water cooling* radiator rata-rata SuhuCore1 32,25°C, SuhuCore2 31,5°C dan SuhuPackage 34,25°C pada saat sebelum melakukan uji *benchmark*. Ketika uji *benchmark* berjalan rata-rata SuhuCore1 52,25°C, SuhuCore2 51°C dan SuhuPackage 54,25°C dengan menggunakan *water cooling* peltier dengan logika *fuzzy*, sedangkan dengan *air cooling* rata-rata SuhuCore1 65°C, SuhuCore2 64,25°C dan SuhuPackage 67,25°C serta dengan *water cooling* radiator rata-rata SuhuCore1 57°C, SuhuCore2 59°C dan SuhuPackage 59,25°C dengan rata-rata suhu ruang 30°C saat pengujian berlangsung. Dari hasil pengujian dengan membandingkan alat pada penelitian ini, *water cooling* peltier dengan logika *fuzzy* lebih baik dan efektif dalam mendinginkan prosesor komputer.

Kata Kunci: *Cooling* Komputer, Logika *Fuzzy*, Mikrokontroler

ABSTRACT

Slow computer problems or lag can be caused by overheating of computer hardware such as the processor, if there is overheating the computer will reduce the performance of the processor performance. In general, computers are equipped with coolers such as air cooling to avoid overheating, but for heavy and intensive work, air cooling is still not effective in cooling computer processors, so in this study an automatic computer cooling device was created that was effective in cooling computers. This tool uses a peltier to cool the coolant that flows on the cold side of the peltier. To control the Peltier, fuzzy logic is applied to control the temperature of the computer processor. This fuzzy logic will be embedded into the microcontroller. Cold generated based on the information provided by the computer to the microcontroller via serial communication. Based on the benchmark test, the average Core1 Temperature is 26,75°C, Core2 Temperature is 25,25°C and Package Temperature is 29°C using water cooling Peltier with fuzzy logic, while with water cooling the average Core1 Temperature is 33,5°C, Core2 Temperature is 33°C and Package Temperature is 35,25°C as well as with The average water cooling radiator has a Core1 temperature of 32,25°C, a Core2 temperature of 31,5°C and a Package Temperature of 34,25°C before performing the benchmark test. When the benchmark test runs, the average Core1 Temperature is 52,25°C, Core2 Temperature is 51°C and Package Temperature is 54,25°C using water cooling Peltier with fuzzy logic, while with water cooling the average TemperatureCore1 is 65°C,Core2Temperature is 64,25°C and PackageTemp is 67,25°C as well as with water The average cooling radiator temperature is Core1 57°C, Core2 Temperature 59°C and Package Temperature 59,25°C with an average room temperature of 30°C during the test. From the test results by comparing the tools in this study, water cooling Peltier with fuzzy logic is better and more effective in cooling computer processors.

Keyword: Computer Cooling, Fuzzy Logic, Microcontroller

KATA PENGANTAR

Syukur Alhamdulillah kita ucapkan kehadiran Allah SWT, di mana atas rahmat dan karunianya, penulis dapat menyelesaikan skripsi ini. Shalawat dan salam tercurahkan kepada baginda Nabi Muhammad SAW.

Penulisan skripsi ini bertujuan sebagai persyaratan untuk memperoleh gelar sarjana program studi Ilmu Komputer Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara. Dengan Judul yang diteliti oleh penulis ialah “Penerapan Logika *Fuzzy* Pada Perancangan *Cooling* Komputer Otomatis Menggunakan Peltier Berbasis Mikrokontroler Atmega 328”.

Dalam penulisan skripsi ini, penulis sangat banyak menerima bantuan, dukungan, nasehat-nasehat serta bimbingan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Syahrin Harahap, MA selaku Rektor Universitas Islam Negeri Sumatera Utara Medan.
2. Bapak Dr. Mhd Syahnan, M.A selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara Medan.
3. Bapak Ilka Zufria, M.Kom selaku Ketua Program Studi Ilmu Komputer Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara Medan.
4. Bapak Rakhmat Kurniawan R, S.T., M.Kom selaku Sekretaris Program Studi Ilmu Komputer, Pembimbing Akademik dan Pembimbing Skripsi I yang senantiasa memberikan arahan dan motivasi selama menyelesaikan skripsi.
5. Bapak Abdul Halim Hasugian, M.Kom selaku Pembimbing Skripsi yang selalu memberikan bimbingan selama menyelesaikan skripsi ini.
6. Kedua orang tua penulis yang tiada hentinya mendoakan di setiap sujud sholatnya, mengingatkan di sepanjang waktu dan memberi motivasi bagi penulis.
7. Adik penulis yang selalu memberikan dukungan dan bantuan kepada penulis.

8. Seluruh dosen dan pegawai Program Studi Ilmu Komputer maupun di luar Ilmu Komputer yang pernah memberikan ilmunya kepada penulis.
9. Teman-teman seperjuangan kelas Ilmu Komputer 1 angkatan 2017 dan seluruh teman-teman angkatan 2017 yang sedang berjuang untuk menyelesaikan tugas akhir ini.
10. Kakak dan adik-adik penulis di Ilmu Komputer semua angkatan sebagai tempat berbagi ilmu pengetahuan.
11. Serta semua pihak yang telah memberikan bantuan dengan tulus dan dukungan dalam penyusunan tugas akhir ini yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa skripsi ini belum sempurna dan masih banyak kekurangan. Untuk itu penulis memohon maaf atas kesalahan dan kekurangan yang terdapat didalamnya dan penulis berharap semoga skripsi ini dapat bermanfaat dan menambah wawasan keilmuan. Kritik dan saran yang sifatnya membangun sangat penulis harapkan untuk perbaikan dimasa yang akan datang.

Medan, 17 Agustus 2021
Penulis,

Rizky Pratama Putra

DAFTAR ISI

Halaman

ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	viii
DAFTAR TABEL.....	xi
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	5
1.5 Manfaat Penelitian.....	5
BAB II TINJAUAN PUSTAKA.....	6
2.1 Komputer.....	6
2.1.1 Penyebab <i>Overheat</i> Pada Komputer	7
2.2 Mikrokontroler	8
2.2.1 Jenis – Jenis Mikrokontroler	9
2.2.2 Mikrokontroler ATmega 328.....	10
2.3 Logika <i>Fuzzy</i>	11
2.3.1 Himpunan <i>Fuzzy</i>	12
2.3.2 Fungsi Keanggotaan.....	13

2.3.3 Operator Logika <i>Fuzzy</i>	16
2.4 Sistem Infrensi <i>Fuzzy</i> Sugeno	17
2.5 Peltier	18
2.6 <i>Power Supply</i>	19
2.7 <i>Waterblock</i>	21
2.8 <i>Fan</i>	21
2.9 Heatsink.....	22
2.10 Pompa Air DC.....	23
2.11 Mosfet IRF520	24
2.12 LED RGB	24
2.13 <i>Stepdown</i>	25
2.14 Kabel Data Mini USB	26
2.15 Kabel Pelangi	26
2.16 Microsoft Visual Studio	27
2.17 Arduino IDE	27
2.18 <i>Flowchart</i>	28
2.19 <i>Pulse Width Modulation (PWM)</i>	30
2.20 Komunikasi Serial	31
2.21 Penelitian Terdahulu	32
BAB III METODOLOGI PENELITIAN	34
3.1 Tempat dan Waktu Penelitian	34
3.1.1 Tempat Penelitian.....	34
3.1.2 Waktu dan Jadwal Pelaksanaan Penelitian	34
3.2 Bahan dan Alat Penelitian	34

3.2.1 Perangkat Keras	34
3.2.2 Perangkat Lunak.....	36
3.3 Cara Kerja	37
3.3.1 Perencanaan.....	37
3.3.2 Teknik Pengumpulan Data.....	38
3.3.3 Analisa Kebutuhan.....	38
3.3.4 Perancangan	39
3.3.5 Pengujian.....	46
3.3.6 Penerapan	47
BAB IV HASIL DAN PEMBAHASAN.....	48
4.1 Pembahasan.....	48
4.1.1 Analisis Data	48
4.1.2 Representasi Data.....	49
4.1.3 Hasil Analisis Data.....	59
4.1.4 Perancangan	60
4.2 Hasil	71
4.2.1 Pengujian.....	71
4.2.2 Penerapan	87
BAB V KESIMPULAN DAN SARAN	88
5.1 Kesimpulan.....	88
5.2 Saran.....	89
DAFTAR PUSTAKA	90
LAMPIRAN-LAMPIRAN	

DAFTAR GAMBAR

Gambar	Judul Gambar	Halaman
Gambar 2.1	Pengantar Komputer.....	7
Gambar 2. 2	Representasi Linear Naik	14
Gambar 2. 3	Representasi Linear Turun	14
Gambar 2. 4	Kurva Trapesium.....	15
Gambar 2. 5	Daerah 'Bahu' pada variabel Temperatur	16
Gambar 2. 6	Skema elemen peltier	19
Gambar 2. 7	Power Supply	20
Gambar 2. 8	DC Fan	22
Gambar 2. 9	Heatsink dari bahan alumunium.....	23
Gambar 2. 10	Pompa Air DC.....	23
Gambar 2. 11	Mosfet IRF520	24
Gambar 2. 12	LED RGB.....	25
Gambar 2. 13	Stepdown.....	25
Gambar 2. 14	Kabel data Mini USB	26
Gambar 2. 15	Kabel Pelangi	27
Gambar 2. 16	Tampilan Arduino IDE.....	28
Gambar 2. 17	Pulse Witdh Modulation.....	30
Gambar 3. 1	Diagram blok sistem.....	40
Gambar 3. 2	Flowchart sistem pada mikrokontroler.....	41
Gambar 3. 3	Flowchart sistem pada logika fuzzy	42
Gambar 3. 4	Flowchart sistem pada software interface	43
Gambar 3. 5	Rancangan perangkat keras.....	44

Gambar 3. 6 Layout software interface.....	45
Gambar 4. 1 Fungsi Keanggotaan SuhuCore1	50
Gambar 4. 2 Fungsi Keanggotaan SuhuCore2.....	51
Gambar 4. 3 Fungsi Keanggotaan Output Fuzzy	58
Gambar 4. 4 Flowchart Keseluruhan Sistem Pada Alat.....	60
Gambar 4. 5 Rangkaian alat tampak depan.....	61
Gambar 4. 6 Rangkaian alat tampak atas	62
Gambar 4. 7 Rangkain alat tampak samping	62
Gambar 4. 8 Rangkaian alat tampak belakang.....	62
Gambar 4. 9 Tampilan Software Interface.....	64
Gambar 4. 10 Komunikasi serial berhasil ditandai dengan gauge yang bergerak	72
Gambar 4. 11 Pembacaan suhu oleh mikrokontroler berhasil ditandai LED hidup dan alat bekerja	73
Gambar 4. 12 Hasil metode logika fuzzy dapat dilihat pada software interface...	74
Gambar 4. 13 Uji benchmark menggunakan Cinebench R15 pada air cooling....	75
Gambar 4. 14 Uji benchmark menggunakan Cinebench R15 pada water cooling dengan radiator.....	75
Gambar 4. 15 Uji benchmark menggunakan Cinebench R15 pada water cooling peltier dengan logika fuzzy	76
Gambar 4. 16 Uji benchmark menggunakan CPUZ pada air cooling.....	78
Gambar 4. 17 Uji benchmark menggunakan CPUZ pada water cooling dengan radiator	78
Gambar 4. 18 Uji benchmark menggunakan CPUZ pada water cooling peltier dengan logika fuzzy	79
Gambar 4. 19 Uji benchmark menggunakan 3DMark pada air cooling	81

Gambar 4. 20 Uji benchmark menggunakan 3DMark pada water cooling dengan radiator	81
Gambar 4. 21 Uji benchmark menggunakan 3DMark pada water cooling peltier dengan logika fuzzy	82
Gambar 4. 22 Uji benchmark menggunakan Blender Benchmark Launcher pada air cooling.....	84
Gambar 4. 23 Uji benchmark menggunakan Blender Benchmark Launcher pada water cooling dengan radiator.....	84
Gambar 4. 24 Uji benchmark menggunakan Blender Benchmark Launcher pada water cooling peltier dengan logika fuzzy	85

DAFTAR TABEL

Tabel	Judul Tabel	Halaman
Tabel 2. 1	Simbol dan Fungsi Flowchart	29
Tabel 3. 1	Perangkat Keras	35
Tabel 3. 2	Perangkat Lunak	37
Tabel 4. 1	Himpunan Fuzzy	49
Tabel 4. 2	Perbandingan suhu dengan uji benchmark menggunakan Cinebench .	77
Tabel 4. 3	Perbandingan suhu dengan uji benchmark menggunakan CPUZ.....	80
Tabel 4. 4	Perbandingan suhu dengan uji benchmark menggunakan 3DMark.....	83
Tabel 4. 5	Perbandingan suhu dengan uji benchmark menggunakan Blender Benchmark Launcher	86

DAFTAR LAMPIRAN

Lampiran	Judul Lampiran
1	Listing Program Mikrokontroler Menggunakan Arduino IDE
2	Listing Program <i>Software Interface</i> Menggunakan Microsoft Visual Studio
3	Penggabungan Alat
4	Kartu Bimbingan
5	Daftar Riwayat Hidup

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan di berbagai bidang pada saat ini tidak luput dari adanya perkembangan ilmu pengetahuan dan teknologi yang berkembang pesat pada saat ini (Fadhillah et al., 2019). Perkembangan ilmu pengetahuan dan teknologi ini sudah dijelaskan dalam firman Allah SWT dalam surah *Ar-Rahman* ayat 33.

يَمْعَشِرَ الْجِنِّ وَالْإِنْسِ إِنَّ اسْتَطَعْتُمْ أَنْ تَنْفُذُوا مِنْ أَقْطَارِ السَّمَوَاتِ وَالْأَرْضِ فَانفُذُوا لَا تَنْفُذُونَ إِلَّا بِسُلْطَانٍ ۝٣٣

Artinya: Wahai golongan jin dan manusia! Jika kamu sanggup menembus (melintasi) penjuru langit dan bumi, maka tembuslah. Kamu tidak akan mampu menembusnya kecuali dengan kekuatan (dari Allah). (QS: Ar-Rahman: 33)

Surah Ar-Rahman ayat 33 tersebut mengisaratkan setiap manusia harus menuntut ilmu pengetahuan dan teknologi untuk bekal menjalani kehidupan di dunia. Dari ilmu pengetahuan dan teknologi yang dimiliki tersebut nanti diharapkan mampu melalui perkembangan dan kemajuan zaman. Teknologi adalah media penyediaan barang-barang yang dibutuhkan untuk kebutuhan hidup manusia (Lubis, I. & Safi, 2018), sehingga manusia selalu berusaha untuk mengembangkan dan memanfaatkan ilmu pengetahuan dan teknologi untuk kemudahan dalam berbagai hal. Contoh pada peralatan elektronik dan instrumentasi dari penggunaan rangkaian elektronik, baik rangkaian elektronik yang berbasis sistem analog maupun rangkaian elektronik yang berbasis sistem digital yang dapat digunakan di berbagai hal (Fadhillah et al., 2019).

Pada saat ini banyak perangkat-perangkat teknologi yang populer digunakan masyarakat yang salah satunya yaitu komputer. Komputer merupakan serangkaian mesin elektronik yang terdiri dari jutaan komponen yang dapat saling bekerja sama, serta membentuk sebuah sistem kerja yang rapi, teliti dan sistem ini juga secara otomatis bekerja berdasarkan intruksi yang diberikan (Yahfizham, 2019).

Komputer digunakan untuk mengolah data, desain grafis, simulasi, analisis, dan sebagainya.

Pada saat ini komputer sudah menjadi sebuah kebutuhan khusus, hampir semua dari berbagai kalangan masyarakat telah menggunakan komputer untuk membantu menyelesaikan setiap pekerjaannya. Meskipun begitu komputer juga memiliki berbagai macam permasalahan, seperti komputer lambat atau lag. Menurut Arif Rachman Hakim pada tulisan artikelnya menjelaskan bahwa salah satu penyebab komputer lambat disebabkan karena terjadinya *overheat* pada *hardware* komputer, semakin lama personal komputer digunakan maka dari waktu ke waktu suhunya akan semakin tinggi sedikit demi sedikit. Akibat dari *overheat* ini, performa CPU akan menurun dikarenakan CPU dirancang untuk bekerja dalam keadaan dingin, jadi waktu suhunya *overheat* motherboard akan secara otomatis menurunkan kecepatan agar mampu menurunkan suhunya kembali ke angka normal (Hakim, 2020). Menurut Marwanto pada tulisan artikelnya juga menjelaskan bahwa salah satu penyebab komputer lambat dikarenakan masalah perangkat keras yang *overheat* seperti pada hardisk, VGA card dan prosesor atau CPU secara umum bisa menyebabkan menurunnya kinerja dari komputer yang pada akhirnya komputer bisa menjadi hang, lambat atau bahkan sering restart (Marwanto, 2018).

Pada umumnya komputer sudah dilengkapi dengan pendingin udara untuk mendinginkan CPU atau prosesornya agar terhindar dari *overheat*. Terdapat dua model pendingin CPU yaitu, pendingin *air cooling* atau pendingin udara merupakan jenis sistem pendingin yang masih umum digunakan pada komputer saat ini, karena *air cooling* sangat ideal jika ingin membangun pendingin komputer dengan biaya lebih rendah, mudah dipasang dan hanya membutuhkan sedikit perawatan meskipun begitu *air cooling* tidak ideal untuk performa tinggi seperti menggunakan komputer untuk bermain game canggih atau beban kerja yang intensif, *air cooling* tidak cukup untuk mendinginkan komponen komputer sehingga *overheating* dapat membatasi potensi komputer dan bahkan menyebabkan komponen gagal. Sedangkan pendingin dengan *water cooling* pada dasarnya merupakan radiator untuk CPU di dalam komputer, *water cooling* ini memiliki kinerja yang luar biasa.

Water cooling memungkinkan untuk mendorong komponen komputer ke batas normal sambil mempertahankan suhu yang aman sehingga dengan kata lain *water cooling* ini merupakan pengaturan panas terbaik. Cara kerja dari *water cooling* ini menyalurkan cairan melalui *water blocks* yang berada di atas CPU yang sedang didinginkan, kemudian cairan yang relatif lebih dingin bersikulasi melalui *water block* mengumpulkan panas dari CPU lalu ketika cairan menjadi panas akan di pompa ke radiator di bagian belakang casing, dimana kipas mentransfer panas menjadi udara dingin. Cairan yang telah didinginkan dikembalikan ke *water blocks* untuk dipanaskan, proses ini terus berulang (Risyan, 2021).

Menurut Resa Risyan pada kesimpulan artikelnya yang berjudul perbedaan *water cooling* dan *air cooling* pada PC, menjelaskan bahwa *water cooling* lebih efisien daripada *air cooling* untuk menjaga suhu dari CPU agar tetap dingin (Risyan, 2021). Hal ini juga disebutkan oleh salah satu perusahaan prosesor terbesar dunia yaitu Intel lewat sebuah artikelnya menjelaskan bahwa, jika mendinginkan temperatur serendah mungkin maka pendingin cair atau *water cooling* kemungkinan merupakan pilihan terbaik (Intel, 2020).

Pada penelitian ini penulis membuat sebuah alat *cooling* komputer otomatis dengan menggunakan peltier dengan judul penelitian yaitu, “Penerapan Logika *Fuzzy* Pada Perancangan *Cooling* Komputer Otomatis Menggunakan Peltier Berbasis Mikrokontroler ATMEGA 328”. Alat ini nantinya menggunakan prinsip kerja dari *water cooling* namun perbedaannya dengan alat yang akan dibuat pada penelitian ini yaitu pada proses pendinginan cairannya, yang dimana pendinginan cairan ini tidak menggunakan radiator tetapi menggunakan peltier yang dikontrol oleh mikrokontroler. Pada *cooling* komputer otomatis ini mikrokontroler berperan penting sebagai pusat dari pengolahan data.

Pada *cooling* komputer otomatis ini juga menerapkan Logika *Fuzzy* sebagai pengontrol suhu dari komputer. Logika *Fuzzy* merupakan suatu cara yang tepat untuk memetakan suatu ruang input ke dalam suatu ruang output. Logika *Fuzzy* memungkinkan nilai keanggotaan antara 0 dan 1, dalam bentuk linguistik konsep tidak pasti seperti dingin, hangat dan panas. Logika *Fuzzy* berhubungan dengan dengan set *Fuzzy* dan teori kemungkinan (WS, 2015).

Implementasi dari *cooling* komputer otomatis ini menggunakan cairan coolant dengan Peltier dalam mendinginkan komputer. Dalam *cooling* komputer otomatis ini, Peltier berperan sebagai media dalam mendinginkan cairan yang mengalir ke *waterblock* yang menempel disisi dingin dari peltier. Dingin yang dihasilkan dari *cooling* komputer otomatis ini berdasarkan informasi suhu dari komputer yang dikirimkan ke Mikrokontroler melalui komunikasi serial. Dengan adanya alat ini diharapkan dalam melakukan pendinginan komputer akan jauh lebih efektif dan akan lebih baik dalam menurunkan dan menjaga kestabilan suhu dari komputer dibandingkan dengan *water cooling* pada umumnya, sehingga terhindar dari panas yang berlebih atau *overheat* serta menjaga performa komputer agar tetap stabil serta menjaga keawetan dari komputer tersebut.

1.2 Rumusan Masalah

Berdasarkan uraian dari latar belakang di atas, rumusan masalah yaitu sebagai berikut:

1. Bagaimana merancang *cooling* komputer otomatis yang dapat mendinginkan komputer secara efektif agar dapat menjaga performa kinerja dari prosesor komputer?
2. Bagaimana menerapkan logika *fuzzy* pada *cooling* komputer otomatis?

1.3 Batasan Masalah

Pada penelitian ini terdapat beberapa batasan masalah, sebagai berikut:

1. Alat pada penelitian ini, hanya dapat diterapkan pada *Personal Computer* (PC).
2. Pada penelitian ini, pendinginan dilakukan hanya untuk prosesor.
3. Pada penelitian ini, data suhu didapat langsung dari komputer yang dikirimkan ke mikrokontroler melalui Komunikasi serial.
4. Pada penelitian ini, digunakan metode *fuzzy* Sugeno.
5. Pada penelitian ini, dalam mendinginkan prosesor menggunakan cairan yang kemudian cairannya didinginkan menggunakan peltier.
6. Pada penelitian ini, peltier dikontrol menggunakan Mosfet IRF520 yang diatur oleh mikrokontroler.

1.4 Tujuan Penelitian

Berdasarkan rumusan penelitian tersebut, maka tujuan dari penelitian ini sebagai berikut:

1. Merancang alat *cooling* otomatis yang efektif dalam mendinginkan komputer agar dapat menjaga performa kinerja prosesor komputer sehingga terhindar dari *overheat*.
2. Menerapkan logika *fuzzy* pada *cooling* komputer otomatis.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini yaitu untuk melakukan pendinginan komputer akan jauh lebih efektif, sehingga terhindar dari panas yang berlebih atau *overheat* dan menjaga performa komputer agar tetap stabil serta menjaga keawetan dari komputer tersebut.

BAB II

TINJAUAN PUSTAKA

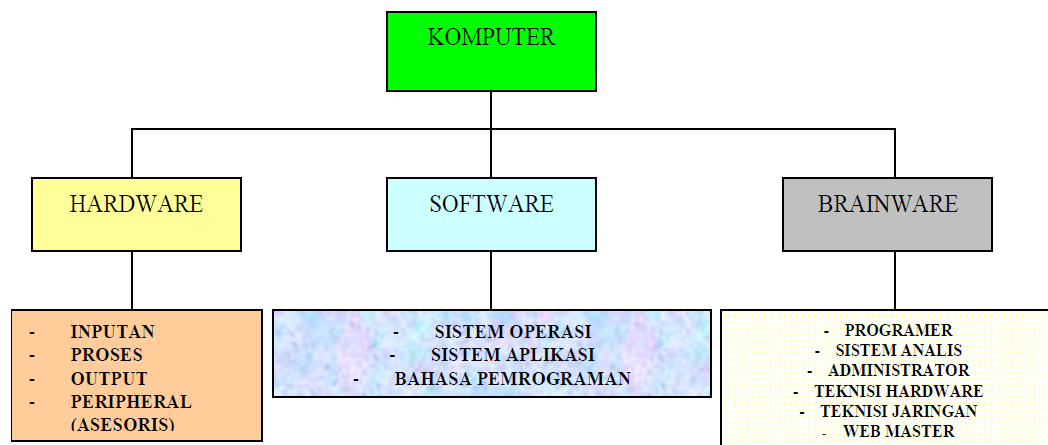
2.1 Komputer

Sejak pertengahan 1950-an, komputer telah digunakan secara komersial. Sejauh ini, perangkat keras dan perangkat lunak telah membuat kemajuan besar. Pesatnya perkembangan teknologi informasi dan komunikasi (TIK) mengakibatkan penggunaan yang sangat sedikit pulpen dan kertas. TIK membuat terminologi virtual menjadi akrab dengan pekerjaan kantor modern. Penulisan atau pencatatan dan laporan sekarang berbasis TIK. TIK telah mengambil alih fungsi perlengkapan alat tulis kantor lama.

Komputer pada awalnya adalah mesin berukuran ruangan yang dapat melakukan tugas-tugas sederhana, seperti aritmatika, dengan sangat cepat. Rangkaian tersebut terdiri dari tabung vakum, ukuran tabung vakum setara dengan bola lampu kecil. Transistor dan chip wafer silikon menggantikan komponen elektronik tabung vakum menyebabkan biaya produksi yang lebih rendah, yang menyebabkan peningkatan permintaan komputer. Ketika ukuran fisik komponen ini berkurang, kecepatan pemrosesan komputer terus meningkat. Data kecil menggandakan kecepatan pemrosesan dan mengurangi biaya produksi (miniaturisasi).

Menurut Yahfizham, komputer adalah rangkaian mesin elektronik yang terdiri dari jutaan komponen dapat bekerja sama membentuk sistem kerja yang teratur dan kompleks. Sistem digunakan untuk secara otomatis melakukan pekerjaan sesuai dengan instruksi (program) yang diberikan. Filosofinya adalah bahwa komputer adalah benda mati yang dapat digunakan orang untuk membantu orang dalam meringankan pekerjaan. Karena manusia adalah makhluk hidup dan perangkat keras dan perangkat lunak merupakan benda mati, sehingga tujuan pembuatan *hardware* dan *software* adalah untuk manusia. Dalam hal ini, istilah "manusia" mengacu pada *brainware* komputer. Kata perangkat keras komputer mengacu pada sesuatu yang berwujud yang dapat dipegang, dipindahkan, dan di-

amati. Perangkat lunak komputer adalah rangkaian instruksi (program/proses) yang memungkinkan untuk mengotomatiskan tugas dengan mengolah atau memproses serangkaian instruksi (data) (Yahfizham, 2019).



Gambar 2.1 Pengantar Komputer

(Yahfizham, 2019)

2.1.1 Penyebab *Overheat* Pada Komputer

Dalam mode siaga, suhu CPU harus antara 39 dan 49 derajat Celcius (tidak menjalankan program berat), saat menjalankan program berat (seperti game dan perangkat lunak lain), suhu prosesor normal adalah antara 60-an derajat Celcius. Saat komputer dalam mode standby menampilkan 60 derajat Celcius ini berarti terdapat masalah dengan CPU. Hal ini dapat menyebabkan panas berlebih atau *overheat*. Ini biasanya disebabkan oleh perangkat keras yang tidak berfungsi.. Ada banyak hal yang dapat menyebabkan CPU komputer menjadi *overheat*. Bahkan terkadang karena berbagai faktor yang sangat berbahaya, prosesor akan menjadi terlalu panas. Merusak seluruh sistem komputer dan perangkat lunak. Sejumlah faktor dapat menyebabkan CPU menjadi terlalu panas:

1. Komputer digunakan selama lebih dari 24 jam tanpa berhenti
2. Terdapat virus komputer dan *malware*
3. Komputer di *overlocking*
4. Penggunaan prosesor hingga 100%

5. Mengoprasikan aplikasi-aplikasi yang berat
6. Mengeringnya thermal pasta
7. Terjadi penyumbatan pada fan CPU
8. Tidak kompatibel spesifikasi perangkat keras (Sutiono, 2016)
9. Temperatur ruangan yang tinggi (Wijaya, 2020)

2.2 Mikrokontroler

Mikrokontroler merupakan sebuah sistem komputer fungsional dalam sebuah chip dan mikrokontroler ini berisi sebuah memori (sejumlah kecil RAM, memori program atau keduanya), prosesor dan perangkat keras input dan output. Mikrokontroler merupakan salah satu bagian utama dari sistem komputer. Meskipun bentuk mikrokontroler jauh lebih kecil daripada komputer pribadi dan mainframe dan terdiri dari elemen dasar yang sama. Singkatnya, komputer menghasilkan hasil tertentu berdasarkan masukan yang diterimanya dan program yang sedang dijalankan.

Seperti kebanyakan komputer, mikrokontroler adalah alat yang beroperasi sesuai dengan instruksi. Dengan kata lain, bagian terpenting dan dasar dari sistem komputer adalah program itu sendiri yang dibuat oleh programmer. Program ini menginstruksikan komputer untuk melakukan rangkaian panjang tindakan sederhana untuk melakukan tugas yang lebih kompleks yang ingin dilakukan oleh programmer.

Dengan kata lain, mikrokontroler adalah perangkat elektronik digital yang memiliki input dan output, dan dikendalikan oleh program yang dapat ditulis dan dihapus dengan cara khusus, seperti halnya kerja dari mikrokontroler yaitu membaca dan menulis data. Mikrokontroler adalah komputer pada chip yang digunakan untuk mengontrol perangkat elektronik, dengan menitik beratkan pada efisiensi dan efektivitas biaya. Secara harfiah, ini bisa menjadi "pengontrol kecil" di mana sistem elektronik yang sebelumnya membutuhkan banyak komponen lain (seperti TTL, CMOS dan IC) dapat beroperasi di dalamnya dikurangi/diperkecil ukurannya dan berakhir di kontrol terpusat oleh mikrokontroler.

2.2.1 Jenis – Jenis Mikrokontroler

Terdapat dua jenis mikrokontroler berdasarkan teknis. Pembagian ini didasarkan pada kompleksitas instruksi yang dapat diterapkan pada mikrokontroler, yaitu RISC dan CISC, masing-masing memiliki keturunan atau keluarganya sendiri-sendiri.

1. *Reduced Instruction Set Computer* (RISC) memiliki fasilitas yang lebih baik tetapi instruksi yang dimiliki terbatas.
2. *Complex Instruction Set Computer* (CISC) merupakan sekumpulan instruksi yang kompleks. Bisa dikatakan instruksinya relatif lengkap, namun memiliki fasilitas yang secukupnya.

Akan tetapi jenis mikrokontroler yang umum digunakan, yaitu:

1. Seri MCS51

Mikrokontroler ini merupakan bagian dari mikrokontroler seri CISC. Kebanyakan instruksi dijalankan dalam 12 siklus jam. Arsitektur harvard didasarkan pada mikrokontroler. Meskipun awalnya dirancang untuk aplikasi mikrokontroler chip tunggal, mode perusahaan memberi izin sebuah ROM adalah 64KB dan RAM 64KB. Alamat ditentukan oleh jalur terpisah yang digunakan untuk memilih sirkuit mikro untuk mengakses program dan penyimpanan data. Salah satu fungsi mikrokontroler 8051 adalah mesin pemroses logika terintegrasi yang memungkinkan pelaksanaan operasi logika boolean secara langsung dan efisien dalam register internal dan RAM dengan kecepatan bit. Oleh karena itu perancangan PLC awal (*Programmable Logic Control*) menggunakan MCS51.

2. Seri AVR

Mikrokontroler Alv and Vegard's Risc processor singkatan dari AVR adalah mikrokontroler RISC 8-bit. Karena RISC ini, satu masa clock dalam RISC inilah sebagian besar kode instruksi dalam RISC dikemas.. AVR adalah sejenis mikrokontroler yang paling umum digunakan dalam elektronik dan instrumentasi. Secara umum, AVR dapat dibagi menjadi 4 kategori. Pada

dasarnya, setiap kelas berbeda dalam hal memori, otorisasi, dan fungsionalitas. Keempat kategori tersebut adalah seri ATTiny, seri AT90Sxx, seri ATmega dan AT86RFxx.

3. Seri PIC

PIC awalnya merupakan singkatan dari Programmable Interface Controller. Divisi Mikroelektronika Instrumen Umum disebut PIC1640. Sekarang Microchipt telah mengumumkan pembuatan PIC keenamnya. PIC sama-sama populer di kalangan pengembang dan penghobi karena harganya murah, dapat digunakan secara luas, memiliki database aplikasi yang besar, dan dapat diprogram melalui koneksi serial.

4. Seri ARM Cortex-M0

ARM adalah prosesor komputer set instruksi (RISC) 32-bit yang dikembangkan oleh ARM Holdings. ARM adalah singkatan dari Advanced RISC Machine (sebelumnya dikenal sebagai Acorn RISC Machine). (Syahwill, 2013).

2.2.2 Mikrokontroler ATmega 328

ATmega 328 merupakan versi mikrokontroler atmel dengan arsitektur RISC, dimana setiap proses eksekusi datanya lebih cepat daripada arsitektur CISC.

ATmega 328 memiliki fungsi yang bermacam-macam, diantaranya:

1. 130 instruksi, hampir semua instruksi dijalankan dalam satu clock cycle.
2. Ini memiliki register tujuan umum 32 x 8-bit.
3. Pada frekuensi clock 16 MHz, kecepatan akses hingga 16 MIPS.
4. Ini memiliki 32 KB Flash, dan Arduino memiliki bootloader yang menggunakan 2 KB Flash sebagai bootloadernya.
5. Memiliki 1 KB EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebagai tempat penyimpanan semi permanen, karena meskipun perangkat dimatikan, EEPROM masih dapat menyimpan data.
6. Ini memiliki 2 KB SRAM (*Static Random Access Memory*).

7. Ini memiliki pin I/O digital 14-pin, 6 pin di antaranya dapat digunakan sebagai pin PWM (*Pulse Width Modulation*).
8. Memiliki antarmuka SPI serial master/slave.

Mikrokontroler ATmega 328 memiliki arsitektur Harvard yang memisahkan memori kode dan memori data sehingga dapat memaksimalkan kinerja mikrokontroler. Instruksi dalam memori program dijalankan dalam satu thread, ketika instruksi dijalankan, instruksi selanjutnya akan diambil dari memori program. Konsep ini memungkinkan instruksi dieksekusi di setiap siklus clock. Sebanyak 32 x 8-bit register digunakan untuk mendukung operasi di ALU (Arithmetic Logic Unit) yang dapat digunakan dalam satu siklus. Dalam mode pengalamatan tidak langsung, total enam register multiguna ini dapat digunakan sebagai tiga register penunjuk 16-bit untuk mendapatkan data dari area memori data. Tiga register penunjuk 16-bit diberi nama X (kombinasi R26 dan R27), register Y (kombinasi R28 dan R29) dan register Z masing-masing dalam format 16-bit. Tiap-tiap alamat memori program mencakup dari instruksi 16 bit atau 32 bit.

Selain register yang disebut di atas, ada register lain yang dipetakan menggunakan metode I/O yang dipetakan memori, yang memiliki ukuran 64 byte. Beberapa register ini digunakan untuk fungsi khusus antar pengguna, seperti timer/counter, interrupt, ADC, USART, SPI, EEPROM dan fungsi I/O lainnya. Register ini menempati memori di alamat 0x20h-0x5fh.(Sriani, 2019).

2.3 Logika Fuzzy

Soft Computing merupakan salah satu komponen dari logika fuzzy. Pada tahun 1965, Profesor Lotfi Asker Zadeh menerbitkan artikel ilmiahnya yang berjudul "*Fuzzy sets*". Terobosan baru adalah konsep memperluas "himpunan" klasik ke himpunan fuzzy, karena klasik (*crisp set*) adalah kasus khusus dari himpunan kabur (Pusadan, 2014). Logika *fuzzy* bukanlah logika kabur, melainkan logika yang digunakan untuk mendeskripsikan ketidakpastian. Logika *fuzzy* didasarkan pada gagasan bahwa segala sesuatu memiliki nilai derajat, yang disebut derajat keanggotaan, dan nilainya berkisar dari 0 hingga 1. Ini berbeda dari

himpunan tegas dengan nilai 0 atau 1. Logika *fuzzy* adalah logika yang samar atau ambigu (*fuzzyness*). Dalam teori logika fuzzy, suatu nilai dapat dianggap benar dan salah pada saat bersamaan, tetapi keberadaan suatu nilai bergantung pada bobot keanggotaan yang dimilikinya. Logika *fuzzy* digunakan untuk menerjemahkan koleksi yang diekspresikan dalam bahasa (linguistik). Misalnya, kecepatan kendaraan dinyatakan sebagai lambat, cukup cepat, cepat, dan sangat cepat. Jadi logika *fuzzy* ini digunakan untuk menunjukkan sejauh mana sebuah nilai itu benar dan sejauh mana nilai itu salah. Logika *fuzzy* adalah cara yang baik untuk mengalokasikan ruang masukan ke ruang keluaran, ini memiliki makna yang kontinyu. Derajat keanggotaan dan derajat kebenaran menyatakan *fuzzy*. (Arindya, 2017).

Ada banyak alasan untuk menggunakan logika *fuzzy*, yaitu:

1. Logika *fuzzy* mudah dipahami.
2. Logika *fuzzy* memiliki toleransi data yang tidak tepat.
3. Logika *fuzzy* sangat fleksibel.
4. Logika *fuzzy* dapat langsung membuat dan menerapkan pengalaman ahli tanpa melalui proses pembelajaran.
5. Logika *fuzzy* dapat memodelkan fungsi nonlinier yang sangat kompleks.
6. Logika *fuzzy* berdasarkan dari bahasa alami.
7. Logika *fuzzy* dapat bekerjasama dengan teknik kontrol konvensional. (Kusumadewi & Purnomo, 2010).

2.3.1 Himpunan *Fuzzy*

Hal-hal yang perlu diperhatikan untuk mengetahui sistem *fuzzy*, diantaranya:

1. Variabel *Fuzzy*

Variabel *fuzzy* merupakan variabel yang perlu dianalisis dalam sistem *fuzzy*.

Contoh: Umur, suhu, permintaan, jarak.

2. Himpunan *Fuzzy*

Himpunan *fuzzy* adalah suatu kelompok yang merepresentasikan keadaan suatu kelompok atau keadaan tertentu suatu variabel *fuzzy*. Contoh: pada variabel

temperatur dibagi menjadi lima *set fuzzy* yaitu dingin, sejuk, normal, hangat dan panas.

3. Semesta Pembicaraan

Semesta pembicaraan merupakan nilai konvensional yang dapat digunakan untuk variabel *fuzzy*. Semesta pembicaraan yang dipertimbangkan adalah kumpulan bilangan real yang meningkat (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan mungkin berbentuk bilangan bulat positif atau negatif. Terkadang nilai semesta pembicaraan ini tidak terbatas pada batas tertingginya. Misalkan: variabel suhu merupakan $[0 \ 40]$.

4. Domain

Domain himpunan *fuzzy* merupakan nilai universal yang valid di semesta pembicaraan dan dapat digunakan dengan himpunan *fuzzy*. Sama seperti di semesta pembicaraan, domain adalah rangkaian bilangan real, dan selalu meningkat (meningkat) secara monoton dari kiri ke kanan. Nilai domain dapat berupa positif atau negatif. Contoh: dingin = $[0 \ 20]$, sejuk = $[15 \ 25]$, normal = $[25 \ 35]$ dan panas = $[30 \ 40]$. (WS, 2015).

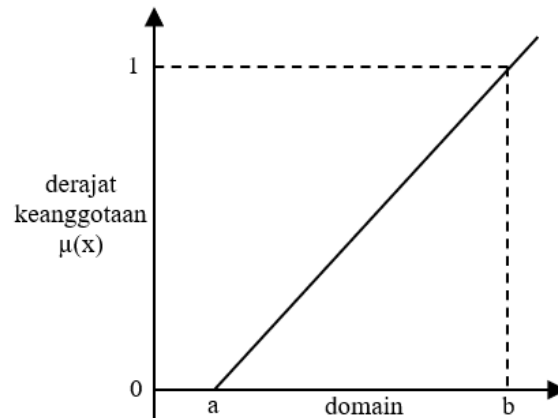
2.3.2 Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menandakan pemetaan titik-titik masukan data pada nilai keanggotaannya (tak jarang pula dianggap menggunakan derajat keanggotaan) yang mempunyai interval antara 0 hingga 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang dapat digunakan diantaranya, yaitu:

1. Representasi Linear

Dalam representasi linier, sebaran derajat keanggotaan masukan digambarkan sebagai garis lurus. Metode ini adalah pilihan termudah dan terbaik untuk menyelesaikan konsep *fuzzy*. Himpunan *fuzzy* linier memiliki 2 keadaan. Pertama, kenaikan yang ditentukan dimulai dari domain dengan derajat

keanggotaan 0, lalu berpindah ke kanan menuju domain dengan derajat keanggotaan yang lebih tinggi.



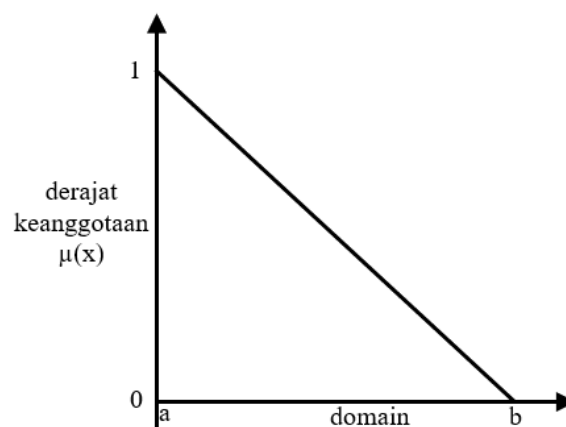
Gambar 2. 2 Representasi Linear Naik

(Kusumadewi & Purnomo, 2010)

Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \leq a \\ (x - a)/(b - a); & a \leq x \leq b \\ 1; & x \geq b \end{cases}$$

Representasi linier kedua berlawanan dengan tampilan linier pertama. Garis lurus dimulai dari sisi kiri dengan domain derajat keanggotaan tertinggi dan kemudian turun ke domain derajat keanggotaan terendah.



Gambar 2. 3 Representasi Linear Turun

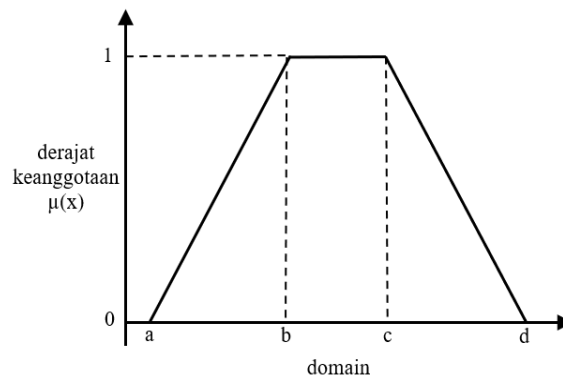
(Kusumadewi & Purnomo, 2010)

Fungsi Keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \geq b \\ (b-x)/(b-a); & a \leq x \leq b \\ 1; & x \leq a \end{cases}$$

2. Representasi Kurva Trapesium

Gambar 2.4 menggambarkan sebuah gambar kurva trapesium, dimana kurva Kurva trapesium mirip dengan bentuk segitiga. Tetapi terdapat beberapa titik saja yang memiliki nilai keanggotaan 1.



Gambar 2. 4 Kurva Trapesium

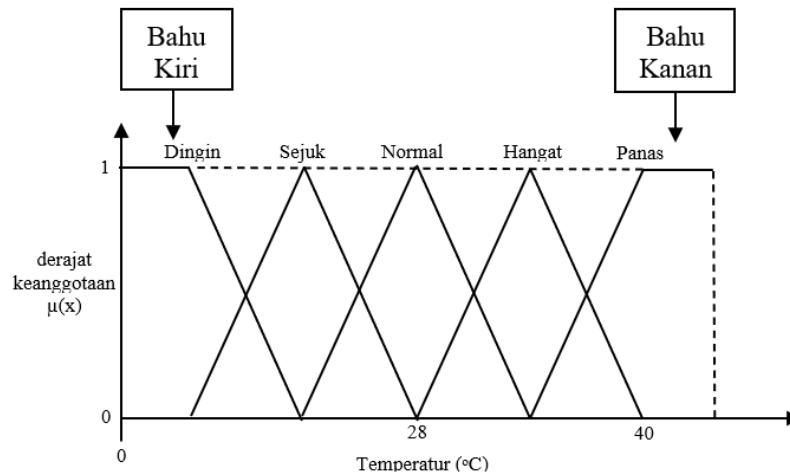
(Kusumadewi & Purnomo, 2010)

Fungsi keanggotaan:

$$\mu\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq d \\ (x-a)/(b-a); & a \leq x \leq b \\ 1; & b \leq x \leq c \\ (d-x)/(d-c); & x \geq d \end{cases}$$

3. Representasi Kurva Bentuk Bahu

Daerah di tengah variabel yang diwakili dalam bentuk segitiga, pada sisi kanan dan kiri akan naik dan turun (misalnya: Dingin pindah ke Sejuk pindah ke Hangat dan pindah ke Panas). Namun, satu sisi variabel tidak selalu berubah. Misalnya, jika sudah mencapai keadaan panas, kenaikan suhu akan terus menjadi panas. Untuk mengakhiri suatu variabel daerah *fuzzy*, satu himpunan *fuzzy* bahu bukan segitiga yang digunakan. Bahu kiri berpindah dari benar ke salah, dan bahu kanan berpindah dari salah ke benar. Gambar 2.6 menggambarkan variabel suhu dalam kaitannya dengan daerah bahu. (Kusumadewi & Purnomo, 2010).



Gambar 2. 5 Daerah 'Bahu' pada variabel Temperatur

(Kusumadewi & Purnomo, 2010)

2.3.3 Operator Logika Fuzzy

Seperti himpunan biasa, ada berbagai operasi khusus untuk menggabungkan dan mengubah himpunan *fuzzy*. Nilai keanggotaan yang dihasilkan oleh operasi dua himpunan biasanya disebut dengan fire strength atau predikat. Lotfi Zadeh menciptakan 3 operator, yaitu:

1. Operator *AND*

Operator ini berhubungan dengan operasi interaktif pada himpunan. Predikat yang merupakan hasil operasi menggunakan operator *AND*, diperoleh dengan mengambil nilai keanggotaan terkecil di antara elemen-elemen himpunan yang berkaitan.

$$\mu_{A \cap B} = \min(\mu_A[x], \mu_B[y])$$

2. Operator *OR*

Operator ini berhubungan dengan operasi gabungan pada himpunan. Predikat α adalah hasil operasi dengan operator *NOT* yang diperoleh dengan mengambil nilai anggota terbesar di antara elemen-elemen himpunan yang berkaitan.

$$\mu_{A \cup B} = \max(\mu_A[x], \mu_B[y])$$

3. Operator *NOT*

perator ini menangani komplemen dari suatu himpunan. Predikat α adalah hasil operasi dengan operator *NOT* yang diperoleh dengan mengurangkan nilai keanggotaan suatu elemen dalam himpunan yang berkaitan dari 1. (WS, 2015)

$$\mu_{A'} = 1 - \mu_A[x]$$

2.4 Sistem Infrensi *Fuzzy* Sugeno

Gaya penalaran Sugeno hampir identik dengan penalaran Mamdani, Satu-satunya perbedaan adalah bahwa keluaran sistem (konsekuen) tidak berupa himpunan *fuzzy*, melainkan konstanta atau persamaan linier. Metode ini diperkenalkan oleh Takagi, Sugeni dan Kang pada tahun 1985, sehingga metode ini sering juga dinamakan dengan metode TSK (Kusumadewi & Purnomo, 2010). Sistem inferensi *fuzzy* Sugeno ini memerlukan empat tahapan, yaitu:

1. Fuzzyfikasi

Fuzzyfikasi merupakan proses mengubah nilai variabel numerik ke nilai variabel linguistik. Dengan kata lain, fuzzifikasi merupakan pemetaan dari ruang input ke himpunan fuzzy yang didefinisikan pada semesta pembicaraan variabel input. (Wati, 2011)

2. Pembentukan aturan *fuzzy*

Aturan *fuzzy* mendefinisikan hubungan antara fungsi keanggotaan dan bentuk fungsi keanggotaan hasil. Metode Sugeno menghasilkan keluaran sistem (konsekuen) sebagai konstanta atau persamaan linier dan sistem tidak berupa himpunan *fuzzy*. Dalam Metode Sugeno terbagi menjadi dua model *fuzzy*, yaitu:

a. *Fuzzy* Sugeno orde nol

Bentuk umum dari model fuzzy sugeno orde nol, yaitu:

$$IF = (x_1 \text{ is } A_1) \text{ o } (x_2 \text{ is } A_2) \text{ o } \dots \text{ o } (x_N \text{ is } A_N) THEN z = k$$

Dimana A_i adalah himpunan fuzzy ke- i anteseden dan k adalah konstanta tegas konsekuen.

b. *Fuzzy* Sugeno orde satu

Bentuk secara umum dari model fuzzy sugeno orde satu, yaitu:

$$IF = (x_1 \text{ is } A_1) o (x_2 \text{ is } A_2) o \dots (x_N \text{ is } A_N) THEN z$$

$$= p_1 * x_1 + \dots + p_N * x_N + q$$

Di mana A_i adalah himpunan *fuzzy* ke- i anteseden, p_i adalah konstanta tegas dari ke- i , dan q adalah konstanta konsekuen.

3. Komposisi Aturan

Inferensi didapat dari pengumpulan dan korelasi aturan yaitu menghitung hasil dari $\sum_{r=1}^R \alpha_r z_r$ dengan R jumlah aturan, α_r *fire strength* ke- r dan z_r output pada anteseden aturan ke- r . (Irfan, 2016)

4. Defuzzifikasi

Defuzzifikasi merupakan pemetaan dari ruang aksi kendali *fuzzy* yang didefinisikan pada semesta pembicaraan keluaran ke ruang aksi kendali *nonfuzzy* (numerik) (Wati, 2011). Pada proses ini keluaran berupa bilangan *crisp*. Defuzzifikasi dilakukan dengan cara mencari nilai rata-ratanya yaitu

$$\frac{\sum_{r=1}^R \alpha_r z_r}{\sum_{r=1}^R \alpha_r} \text{ (Irfan, 2016).}$$

2.5 Peltier

Komponen listrik yang dapat menghasilkan suhu rendah di satu sisi dan suhu tinggi di sisi lain ketika arus dialirkan merupakan pengertian dari elemen peltier atau *thermoelectric cooler* (TEC). Kenaikan atau penurunan suhu tergantung pada arah arus. Aplikasi yang sering digunakan dalam sistem Peltier menggunakan suhu bagian dingin yang diperoleh, yaitu sebagai pendingin untuk prosesor, pendingin udara mikro, pendingin di dispenser, pendingin minuman, dan pengatur suhu akuarium. Pada saat yang sama, suhu komponen bersuhu tinggi dapat dikurangi dengan memasang heatsink dan fan. Nilai koefisien kinerja Peltier (COP) merupakan perbandingan kalori yang dihasilkan oleh unsur Peltier terhadap energi yang diberikan. (Lubis, Z. et al., 2017). Komponen pendingin termoelektrik (TEC) terdiri dari semikonduktor tipe-P dan tipe-N, yang masing-masing dihubungkan secara elektrik secara seri tetapi secara termis dihubungkan secara paralel. Pendingin termoelektrik berdasarkan efek Peltier Ketika arus searah dialirkan ke elemen Peltier, satu sisi elemen Peltier akan mendingin (menyerap panas), dan sisi

lainnya akan memanaskan (melepaskan panas). Hal tersebut dapat dilihat dari uraian Gambar 2.7. Skema elemen peltier, Energi dalam bentuk energi listrik merupakan panas yang dilepaskan ke lingkungan dikurangi panas yang diserap.



Gambar 2. 6 Skema elemen peltier
(Agus Salim & Indarto, 2018)

Seperti yang ditunjukkan pada Gambar 2.7, beberapa bentuk energi listrik dihasilkan oleh panas. Sisi dingin unsur Peltier didinginkan dari tingkat energi yang lebih rendah dalam semikonduktor P ke tingkat energi yang lebih tinggi, yaitu tingkat energi yang lebih tinggi, oleh aliran elektron. Semikonduktor tipe-N. Agar elektron tipe-P mengalir pada tingkat energi yang lebih rendah, elektron akan menyerap energi dan mendinginkan sisi-sisinya. (Agus Salim & Indarto, 2018).

2.6 Power Supply

Alat listrik yang dapat memberikan energi ke peralatan listrik atau alat elektronik lainnya disebut dengan *power supply* atau dalam bahasa Indonesia disebut dengan catu daya. Intinya, catu daya ini memerlukan penggunaan sumber energi, yang kemudian diubah menjadi energi listrik untuk digunakan oleh perangkat elektronik lainnya. Oleh karena itu, catu daya terkadang disebut sebagai konverter. (Cholish et al., 2017).



Gambar 2. 7 *Power Supply*

(Kho, 2014)

Secara umum, catu daya dapat dibagi menjadi tiga kategori berdasarkan fungsinya, bentuk mekanis, dan metode konversinya. Berikut ini adalah uraian singkat dari ketiga kategori, yaitu:

1. Berdasarkan Fungsi

Berdasarkan fungsinya, catu daya dapat dibagi menjadi *Regulated Power Supply*, *Unregulated Power Supply*, dan *Adjustable Power Supply*.

- a. *Regulated Power Supply* merupakan catu daya ini dapat menjaga kestabilan tegangan dan arus walaupun terjadi perubahan sumber tegangan dan arus listrik.
- b. *Unregulated Power Supply* merupakan catu daya yang dimana tegangan atau arus listriknya dapat bervariasi dengan perubahan beban atau sumber tegangan listriknya terjadi perubahan.
- c. *Adjustable Power Supply* merupakan Catu daya yang tegangan atau arusnya dapat disesuaikan dengan tombol mekanis. Ada dua jenis *Adjustable Power Supply*, yaitu *Regulated Adjustable Power Supply* dan *Unregulated Adjustable Power Supply*.

2. Berdasarkan Bentuk

Pada perangkat elektronik seperti televisi, monitor komputer, komputer desktop dan pemutar DVD, *power supply* biasanya terletak di dalam atau diletakkan ke dalam perangkat tersebut, sehingga kita sebagai konsumen tidak dapat melihatnya secara langsung. Karenanya, hanya kabel daya yang terlihat dari luar.

Catu daya ini disebut dengan catu daya internal (*Built-in*). Namun ada juga peralatan elektronik yang menggunakan catu daya yang berdiri sendiri (*stand alone*), misalnya Pengisi daya seluler dan adaptor komputer portabel. Ada juga catu daya yang *stand alone* yang cukup besar dan dapat menyesuaikan tegangannya sesuai kebutuhan.

3. Berdasarkan Metode Konversi

Menurut metode konversi, catu daya dapat dibagi menjadi catu daya linier dan catu daya switching. Catu daya linier secara langsung mengubah tegangan saluran dari tegangan input sedangkan catu daya switching harus terlebih dahulu mengubah tegangan input menjadi AC atau DC. (Kho, 2014).

2.7 *Waterblock*

Waterblock adalah bagian dari heatsink sistem pendingin air. Komponen ini terdiri dari dua bagian, aluminium dan tembaga, yang digabungkan menjadi satu bagian untuk membentuk blok pendingin yang berisi rongga tempat bahan cair mengalir. Setiap rongga (saluran) memiliki saluran masuk dan keluar di ujungnya. Keluar, seperti sambungan dengan selang atau pipa, yang membuang material air pendingin melalui sistem pendingin air lainnya. Saluran atau rongga tersebut biasanya berbentuk seperti pola zigzag, untuk dapat mencapai area perpindahan panas efektif yang lebih besar. (Riandika et al., 2018).

2.8 *Fan*

Desain kipas menjadi semakin beragam dalam hal ukuran, peletakan posisi, dan fungsi. Pendingin udara, penyegar udara, ventilasi (*exhaust fan*), pengering (biasanya dengan komponen yang menghasilkan panas) merupakan fungsi umum dari kipas. *Fan mini* (*fan* yang dipegang menggunakan tangan berdayakan energi baterai) merupakan salah satu varian ukuran dari kipas angin, komputer juga menggunakan fan yang digunakan sebagai kipas untuk mendinginkan prosesor, catu daya, dan casing. Agar tidak melebihi batas suhu yang ditetapkan maka digunakan kipas angin untuk menjaga suhu udara. Kipas angin juga dipasang di bagian bawah

laptop untuk memberikan udara dan membantu menurunkan suhu laptop. Kecepatan hembusan kipas dapat diubah dengan tiga cara: melalui remote control, pemutar dan tali penarik. Bilah kipas dibagi menjadi dua bagian yaitu *centrifugal* (arah angin mengalir searah poros kipas) dan *axial* (arah angin sejajar dengan arah poros kipas). (Arifin et al., 2017).



Gambar 2. 8 DC Fan

(Arifin et al., 2017)

2.9 Heatsink

Heatsink merupakan plat-plat yang terbuat dari aluminium atau sirip tembaga, dengan sirip untuk meningkatkan area kontakannya dengan udara atau pendingin lainnya. (Pawawoi & Zulfahmi, 2019). Heatsink mengadopsi konsep perpindahan panas, yang dapat mentransfer panas dari lingkungan bersuhu tinggi ke lingkungan bersuhu rendah. Kinerja heat sink bergantung pada material dan geometrinya. Distribusi panas pada kotak pendingin hampir seragam di semua bagian kotak pendingin. Komponen pendingin akan mempengaruhi distribusi panas yang dihasilkan. Oleh karena itu, untuk material yang baik dengan konduktivitas termal yang tinggi, dalam hal ini pendingin berbahan dasar tembaga lebih baik daripada aluminium. (Arifin et al., 2017).



Gambar 2. 9 Heatsink dari bahan alumunium

(Pawawoi & Zulfahmi, 2019)

2.10 Pompa Air DC

Untuk mengangkat benda cair dari rendah ke tinggi digunakanlah alat mekanis yaitu pompa, atau untuk mentransfer benda cair dari area bertekanan rendah ke area bertekanan tinggi, dan digunakan sebagai penguat aliran dalam sistem perpipaan jaringan. Pompa juga dapat digunakan dalam proses yang membutuhkan tekanan hidrolik tinggi. Ini bisa ditemukan di alat berat. Mesin sedang beroperasi membutuhkan bobot tekanan buang yang lebih tinggi dan tekanan isap yang lebih rendah. Karena tekanan rendah di sisi hisap pompa, cairan naik dari kedalaman tertentu dan karena disisi tekanan buang tinggi maka, cairan naik ke ketinggian yang diperlukan. (Arifin et al., 2017).



Gambar 2. 10 Pompa Air DC

(Arifin et al., 2017)

2.11 Mosfet IRF520

IRF520N adalah Mosfet dengan daya N-Channel. Mosfet dapat mengalihkan beban dengan arus DC hingga 9,2A dan bekerja di bawah 100V. Mosfet ini memiliki tegangan ambang batas gerbang rendah hanya 4V, ini berarti Mosfet dapat dihidupkan meskipun dengan 5V dari pin GPIO mikrokontroler seperti Arduino. Tetapi ini tidak berarti bahwa Mosfet akan menyala sepenuhnya hanya dengan 5V, dibutuhkan sekitar 10V yang disuplai ke pin gerbang untuk menyala sepenuhnya dan memasok arus kolektor 9,2A. Selain itu Mosfet juga memiliki kecepatan switching yang baik dan karenanya dapat digunakan di rangkaian konverter DC-DC (COMPONENTS101, 2019).

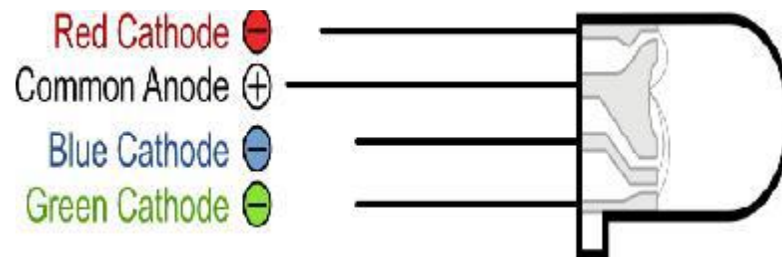


Gambar 2. 11 Mosfet IRF520
(COMPONENTS101, 2019)

2.12 LED RGB

LED RGB (*red, green, blue*) adalah LED yang mencakup tiga warna yang menyatu dalam satu paket. Terlihat pada gambar 2. 12 merupakan bentuk dari LED RGB yang mencakup dari LED *red, green dan blue*. Pada gambar 2. 12 LED RGB mempunyai jenis *common anode* dimana terminal anode adalah LED *red, green, dan blue* yang terhubung dan katoda dalam keadaan terbuka, penyesuaian warna hidup LED dapat dilakukan dengan mengontrol arus yang disuplai ke LED, sehingga mempengaruhi kecerahan LED. Kemudian pemberian arus yang berbeda pada setiap terminal katoda LED RGB akan menghasilkan LED dengan warna yang berbeda. Pengaturan arus dapat dilakukan dengan menggunakan metode

modulasi lebar pulsa (PWM). PWM adalah metode modulasi pulsa yang dilakukan dengan mengatur siklus kerja gelombang persegi. Oleh karena itu, metode ini dapat digunakan untuk memeriksa daya rata-rata yang dikirim ke setiap LED RGB. (Kurnia Utama et al., 2018).



Gambar 2. 12 LED RGB

(Kurnia Utama et al., 2018)

2.13 Stepdown

StepDown LM2596 DC ke DC adalah konverter untuk menurunkan dan mengubah tegangan *input* DC menjadi tegangan DC. Spesifikasi *Stepdown* LM2596:

1. Tegangan masukan : DC 3V-40V
2. Tegangan keluaran : DC 1.5V-35V
3. Arus maksimal : 3A
4. Skala papan PCB : 42mm x 20mm x 14mm (Hamdani et al., 2019)



Gambar 2. 13 Stepdown

(Hamdani et al., 2019)

2.14 Kabel Data Mini USB

Port USB saat ini biasanya sudah terdapat pada komputer atau laptop. USB adalah singkatan dari *Universal Serial Bus*, yang merupakan konektor untuk perangkat elektronik eksternal seperti printer, mouse, dan keyboard komputer. USB juga dapat digunakan sebagai media perantara untuk mentransfer data ke komputer atau laptop dari perangkat HDD eksternal, FDD hingga *smartphone*. Jika diperhatikan, semua *port* USB cenderung berukuran sama di komputer atau di laptop. Tetapi pada saat ini ada banyak jenis USB, salah satunya adalah kabel data mini USB yang ukurannya lebih kecil dari USB tipe A. Tipe USB yang umumnya digunakan di berbagai perangkat *mobile* seperti *smartphone* adalah tipe Mini USB. (Fauziah jihan, 2021).

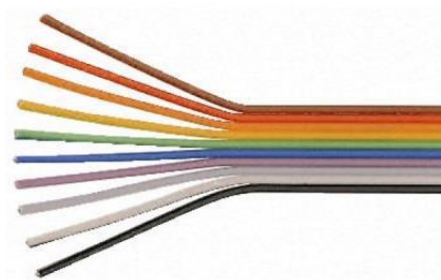


Gambar 2. 14 Kabel data Mini USB

(Fauziah jihan, 2021)

2.15 Kabel Pelangi

Kabel pelangi adalah kabel yang digunakan untuk menghubungkan arus listrik antar komponen. Kabel pelangi ini memiliki 10 warna yang berbeda dimana perbedaan warna ini sangat berguna untuk membedakan jenis arus listrik yang disambungkan antar komponen agar tidak terjadi kesalahan dalam perakitan komponen pada robot yang dapat merusak komponen tersebut. (Rustan et al., 2017).



Gambar 2. 15 Kabel Pelangi

(Rustan et al., 2017)

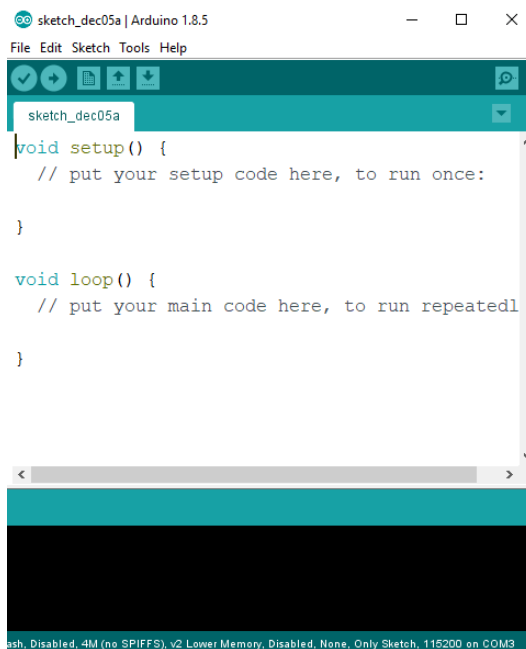
2.16 Microsoft Visual Studio

Microsoft Corporation telah membuat perangkat lunak IDE yaitu Microsoft Visual Studio. Microsoft Visual Studio bisa dipakai untuk mengembangkan aplikasi dalam kode asli (seperti bahasa *machine* yang berjalan pada Windows) atau kode yang dikelola (sebagai bentuk Bahasa Microsoft *Intermediate .Net Framework*). Bahasa pemrograman yang mendukung Microsoft Visual Studio berbeda-beda, seperti bahasa pemrograman C++, visual basic dan visual C#. Selain itu bahasa pemrograman seperti M, Python, dan Ruby juga didukung oleh microsoft visual studio yang semuanya diperoleh dari paket *add-on* terpisah dari microsoft visual studio. (Putri & Azpar, 2017).

2.17 Arduino IDE

Perangkat lunak Arduino merupakan aplikasi yang ditulis dengan Java. Ini pertama kali dikembangkan pada tahun 2005 pada tanggal 25 Agustus 2005 untuk versi 0001. Perangkat lunak ini ditujukan untuk pemrograman mikrokontroler untuk papan AVR yang diproduksi oleh Atmel. Arduino tunduk pada GPL (*General Public License*), yang berarti bahwa itu adalah perangkat lunak *open source* yang dapat disalin dan dikembangkan oleh siapa saja. Terlihat pada gambar 2.16 merupakan tampilan dari arduino IDE yang dimana Penggunaan perangkat lunak

ini mencakup dari dua bagian utama yaitu “*void setup()*” dan “*void loop()*”. “*void setup()*” adalah bagian dari program yang akan dieksekusi sekali selama eksekusi program, biasanya digunakan sebagai lokasi untuk menginisialisasi variabel. Sementara itu perintah “*void loop()*” adalah bagian dari program yang dieksekusi berulang-ulang, disinilah program utama dijalankan. (Rustan et al., 2017)

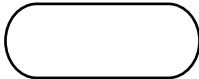
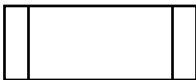

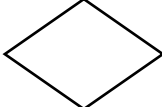

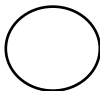





Gambar 2. 16 Tampilan Arduino IDE

2.18 Flowchart

Gambar atau grafik yang menunjukkan urutan dan hubungan antara proses dengan instruksinya merupakan pengertian dari *flowchart*. Gambar dari *flowchart* ini diwakili oleh simbol-simbol. Oleh karena itu, setiap simbol menggambarkan proses tertentu. Sementara itu kaitan antara proses digambarkan dalam bentuk garis penghubung. (Suhardi, 2019). Berikut ini merupakan fungsi dari masing-masing simbol dari *flowchart*:

Tabel 2. 1 Simbol dan Fungsi *Flowchart*
(Santoso & Nurmalina, 2017)

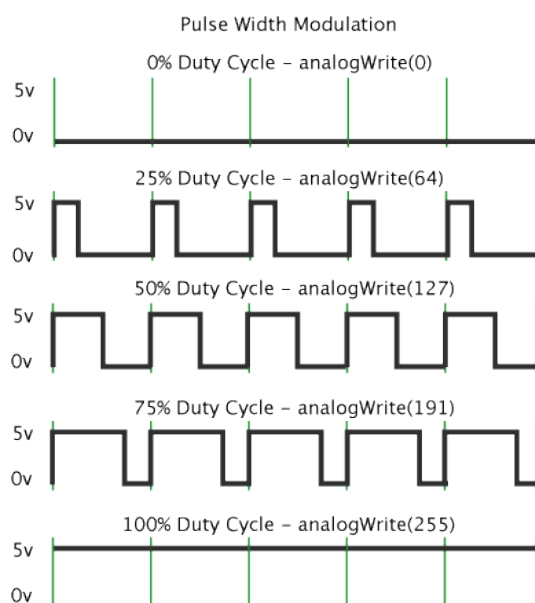
Simbol	Fungsi
	Digunakan untuk mulai dan akhir pada program
	Digunakan untuk <i>sub</i> program
	Digunakan untuk proses pengolahan data
	Digunakan untuk pengkondisian atau pilihan
	Digunakan untuk input dan output dari program
	Digunakan untuk menyambung bagian <i>flowchart</i> pada halaman yang sama
	Digunakan untuk menyambung bagian <i>flowchart</i> pada halaman yang berbeda
	Digunakan untuk inisialisasi program
	Digunakan untuk penghubung antar program

2.19 Pulse Width Modulation (PWM)

PWM (*Pulse Width Modulation*) adalah suatu metode pengontrolan kecepatan putaran, dalam hal ini motor DC untuk mengendalikan robot. PWM dapat dibangkitkan dengan empat cara, yaitu:

- Proses analog
- Proses digital
- IC diskrit
- Mikrokontroler

Proses PWM dijalankan oleh mikrokontroler. Proses PWM ini mengatur lebar atau sempitnya periode pulsa efektif yang dikirim dari mikrokontroler ke *driver* motor. Pada pengaturan kecepatan robot, nilai PWM dimulai dari 0 hingga 255. Secara analog besaran PWM dihitung dalam persentase, nilai ini diperoleh dengan perbandingan: $T_{high} / (T_{high} + T_{low}) * 100\%$. Dimana T adalah periode atau waktu tempuh pulsa, dibagi menjadi puncak positif (T_{high}) dan puncak negatif (T_{low}).



Gambar 2. 17 *Pulse Witdh Modulation*

(Arduino.cc, 2018)

Semakin rapat periode antar pulsa, maka semakin tinggi frekuensinya, yang berarti kecepatannya meningkat. Semakin lebar jarak antara pulsa, maka semakin rendah frekuensinya, yang berarti kecepatannya menurun. Kondisi pemberian kecepatan harus disesuaikan dengan kondisi rute yang akan diikuti oleh robot. Misalnya, jika jalan lurus, menanjak atau menurun, harus mendapatkan nilai PWM yang sesuai (Sujarwata, 2016).

2.20 Komunikasi Serial

Komunikasi serial merupakan komunikasi yang datanya dikirimkan perbit secara berurutan dan bergantian. Keuntungan dari jenis komunikasi ini dibandingkan dengan komunikasi paralel, hanya diperlukan satu jalur dan beberapa kabel. Komunikasi serial pada dasarnya adalah komunikasi yang mengirimkan data dalam satuan bit, sehingga lebih lambat dari komunikasi paralel. Dengan kata lain, komunikasi serial merupakan salah satu metode transmisi data yang selalu dikirimkan melalui rangkaian kabel. Pada prinsipnya komunikasi serial merupakan kasus khusus dari komunikasi paralel, dan nilainya $n = 1$, atau dengan kata lain merupakan bentuk komunikasi paralel hanya dengan beberapa kabel, dan hanya satu bit data yang ditransmisikan pada waktu yang bersamaan. Ada dua jenis komunikasi serial yaitu serial asinkron dan serial sinkron. Serial sinkron adalah sejenis komunikasi yang hanya ada satu pihak (pengirim atau penerima) yang menghasilkan jam dan mengirimkan jam bersama dengan data. Komunikasi keyboard adalah contoh penggunaan antarmuka serial sinkron. Serial asinkron merupakan dua frekuensi clock yang harus sama dan disinkronkan. Setelah sinkronisasi, pengirim mengirimkan data sesuai frekuensinya. Jam pemancar dan penerima membaca data sesuai dengan frekuensi jam penerima. Contoh penggunaan antarmuka serial asinkron adalah *Universal Asynchronous Receiver Transmitter* (UART) yang digunakan pada antarmuka serial port (COM) komputer. Dibandingkan dengan antarmuka pada saluran paralel, antarmuka serial lebih rumit / sulit karena: 1. Dari banyak perangkat keras: perangkat tambahan bernama UART (*Universal Asynchronous Receiver / Transmitter*) digunakan untuk mengubah data

dari paralel ke Serial dan sebaliknya 2. Mengenai perangkat lunak: gunakan atau melibatkan lebih banyak register. Di sisi lain, antarmuka serial memiliki banyak keunggulan dibandingkan antarmuka paralel, antara lain: 1. Kabel komunikasi serial bisa lebih panjang dari kabel paralel; data dalam komunikasi serial adalah- Tegangan -3 hingga -25 volt dikirim ke logika '1', dan logika '0' dikirim ke tegangan +3 hingga +25 volt ke logika '0', sehingga tegangan melalui komunikasi serial memiliki perubahan tegangan maksimum 50 volt dan hanya 5 volt untuk komunikasi paralel. Ini dapat menyebabkan gangguan saat menggunakan kabel yang lebih panjang. Lebih mudah dipecahkan daripada paralel. 2. Lebih sedikit kabel serial; Anda dapat menghubungkan dua komputer jarak jauh dengan hanya tiga kabel untuk mengkonfigurasi modem sepenuhnya, yaitu TXD (saluran transmisi), RXD (saluran penerima) dan *Ground*. Bayangkan jika menggunakan teknologi paralel, ada 20-25 kabel. Setiap komputer dengan kemampuan komunikasi serial harus membayar "biaya" yang sedikit lebih mahal untuk antarmuka serial. 3. Banyak perangkat modern (PDA, manajer, telepon seluler, dll.) Menggunakan teknologi inframerah. Warna merah menunjukkan transmisi data, dalam hal ini pengiriman data adalah serial. IRDA-1 (spesifikasi inframerah pertama) dapat mengirim data dengan kecepatan 115,2 kbps, dan perangkat UART mendukung konsep serial. Untuk menghemat daya, panjang pulsa hanya dipersingkat menjadi 3/16 dari standar RS-232. 4. Untuk teknologi sistem tertanam, banyak mikrokontroler dilengkapi dengan komunikasi serial (seri RISC dan CISC) atau antarmuka komunikasi serial (SCI). Ketika SCI dibangun menjadi 1C, mikrokontroler mengurangi jumlah output, jadi hanya diperlukan 2 pin utama TxD dan RxD (tidak di-ground). (Triansah, 2017).

2.21 Penelitian Terdahulu

Penelitian Asrofi dan Bambang Eka Purnama dengan judul "Rancang Bangun Alat Kontrol Otomatis Pendingin Komputer Berbasis Mikrokontroler ATMEGA8L". Pada penelitian ini menggunakan metode literatur, observasi, perancangan sistem alat, pembangunan alat, pengujian sistem kerja alat dan

implementasi alat dan pada penelitian juga ini menggunakan sebuah mikrokontroler ATMEGA8L sebagai pengontrol keseluruhan sistem, sebuah sensor LM35 digunakan sebagai pendeteksi suhu, sebuah relay yang digunakan untuk mengontrol kipas, sebuah LCD 16x2 yang digunakan untuk menampilkan informasi yang di dapat dari sensor suhu dan sebuah kipas yang digunakan untuk media pendingin komputer. (Asrofi, 2013).

BAB III

METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Adapun tempat dan waktu penelitian Penerapan Logika *Fuzzy* Pada Perancangan *Cooling* Komputer Otomatis Menggunakan Peltier Berbasis Mikrokontroler Atmega 328, sebagai berikut:

3.1.1 Tempat Penelitian

Tempat penelitian Penerapan Logika *Fuzzy* Pada Perancangan *Cooling* Komputer Otomatis Menggunakan Peltier Berbasis Mikrokontroler Atmega 328, dilakukan di Laboratorium Robotik Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara.

3.1.2 Waktu dan Jadwal Pelaksanaan Penelitian

Waktu dan jadwal pelaksanaan penelitian Penerapan Logika *Fuzzy* Pada Perancangan *Cooling* Komputer Otomatis Menggunakan Peltier Berbasis Mikrokontroler Atmega 328, yaitu pada bulan Maret sampai dengan bulan Juli 2021.

3.2 Bahan dan Alat Penelitian

Adapun bahan dan alat dalam penelitian Penerapan Logika *Fuzzy* Pada Perancangan *Cooling* Komputer Otomatis Menggunakan Peltier Berbasis Mikrokontroler Atmega 328 yang digunakan, sebagai berikut:

3.2.1 Perangkat Keras

Adapun perangkat keras yang digunakan dalam penelitian Penerapan Logika *Fuzzy* Pada Perancangan *Cooling* Komputer Otomatis Menggunakan Peltier berbasis Mikrokontroler Atmega 328, sebagai berikut:

Tabel 3. 1 Perangkat Keras

No.	Nama Perangkat Keras	Jumlah	Spesifikasi
1.	Mikrokontroler Atmega 328	1 buah	Pin I/O <i>digital</i> 14 (6 diantara pin PWM), 6 pin <i>analog</i> , SRAM 2KB, EPROM 1KB, <i>flash</i> memori 32KB, <i>clock speed</i> 16MHz
2.	Mosfet IRF520	2 buah	Tegangan <i>output</i> 0-24V, tegangan <i>input</i> 3,3V/5V, maksimum arus <i>output</i> 5A
3.	Peltier	1 buah	Model TEC1-12706, tegangan kerja 0-15V, daya max 91,2W, dimensi 4cm x 4cm x 4mm, berat 25gr
4.	Water blok	1 buah	Dimensi 40x40x12mm, bahan alumunium, warna silver
5.	Water blok CPU	1 buah	Bahan POM, benang tipe G $\frac{1}{4}$, diameter luar 12.9mm, warna hitam, berat 155gr
6.	<i>Heatsink</i>	1 buah	<i>Heatsink</i> 2 pipe, bahan alumunium
7.	Fan	2 buah	Dimensi 100 x 100 mm, tegangan <i>input</i> 12V
8.	Power Supply 12V	1 buah	Tegangan <i>input</i> 110-220V AC, tegangan <i>output</i> 12V DC, daya maksimal 10A, dimensi 20x10x4CM

9.	Pompa Air DC	1 buah	Tegangan <i>input</i> 0-12V
10.	LED RGB	1 buah	Ukuran 5mm, jumlah pin 4 pin, <i>Red voltage</i> 1.8-2.0V, <i>Green voltage</i> 3.2-3.4V, <i>Blue voltage</i> 3.2-3.4V
11.	Selang Air	2 meter	Panjang 200cm, warna transparant
12.	Wadah Air	1 buah	Kapasitas 1000ml, tinggi 23,5cm, diameter 8,4cm, berat 70gr
13.	Kabel Data Mini USB	1 buah	Panjang 30cm, warna biru transparant
14.	Kabel Pelangi	1 meter	Panjang 100cm, warna <i>rainbow</i> , isi 10p
15.	Step Down	1 buah	<i>Input voltage</i> 3.2-35V, <i>ouput voltage</i> 1.25-35V, maksimum arus <i>output</i> 3A
16.	Coolant	500 ml	Isi 1000ml, warna hijau, prediluted 33%
17.	Komputer	1 unit	Prosesor <i>core</i> i3 6100, memori 8GB DDR4, SSD 120GB, <i>socket</i> 1151

3.2.2 Perangkat Lunak

Adapun perangkat lunak yang digunakan dalam penelitian Penerapan Logika *Fuzzy* Pada Perancangan *Cooling* Komputer Otomatis Menggunakan Peltier Berbasis Mikrokontroler Atmega 328, sebagai berikut:

Tabel 3. 2 Perangkat Lunak

No.	Nama Perangkat Lunak
1.	Arduino IDE
2.	Microsoft Visual Studio 2015

3.3 Cara Kerja

Adapun cara kerja dalam penelitian Penerapan Logika *Fuzzy* Pada Perancangan *Cooling* Komputer Otomatis Menggunakan Peltier Berbasis Mikrokontroler Atmega 328, sebagai berikut:

3.3.1 Perencanaan

Dalam proses perencanaan penelitian ini, terdapat beberapa tahapan yang dilakukan peneliti dalam menyelesaikan penelitian ini, sebagai berikut:

1. Pengumpulan Data

Pada penelitian ini terdapat dua teknik yang dilakukan dalam melakukan pengumpulan data, yaitu dengan studi Observasi dan studi Literatur.

2. Analisa Kebutuhan

Pada penelitian ini analisa kebutuhan dapat mencakup tentang kebutuhan sistem.

3. Perancangan

Pada perancangan ini peneliti melakukan perancangan yang mencakup perancangan skema alat, *layout* diagram alat, proses *fuzzy* pada alat, *layout interface* dan *flowchart*.

4. Pengujian

Pada tahapan ini peneliti melakukan pengujian untuk menguji kinerja alat dan sistem yang telah dibuat dan untuk mengetahui tingkat keberhasilan dan kegagalan dari alat dan sistem yang telah dirancang.

5. Penerapan

Penelitian ini nantinya dapat diterapkan pada komputer yang mengalami permasalahan lambat atau lag yang disebabkan oleh *overheat*.

3.3.2 Teknik Pengumpulan Data

Terdapat dua teknik pengumpulan data yang dilakukan peneliti untuk menyelesaikan penelitian ini, sebagai berikut:

1. Studi Observasi

Pada penelitian ini peneliti melakukan pengumpulan data yang salah satunya dengan teknik studi Observasi, yang dimana teknik ini dilakukan melalui pengamatan secara langsung di lapangan dengan pengujian alat pada komputer milik peneliti sendiri untuk mendapatkan hasil yang nyata. Pada hakikatnya observasi adalah sebuah kegiatan dengan menggunakan pancaindra, baik itu penciuman, pendengaran, penglihatan, pengecap dan peraba untuk mendapatkan informasi yang diperlukan untuk menjawab masalah penelitian.

2. Studi Literatur

Pada penelitian ini peneliti melakukan pengumpulan data yang salah satunya dengan teknik studi Literatur, yang dimana teknik ini dilakukan dengan melakukan pencarian referensi teori yang relevan dengan kasus dan permasalahan yang ditemukan. Pencarian referensi dapat dilakukan dengan studi pustaka dengan beberapa buku, jurnal, artikel dan tugas akhir yang berhubungan dengan penerapan logika *fuzzy* pada perancangan *cooling* komputer otomatis menggunakan peltier berbasis mikrokontroler atmega 328 yang dapat memperkuat dasar teori dan permasalahan dalam penelitian ini.

3.3.3 Analisa Kebutuhan

Pada bagian ini dilakukan analisa kebutuhan yang akan menguraikan mengenai kebutuhan yang wajib dimiliki oleh sistem. Pada analisa kebutuhan ini terdapat dua bagian, sebagai berikut:

1. Analisa Kebutuhan Fungsional

Didalam analisa kebutuhan fungsional ini terdapat beberapa fungsi utama sistem, sebagai berikut:

- a. Sistem dapat mendeteksi suhu pada komputer melalui komunikasi serial antara mikrokontroler dengan komputer.

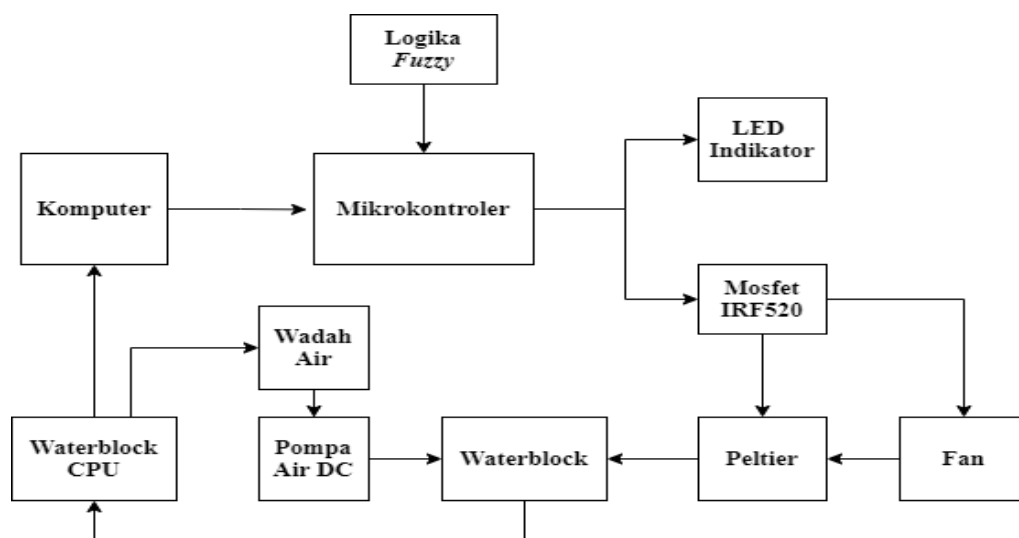
- b. Sistem dapat menampilkan data suhu dari komputer melalui *Software interface*.
 - c. Sistem dapat dihidupkan atau dimatikan melalui *software interface*.
 - d. Sistem dapat menurunkan suhu komputer secara otomatis melalui peltier yang diatur dari mikrokontroler melalui Mosfet IRF520.
2. Analisa Kebutuhan Non Fungsional
- Untuk mendukung kinerja dari sistem ini terdapat beberapa non fungsional yang dimiliki sistem, sebagai berikut:
- a. Sistem memiliki lampu indikator untuk menunjukkan nilai dari suhu tertentu dan indikator dari komputer dan mikrokontroler terhubung atau tidak.
 - b. Sistem *software interface* dapat dengan mudah dioperasikan oleh pengguna.
 - c. Pada sistem *software interface* pengguna juga dapat melihat load dari CPU.

3.3.4 Perancangan

Pada penelitian ini, proses perancangan mencakup empat bagian yaitu perancangan sistem, *flowchart* sistem, perancangan perangkat keras dan perancangan perangkat lunak.

1. Perancangan sistem

Pada tahap perancangan sistem ini akan digambarkan dalam bentuk diagram blok, seperti yang terlihat pada gambar 3.1. Perancangan sistem ini bertujuan untuk memberikan gambaran secara umum dari sistem yang dibuat.



Gambar 3. 1 Diagram blok sistem

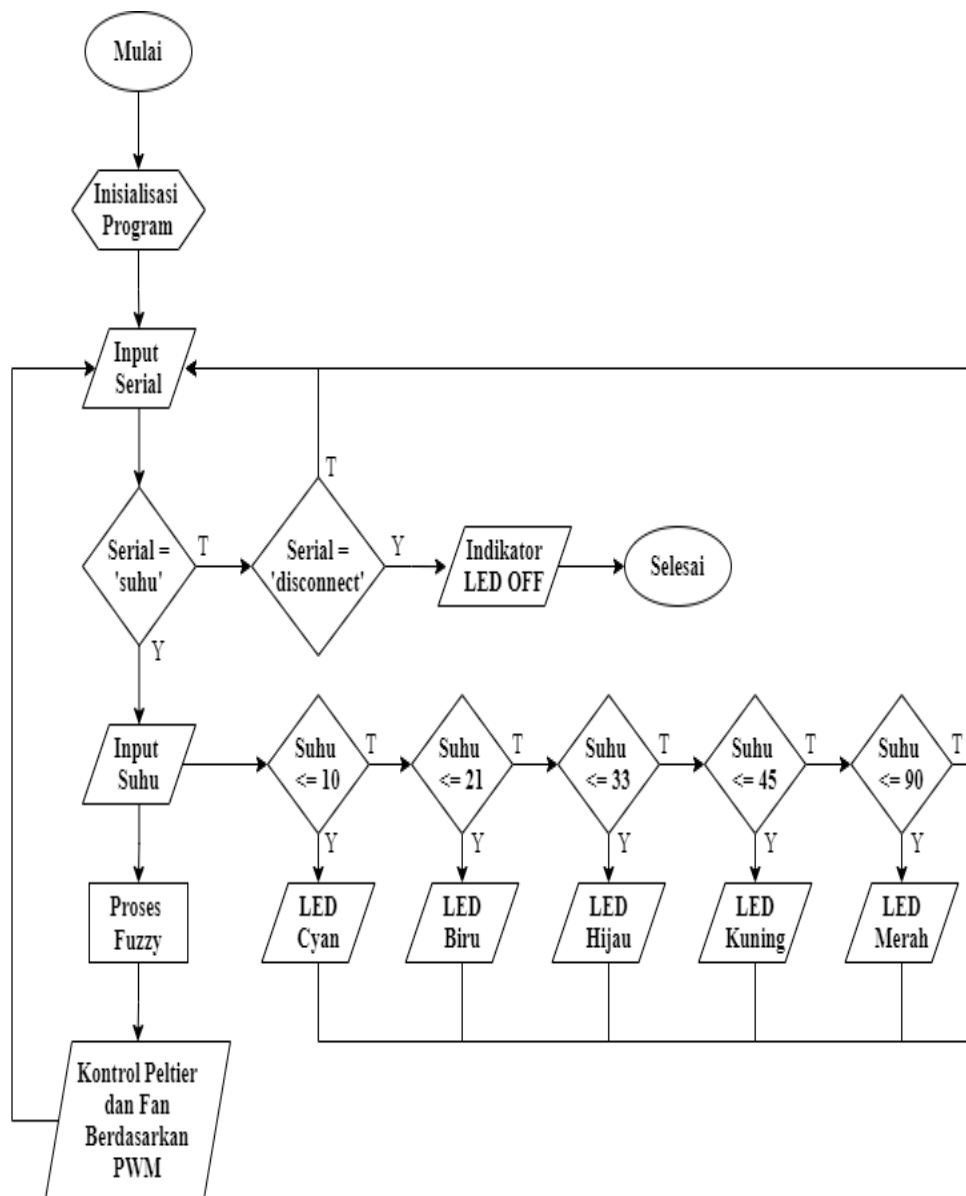
Dari diagram blok diatas dapat dijelaskan bahwa komputer berfungsi sebagai objek yang akan dilakukan pendinginan terhadap prosesoranya dan komputer juga berfungsi sebagai pengirim data suhu dari prosesoranya ke mikrokontroler, kemudian mikrokontroler ini berfungsi sebagai pusat pengolahan data suhu yang diterima dari komputer dan didalam mikrokontroler tersebut juga ditanamkan sebuah metode logika *fuzzy* yang berfungsi sebagai pengontrol suhu dari komputer, hasil *output* dari *fuzzy* inilah yang nantinya akan berupa nilai PWM untuk mengatur kecepatan *fan* dan untuk melakukan pengontrolan terhadap peltier yang dikendalikan oleh Mosfet IRF520. Pada diagram diatas LED berfungsi sebagai indikator dari nilai suhu dan indikator dari komputer dan mikrokontroler terhubung atau tidak. Dari diagram diatas peltier berfungsi sebagai elemen pemberi suhu dingin terhadap cairan yang mengalir ke *waterblock*. *Waterblock* tersebut akan menempel disisi dingin dari peltier. Cairan yang mengalir dari *waterblock* nantinya akan dialirkan ke *waterblock* CPU yang kemudian cairan tersebut dimasukkan kembali ke wadah penampung cairan lalu dialirkan kembali ke *waterblock* yang menempel di isi dingin dari peltier dan nantinya *waterblock* CPU akan diletakkan di atas prosesor komputer. Selanjutnya pompa air disini berfungsi sebagai penyerap dan pendorong cairan yang terdapat di wadah penampung cairan yang kemudian di alirkan ke *waterblock*, sehingga cairan dapat bersikulasi.

2. Flowchart sistem

Flowchart bertujuan untuk melihat proses dari cara kerja sistem yang dibuat. Pada tahapan ini akan dibuat tiga buah *flowchart* sistem, yaitu *flowchart* sistem pada mikrokontroler, *flowchart* sistem pada logika *fuzzy* dan *flowchart* sistem pada *software interface*.

a. Flowchart sistem pada mikrokontroler

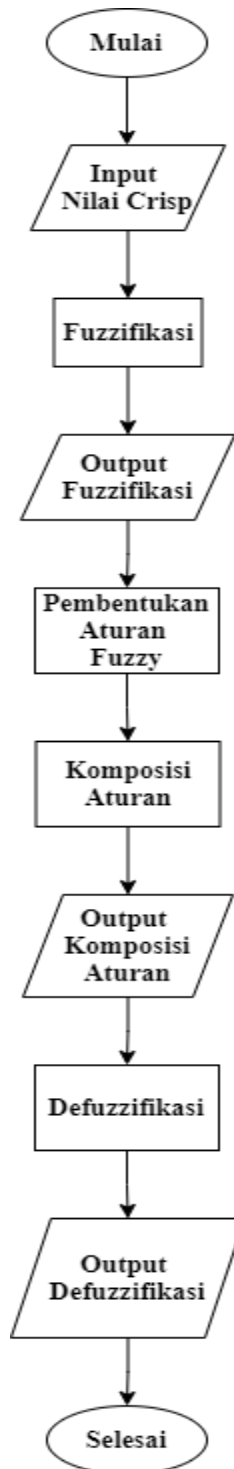
Berikut ini merupakan gambar dari *flowchart* sistem pada mikrokontroler yang ditunjukkan pada gambar 3.2.



Gambar 3. 2 *Flowchart* sistem pada mikrokontroler

b. *Flowchart* sistem pada logika *fuzzy*

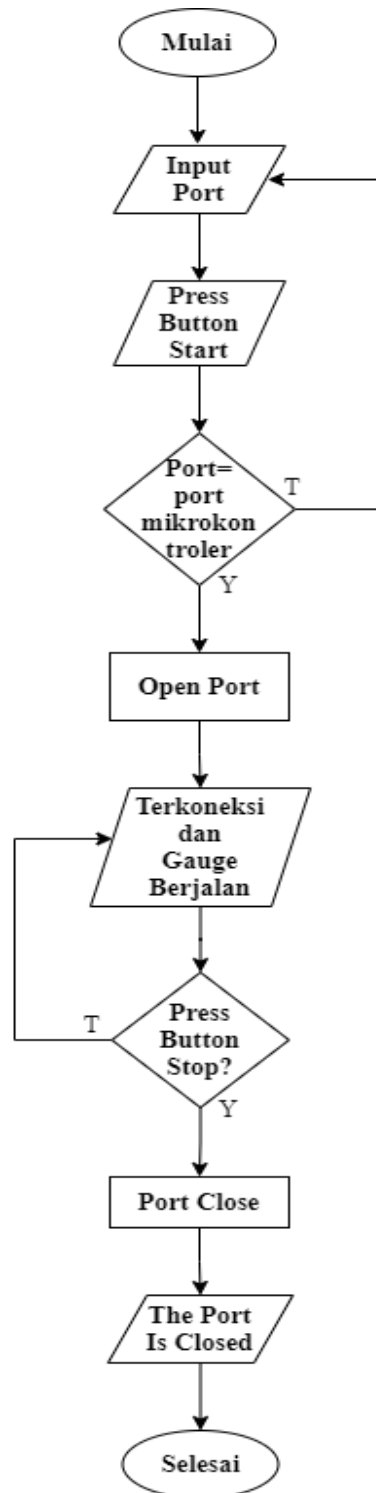
Berikut ini merupakan gambar dari *flowchart* sistem pada logika *fuzzy* yang ditunjukkan pada gambar 3.3.



Gambar 3. 3 *Flowchart* sistem pada logika *fuzzy*

c. *Flowchart* sistem pada *software interface*

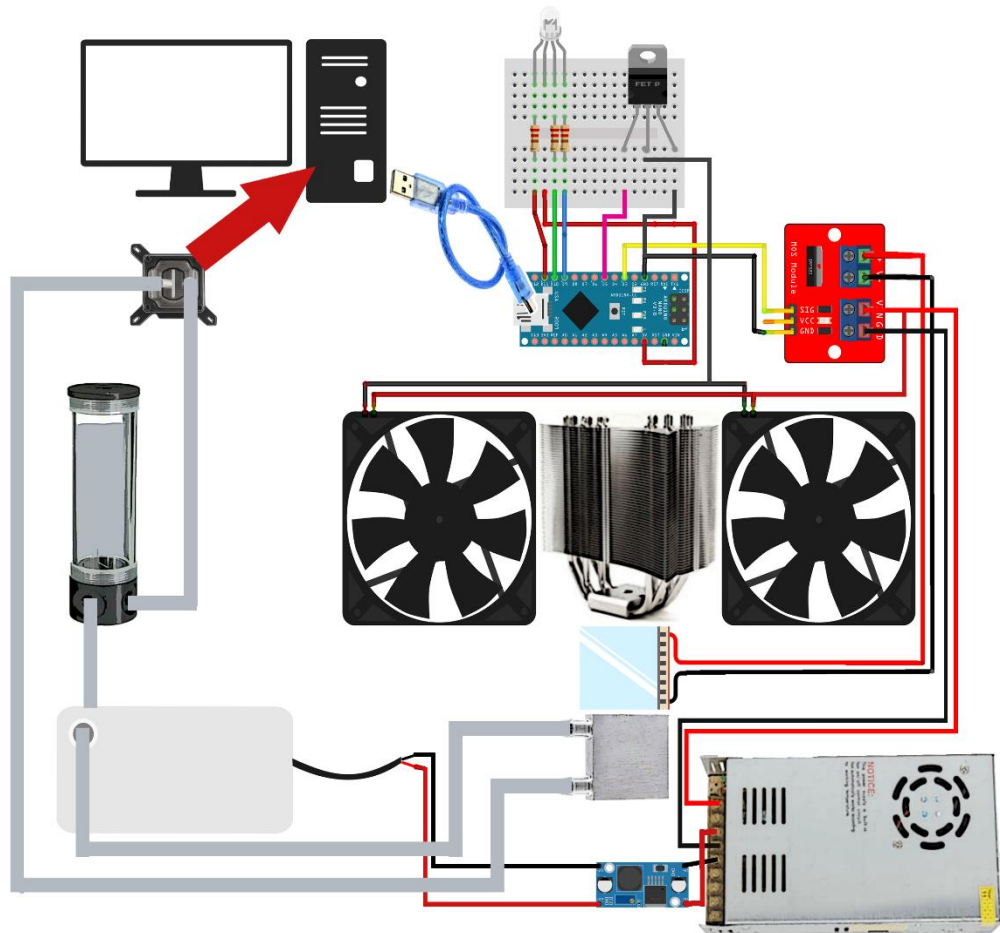
Berikut ini merupakan gambar dari *flowchart* sistem pada *software interface* yang ditunjukkan pada gambar 3.4.



Gambar 3. 4 *Flowchart* sistem pada *software interface*

3. Perancangan perangkat keras

Adapun perancangan perangkat keras pada penelitian ini ditunjukkan pada gambar 3.5.



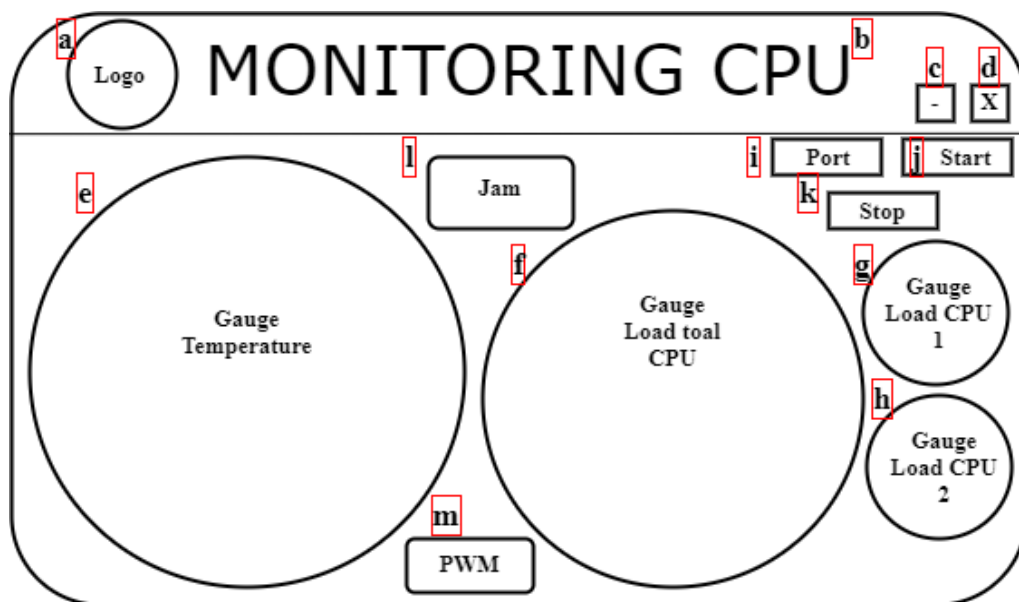
Gambar 3. 5 Rancangan perangkat keras

Berikut ini merupakan penjelasan dari gambar rancangan perangkat keras diatas:

- Komputer dihubungkan dengan mikrokontroler yang bertujuan sebagai pengirim data suhu dari komputer ke mikrokontroler.
- LED dihubungkan dengan mikrokontroler yang bertujuan sebagai indikator komputer dan mikrokontroler terhubung atau tidak serta sebagai indikator suhu.
- Mosfet IRF520 dihubungkan dengan mikrokontroler yang bertujuan sebagai output dari mikrokontroler yang berupa nilai PWM untuk mengatur kecepatan *fan* dan pengontrol peltier.

- d. *Fan* berfungsi untuk mendinginkan suhu sisi panas dari peltier.
 - e. Peltier berfungsi sebagai pendingin cairan yang mengalir dari *waterblock*.
 - f. *Waterblock* berfungsi sebagai penyerap panas atau dingin.
 - g. Pompa air DC berfungsi sebagai penyerap dan pendorong cairan dari wadah cairan.
 - h. *Power supply* berfungsi sebagai penyuplai tegangan ke komponen Mosfet IRF520
 - i. Wadah cairan berfungsi sebagai penampung dari cairan.
4. Perancangan perangkat lunak

Adapun perancangan perangkat lunak untuk *software interface* pada penelitian ini ditunjukkan pada gambar 3.6.



Gambar 3. 6 *Layout software interface*

Berikut penjelasan dari layout dari *software interface* diatas:

- a. *Picture box* berfungsi untuk menampilkan logo dari *software interface*
- b. *Label text* berfungsi untuk menampilkan judul.
- c. *Button x* berfungsi untuk menutup *software interfece*.
- d. *Button –* berfungsi untuk meperkecil window *software interface*.
- e. *Gauge* temperature berfungsi untuk menampilkan nilai dari suhu komputer.
- f. *Gauge load* total CPU berfungsi untuk menampilkan total beban kerja dari CPU.

- g. *Gauge load* CPU 1 berfungsi untuk menampilkan beban kerja dari core CPU 1.
- h. *Gauge load* CPU 2 berfungsi untuk menampilkan beban kerja dari core CPU 2.
- i. *Combobox port* berfungsi untuk menampilkan port dari mikrokontroler.
- j. *Button start* berfungsi untuk menghubungkan komputer dengan mikrokontroler.
- k. *Button stop* berfungsi untuk memutuskan hubungan komputer dengan mikrokontroler.
- l. *Label text* berfungsi untuk menampilkan jam.
- m. *Label text* berfungsi untuk menampilkan hasil logika *fuzzy* yang berupa PWM.

3.3.5 Pengujian

Pada proses pengujian alat *cooling* komputer otomatis ini dilakukan dengan penginstalasian alat ke sebuah komputer, yang selanjutnya melakukan uji coba komunikasi serial antara komputer dengan mikrokontroler melalui *software interface*, setelah terhubung antara komputer dan mikrokontroler selanjutnya melakukan uji coba pembacaan suhu oleh *software interface*, setelah itu melakukan pengujian pembacaan suhu yang diterima oleh mikrokontroler dari komputer dengan ditandai lampu LED hidup. Setelah pembacaan suhu berhasil selanjutnya melakukan proses *fuzzy* yang dilakukan dengan empat tahapan yaitu, fuzzifikasi, pembentukan aturan *fuzzy*, komposisi aturan dan defuzzifikasi, hasil dari defuzzifikasi ini nantinya akan berupa PWM untuk mengatur kecepatan fan dan pengontrol dari peltier untuk menstabilkan suhu pada komputer dan hasil dari defuzzifikasi ini dapat dilihat pada *software interface*. Kemudian pengujian juga dilakukan dengan membandingkan alat yang dibuat pada penelitian ini dengan pendingin komputer pada umumnya seperti air cooling dan water cooling dengan radiator.

3.3.6 Penerapan

Penerapan alat cooling komputer otomatis ini menggunakan mikrokontroler Atmega 328 dan *software interface* untuk mengontrol suhu pada komputer. Nantinya alat ini akan diterapkan pada komputer yang mengalami permasalahan lambat atau lag yang disebabkan oleh *overheat*. Alat ini juga menggunakan komunikasi serial sebagai penghubung antara mikrokontroler dengan komputer dan pada alat ini juga pengguna dapat melihat nilai suhu dari *software interface*.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pembahasan

Adapun beberapa tahapan yang akan dibahas pada penelitian ini yaitu terkait dengan analisis data, representasi data, hasil analisis data dan perancangan sebagai berikut.

4.1.1 Analisis Data

Adapun bagian ini yang dibutuhkan dalam melakukan pembuatan pada penelitian penerapan logika *fuzzy* pada perancangan *cooling* komputer otomatis menggunakan Peltier berbasis Mikrokontroler Atmega 328 ini berupa analisis sistem *fuzzy*, analisis perangkat lunak dan analisis perangkat keras.

1. Analisis Sistem *Fuzzy*

Sistem *fuzzy* pada penelitian ini yaitu suhu dari prosesor komputer berdasarkan masing-masing core dari prosesor tersebut. Sistem ini nantinya akan berfungsi sebagai pengontrol suhu dari komputer, yang suhu dari komputer dikirim ke mikrokontroler melalui komunikasi serial.

2. Analisis Perangkat Lunak

Perangkat lunak yang digunakan untuk mendukung pembuatan alat pada penelitian ini terdiri dari Arduino IDE yang digunakan untuk memprogram mikrokontroler Atmega 328 dan Microsoft Visual Studio 2015 digunakan untuk memprogram *software interface* sebagai media penghubung antara komputer dengan alat pada penelitian ini.

3. Analisis Perangkat Keras

Perangkat keras yang digunakan untuk mendukung pembuatan alat pada penelitian ini terdiri dari komputer sebagai objek yang prosesornya akan di dinginkan, mikrokontroler Atmega 328 digunakan sebagai pusat pengolahan data, peltier digunakan sebagai media pendingin cairan yang mengalir melalui waterblock yang menempel di sisi dingin dari peltier dan waterblock CPU yang

menempel disisi permukaan atas dari prosesor, fan dan heatsink digunakan sebagai media pendingin dari peltier yang di tempelkan disisi panas dari peltier tersebut, mosfet IRF520 digunakan sebagai pengotrol peltier dan fan, LED RGB digunakan sebagai indikator dari besaran suhu serta indikator apabila komputer terhubung atau tidak dengan mikrokontroler, pompa air digunakan sebagai pendorong dan penyerap cairan agar cairan dapat bersikulasi dan power supply digunakan sebagai penyuplai daya ke komponen-komponen pada alat ini.

4.1.2 Representasi Data

Pada bagian ini terdapat beberapa tahapan yang digunakan untuk merepresentasikan data diantaranya fuzzifikasi, pembentukan aturan *fuzzy*, komposisi aturan dan defuzzifikasi.

1. Fuzzifikasi

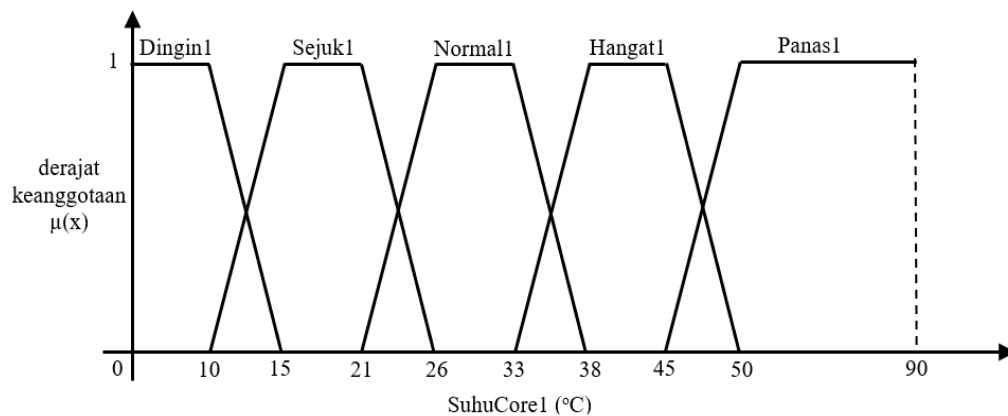
Adapun pada tahap fuzzifikasi ini, nilai masukkan tegas diambil dari suhu prosesor komputer berdasarkan masing-masing *Core*. Pada uji coba ini digunakan prosesor *Core i3 6100* yang memiliki 2 *Core*. Masing-masing *Core* tersebut nantinya akan mengirim nilai suhunya ke mikrokontroler lewat komunikasi serial. Setelah di terima mikrokontroler nilai masing-masing *Core* tersebut selanjutnya diproses dalam tahapan fuzzifikasi untuk mendapatkan nilai dari derajat keanggotaan. Untuk mendapatkan nilai dari derajat keanggotaan, maka terlebih dahulu menentukan himpunan *fuzzy* yang dapat dilihat pada tabel 4.1 berikut.

Tabel 4. 1 Himpunan *Fuzzy*

Variabel	Himpunan <i>Fuzzy</i>	Semesta Pembicaraan	Domain
SuhuCore1	Dingin1	[0 90]	[0 10]
	Sejuk1		[11 21]
	Normal1		[22 33]
	Hangat1		[34 45]

	Panas1		[46 90]
SuhuCore2	Dingin2	[0 90]	[0 10]
	Sejuk2		[11 21]
	Normal2		[22 33]
	Hangat2		[34 45]
	Panas2		[46 90]

Selanjutnya untuk mendapatkan nilai derajat keanggotaan di butuhkan juga fungsi keanggotaan. Pada penelitian ini digunakan fungsi keanggotaan diantaranya yaitu kurva bahu dan kurva trapesium. Berikut terlihat pada gambar 4.1 merupakan gambar fungsi keanggotaan yang menggunakan kurva bahu dan kurva trapesium untuk variabel SuhuCore1 dan terlihat juga pada gambar 4.2 merupakan gambar fungsi keanggotaan yang menggunakan kurva bahu dan kurva trapesium untuk variabel SuhuCore2.



Gambar 4. 1 Fungsi Keanggotaan SuhuCore1

Dari kurva diatas dapat dibuat formula derajat keanggotaan sebagai berikut:

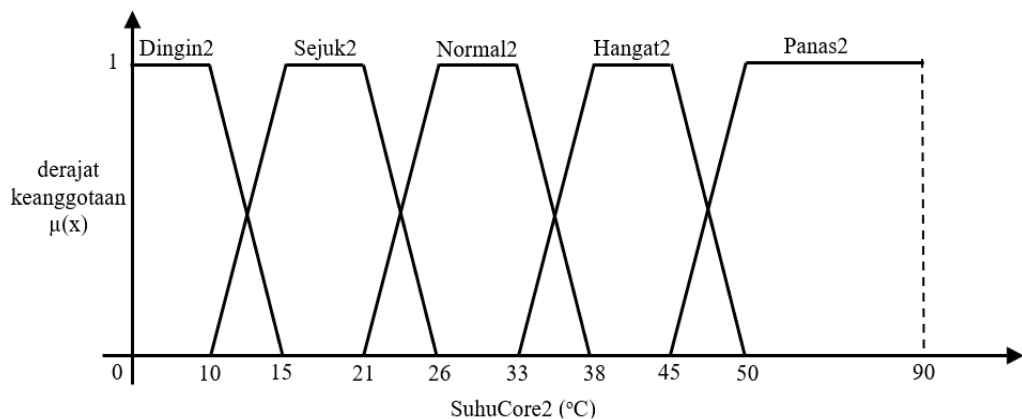
$$\mu_{Dingin1} = \begin{cases} 0, & x \geq 15 \\ \frac{(15 - x)}{(15 - 10)}, & 10 \leq x \leq 15 \\ 1, & x \leq 10 \end{cases}$$

$$\mu_{Sejuk1} = \begin{cases} 0, & x \leq 10 \text{ atau } x \geq 26 \\ \frac{(x-10)}{(15-10)}, & 10 \leq x \leq 15 \\ \frac{(26-x)}{(26-21)}, & 21 \leq x \leq 26 \\ 1, & 15 \leq x \leq 21 \end{cases}$$

$$\mu_{Normal1} = \begin{cases} 0, & x \leq 21 \text{ atau } x \geq 38 \\ \frac{(x-21)}{(26-21)}, & 21 \leq x \leq 26 \\ \frac{(38-x)}{(38-33)}, & 33 \leq x \leq 38 \\ 1, & 26 \leq x \leq 33 \end{cases}$$

$$\mu_{Hangat1} = \begin{cases} 0, & x \leq 33 \text{ atau } x \geq 50 \\ \frac{(x-33)}{(38-33)}, & 33 \leq x \leq 38 \\ \frac{(50-x)}{(50-45)}, & 45 \leq x \leq 50 \\ 1, & 38 \leq x \leq 45 \end{cases}$$

$$\mu_{Panas1} = \begin{cases} 0, & x \leq 45 \\ \frac{(x-21)}{(26-21)}, & 21 \leq x \leq 26 \\ 1, & x \geq 50 \end{cases}$$



Gambar 4. 2 Fungsi Keanggotaan SuhuCore2

Dari kurva diatas dapat dibuat formula derajat keanggotaan sebagai berikut:

$$\mu_{Dingin2} = \begin{cases} 0, & x \geq 15 \\ \frac{(15-x)}{(15-10)}, & 10 \leq x \leq 15 \\ 1, & x \leq 10 \end{cases}$$

$$\mu_{Sejuk2} = \begin{cases} 0, & x \leq 10 \text{ atau } x \geq 26 \\ \frac{(x-10)}{(15-10)}, & 10 \leq x \leq 15 \\ \frac{(26-x)}{(26-21)}, & 21 \leq x \leq 26 \\ 1, & 15 \leq x \leq 21 \end{cases}$$

$$\mu_{Normal2} = \begin{cases} 0, & x \leq 21 \text{ atau } x \geq 38 \\ \frac{(x-21)}{(26-21)}, & 21 \leq x \leq 26 \\ \frac{(38-x)}{(38-33)}, & 33 \leq x \leq 38 \\ 1, & 26 \leq x \leq 33 \end{cases}$$

$$\mu_{Hangat2} = \begin{cases} 0, & x \leq 33 \text{ atau } x \geq 50 \\ \frac{(x-33)}{(38-33)}, & 33 \leq x \leq 38 \\ \frac{(50-x)}{(50-45)}, & 45 \leq x \leq 50 \\ 1, & 38 \leq x \leq 45 \end{cases}$$

$$\mu_{Panas2} = \begin{cases} 0, & x \leq 45 \\ \frac{(x-21)}{(26-21)}, & 21 \leq x \leq 26 \\ 1, & x \geq 50 \end{cases}$$

Diberikan satu contoh kasus untuk mencari nilai derajat keanggotaan, yang dimana diberikan nilai tegas SuhuCore1 sebesar 25°C dan SuhuCore2 sebesar 23°C. Kemudian nilai tersebut dimasukkan ke dalam fungsi keanggotaan pada tiap-tiap himpunan *fuzzy*, sebagai berikut:

a. Fungsi keanggotaan SuhuCore1

- Himpunan *fuzzy* Dingin1, $\mu_{Dingin1}[25] = 0$
- Himpunan *fuzzy* Sejuk1, $\mu_{Sejuk1}[25] = \frac{(26-25)}{(26-21)} = \frac{1}{5} = 0,2$
- Himpunan *fuzzy* Normal1, $\mu_{Normal1}[25] = \frac{(25-21)}{(26-21)} = \frac{4}{5} = 0,8$
- Himpunan *fuzzy* Hangat1, $\mu_{Hangat1}[25] = 0$
- Himpunan *fuzzy* Panas1, $\mu_{Panas1}[25] = 0$

b. Fungsi keanggotaan SuhuCore2

- Himpunan *fuzzy* Dingin2, $\mu_{Dingin2}[23] = 0$

- Himpunan *fuzzy* Sejuk2, $\mu_{\text{Sejuk2}}[23] = \frac{(26-23)}{(26-21)} = \frac{3}{5} = 0,6$
- Himpunan *fuzzy* Normal2, $\mu_{\text{Normal2}}[23] = \frac{(23-21)}{(26-21)} = \frac{2}{5} = 0,4$
- Himpunan *fuzzy* Hangat2, $\mu_{\text{Hangat2}}[23] = 0$
- Himpunan *fuzzy* Panas2, $\mu_{\text{Panas2}}[23] = 0$

2. Pembentukan Aturan *Fuzzy*

Tahapan pembentukan aturan *fuzzy* ini menggunakan *fuzzy* Sugeno dengan model pembentukan aturan *fuzzy* orde nol. Adapun pada alat cooling komputer otomatis ini terdapat 25 aturan *fuzzy* sebagai berikut:

[Aturan 1] *If* (SuhuCore1 is Dingin1) *And* (SuhuCore2 is Dingin2) *Then* 150

[Aturan 2] *If* (SuhuCore1 is Dingin1) *And* (SuhuCore2 is Sejuk2) *Then* 150

[Aturan 3] *If* (SuhuCore1 is Dingin1) *And* (SuhuCore2 is Normal2) *Then* 200

[Aturan 4] *If* (SuhuCore1 is Dingin1) *And* (SuhuCore2 is Hangat2) *Then* 200

[Aturan 5] *If* (SuhuCore1 is Dingin1) *And* (SuhuCore2 is Panas2) *Then* 255

[Aturan 6] *If* (SuhuCore1 is Sejuk1) *And* (SuhuCore2 is Dingin2) *Then* 150

[Aturan 7] *If* (SuhuCore1 is Sejuk 1) *And* (SuhuCore2 is Sejuk2) *Then* 150

[Aturan 8] *If* (SuhuCore1 is Sejuk 1) *And* (SuhuCore2 is Normal2) *Then* 200

[Aturan 9] *If* (SuhuCore1 is Sejuk 1) *And* (SuhuCore2 is Hangat2) *Then* 200

[Aturan 10] *If* (SuhuCore1 is Sejuk 1) *And* (SuhuCore2 is Panas2) *Then* 255

[Aturan 11] *If* (SuhuCore1 is Normal1) *And* (SuhuCore2 is Dingin2) *Then* 200

[Aturan 12] *If* (SuhuCore1 is Normal 1) *And* (SuhuCore2 is Sejuk2) *Then* 200

[Aturan 13] *If* (SuhuCore1 is Normal 1) *And* (SuhuCore2 is Normal2) *Then* 255

[Aturan 14] *If* (SuhuCore1 is Normal 1) *And* (SuhuCore2 is Hangat2) *Then* 255

[Aturan 15] *If* (SuhuCore1 is Normal 1) *And* (SuhuCore2 is Panas2) *Then* 255

[Aturan 16] *If* (SuhuCore1 is Hangat1) *And* (SuhuCore2 is Dingin2) *Then* 200

[Aturan 17] *If* (SuhuCore1 is Hangat 1) *And* (SuhuCore2 is Sejuk2) *Then* 200

[Aturan 18] *If* (SuhuCore1 is Hangat 1) *And* (SuhuCore2 is Normal2) *Then* 255

[Aturan 19] *If* (SuhuCore1 is Hangat 1) *And* (SuhuCore2 is Hangat2) *Then* 255

[Aturan 20] *If* (SuhuCore1 is Hangat 1) *And* (SuhuCore2 is Panas2) *Then* 255

[Aturan 21] *If* (SuhuCore1 is Panas1) *And* (SuhuCore2 is Dingin2) *Then* 255

[Aturan 22] *If* (SuhuCore1 is Panas1) *And* (SuhuCore2 is Sejuk2) *Then* 255

[Aturan 23] *If (SuhuCore1 is Panas1) And (SuhuCore2 is Normal2) Then 255*

[Aturan 24] *If (SuhuCore1 is Panas1) And (SuhuCore2 is Hangat2) Then 255*

[Aturan 25] *If (SuhuCore1 is Panas1) And (SuhuCore2 is Panas2) Then 255*

3. Komposisi Aturan

Pada tahap komposisi aturan alat cooling komputer otomatis ini menggunakan fungsi implikasi *MIN* untuk mencari α -predikat dari setiap aturan. Kemudian masing-masing nilai α -predikat ini digunakan untuk menghitung keluaran hasil inferensi secara tegas (*crisp*) pada masing-masing rule ($z_1, z_2, z_3, \dots, z_n$). Pada contoh diatas maka di dapat komposisi aturan sebagai berikut:

[Aturan 1] *If (SuhuCore1 is Dingin1) And (SuhuCore2 is Dingin2) Then 150*

$$\begin{aligned}\alpha - \text{predikat} &= \mu_{Dingin1} \cap \mu_{Dingin2} \\ &= \min (\mu_{Dingin1}[25], \mu_{Dingin2}[23]) \\ &= \min (0; 0) \\ &= 0\end{aligned}$$

[Aturan 2] *If (SuhuCore1 is Dingin1) And (SuhuCore2 is Sejuk2) Then 150*

$$\begin{aligned}\alpha - \text{predikat} &= \mu_{Dingin1} \cap \mu_{Sejuk2} \\ &= \min (\mu_{Dingin1}[25], \mu_{Sejuk2}[23]) \\ &= \min (0; 0,6) \\ &= 0\end{aligned}$$

[Aturan 3] *If (SuhuCore1 is Dingin1) And (SuhuCore2 is Normal2) Then 200*

$$\begin{aligned}\alpha - \text{predikat} &= \mu_{Dingin1} \cap \mu_{Normal2} \\ &= \min (\mu_{Dingin1}[25], \mu_{Normal2}[23]) \\ &= \min (0; 0,4) \\ &= 0\end{aligned}$$

[Aturan 4] *If (SuhuCore1 is Dingin1) And (SuhuCore2 is Hangat2) Then 200*

$$\begin{aligned}\alpha - \text{predikat} &= \mu_{Dingin1} \cap \mu_{Hangat2} \\ &= \min (\mu_{Dingin1}[25], \mu_{Hangat2}[23]) \\ &= \min (0; 0) \\ &= 0\end{aligned}$$

[Aturan 5] *If (SuhuCore1 is Dingin1) And (SuhuCore2 is Panas2) Then 255*

$$\alpha - \text{predikat} = \mu_{Dingin1} \cap \mu_{Panas2}$$

$$\begin{aligned}
&= \min (\mu_{Dingin1}[25], \mu_{Panas2}[23]) \\
&= \min (0; 0) \\
&= 0
\end{aligned}$$

[Aturan 6] *If (SuhuCore1 is Sejuk1) And (SuhuCore2 is Dingin2) Then 150*

$$\begin{aligned}
\alpha - \text{predikat} &= \mu_{Sejuk1} \cap \mu_{Dingin2} \\
&= \min (\mu_{Sejuk1}[25], \mu_{Dingin2}[23]) \\
&= \min (0,2; 0) \\
&= 0
\end{aligned}$$

[Aturan 7] *If (SuhuCore1 is Sejuk 1) And (SuhuCore2 is Sejuk2) Then 150*

$$\begin{aligned}
\alpha - \text{predikat} &= \mu_{Sejuk1} \cap \mu_{Sejuk2} \\
&= \min (\mu_{Sejuk1}[25], \mu_{Sejuk2}[23]) \\
&= \min (0,2; 0,6) \\
&= 0,2
\end{aligned}$$

[Aturan 8] *If (SuhuCore1 is Sejuk 1) And (SuhuCore2 is Normal2) Then 200*

$$\begin{aligned}
\alpha - \text{predikat} &= \mu_{Sejuk1} \cap \mu_{Normal2} \\
&= \min (\mu_{Sejuk1}[25], \mu_{Normal2}[23]) \\
&= \min (0,2; 0,4) \\
&= 0,2
\end{aligned}$$

[Aturan 9] *If (SuhuCore1 is Sejuk 1) And (SuhuCore2 is Hangat2) Then 200*

$$\begin{aligned}
\alpha - \text{predikat} &= \mu_{Sejuk1} \cap \mu_{Hangat2} \\
&= \min (\mu_{Sejuk1}[25], \mu_{Hangat2}[23]) \\
&= \min (0,2; 0) \\
&= 0
\end{aligned}$$

[Aturan 10] *If (SuhuCore1 is Sejuk 1) And (SuhuCore2 is Panas2) Then 255*

$$\begin{aligned}
\alpha - \text{predikat} &= \mu_{Sejuk1} \cap \mu_{Panas2} \\
&= \min (\mu_{Sejuk1}[25], \mu_{Panas2}[23]) \\
&= \min (0,2; 0) \\
&= 0
\end{aligned}$$

[Aturan 11] *If (SuhuCore1 is Normal1) And (SuhuCore2 is Dingin2) Then 200*

$$\alpha - \text{predikat} = \mu_{Normal1} \cap \mu_{Dingin2}$$

$$\begin{aligned}
&= \min (\mu_{Normal1}[25], \mu_{Dingin2}[23]) \\
&= \min (0,8; 0) \\
&= 0
\end{aligned}$$

[Aturan 12] *If (SuhuCore1 is Normal 1) And (SuhuCore2 is Sejuk2) Then 200*

$$\begin{aligned}
\alpha - \text{predikat} &= \mu_{Normal1} \cap \mu_{Sejuk2} \\
&= \min (\mu_{Normal1}[25], \mu_{Sejuk2}[23]) \\
&= \min (0,8; 0,6) \\
&= 0,6
\end{aligned}$$

[Aturan 13] *If (SuhuCore1 is Normal 1) And (SuhuCore2 is Normal2) Then 255*

$$\begin{aligned}
\alpha - \text{predikat} &= \mu_{Normal1} \cap \mu_{Normal2} \\
&= \min (\mu_{Normal1}[25], \mu_{Normal2}[23]) \\
&= \min (0,8; 0,4) \\
&= 0,4
\end{aligned}$$

[Aturan 14] *If (SuhuCore1 is Normal 1) And (SuhuCore2 is Hangat2) Then 255*

$$\begin{aligned}
\alpha - \text{predikat} &= \mu_{Normal1} \cap \mu_{Hangat2} \\
&= \min (\mu_{Normal1}[25], \mu_{Hangat2}[23]) \\
&= \min (0,8; 0) \\
&= 0
\end{aligned}$$

[Aturan 15] *If (SuhuCore1 is Normal 1) And (SuhuCore2 is Panas2) Then 255*

$$\begin{aligned}
\alpha - \text{predikat} &= \mu_{Normal1} \cap \mu_{Panas2} \\
&= \min (\mu_{Normal1}[25], \mu_{Panas2}[23]) \\
&= \min (0,8; 0) \\
&= 0
\end{aligned}$$

[Aturan 16] *If (SuhuCore1 is Hangat1) And (SuhuCore2 is Dingin2) Then 200*

$$\begin{aligned}
\alpha - \text{predikat} &= \mu_{Hangat1} \cap \mu_{Dingin2} \\
&= \min (\mu_{Hangat1}[25], \mu_{Dingin2}[23]) \\
&= \min (0; 0) \\
&= 0
\end{aligned}$$

[Aturan 17] *If (SuhuCore1 is Hangat 1) And (SuhuCore2 is Sejuk2) Then 200*

$$\alpha - \text{predikat} = \mu_{Hangat1} \cap \mu_{Sejuk2}$$

$$\begin{aligned}
&= \min (\mu_{Hangat1}[25], \mu_{Sejuk2}[23]) \\
&= \min (0; 0,6) \\
&= 0
\end{aligned}$$

[Aturan 18] *If (SuhuCore1 is Hangat 1) And (SuhuCore2 is Normal2) Then 255*

$$\begin{aligned}
\alpha - \text{predikat} &= \mu_{Hangat1} \cap \mu_{Normal2} \\
&= \min (\mu_{Hangat1}[25], \mu_{Normal2}[23]) \\
&= \min (0; 0,4) \\
&= 0
\end{aligned}$$

[Aturan 19] *If (SuhuCore1 is Hangat 1) And (SuhuCore2 is Hangat2) Then 255*

$$\begin{aligned}
\alpha - \text{predikat} &= \mu_{Hangat1} \cap \mu_{Hangat2} \\
&= \min (\mu_{Hangat1}[25], \mu_{Hangat2}[23]) \\
&= \min (0; 0) \\
&= 0
\end{aligned}$$

[Aturan 20] *If (SuhuCore1 is Hangat 1) And (SuhuCore2 is Panas2) Then 255*

$$\begin{aligned}
\alpha - \text{predikat} &= \mu_{Hangat1} \cap \mu_{Panas2} \\
&= \min (\mu_{Hangat1}[25], \mu_{Panas2}[23]) \\
&= \min (0; 0) \\
&= 0
\end{aligned}$$

[Aturan 21] *If (SuhuCore1 is Panas1) And (SuhuCore2 is Dingin2) Then 255*

$$\begin{aligned}
\alpha - \text{predikat} &= \mu_{Panas1} \cap \mu_{Dingin2} \\
&= \min (\mu_{Panas1}[25], \mu_{Dingin2}[23]) \\
&= \min (0; 0) \\
&= 0
\end{aligned}$$

[Aturan 22] *If (SuhuCore1 is Panas1) And (SuhuCore2 is Sejuk2) Then 255*

$$\begin{aligned}
\alpha - \text{predikat} &= \mu_{Panas1} \cap \mu_{Sejuk2} \\
&= \min (\mu_{Panas1}[25], \mu_{Sejuk2}[23]) \\
&= \min (0; 0,6) \\
&= 0
\end{aligned}$$

[Aturan 23] *If (SuhuCore1 is Panas1) And (SuhuCore2 is Normal2) Then 255*

$$\alpha - \text{predikat} = \mu_{Panas1} \cap \mu_{Normal2}$$

$$\begin{aligned}
 &= \min (\mu_{Panas1}[25], \mu_{Normal2}[23]) \\
 &= \min (0; 0,4) \\
 &= 0
 \end{aligned}$$

[Aturan 24] *If (SuhuCore1 is Panas1) And (SuhuCore2 is Hangat2) Then 255*

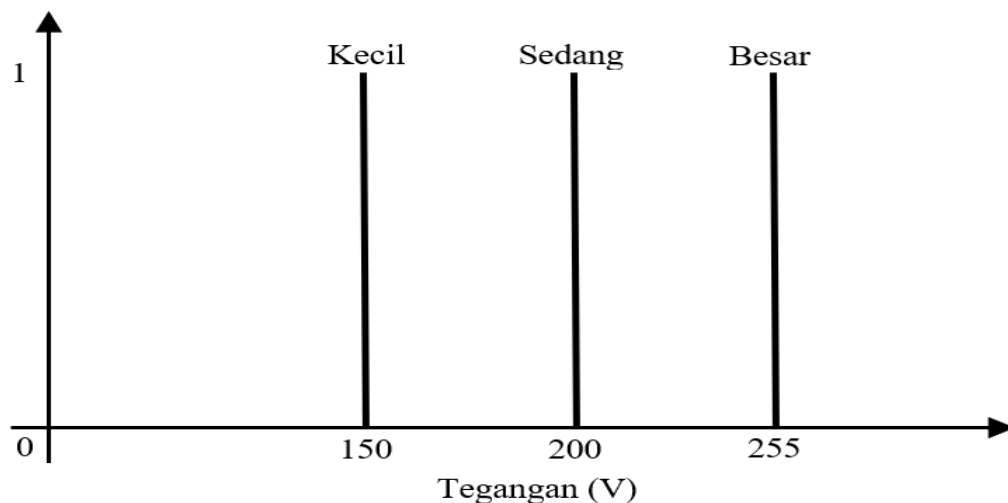
$$\begin{aligned}
 \alpha - predikat &= \mu_{Panas1} \cap \mu_{Hangat2} \\
 &= \min (\mu_{Panas1}[25], \mu_{Hangat2}[23]) \\
 &= \min (0; 0) \\
 &= 0
 \end{aligned}$$

[Aturan 25] *If (SuhuCore1 is Panas1) And (SuhuCore2 is Panas2) Then 255*

$$\begin{aligned}
 \alpha - predikat &= \mu_{Panas1} \cap \mu_{Panas2} \\
 &= \min (\mu_{Panas1}[25], \mu_{Panas2}[23]) \\
 &= \min (0; 0) \\
 &= 0
 \end{aligned}$$

4. Defuzzifikasi

Terlihat pada gambar 4.3 merupakan fungsi keanggotaan dari *output fuzzy* dalam bentuk singleton.



Gambar 4. 3 Fungsi Keanggotaan Output *Fuzzy*

Pada tahap ini proses defuzzifikasi *fuzzy* Sugeno digunakan metode rata-rata (*average*). Dari contoh diatas maka di dapat hasil defuzzifikasi sebagai berikut:

$$\begin{aligned}
& apred_1 * z_1 + apred_2 * z_2 + apred_3 * z_3 + apred_4 * z_4 \\
& + apred_5 * z_5 + apred_6 * z_6 + apred_7 * z_7 + apred_8 * z_8 \\
& + apred_9 * z_9 + apred_{10} * z_{10} + apred_{11} * z_{11} + apred_{12} * z_{12} \\
& + apred_{13} * z_{13} + apred_{14} * z_{14} + apred_{15} * z_{15} + apred_{16} * z_{16} \\
& + apred_{17} * z_{17} + apred_{18} * z_{18} + apred_{19} * z_{19} + apred_{20} * z_{20} \\
& + apred_{21} * z_{21} + apred_{22} * z_{22} + apred_{23} * z_{23} + apred_{24} * z_{24} \\
& + apred_{25} * z_{25} \\
z = & \frac{apred_1 + apred_2 + apred_3 + apred_4 + apred_5 + apred_6}{apred_7 + apred_8 + apred_9 + apred_{10} + apred_{11} + apred_{12} + apred_{13} + apred_{14} + apred_{15} + apred_{16} + apred_{17} + apred_{18} + apred_{19} + apred_{20} + apred_{21} + apred_{22} + apred_{23} + apred_{24} + apred_{25}} \\
& \frac{0 * 150 + 0 * 150 + 0 * 200 + 0 * 200 + 0 * 255 + 0 * 150 + 0,2 * 150 + 0,2 * 200 + 0 * 200 + 0 * 255 + 0 * 200 + 0,6 * 200 + 0,4 * 255 + 0 * 255 + 0 * 255 + 0 * 200 + 0 * 200 + 0 * 255 + 0 * 255 + 0 * 255 + 0 * 255 + 0 * 255 + 0 * 255}{0 + 0 + 0 + 0 + 0 + 0 + 0,2 + 0,2 + 0 + 0 + 0 + 0,6 + 0,4 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0} \\
z = & \frac{292}{1,4} \\
z = & 208,57
\end{aligned}$$

Berdasarkan contoh diatas apabila SuhuCore1 diberikan nilai sebesar 25°C dan SuhuCore2 sebesar 23°C, maka diperoleh hasil defuzzifikasi sebesar 208,57. Hasil dari defuzzifikasi inilah nantinya akan berbentuk nilai PWM yang digunakan untuk mengontrol peltier dan *fan* guna menjaga kestabilan suhu pada prosesor komputer.

4.1.3 Hasil Analisis Data

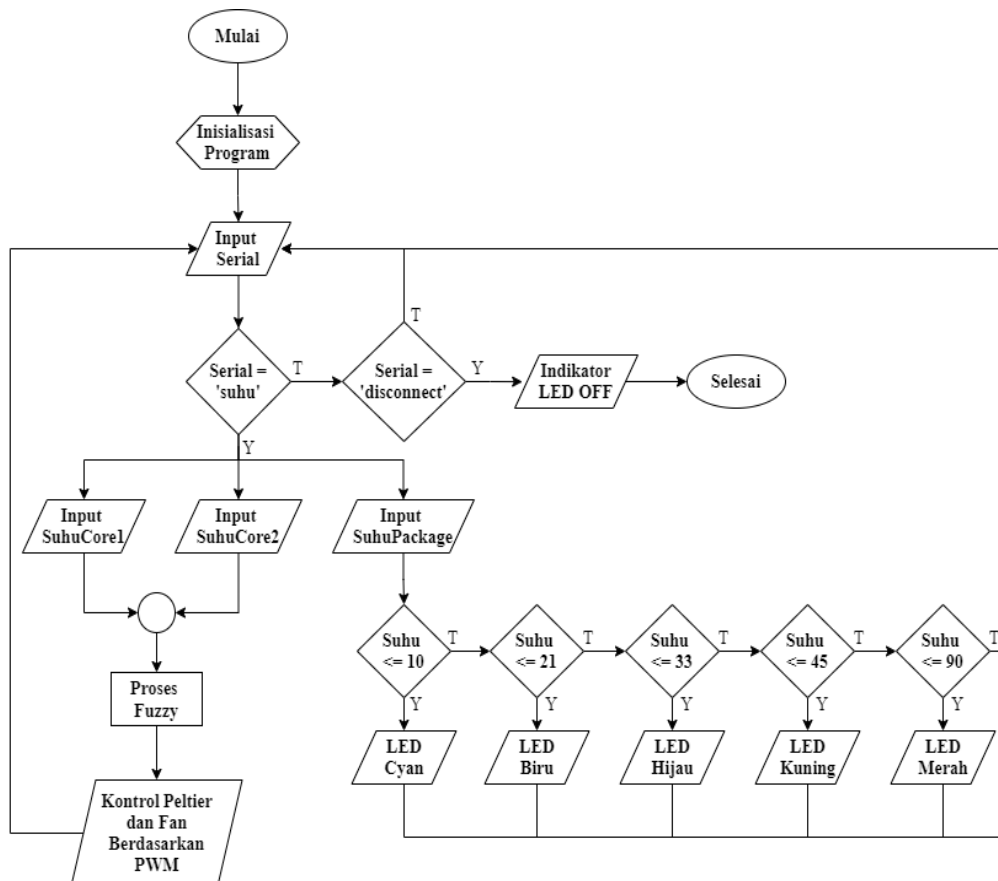
Setelah menganalisis data diatas, maka dapat diketahui hasil dari analisis data yang meliputi baik tidaknya setiap komponen dan faktor-faktor apa saja yang dapat mempengaruhinya. Selain itu, berdasarkan hasil dari representasi data diatas dengan menggunakan metode logika *fuzzy* sugeno dapat diketahui bahwa hasil dari tiap-tiap tahapan seperti fuzzifikasi, pembentukan aturan *fuzzy*, komposisi aturan serta defuzzifikasi digunakan untuk mengontrol suhu dari prosesor pada komputer sehingga komputer dapat menjaga performanya dalam keadaan stabil.

4.1.4 Perancangan

Pada bagian perancangan ini terdapat beberapa tahapan yaitu, perancangan *flowchart*, perancangan alat, perancangan *software interface* dan perancangan program logika *fuzzy*.

1. Perancangan *Flowchart*

Pada tahapan perancangan *flowchart* ini dilakukan penjelasan alur dari sistem kerja alat dalam bentuk diagram alir yang bertujuan untuk menggambarkan hubungan dan urutan proses dalam suatu sistem. Berikut ini merupakan *flowchart* dari sistem alat cooling komputer otomatis yang terlihat pada gambar 4.4 dibawah ini.



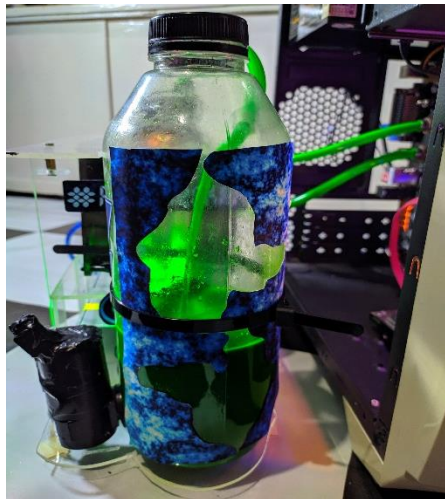
Gambar 4. 4 *Flowchart* Keseluruhan Sistem Pada Alat

Berdasarkan *flowchart* diatas cara kerja dari alat cooling komputer otomatis ini yaitu, diawali dengan menginisialisasi program terlebih dahulu selanjutnya membaca inputan serial yang dikirim dari *software interface* melalui komunikasi

serial antara komputer dan mikrokontroler, kemudian apabila mikrokontroler membaca karakter “tagar (#)” yang artinya suhu maka SuhuCore1, SuhuCore2 dan SuhuPackage dari prosesor komputer akan terbaca oleh mikrokontroler, setelah itu SuhuCore1 dan SuhuCore2 di proses pada logika *fuzzy* yang kemudian hasil dari proses logika fuzzy tersebut berupa nilai PWM untuk mengotrol peltier dan fan setelah itu kembali lagi membaca *inputan* serial. Sedangkan untuk SuhuPackage nantinya akan dikonversi menjadi indikator warna LED berdasarkan pada SuhuPackage tertentu, setelah itu kembali lagi membaca inputan serial. Kemudian apabila mikrokontroler membaca kata “dis” yang artinya *disconnect* maka indikator dari LED akan OFF dan selesai.

2. Perancangan Alat

Pada tahap perancangan alat penerapan logika *fuzzy* pada perancangan *cooling* komputer otomatis menggunakan peltier berbasis mikrokontroler Atmega 328 ini, menggunakan beberapa komponen-komponen perangkat keras yang terdiri dari mikrokontroler Atmega 328, peltier, *waterblock*, *waterblock* CPU, *heatsink*, *fan*, mosfet IRF 520, *step down*, *power supply*, kabel, LED, resistor, pompa air DC, wadah cairan dan selang air.

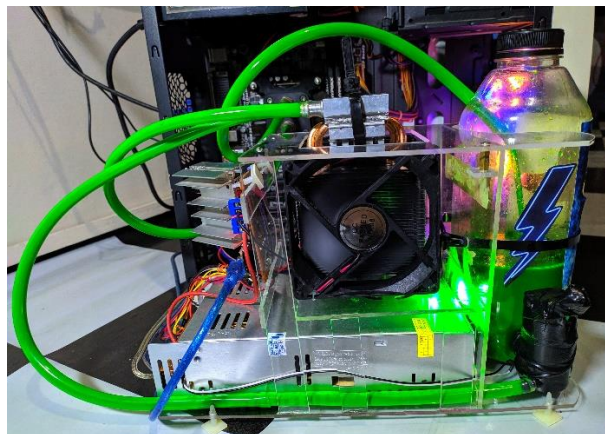


Gambar 4. 5 Rangkaian alat tampak depan

Pada gambar 4.5 terlihat rangkaian alat tampak depan yang terdiri dari wadah cairan dan pompa air DC, yang dimana wadah cairan digunakan untuk menampung cairan dan mensirkulasikan cairan tersebut dengan menggunakan pompa air DC.

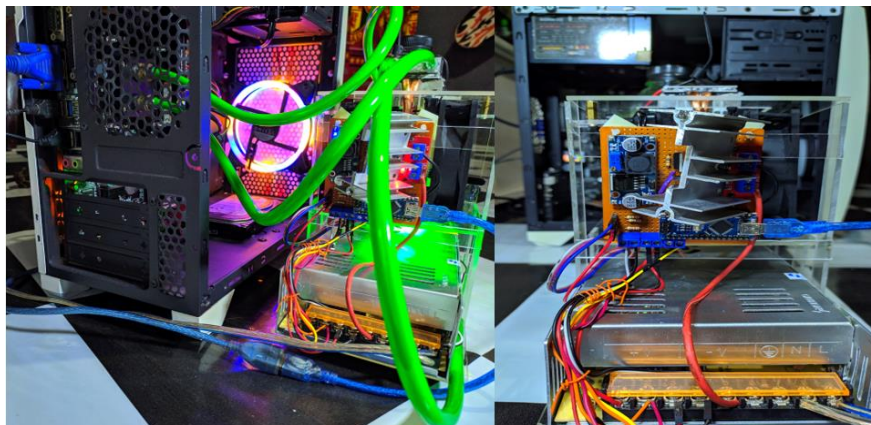


Gambar 4. 6 Rangkaian alat tampak atas



Gambar 4. 7 Rangkain alat tampak samping

Pada gambar 4.6 dan gambar. 4.7 terlihat rangkaian alat tampak atas dan samping yang terdiri dari beberapa komponen seperti *waterblock*, *peltier*, *heatsink*, *fan*, *waterblock CPU*, *led* dan *selang*.



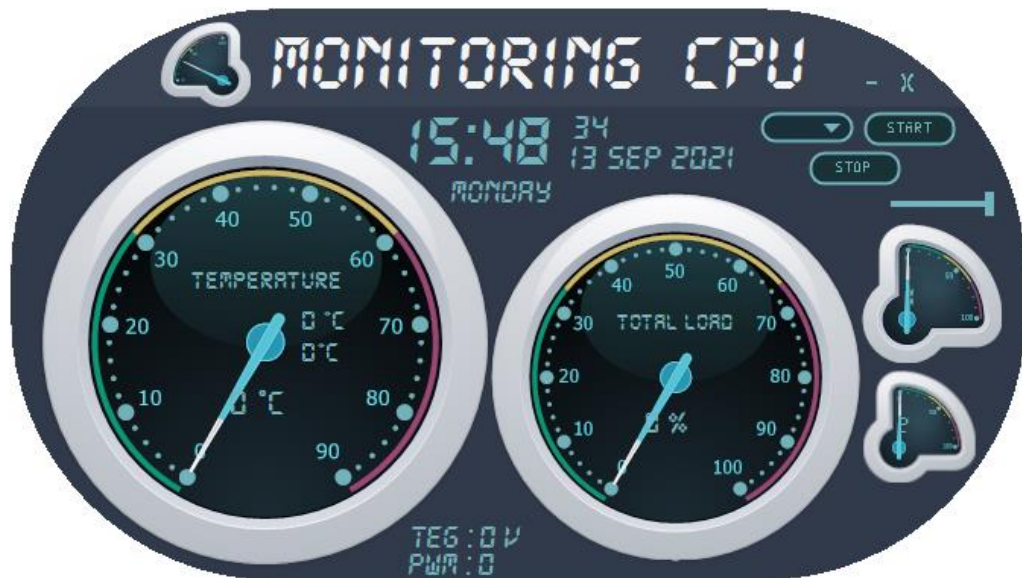
Gambar 4. 8 Rangkaian alat tampak belakang

Pada gambar 4.8 terlihat rangkaian alat tampak belakang yang dimana terlihat beberapa komponen seperti mikrokontroler Atmega 328, mosfet IRF520, *step down*, resistor, yang semua komponen tersebut di letakkan dalam satu papan PCB. Serta pada gambar di 4.8 juga terlihat komponen seperti *power supply* dan kabel.

Prinsip kerja dari alat ini, diawali dengan menghubungkan *power supply* ke catu daya, kemudian menghubungkan kabel USB antara mikrokontroler dan komputer, setelah itu menghubungkan mikrokontroler dan komputer dengan menggunakan *software interface* melalui komunikasi serial. Apabila mikrokontroler dan komputer telah terhubung maka ditandai dengan hidupnya lampu indikator dan alat bekerja. Setelah terhubung komputer akan mengirim data suhu dari prosesornya dari masing-masing core ke mikrokontroler, kemudian data suhu tersebut akan diolah datanya oleh mikrokontroler dan dilakukan proses logika *fuzzy* yang hasil dari logika *fuzzy* tersebut nantinya digunakan untuk mengotrol peltier dan *fan* guna untuk mengendalikan dan menstabilkan suhu dari prosesor komputer tersebut.

3. Perancangan Software

Pada tahap perancangan *software interface* ini digunakan *software* Microsoft Visual Studio untuk membantu melakukan perancangan dan pembuatan *software interface* pada penelitian ini. Terlihat pada gambar 4.9 merupakan tampilan dari *software interface* pada penelitian ini. *Software interface* ini digunakan sebagai media penghubung antara komputer dan mikrokontroler. Cara kerja dari *software interface* ini, diawali dengan memilih *port* dari mikrokontroler setelah itu menekan tombol *start*, ketika tombol *start* ditekan maka komputer dan mikrokontroler terhubung ditandai dengan *gauge temperature* dan *load* bergerak serta nilai dari tegangan dan PWM tidak menunjukkan angka nol lagi. Kemudian apabila tombol *stop* ditekan maka hubungan antara komputer dan mikrokontroler akan terputus serta ditandai dengan *gauge* yang berhenti bergerak.



Gambar 4. 9 Tampilan *Software Interface*

4. Perancangan Program Logika *Fuzzy*

Pada tahapan perancangan program logika *fuzzy* alat penerapan logika *fuzzy* pada perancangan *cooling* komputer otomatis menggunakan peltier berbasis mikrokontroler Atmega 328 ini menggunakan aplikasi Arduino IDE. Program ditulis dalam bahasa pemrograman C. Proses dalam metode logika *fuzzy* ini terdiri dari tahap fuzzifikasi, pembentukan aturan *fuzzy* serta komposisi aturan dan defuzzifikasi. Berikut program dari logika *fuzzy* pada penelitian ini:

```
//Fuzzifikasi SuhuCore1
unsigned char suhuDingin(){
    if (suhu <= 10){
        dingin = 1;
    }
    else if (suhu >= 10 && suhu <= 15){
        dingin = (15 - suhu)/5;
    }
    else if (suhu >= 15){
        dingin = 0;
    }
    return dingin;
}

unsigned char suhuSejuk(){
```

```
if (suhu <= 10 || suhu >= 26) {
    sejuk = 0;
}
else if (suhu >= 10 && suhu <= 15){
    sejuk = (suhu - 10)/5;
}
else if (suhu >= 21 && suhu <= 26){
    sejuk = (26 - suhu)/5;
}
else if (suhu >= 15 && suhu <= 21){
    sejuk = 1;
}
return sejuk;
}

unsigned char suhuNormal(){
    if (suhu <= 21 || suhu >= 38) {
        normal = 0;
    }
    else if (suhu >= 21 && suhu <= 26){
        normal = (suhu - 21)/5;
    }
    else if (suhu >= 33 && suhu <= 38){
        normal = (38 - suhu)/5;
    }
    else if (suhu >=26 && suhu <=33){
        normal = 1;
    }
    return normal;
}

unsigned char suhuHangat(){
    if (suhu <= 33 || suhu >= 50) {
        hangat = 0;
    }
    else if (suhu >= 33 && suhu <= 38){
        hangat = (suhu - 33)/5;
    }
}
```

```
else if (suhu >= 45 && suhu <= 50){
    hangat = (50 - suhu)/5;
}
else if (suhu >=38 && suhu <=45){
    hangat = 1;
}
return hangat;
}

unsigned char suhuPanas(){
    if (suhu >= 50 ){
        panas = 1;
    }
    else if (suhu >= 45 && suhu <= 50){
        panas = (suhu - 45)/5;
    }
    else if ( suhu <= 45){
        panas = 0;
    }
    return panas;
}

//Fuzzifikasi SuhuCore2
unsigned char suhuDingin2(){
    if (suhu2 <= 10){
        dingin2 = 1;
    }
    else if (suhu2 >= 10 && suhu2 <= 15){
        dingin2 = (15 - suhu2)/5;
    }
    else if (suhu2 >= 15){
        dingin2 = 0;
    }
    return dingin2;
}

unsigned char suhuSejuk2(){
```

```
if (suhu2 <= 10 || suhu2 >= 26) {
    sejuk2 = 0;
}
else if (suhu2 >= 10 && suhu2 <= 15){
    sejuk2 = (suhu2 - 10)/5;
}
else if (suhu2 >= 21 && suhu2 <= 26){
    sejuk2 = (26 - suhu2)/5;
}
else if (suhu2 >=15 && suhu2 <=21){
    sejuk2 = 1;
}
return sejuk2;
}

unsigned char suhuNormal2(){
    if (suhu2 <= 21 && suhu2 >= 38 ) {
        normal2 = 0;
    }
    else if (suhu2 >= 21 && suhu2 <= 26){
        normal2 = (suhu2 - 21)/5;
    }
    else if (suhu2 >= 33 && suhu2 <= 38){
        normal2 = (38 - suhu2)/5;
    }
    else if (suhu2 >=26 && suhu2 <=33){
        normal2 = 1;
    }
    return normal2;
}

unsigned char suhuHangat2(){
    if (suhu2 <= 33 || suhu2 >= 50) {
        hangat2 = 0;
    }
    else if (suhu2 >= 33 && suhu2 <= 38){
        hangat2 = (suhu2 - 33)/5;
    }
}
```

```
else if (suhu2 >= 45 && suhu2 <= 50){
    hangat2 = (50 - suhu2)/5;
}
else if (suhu2 >=38 && suhu2 <=45){
    hangat2 = 1;
}
return hangat2;
}

unsigned char suhuPanas2(){
    if (suhu2 >= 50 ){
        panas2 = 1;
    }
    else if (suhu2 >= 45 && suhu2 <= 50){
        panas2 = (suhu2 - 45)/5;
    }
    else if ( suhu2 <= 45){
        panas2 = 0;
    }
    return panas2;
}

void fuzzyfikasi(){
    suhuDingin();
    suhuSejuk();
    suhuNormal();
    suhuHangat();
    suhuPanas();
    suhuDingin2();
    suhuSejuk2();
    suhuNormal2();
    suhuHangat2();
    suhuPanas2();
}

//Pembentukan aturan fuzzy dan komposisi aturan
void rule(){
    fuzzyfikasi();
```

```
rule1 = 150;
mrule1 = min(dingin, dingin2);

rule2 = 150;
mrule2 = min(dingin, sejuk2);

rule3 = 200;
mrule3 = min(dingin, normal2);

rule4 = 200;
mrule4 = min(dingin, hangat2);

rule5 = 255;
mrule5 = min(dingin, panas2);

rule6 = 150;
mrule6 = min(sejuk, dingin2);

rule7 = 150;
mrule7 = min(sejuk, sejuk2);

rule8 = 200;
mrule8 = min(sejuk, normal2);

rule9 = 200;
mrule9 = min(sejuk, hangat2);

rule10 = 255;
mrule10 = min(sejuk, panas2);

rule11 = 200;
mrule11 = min(normal, dingin2);

rule12 = 200;
mrule12 = min(normal, sejuk2);

rule13 = 255;
mrule13 = min(normal, normal2);
```

```
rule14 = 255;
mrule14 = min(normal, hangat2);

rule15 = 255;
mrule15 = min(normal, panas2);

rule16 = 200;
mrule16 = min(hangat, dingin2);

rule17 = 200;
mrule17 = min(hangat, sejuk2);

rule18 = 255;
mrule18 = min(hangat, normal2);

rule19 = 255;
mrule19 = min(hangat, hangat2);

rule20 = 255;
mrule20 = min(hangat, panas2);

rule21 = 255;
mrule21 = min(panas, dingin2);

rule22 = 255;
mrule22 = min(panas, sejuk2);

rule23 = 255;
mrule23 = min(panas, normal2);

rule24 = 255;
mrule24 = min(panas, hangat2);

rule25 = 255;
mrule25 = min(panas, panas2);

}
```

```

//Defuzzifikasi
void defuzifikasi(){
    rule();
    float jumlah, jumlah2;

    jumlah = (mrule1 * rule1) + (mrule2 * rule2) +
    (mrule3 * rule3) + (mrule4 * rule4) + (mrule5 *
    rule5)+ (mrule6 * rule6)+ (mrule7 * rule7)+ (mrule8
    * rule8)+ (mrule9 * rule9)+ (mrule10 * rule10)+
    (mrule11 * rule11)+ (mrule12 * rule12)+ (mrule13
    * rule13)+ (mrule14 * rule14)+ (mrule15 * rule15)+
    (mrule16 * rule16)+ (mrule17 * rule17)+ (mrule18
    * rule18)+ (mrule19 * rule19)+ (mrule20 * rule20)+
    (mrule21 * rule21)+ (mrule22 * rule22)+ (mrule23
    * rule23)+ (mrule24 * rule24)+ (mrule25 *
    rule25);

    jumlah2 = (mrule1 + mrule2 + mrule3 + mrule4 + mrule5
    + mrule6 + mrule7 + mrule8 + mrule9 + mrule10 + mrule11
    + mrule12 + mrule13 + mrule14 + mrule15 + mrule16 +
    mrule17 + mrule18 + mrule19 + mrule20 + mrule21 +
    mrule22 + mrule23 + mrule24 + mrule25);

    out = jumlah/jumlah2;
}

```

4.2 Hasil

Terdapat beberapa tahapan yang akan dibahas pada bagian hasil yang di dapat berdasarkan penelitian ini yaitu pengujian dan penerapan sebagai berikut.

4.2.1 Pengujian

Pada tahap pengujian ini diawali dengan penginstalasian alat ke sebuah komputer, selanjutnya dilakukan beberapa pengujian, diantaranya:

1. Melakukan pengujian komunikasi serial

Sebelum melakukan pengujian komunikasi serial terlebih dahulu menghubungkan komputer dan mikrokontroler dengan kabel USB, kemudian

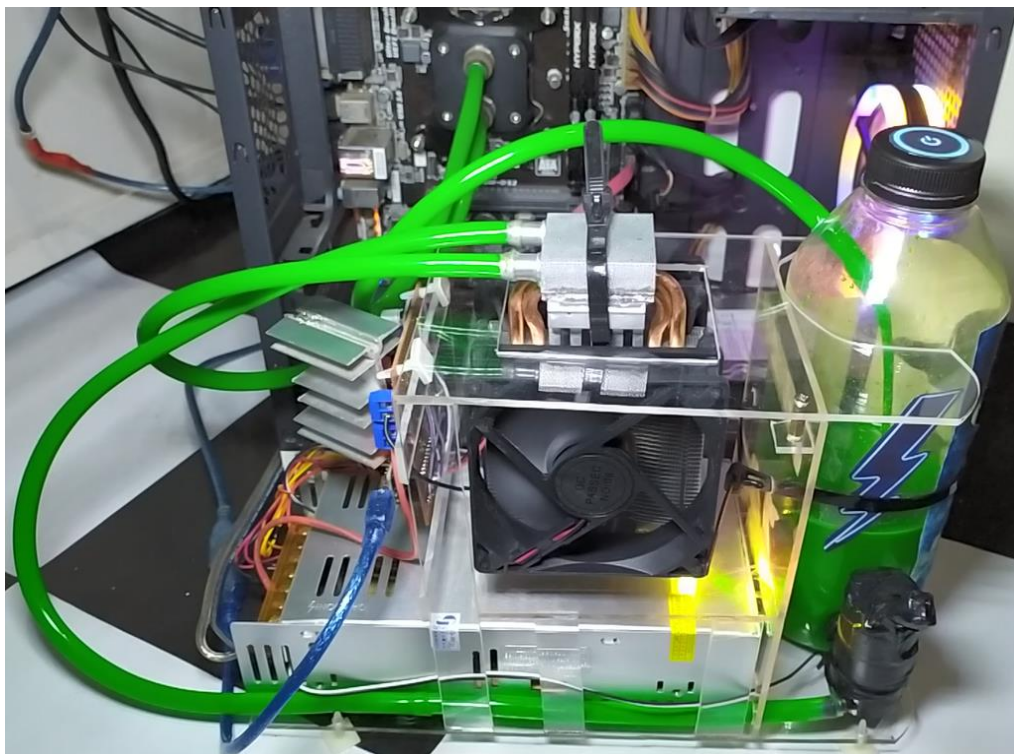
menjalankan *software interface*, selanjutnya pada *software interface* dilakukan pengimputan *port* dari mikrokontroler dan setelah itu menekan tombol *start*, apabila *gauge* bergerak dan tidak menunjukkan angka nol lagi maka komputer dan mikrokontroler telah terhubung dan dapat dipastikan pengujian komunikasi serial berhasil seperti yang terlihat pada gambar 4.10.



Gambar 4. 10 Komunikasi serial berhasil ditandai dengan gauge yang bergerak

2. Melakukan pengujian pembacaan suhu oleh mikrokontroler

Setelah komputer dan mikrokontroler berhasil terhubung selanjutnya melakukan pengujian pembacaan suhu yang diterima oleh mikrokontroler dari komputer. Apabila suhu dari komputer berhasil di terima dan dapat dibaca oleh mikrokontroler maka di tandai dengan indikator lampu LED yang hidup dengan menunjukkan warna-warna tertentu sesuai dengan besaran nilai suhu prosesor komputer yang telah di tentukan sebelumnya serta komponen alat bekerja secara keseluruhan. Terlihat pada gambar 4.11 merupakan gambar pembacaan suhu oleh mikrokontroler yang berhasil dengan ditandai lampu LED yang hidup mengeluarkan cahaya warna kuning dan komponen alat bekerja seluruhnya.



Gambar 4. 11 Pembacaan suhu oleh mikrokontroler berhasil ditandai LED hidup dan alat bekerja

3. Melakukan pengujian metode logika *fuzzy*

Selanjutnya dilakukan pengujian metode logika *fuzzy*, apabila komputer dan mikrokontroler sudah terhubung maka nilai suhu yang di terima oleh mikrokontroler di proses pada metode logika *fuzzy* yang dimana hasil dari metode logika *fuzzy* ini berupa nilai PWM yang digunakan untuk mengontrol tegangan dari peltier dan *fan*. Untuk besaran nilai dari hasil metode logika *fuzzy* yang berupa nilai PWM dapat juga dilihat pada *software interface*. Pada gambar 4.12 terlihat pada lingkaran merah merupakan besaran nilai PWM dari hasil proses metode logika *fuzzy*.



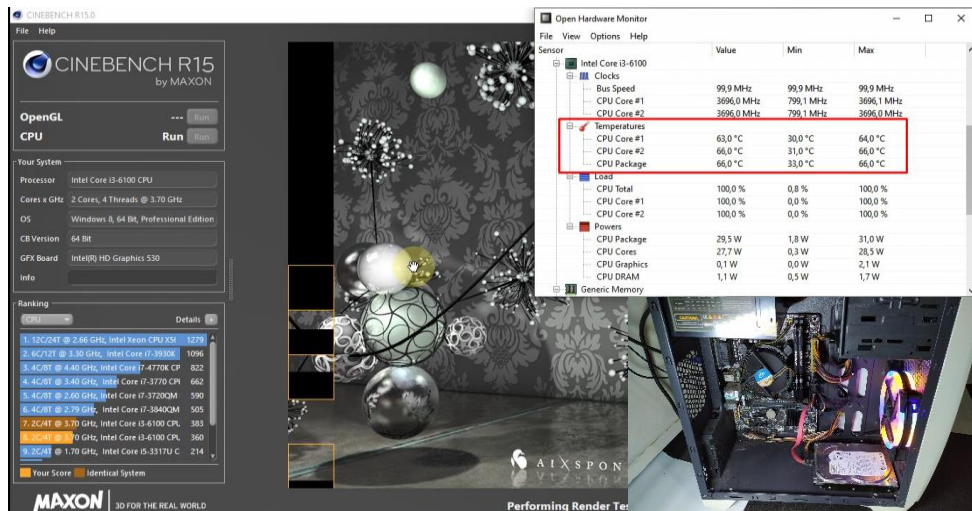
Gambar 4. 12 Hasil metode logika *fuzzy* dapat dilihat pada *software interface*

4. Melakukan pengujian dengan membandingkan alat

Pada tahap pengujian ini akan dilakukan pengujian alat dengan cara membandingkan alat yang dibuat pada penelitian ini dengan alat pendingin komputer pada umumnya seperti *air cooling* dan *water cooling* dengan radiator. Perbandingan dilakukan dengan cara melakukan uji *benchmark* terhadap prosesor komputer guna untuk menaikkan kinerja atau *load* dari prosesor, apabila kinerja atau *load* dari prosesor meningkat maka suhu dari prosesor akan terus meningkat. Uji *benchmark* akan dilakukan dengan empat software yaitu, Cinebench R15, CPUZ, 3DMark, Blender *Benchmark Launcher*.

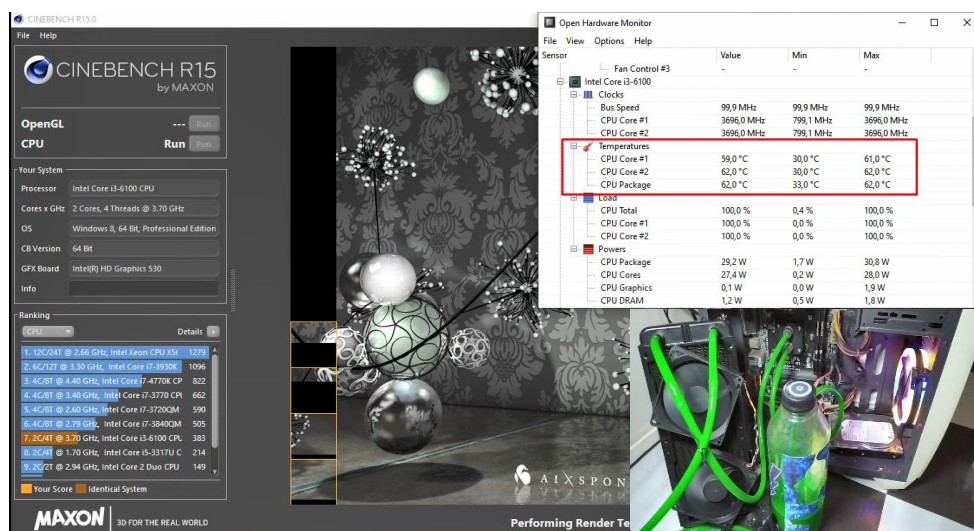
a. Cinebench R15

Tahap pertama dilakukan uji *benchmark* dengan pendingin *air cooling*, terlihat pada gambar 4.13. Uji *benchmark* dilakukan dengan 2 kali *run* dan saat dilakukan uji *benchmark* suhu ruang berkisar 29°C dan suhu prosesor dari SuhuCore1 33°C, SuhuCore2 32°C dan SuhuPackage 35 °C sebelum dilakukan uji *benchmark*.



Gambar 4. 13 Uji *benchmark* menggunakan Cinebench R15 pada *air cooling*

Saat uji *benchmark* berjalan menggunakan pendingin *air cooling* suhu prosesor dari SuhuCore1 mencapai 64°C, SuhuCore2 mencapai 66°C dan SuhuPackage mencapai 66°C. Tahap kedua dilakukan uji *benchmark* menggunakan *water cooling* dengan radiator, terlihat pada gambar 4.14. Uji *benchmark* dilakukan dengan 2 kali *run* dengan cairan yang digunakan sebanyak 500 ml dan saat dilakukan uji *benchmark* suhu ruang berkisar 30°C dan suhu prosesor dari SuhuCore1 32°C, SuhuCore2 31°C dan SuhuPackage 34°C sebelum dilakukan uji *benchmark*.



Gambar 4. 14 Uji *benchmark* menggunakan Cinebench R15 pada *water cooling* dengan radiator

Saat uji *benchmark* berjalan menggunakan pendingin *water cooling* dengan radiator suhu prosesor dari SuhuCore1 mencapai 59°C, SuhuCore2 mencapai 62°C dan SuhuPackage mencapai 62°C. Tahap ketiga dilakukan uji *benchmark* menggunakan *water cooling* peltier dengan logika *fuzzy*, terlihat pada gambar 4.15. Uji *benchmark* dilakukan dengan 2 kali run dengan cairan yang digunakan sebanyak 500 ml dan saat dilakukan uji *benchmark* suhu ruang berkisar 30°C dan suhu prosesor dari SuhuCore1 27°C, SuhuCore2 27°C dan SuhuPackage 29°C sebelum dilakukan uji *benchmark*.



Gambar 4. 15 Uji *benchmark* menggunakan Cinebench R15 pada *water cooling* peltier dengan logika *fuzzy*

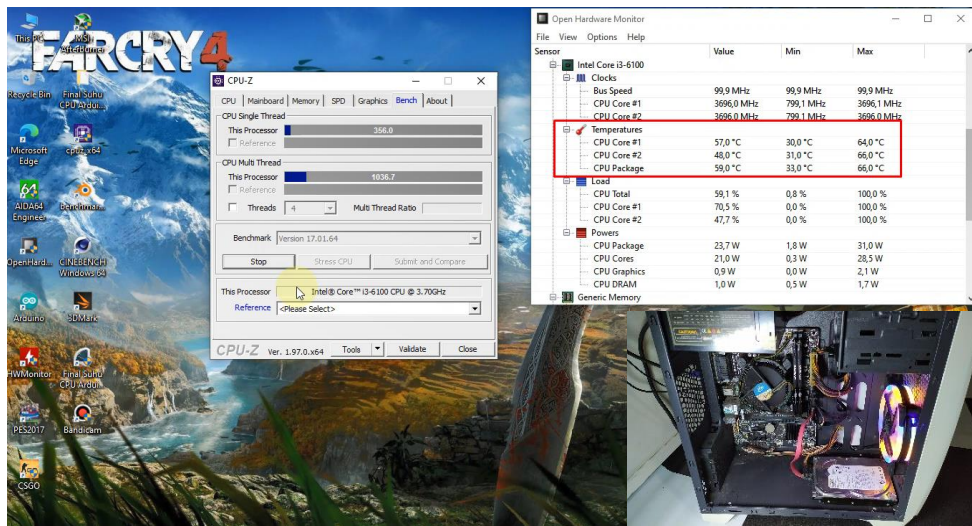
Saat uji *benchmark* berjalan menggunakan pendingin *water cooling* peltier dengan logika *fuzzy* suhu prosesor dari SuhuCore1 mencapai 49°C, SuhuCore2 mencapai 51°C dan SuhuPackage mencapai 52°C. Hasil dari perbandingan uji coba *benchmark* diatas dengan menggunakan beberapa pendingin yang berbeda, maka dapat disimpulkan dalam tabel 4.2 berikut:

Tabel 4. 2 Perbandingan suhu dengan uji *benchmark* menggunakan Cinebench

Suhu (°C)		Jenis Pendingin			
		<i>Air Cooling</i>	<i>Water Cooling Radiator</i>	<i>Water Cooling Peltier Dengan Fuzzy</i>	
Suhu Ruang		29 °C	30 °C	30 °C	
Suhu Sebelum Benchmark	Core1	33 °C	32 °C	27 °C	
	Core2	32 °C	31 °C	27 °C	
	Package	35 °C	34 °C	29 °C	
Suhu Sesudah Benchmark	Core1	39 °C	32 °C	29 °C	
	Core2	40 °C	33 °C	31 °C	
	Package	42 °C	35 °C	32 °C	
Suhu Benchmark Berjalan	Min	Core1	53 °C	53 °C	47 °C
		Core2	53 °C	53 °C	48 °C
		Package	53 °C	53 °C	48 °C
	Max	Core1	64 °C	59 °C	49 °C
		Core2	66 °C	62 °C	51 °C
		Package	66 °C	62 °C	52 °C

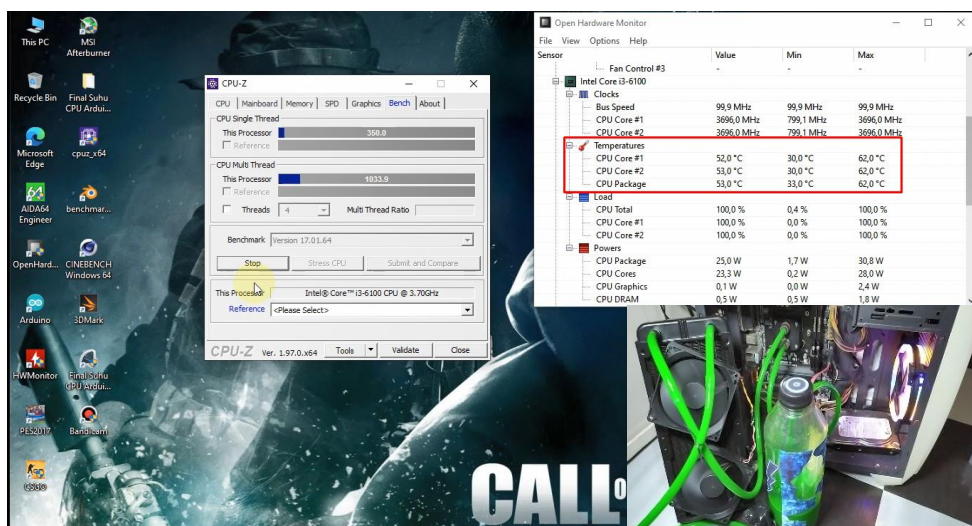
b. CPUZ

Tahap pertama dilakukan uji *benchmark* dengan pendingin *air cooling*, terlihat pada gambar 4.16. Uji *benchmark* dilakukan dengan 2 kali *run* dan saat dilakukan uji *benchmark* suhu ruang berkisar 29°C dan suhu prosesor dari SuhuCore1 34°C, SuhuCore2 34°C dan SuhuPackage 36 °C sebelum dilakukan uji *benchmark*.



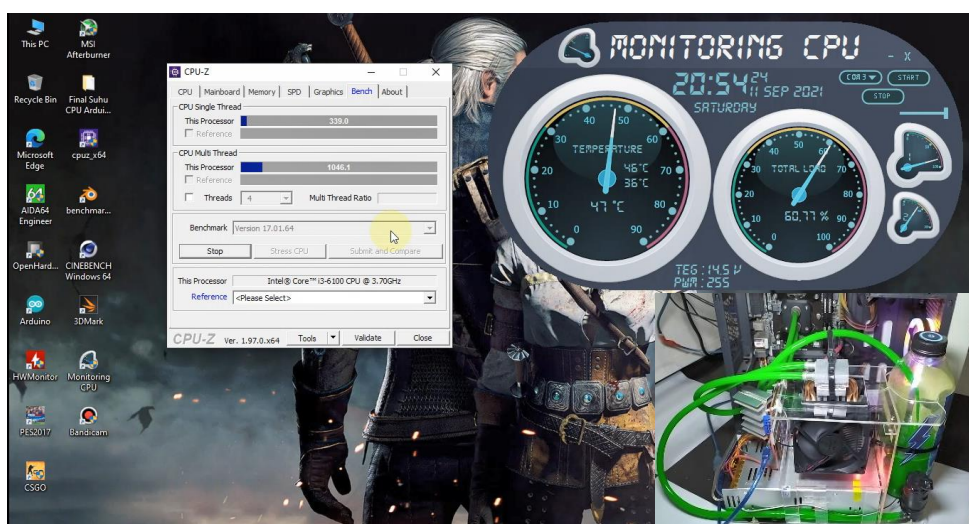
Gambar 4. 16 Uji *benchmark* menggunakan CPUZ pada *air cooling*

Saat uji *benchmark* berjalan menggunakan pendingin *air cooling* suhu prosesor dari SuhuCore1 mencapai 57°C, SuhuCore2 mencapai 48°C dan SuhuPackage mencapai 59°C. Tahap kedua dilakukan uji *benchmark* menggunakan *water cooling* dengan radiator, terlihat pada gambar 4.17. Uji *benchmark* dilakukan dengan 2 kali *run* dengan cairan yang digunakan sebanyak 500 ml dan saat dilakukan uji *benchmark* suhu ruang berkisar 30°C dan suhu prosesor dari SuhuCore1 32°C, SuhuCore2 31°C dan SuhuPackage 34°C sebelum dilakukan uji *benchmark*.



Gambar 4. 17 Uji *benchmark* menggunakan CPUZ pada *water cooling* dengan radiator

Saat uji *benchmark* berjalan menggunakan pendingin *water cooling* dengan radiator suhu prosesor dari SuhuCore1 mencapai 52°C, SuhuCore2 mencapai 53°C dan SuhuPackage mencapai 53°C. Tahap ketiga dilakukan uji *benchmark* menggunakan *water cooling* peltier dengan logika *fuzzy*, terlihat pada gambar 4.18. Uji *benchmark* dilakukan dengan 2 kali run dengan cairan yang digunakan sebanyak 500 ml dan saat dilakukan uji *benchmark* suhu ruang berkisar 30°C dan suhu prosesor dari SuhuCore1 28°C, SuhuCore2 27°C dan SuhuPackage 29°C sebelum dilakukan uji *benchmark*.



Gambar 4. 18 Uji *benchmark* menggunakan CPUZ pada *water cooling* peltier dengan logika *fuzzy*

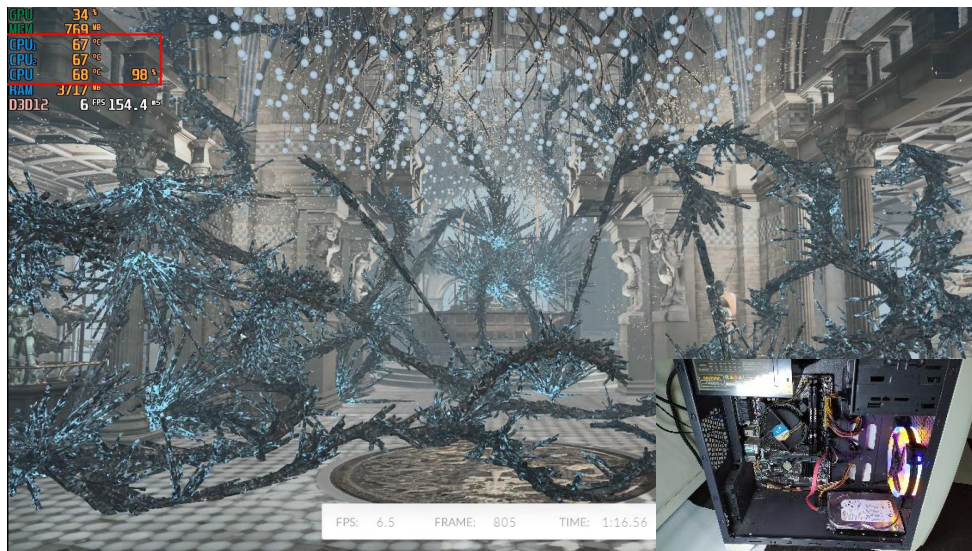
Saat uji *benchmark* berjalan menggunakan pendingin *water cooling* peltier dengan logika *fuzzy* suhu prosesor dari SuhuCore1 mencapai 46°C, SuhuCore2 mencapai 36°C dan SuhuPackage mencapai 47°C. Hasil dari perbandingan uji coba *benchmark* diatas dengan menggunakan beberapa pendingin yang berbeda, maka dapat disimpulkan dalam tabel 4.3 berikut:

Tabel 4. 3 Perbandingan suhu dengan uji *benchmark* menggunakan CPUZ

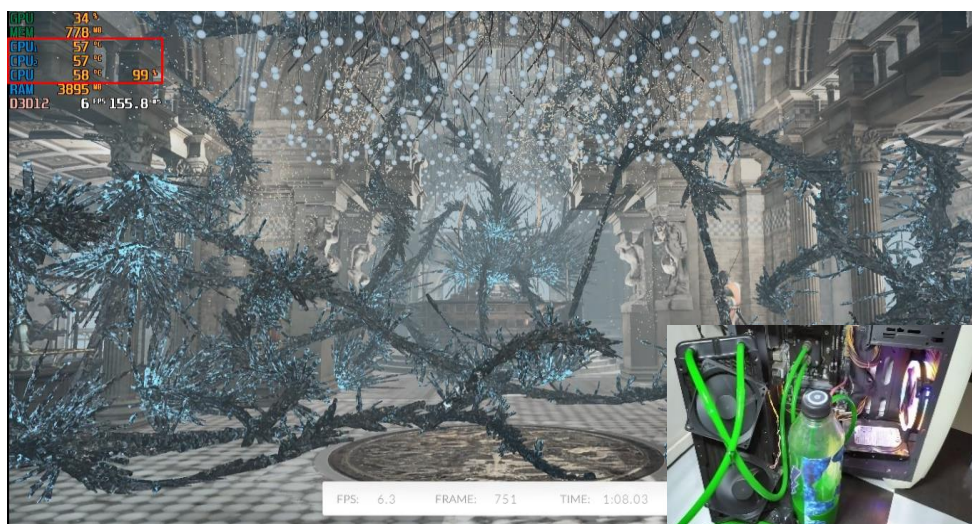
Suhu (°C)		Jenis Pendingin			
		<i>Air Cooling</i>	<i>Water Cooling Radiator</i>	<i>Water Cooling Peltier Dengan Fuzzy</i>	
Suhu Ruang		29 °C	30 °C	30 °C	
Suhu Sebelum Benchmark	<i>Core1</i>	34 °C	32 °C	28 °C	
	<i>Core2</i>	34 °C	31 °C	27 °C	
	<i>Package</i>	36 °C	34 °C	29 °C	
Suhu Sesudah Benchmark	<i>Core1</i>	39 °C	33 °C	28 °C	
	<i>Core2</i>	38 °C	33 °C	28 °C	
	<i>Package</i>	41 °C	35 °C	30 °C	
Suhu Benchmark Berjalan	<i>Min</i>	<i>Core1</i>	49 °C	52 °C	43 °C
		<i>Core2</i>	50 °C	47 °C	45 °C
		<i>Package</i>	51 °C	50 °C	45 °C
	<i>Max</i>	<i>Core1</i>	57 °C	52 °C	46 °C
		<i>Core2</i>	48 °C	53 °C	36 °C
		<i>Package</i>	59 °C	53 °C	47 °C

c. 3Dmark

Tahap pertama dilakukan uji *benchmark* dengan pendingin *air cooling*, terlihat pada gambar 4.19. Uji *benchmark* dilakukan dengan 1 kali *run* dan saat dilakukan uji *benchmark* suhu ruang berkisar 29°C dan suhu prosesor dari SuhuCore1 33°C, SuhuCore2 33°C dan SuhuPackage 35 °C sebelum dilakukan uji *benchmark*.

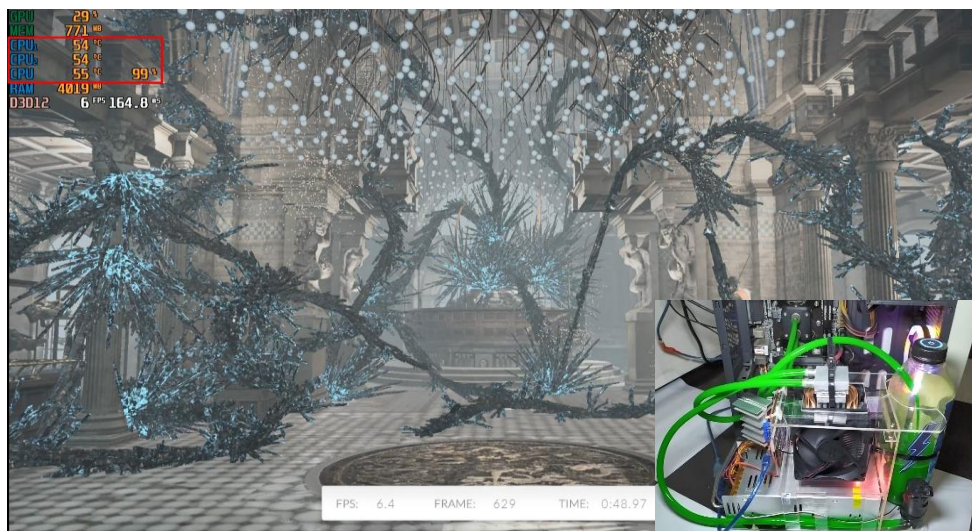


Gambar 4. 19 Uji *benchmark* menggunakan 3DMark pada *air cooling*. Saat uji *benchmark* berjalan menggunakan pendingin *air cooling* suhu prosesor dari SuhuCore1 mencapai 67°C, SuhuCore2 mencapai 67°C dan SuhuPackage mencapai 68°C. Tahap kedua dilakukan uji *benchmark* menggunakan *water cooling* dengan radiator, terlihat pada gambar 4.20. Uji *benchmark* dilakukan dengan 1 kali *run* dengan cairan yang digunakan sebanyak 500 ml dan saat dilakukan uji *benchmark* suhu ruang berkisar 30°C dan suhu prosesor dari SuhuCore1 32°C, SuhuCore2 31°C dan SuhuPackage 34°C sebelum dilakukan uji *benchmark*.



Gambar 4. 20 Uji *benchmark* menggunakan 3DMark pada *water cooling* dengan radiator

Saat uji *benchmark* berjalan menggunakan pendingin *water cooling* dengan radiator suhu prosesor dari SuhuCore1 mencapai 57°C, SuhuCore2 mencapai 57°C dan SuhuPackage mencapai 58°C. Tahap ketiga dilakukan uji *benchmark* menggunakan *water cooling* peltier dengan logika *fuzzy*, terlihat pada gambar 4.21. Uji *benchmark* dilakukan dengan 1 kali *run* dengan cairan yang digunakan sebanyak 500 ml dan saat dilakukan uji *benchmark* suhu ruang berkisar 30°C dan suhu prosesor dari SuhuCore1 26°C, SuhuCore2 26°C dan SuhuPackage 29°C sebelum dilakukan uji *benchmark*.



Gambar 4. 21 Uji *benchmark* menggunakan 3DMark pada *water cooling* peltier dengan logika *fuzzy*

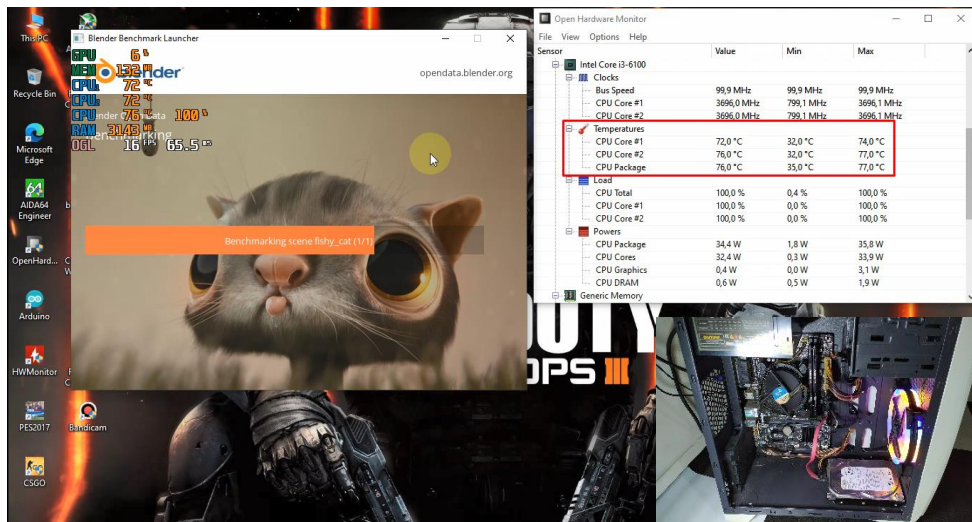
Saat uji *benchmark* berjalan menggunakan pendingin *water cooling* peltier dengan logika *fuzzy* suhu prosesor dari SuhuCore1 mencapai 54°C, SuhuCore2 mencapai 54°C dan SuhuPackage mencapai 55°C. Hasil dari perbandingan uji coba *benchmark* diatas dengan menggunakan beberapa pendingin yang berbeda, maka dapat disimpulkan dalam tabel 4.4 berikut:

Tabel 4. 4 Perbandingan suhu dengan uji *benchmark* menggunakan 3DMark

Suhu (°C)		Jenis Pendingin			
		<i>Air Cooling</i>	<i>Water Cooling Radiator</i>	<i>Water Cooling Peltier Dengan Fuzzy</i>	
Suhu Ruang		29 °C	30 °C	30 °C	
Suhu Sebelum Benchmark	Core1	33 °C	32 °C	26 °C	
	Core2	33 °C	31 °C	26 °C	
	Package	35 °C	34 °C	29 °C	
Suhu Sesudah Benchmark	Core1	42 °C	33 °C	35 °C	
	Core2	42 °C	35 °C	33 °C	
	Package	44 °C	36 °C	35 °C	
Suhu Benchmark Berjalan	Min	Core1	47 °C	37 °C	32 °C
		Core2	47 °C	37 °C	32 °C
		Package	49 °C	43 °C	37 °C
	Max	Core1	67 °C	57 °C	54 °C
		Core2	67 °C	57 °C	54 °C
		Package	68 °C	58 °C	55 °C

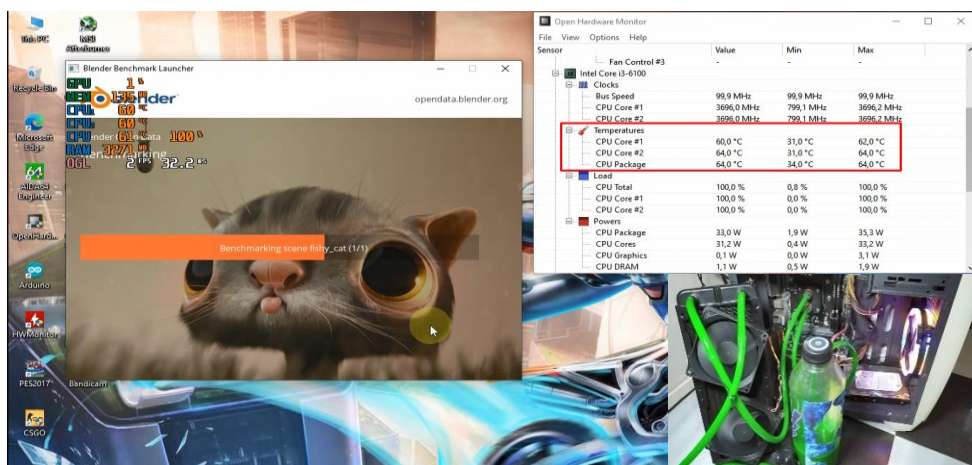
d. Blender *Benchmark Launcher*

Tahap pertama dilakukan uji *benchmark* dengan pendingin *air cooling*, terlihat pada gambar 4.22. Uji *benchmark* dilakukan dengan 1 kali *run* dan saat dilakukan uji *benchmark* suhu ruang berkisar 29°C dan suhu prosesor dari SuhuCore1 34°C, SuhuCore2 33°C dan SuhuPackage 35°C sebelum dilakukan uji *benchmark*.



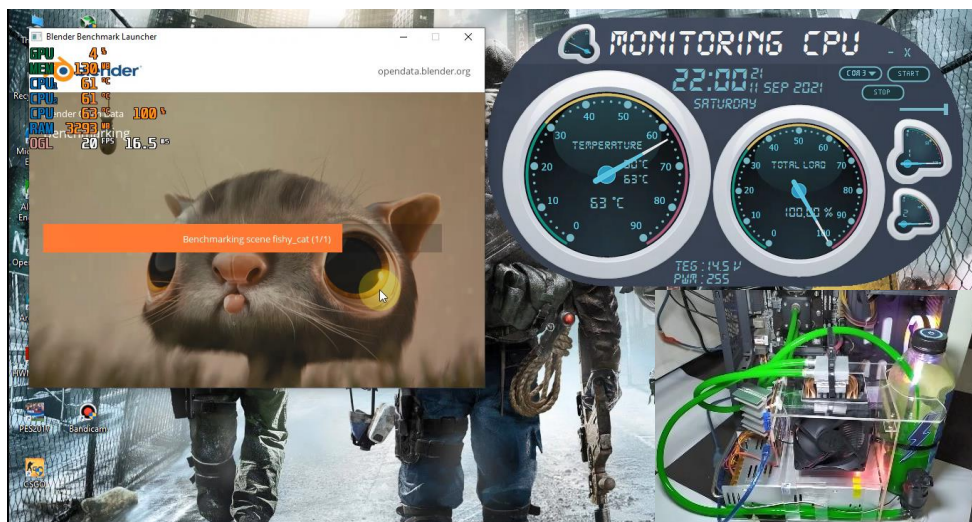
Gambar 4. 22 Uji *benchmark* menggunakan Blender *Benchmark Launcher* pada *air cooling*

Saat uji *benchmark* berjalan menggunakan pendingin *air cooling* suhu prosesor dari SuhuCore1 mencapai 72°C, SuhuCore2 mencapai 76°C dan SuhuPackage mencapai 76°C. Tahap kedua dilakukan uji *benchmark* menggunakan *water cooling* dengan radiator, terlihat pada gambar 4.23. Uji *benchmark* dilakukan dengan 1 kali *run* dengan cairan yang digunakan sebanyak 500 ml dan saat dilakukan uji *benchmark* suhu ruang berkisar 30°C dan suhu prosesor dari SuhuCore1 33°C, SuhuCore2 33°C dan SuhuPackage 35°C sebelum dilakukan uji *benchmark*.



Gambar 4. 23 Uji *benchmark* menggunakan Blender *Benchmark Launcher* pada *water cooling* dengan radiator

Saat uji *benchmark* berjalan menggunakan pendingin *water cooling* dengan radiator suhu prosesor dari SuhuCore1 mencapai 60°C, SuhuCore2 mencapai 64°C dan SuhuPackage mencapai 64°C. Tahap ketiga dilakukan uji *benchmark* menggunakan *water cooling* peltier dengan logika *fuzzy*, terlihat pada gambar 4.24. Uji *benchmark* dilakukan dengan 1 kali *run* dengan cairan yang digunakan sebanyak 500 ml dan saat dilakukan uji *benchmark* suhu ruang berkisar 30°C dan suhu prosesor dari SuhuCore1 26°C, SuhuCore2 27°C dan SuhuPackage 29°C sebelum dilakukan uji *benchmark*.



Gambar 4. 24 Uji *benchmark* menggunakan Blender *Benchmark Launcher* pada *water cooling* peltier dengan logika *fuzzy*

Saat uji *benchmark* berjalan menggunakan pendingin *water cooling* peltier dengan logika *fuzzy* suhu prosesor dari SuhuCore1 mencapai 60°C, SuhuCore2 mencapai 63°C dan SuhuPackage mencapai 63°C. Hasil dari perbandingan uji coba *benchmark* diatas dengan menggunakan beberapa pendingin yang berbeda, maka dapat disimpulkan dalam tabel 4.5 berikut:

Tabel 4. 5 Perbandingan suhu dengan uji *benchmark* menggunakan Blender *Benchmark Launcher*

Suhu (°C)		Jenis Pendingin			
		<i>Air Cooling</i>	<i>Water Cooling Radiator</i>	<i>Water Cooling Peltier Dengan Fuzzy</i>	
Suhu Ruang		29 °C	30 °C	30 °C	
Suhu Sebelum Benchmark	Core1	34 °C	33 °C	26 °C	
	Core2	33 °C	33 °C	27 °C	
	Package	35 °C	35 °C	29 °C	
Suhu Sesudah Benchmark	Core1	42 °C	34 °C	33 °C	
	Core2	43 °C	34 °C	33 °C	
	Package	45 °C	36 °C	35 °C	
Suhu Benchmark Berjalan	Min	Core1	62 °C	54 °C	50 °C
		Core2	64 °C	56 °C	53 °C
		Package	64 °C	56 °C	53 °C
	Max	Core1	72 °C	60 °C	60 °C
		Core2	76 °C	64 °C	63 °C
		Package	76 °C	64 °C	63 °C

Hasil dari pengujian diatas dengan membandingkan ketiga pendingin komputer *air cooling*, *water cooling* dengan radiator dan *water cooling* peltier dengan logika *fuzzy*, *water cooling* peltier dengan logika *fuzzy* menjadi yang paling baik dalam melakukan pendinginan terhadap prosesor komputer dibuktikan dengan melakukan pengujian dengan uji *benchmark* terhadap prosesor dengan empat *software* dengan rata-rata SuhuCore1 52,25°C, SuhuCore2 51°C dan SuhuPackage 54,25°C, sedangkan dengan *air cooling* rata-rata SuhuCore1 65°C, SuhuCore2 64,25°C dan SuhuPackage 67,25°C serta dengan *water cooling*

radiador rata-rata SuhuCore1 57°C, SuhuCore2 59°C dan SuhuPackage 59,25°C dengan rata-rata suhu ruang 30 °C saat pengujian berlangsung.

4.2.2 Penerapan

Penerapan dari alat pada penelitian ini nantinya alat ini akan diterapkan pada komputer-komputer yang mengalami permasalahan lambat atau lag yang disebabkan oleh *overheat* sehingga dengan alat ini komputer dapat menjaga kinerja performa dari prosesor agar tetap stabil.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari pengujian pada penerapan logika *fuzzy* pada perancangan *cooling* komputer otomatis menggunakan peltier berbasis mikrokontroler Atmega 328, dapat ditarik kesimpulan sebagai berikut:

1. Dari hasil penerapan alat *water cooling* peltier dengan logika *fuzzy* pada penelitian ini ke sebuah komputer dengan prosesor *core i3 6100* terbukti lebih baik dan efektif dalam mendingin prosesor komputer dengan rata-rata SuhuCore1 26,75°C, SuhuCore2 25,25°C dan SuhuPackage 29°C, sedangkan dengan *air cooling* rata-rata SuhuCore1 33,5°C, SuhuCore2 33°C dan SuhuPackage 35,25°C serta dengan *water cooling* radiator rata-rata SuhuCore1 32,25°C, SuhuCore2 31,5°C dan SuhuPackage 34,25°C pada saat sebelum melakukan uji *benchmark*. Ketika uji *benchmark* berjalan rata-rata SuhuCore1 52,25°C, SuhuCore2 51°C dan SuhuPackage 54,25°C dengan menggunakan *water cooling* peltier dengan logika *fuzzy*, sedangkan dengan *air cooling* rata-rata SuhuCore1 65°C, SuhuCore2 64,25°C dan SuhuPackage 67,25°C serta dengan *water cooling* radiator rata-rata SuhuCore1 57°C, SuhuCore2 59°C dan SuhuPackage 59,25°C dengan rata-rata suhu ruang 30 °C saat pengujian berlangsung. Uji coba tersebut dilakukan dengan melakukan uji *benchmark* dengan menggunakan empat *software* yang berbeda. Dari hasil uji coba tersebut pendingin pada alat penelitian ini lebih baik dari pada pendingin komputer pada umumnya seperti *air cooling* dan *water cooling* dengan radiator sehingga untuk menjaga performa kinerja dari prosesor komputer lebih baik daripada pendingin komputer lainnya.
2. Dari hasil uji coba pada penelitian ini penerapan logika *fuzzy* pada alat *cooling* komputer otomatis ini berhasil diterapkan untuk mengotrol suhu dari komputer agar tetap stabil.

5.2 Saran

Saran peneliti untuk melakukan pengembangan pada penelitian ini, diantaranya sebagai berikut:

1. Untuk mendapatkan hasil dingin yang lebih maksimal peneliti menyarankan untuk menggunakan heatsink pada sisi panas peltier dengan lebih dari 2 *pipe*.
2. Untuk pengembangan dari alat ini peneliti menyarankan untuk membuat alat ini lebih fleksibel agar dapat berjalan pada semua jenis prosesor terutama lebih dari 2 core.
3. Untuk cairan coolant peneliti menyarankan agar menggunakan coolant mobil yang lebih baik dalam melakukan penyerapan panas.
4. Banyaknya cairan yang digunakan disesuaikan dengan kemampuan dari peltier.

DAFTAR PUSTAKA

- Agus Salim, A. T., & Indarto, B. (2018). Studi Eksperimental Karakterisasi Elemen Termoelektrik Peltier Tipe TEC. *JEECAE (Journal of Electrical, Electronics, Control, and Automotive Engineering)*, 3(1), 179–182. <https://doi.org/10.32486/jeecae.v3i1.211>
- Arduino.cc. (2018, February 5). Arduino Tutorial - PWM. Retrieved June 1, 2021, from <https://www.arduino.cc/en/Tutorial/Foundations/PWM>
- Arifin, J., Dewanti, I. E., & Kurnianto, D. (2017). Prototipe Pendingin Perangkat Telekomunikasi Sumber Arus DC menggunakan Smartphone. *Media ElektriKa*, 10(1), 13–29.
- Arindya, R. (2017). *Mekatronika*. Yogyakarta: Teknosain.
- Asrofi, B. E. P. (2013). Rancang Bangun Alat Kontrol Otomatis Pendingin Komputer Berbasis Mikrokontroler Atmega8L. *Indonesian Jurnal on Networking and Security (IJNS)*, 2(2), 23–29.
- Cholish, C., Rimbawati, R., & Hutasuhut, A. A. (2017). Analisa Perbandingan Switch Mode Power Supply (SMPS) dan Transformator Linear Pada Audio Amplifier. *CIRCUIT: Jurnal Ilmiah Pendidikan Teknik Elektro*, 1(2), 90–102. <https://doi.org/10.22373/crc.v1i2.2079>
- COMPONENTS101. (2019, January 11). IRF520 N-Channel Power MOSFET. Retrieved June 28, 2021, from <https://components101.com/mosfets/irf520-pinout-datasheet-features>
- Fadhillah, A., Insidini, F., Muhathir, & N P, D. (2019). Rancang Bangun Water Level Detection Dengan Sensor Ultrasonik Berbasis Fuzzy Logic. *Journal of Informatics and Telecommunication Engineering*, 3(1), 142–149.
- Fauziah jihan. (2021, October 15). Ini Dia 8 Jenis USB dari Versi Pertama hingga Saat ini. Retrieved November 22, 2021, from <https://carisinyal.com/jenis-jenis-usb/>

- Hakim, A. (2020, April 8). Penyebab dan Cara Mengatasinya Komputer Lag dan Lemot. Retrieved August 3, 2021, from <https://www.multimedialy.com/2020/04/penyebab-komputer-lemot-cara-mengatasinya.html>
- Hamdani, R., Puspita, I. H., & Wildan, B. D. R. W. (2019). Pembuatan Sistem Pengamanan Kendaraan Bermotor Berbasis Radio Frequency Identification (Rfid). *Indept*, 8(2), 56–63.
- Intel. (2020, January 8). Pendingin CPU: Pendingin Cair VS Pendingin Udara. Retrieved August 2, 2021, from <https://www.intel.co.id/content/www/id/id/gaming/resources/cpu-cooler-liquid-cooling-vs-air-cooling.html>
- Irfan, M. (2016). *Implementasi Logika Fuzzy Inference System Metode Sugeno Pada Penentuan Jumlah Produksi Sarung (Studi Kasus: PT. Asaputex Jaya Tegal)*. Semarang.
- Kho, D. (2014, December 7). Pengertian Power Supply dan Jenis-jenisnya. Retrieved May 20, 2021, from <https://teknikelektronika.com/pengertian-power-supply-jenis-catu-daya/>
- Kurnia Utama, Y. A., Tamaji, & Sanjaya, R. H. (2018). DESAIN DAN PENGENDALIAN WARNA MOOD LAMP OTOMATIS BERDASARKAN WAKTU MENGGUNAKAN APLIKASI ANDROID PADA SMARTPHONE. *TECNOSCIENZA*, 2(2), 124–143.
- Kusumadewi, S., & Purnomo, H. (2010). *Aplikasi Logika Fuzzy Untuk pendukung Keputusan* (2nd ed.). Yogyakarta: Graha Ilmu.
- Lubis, I., & Safi, M. (2018). *Smart Economy Kota Tangerang Selatan*. Tangerang: KAMI.
- Lubis, Z., Sitorus, T., & UHSG, T. (2017). Analisa kinerja sistempendingin peltier yang menggunakan sel PVdengan sumber energi radiasi matahari. *Jurnal Energi Dan Manufaktur*, 9(2), 166–173.

- Marwanto. (2018, January 4). 10 Penyebab Komputer Lambat Dan Cara Mengatasinya. Retrieved August 4, 2021, from <https://digitalsolution.co.id/10-penyebab-komputer-lambat-dan-cara-mengatasinya/>
- Pawawoi, A., & Zulfahmi, Z. (2019). Penambahan Sistem Pendingin Heatsink Untuk Optimasi Penggunaan Reflektor Pada Panel Surya. *JURNAL NASIONAL TEKNIK ELEKTRO*, 8(1), 1–7. <https://doi.org/10.25077/jnte.v8n1.607.2019>
- Pusadan, M. (2014). *Pemrograman Matlab Pada Sistem Pakar Fuzzy*. Yogyakarta: Deepublish.
- Putri, N. E., & Azpar, S. (2017). Sistem Informasi Pengolahan Data Pendidikan Anak Usia Dini (PAUD) Terpadu Amalia Syukra Padang. *Edik Informatika*, 2(2), 203–212. <https://doi.org/10.22202/ei.2016.v2i2.1464>
- Riandika, P., Wigraha, N. A., & Nugraha, I. N. P. (2018). PENGARUH KECEPATAN ALIRAN FLUIDA TERHADAP CAPAIAN SUHU OPTIMAL HASIL RANCANGAN COOLBOX ZERO POLLUTION. *Jurnal Pendidikan Teknik Mesin Undiksha*, 6(3), 160–167. <https://doi.org/10.23887/jjtm.v6i3.14989>
- Risyan, R. (2021, June 25). Perbedaan Water Cooling Dan Air Cooling Pada PC. Retrieved August 4, 2021, from <https://www.monitorteknologi.com/perbedaan-water-cooling-dan-air-cooling/>
- Rustan, H., Haryansyah, & M, D. P. (2017). Rekayasa Perangkat Kendali Tangan Robot Menggunakan Modul Wireless NRF. *Journal of Applied Microcontrollers and Autonomous System*, 3(1), 38–46.
- Santoso, S., & Nurmalina, R. (2017). Perencanaan dan Pengembangan Aplikasi Absensi Mahasiswa Menggunakan Smart Card Guna Pengembangan Kampus Cerdas (Studi Kasus Politeknik Negeri Tanah Laut). *Jurnal Integrasi*, 9(1), 84–91.

- Sriani, S. (2019). Pemanfaatan Sistem Pengendali Water Level Control Untuk Budidaya Ikan Gurame Pada Kolam Terpal Menggunakan Logika Fuzzy Berbasis Mikrokontroler. *Elkawnie*, 5(1), 47–57. <https://doi.org/10.22373/ekw.v5i1.3766>
- Suhardi, S. (2019). Keran Air Otomatis Pada Bak Mandi Berbasis Arduino Uno Menggunakan Sensor Ultrasonic. *ALGORITMA: JURNAL ILMU KOMPUTER DAN INFORMATIKA*, 3(1), 48–54. <https://doi.org/10.30829/algorithm.v3i1.4438>
- Sujarwata. (2016). *Belajar Mikrokontroler PIC16C57 Dengan Bahasa Pemrograman PBasic*. Yogyakarta: Deepublish.
- Sutiono. (2016, January 4). CPU Panas - Penyebab, Dampak dan Cara Mengatasinya. Retrieved May 5, 2021, from <https://dosenit.com/hardware/cpu/cpu-panas>
- Syahwill, M. (2013). *Panduan Mudah Simulasi dan Praktik Mikrokontroler Arduino*. Yogyakarta: ANDI.
- Triansah, A. (2017). Autentifikasi Login User pada Perangkat Lunak Menggunakan Arduino dan Enkripsi AES 256. *EXPERT: Jurnal Manajemen Sistem Informasi dan Teknologi*, 7(2), 90–95. <https://doi.org/10.36448/jmsit.v7i2.968>
- Wati, D. (2011). *Sistem Kendali Cerdas: Fuzzy Logic Controller (FLC) Jaringan Syaraf Tiruan (JST), Algoritma Genetik (AG), dan Algoritma Particle Swarm Optimization (PSO)*. Yogyakarta: Graha Ilmu.
- Wijaya, M. (2020, May 10). 6 Cara Mengatasi Prosesor/CPU Panas (Overheat). Retrieved May 21, 2021, from <https://www.nitrotekno.com/cara-mengatasi-prosesor-cpu-panas-overheat/>
- WS, M. (2015). *Membuat Robot Cerdas Berbasis Vision - Menggunakan MATLAB dan CodeVisionAVR*. Yogyakarta: ANDI.
- Yahfizham. (2019). *Dasar-Dasar Komputer*. Medan: Perdana Publishing.

LAMPIRAN-LAMPIRAN

Lampiran 1

Listing Program Mikrokontroler Menggunakan Arduino IDE

```
#define pwm 3
#define fan 5
int red = 9;
int green = 10;
int blue = 11;

String inData;
float suhu, suhu2, suhuPackage, out;
float dingin, sejuk, normal, hangat, panas, dingin2, sejuk2,
normal2, hangat2, panas2;
float lambat, sedang, cepat;
float rule1, rule2, rule3, rule4, rule5, rule6, rule7,
rule8, rule9, rule10,
rule11, rule12, rule13, rule14, rule15, rule16, rule17,
rule18, rule19, rule20,
rule21, rule22, rule23, rule24, rule25;
float mrule1, mrule2, mrule3, mrule4, mrule5, mrule6,
mrule7, mrule8, mrule9, mrule10,
mrule11, mrule12, mrule13, mrule14, mrule15, mrule16,
mrule17, mrule18, mrule19, mrule20,
mrule21, mrule22, mrule23, mrule24, mrule25;

//Fuzzifikasi
unsigned char suhuDingin(){
  if (suhu <= 10){
    dingin = 1;
  }
  else if (suhu >= 10 && suhu <= 15){
    dingin = (15 - suhu)/5;
  }
  else if (suhu >= 15){
    dingin = 0;
  }
}
```

```

    return dingin;
}

unsigned char suhuSejuk(){
    if (suhu <= 10 || suhu >= 26) {
        sejuk = 0;
    }
    else if (suhu >= 10 && suhu <= 15){
        sejuk = (suhu - 10)/5;
    }
    else if (suhu >= 21 && suhu <= 26){
        sejuk = (26 - suhu)/5;
    }
    else if (suhu >= 15 && suhu <= 21){
        sejuk = 1;
    }
    return sejuk;
}

unsigned char suhuNormal(){
    if (suhu <= 21 || suhu >= 38) {
        normal = 0;
    }
    else if (suhu >= 21 && suhu <= 26){
        normal = (suhu - 21)/5;
    }
    else if (suhu >= 33 && suhu <= 38){
        normal = (38 - suhu)/5;
    }
    else if (suhu >=26 && suhu <=33){
        normal = 1;
    }
    return normal;
}

unsigned char suhuHangat(){
    if (suhu <= 33 || suhu >= 50) {
        hangat = 0;
    }
}

```



```

    }
    else if (suhu >= 33 && suhu <= 38){
        hangat = (suhu - 33)/5;
    }
    else if (suhu >= 45 && suhu <= 50){
        hangat = (50 - suhu)/5;
    }
    else if (suhu >=38 && suhu <=45){
        hangat = 1;
    }
    return hangat;
}

unsigned char suhuPanas(){
    if (suhu >= 50 ){
        panas = 1;
    }
    else if (suhu >= 45 && suhu <= 50){
        panas = (suhu - 45)/5;
    }
    else if ( suhu <= 45){
        panas = 0;
    }
    return panas;
}

unsigned char suhuDingin2(){
    if (suhu2 <= 10){
        dingin2 = 1;
    }
    else if (suhu2 >= 10 && suhu2 <= 15){
        dingin2 = (15 - suhu2)/5;
    }
    else if (suhu2 >= 15){
        dingin2 = 0;
    }
    return dingin2;
}

```

```
unsigned char suhuSejuk2(){
    if (suhu2 <= 10 || suhu2 >= 26) {
        sejuk2 = 0;
    }
    else if (suhu2 >= 10 && suhu2 <= 15){
        sejuk2 = (suhu2 - 10)/5;
    }
    else if (suhu2 >= 21 && suhu2 <= 26){
        sejuk2 = (26 - suhu2)/5;
    }
    else if (suhu2 >=15 && suhu2 <=21){
        sejuk2 = 1;
    }
    return sejuk2;
}
```

```
unsigned char suhuNormal2(){
    if (suhu2 <= 21 && suhu2 >= 38 ) {
        normal2 = 0;
    }
    else if (suhu2 >= 21 && suhu2 <= 26){
        normal2 = (suhu2 - 21)/5;
    }
    else if (suhu2 >= 33 && suhu2 <= 38){
        normal2 = (38 - suhu2)/5;
    }
    else if (suhu2 >=26 && suhu2 <=33){
        normal2 = 1;
    }
    return normal2;
}
```

```
unsigned char suhuHangat2(){
    if (suhu2 <= 33 || suhu2 >= 50) {
        hangat2 = 0;
    }
    else if (suhu2 >= 33 && suhu2 <= 38){
        hangat2 = (suhu2 - 33)/5;
    }
}
```

```

    }
    else if (suhu2 >= 45 && suhu2 <= 50){
        hangat2 = (50 - suhu2)/5;
    }
    else if (suhu2 >=38 && suhu2 <=45){
        hangat2 = 1;
    }
    return hangat2;
}

unsigned char suhuPanas2(){
    if (suhu2 >= 50 ){
        panas2 = 1;
    }
    else if (suhu2 >= 45 && suhu2 <= 50){
        panas2 = (suhu2 - 45)/5;
    }
    else if ( suhu2 <= 45){
        panas2 = 0;
    }
    return panas2;
}

void fuzzyfikasi(){
    suhuDingin();
    suhuSejuk();
    suhuNormal();
    suhuHangat();
    suhuPanas();
    suhuDingin2();
    suhuSejuk2();
    suhuNormal2();
    suhuHangat2();
    suhuPanas2();
}

//Pembentukan aturan fuzzy dan Komposisi aturan
void rule(){
    fuzzyfikasi();

```

```
rule1 = 150;
mrule1 = min(dingin, dingin2);

rule2 = 150;
mrule2 = min(dingin, sejuk2);

rule3 = 200;
mrule3 = min(dingin, normal2);

rule4 = 200;
mrule4 = min(dingin, hangat2);

rule5 = 255;
mrule5 = min(dingin, panas2);

rule6 = 150;
mrule6 = min(sejuk, dingin2);

rule7 = 150;
mrule7 = min(sejuk, sejuk2);

rule8 = 200;
mrule8 = min(sejuk, normal2);

rule9 = 200;
mrule9 = min(sejuk, hangat2);

rule10 = 255;
mrule10 = min(sejuk, panas2);

rule11 = 200;
mrule11 = min(normal, dingin2);

rule12 = 200;
mrule12 = min(normal, sejuk2);

rule13 = 255;
mrule13 = min(normal, normal2);
```

```
rule14 = 255;
mrule14 = min(normal, hangat2);

rule15 = 255;
mrule15 = min(normal, panas2);

rule16 = 200;
mrule16 = min(hangat, dingin2);

rule17 = 200;
mrule17 = min(hangat, sejuk2);

rule18 = 255;
mrule18 = min(hangat, normal2);

rule19 = 255;
mrule19 = min(hangat, hangat2);

rule20 = 255;
mrule20 = min(hangat, panas2);

rule21 = 255;
mrule21 = min(panas, dingin2);

rule22 = 255;
mrule22 = min(panas, sejuk2);

rule23 = 255;
mrule23 = min(panas, normal2);

rule24 = 255;
mrule24 = min(panas, hangat2);

rule25 = 255;
mrule25 = min(panas, panas2);

}
```

```

//Defuzzifikasi
void defuzzifikasi(){
    rule();
    float jumlah, jumlah2;

    jumlah = (mrule1 * rule1) + (mrule2 * rule2) + (mrule3 *
    rule3) + (mrule4 * rule4) + (mrule5 * rule5)+ (mrule6 *
    rule6)+ (mrule7 * rule7)+ (mrule8 * rule8)+ (mrule9 *
    rule9)+ (mrule10 * rule10)+ (mrule11 * rule11)+ (mrule12
    * rule12)+ (mrule13 * rule13)+ (mrule14 * rule14)+
    (mrule15 * rule15)+ (mrule16 * rule16)+ (mrule17 *
    rule17)+ (mrule18 * rule18)+ (mrule19 * rule19)+ (mrule20
    * rule20)+ (mrule21 * rule21)+ (mrule22 * rule22)+
    (mrule23 * rule23)+ (mrule24 * rule24)+ (mrule25 *
    rule25);

    jumlah2 = (mrule1 + mrule2 + mrule3 + mrule4 + mrule5 +
    mrule6 + mrule7 + mrule8 + mrule9 + mrule10 + mrule11 +
    mrule12 + mrule13 + mrule14 + mrule15 + mrule16 + mrule17
    + mrule18 + mrule19 + mrule20 + mrule21 + mrule22 +
    mrule23 + mrule24 + mrule25);

    out = jumlah/jumlah2;
}

void nilaiSuhu(){
    while (Serial.available() > 0)
    {
        char recieved = Serial.read();
        inData += recieved;

        if(inData == "DIS")
        {
            analogWrite(pwm, 0);
            analogWrite(fan, 0);
            rgb(0,0,0);
        }

        if (recieved == '#')

```

```
{
  rgb(0,0,0);
  int count;
  count ++;
  if (count==1)
  {
    inData.remove(inData.length() - 1, 1);
    suhu = inData.toFloat();
  }

  else if (count==2)
  {
    inData.remove(inData.length() - 1, 1);
    suhu2 = inData.toFloat();
    Serial.println(out);
  }

  else if (count==3)
  {
    inData.remove(inData.length() - 1, 1);
    suhuPackage = inData.toFloat();

    if (suhuPackage <=10){
      rgb(0,255,255);
    }
    else if (suhuPackage <=21){
      rgb(0,0,255);
    }
    else if (suhuPackage <=33){
      rgb(0,255,0);
    }
    else if (suhuPackage <=45){
      rgb(255,255,0);
    }
    else if (suhuPackage <=90){
      rgb(255,0,0);
    }
  }
}
```

```

        nilaiPWM();
        inData = "";
    }
}

void nilaiPWM(){
    defuzifikasi();
    analogWrite(pwm, out);
    analogWrite(fan, out);
}

void rgb(int nilai_red, int nilai_green, int nilai_blue)
{
    analogWrite(red, 255 - nilai_red);
    analogWrite(green, 255 - nilai_green);
    analogWrite(blue, 255 - nilai_blue);
}

void setup() {
    Serial.begin(9600);
    pinMode(pwm, OUTPUT);
    pinMode(fan, OUTPUT);
    pinMode(red, OUTPUT);
    pinMode(green, OUTPUT);
    pinMode(blue, OUTPUT);
    rgb(0,0,0);
}

void loop() {
    nilaiSuhu();
}

```

Lampiran 2

Listing Program *Software Interface* Menggunakan Microsoft Visual Studio

```

using System;
using System.Collections.Generic;

```



```

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using OpenHardwareMonitor.Hardware;
using System.IO.Ports;

namespace Final_Suhu_CPU_Arduino
{
    public partial class Form1 : Form
    {
        static string data;
        static string data1;
        Computer c = new Computer()
        {
            GPUEnabled = true,
            CPUEnabled = true
        };

        float nilai, nilai1, nilai2;

        private SerialPort port = new SerialPort();

        public Form1()
        {
            InitializeComponent();
            Init();
        }

        private void timer2_Tick(object sender, EventArgs
e)
        {
            label10.Text = DateTime.Now.ToString("HH:mm");
            label11.Text = DateTime.Now.ToString("ss");
            label12.Text = DateTime.Now.ToString("ddd");
        }
    }
}

```

```

        label13.Text = DateTime.Now.ToString("dd MMM
yyyy");
    }

    private void gaugeControl1_Click(object sender,
EventArgs e)
    {

    }

    private void Form1_Load(object sender, EventArgs e)
    {
        timer2.Start();
        c.Open();
    }

    private void gunaMetroTrackBar1_Scroll(object
sender, ScrollEventArgs e)
    {
        gunaMetroTrackBar1.Minimum = 15;
        this.Opacity = gunaMetroTrackBar1.Value * 0.01;
    }

    private void gunaAdvenceButton1_Click(object
sender, EventArgs e)
    {
        try
        {
            if (!port.IsOpen)
            {
                port.PortName = gunaComboBox1.Text;
                port.Open();
                timer1.Interval = 1000;
                timer1.Enabled = true;
            }
        }

        catch (Exception ex)
        {

```

```

        MessageBox.Show(ex.Message);
    }
}

private void gunaAdvenceButton3_Click(object
sender, EventArgs e)
{
    try
    {
        port.Write("DIS");
        port.Close();
        label2.Text = "0";
        label3.Text = "0 V";
        arcScaleNeedleComponent1.Value = 0;
        label4.Text = "0 °C";
        label16.Text = "0 °C";
        label17.Text = "0 °C";
        arcScaleNeedleComponent5.Value = 0;
        arcScaleNeedleComponent4.Value = 0;
        arcScaleNeedleComponent6.Value = 0;
        label15.Text = "0 %";

    }

    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    timer1.Enabled = false;
    data = "";
}

private void Status()
{
    float totalLoad = 0, load1 = 0, load2 = 0;
    float totaltemp = 0, core1 = 0, core2 = 0;
    byte count = 0, count1=0;

    foreach (var hardware in c.Hardware)

```

```

        {
            if(hardware.HardwareType ==
HardwareType.CPU)
            {
                hardware.Update();
                foreach (var sensor in
hardware.Sensors)
                    if (sensor.SensorType ==
SensorType.Temperature)
                        {
                            count1++;
                            if (count1==3)
                                {
                                    totaltemp =
sensor.Value.GetValueOrDefault();

arcScaleNeedleComponent1.Value =totaltemp;
                                    label4.Text =
string.Format("{0:0} °C", totaltemp);

                                }
                            else if (count1 == 1)
                                {
                                    core1 =
sensor.Value.GetValueOrDefault();
                                    label16.Text =
string.Format("{0:0}°C", core1);
                                }
                            else if (count1 == 2)
                                {
                                    core2 =
sensor.Value.GetValueOrDefault();
                                    label17.Text =
string.Format("{0:0}°C", core2);
                                }
                        }
        }
    }

```

```

else if (sensor.SensorType ==
SensorType.Load)
    {
        count++;
        switch (count)
        {
            case 1:
                load1 =
sensor.Value.GetValueOrDefault();

arcScaleNeedleComponent5.Value = load1;
                break;
            case 2:
                load2 =
sensor.Value.GetValueOrDefault();

arcScaleNeedleComponent4.Value = load2;
                break;
            case 3:
                totalLoad =
sensor.Value.GetValueOrDefault();

arcScaleNeedleComponent6.Value = totalLoad;
                label5.Text =
string.Format("{0:0.00} %", totalLoad);
                break;
        }
    }
}

try
{
    port.Write(core1 + "#");
    port.Write(core2 + "#");
    port.Write(totaltemp + "#");
    data1 = port.ReadLine();
    double hasil = Convert.ToDouble(data1,
System.Globalization.CultureInfo.InvariantCulture);

```

```

        label2.Text = hasil.ToString();
        double teg = hasil / 255 * 100;
        double hteg = Math.Round(teg / 100 * 14.50,
1);
        label3.Text = hteg.ToString() + " V";

    }
    catch (Exception ex)
    {
        timer1.Stop();
        MessageBox.Show(ex.Message);
    }
}

private void timer1_Tick(object sender, EventArgs
e)
{
    Status();
}

private void Init()
{
    try
    {
        port.Parity = Parity.None;
        port.StopBits = StopBits.One;
        port.DataBits = 8;
        port.Handshake = Handshake.None;
        port.RtsEnable = true;
        string[] ports = SerialPort.GetPortNames();

        foreach (string port in ports)
        {
            gunaComboBox1.Items.Add(port);
        }
        port.BaudRate = 9600;
    }

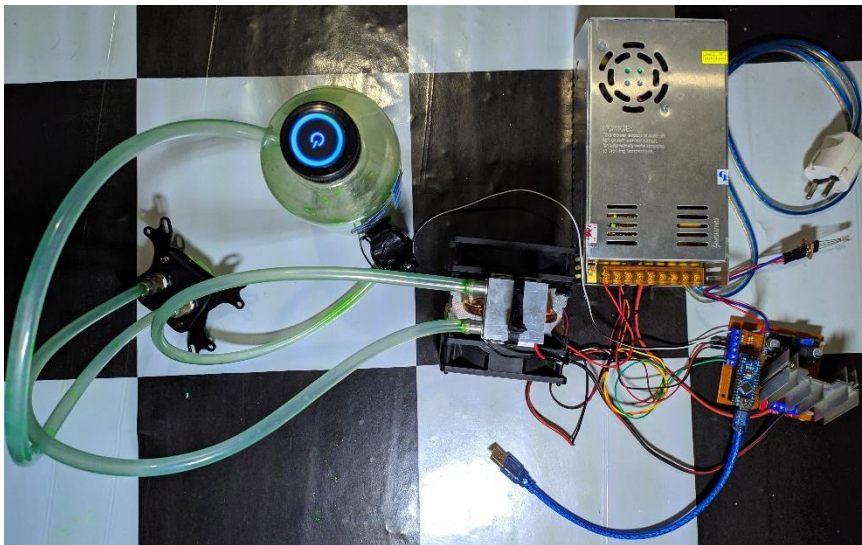
    catch (Exception ex)

```

```
        {  
            MessageBox.Show(ex.Message);  
        }  
    }  
  
    private void gunaAdvenceButton2_Click(object  
sender, EventArgs e)  
    {  
        Close();  
    }  
  
    private void gunaAdvenceButton4_Click(object  
sender, EventArgs e)  
    {  
        this.WindowState = FormWindowState.Minimized;  
    }  
  
    }  
}
```

Lampiran 3

Pengabungan Alat



Lampiran 4

Kartu Bimbingan

KARTU BIMBINGAN SKRIPSI





Semester Gasal/Genap Tahun Akademik/.....

Nama : RIZKY PRATAMA PUTRA	Pembimbing I : RAKHMAT KURNIAWAN, M. Kom
NIM : 0701172060	Pembimbing II : ABDUL HALIM HASUGIAN, M. Kom
Prog. Studi : ILMU KOMPUTER	SK Pembimbing :

Judul Skripsi :
 Penerapan Logika fuzzy pada Perancangan Cooling Komputer
 Otomatis Menggunakan Pelatier Berbasis Mikrokontroler ATMEGA
 328

P E R T	PEMBIMBING I			PEMBIMBING II		
	Tgl.	Materi Bimbingan	Tanda Tangan	Tgl.	Materi Bimbingan	Tanda Tangan
I	14/6/2021	Perbaikan Bab I		18/6-21	Perbaikan Bab I, II, III	
II	18/6/2021	Perbaikan Bab II		23/6-21	Acc Bab I, II, III	
III	21/6/2021	Perbaikan Bab III		29/6-21	Perbaikan Bab IV, V	
IV	24/6/2021	Penyempurnaan Bab III		4/7-21	Acc Bab IV, V lengkap	
V	30/6/2021	Acc Bab I, II, III		11/7-21	Beleas lengkap skripsi	

Buku Laporan Kegiatan Akademik Mahasiswa Fakultas SAINTEK UIN-SU Medan | 28

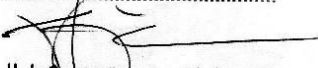
VI	29/ 9-21	Revisi Bab <u>IV</u>				
VII	4/ 10-21	Revisi BAB <u>V</u>				
VIII	11/ 10-21	Revisi Abstrak dan kesimpulan				
IX	25/ 10-21	ACC semua BAB				
X	25/ 10-21	ACC skrang				

Medan, 25 Oktober 2021

An. Dekan

Ketua Jurusan/Program Studi

ILMU KOMPUTER



ILKA ZULFIRA, M. Kom

NIP. 198506042015031006

Catatan: Pada saat bimbingan, kartu ini harus diisi dan ditandatangani oleh pembimbing

Lampiran 5

DAFTAR RIWAYAT HIDUP (*CURRICULUME VITAE*)



DATA PRIBADI

Nama : Rizky Pratama Putra
NIM : 0701172060
Tempat, Tanggal Lahir : Medan, 19 Januari 1999
Jenis Kelamin : Laki-laki
Alamat : Jl. Binjai Gg. Rasmi LK X No. 34
Kel/Desa : Sei Sikambing C II
Kecamatan : Medan Helvetia
Agama : Islam
No. HP : 085372711573
Alamat Email : rizkypratama899@gmail.com
Nama Orang Tua
Ayah : ALM Pairin
Ibu : Lidiya Ariany

PENDIDIKAN FORMAL

2004 – 2010 : SD Muhammadiyah – 12 Medan
2010 – 2013 : SMP Negeri – 18 Medan
2013 – 2016 : SMK Negeri – 9 Medan
2017 – 2021 : Universitas Islam Negeri Sumatera Utara