

## BAB II

### LANDASAN TEORI

#### 2.1 TINJAUAN PUSTAKA

Penelitian tahun 2020 oleh Fajar Ramadhan Pratama dan Peby Wahyu Purnawan mengenai “Perancangan Sistem Kendali *Temperature* Suhu dan Level *Oil* Pada *Prototype* Transformator Tenaga”. Pada penelitian tersebut dibuat *prototype* sistem kendali suhu *oil* yang dapat mengendalikan temperatur suhu *oil* pada tangki transformator agar tidak terjadi *overheat* dengan mengatur kecepatan putaran kipas dan kendali level *oil* yang dapat menjaga level *oil* dalam kondisi selalu full. Sistem kendali *fuzzy* digunakan untuk mengendalikan suhu *oil* dan sistem kendali *on-off* digunakan untuk mengendalikan level *oil* pada prototipe sesuai (*rule base*) yang digunakan. Dari penelitian tersebut didapatkan kesimpulan bahwa untuk proses pengendalian temperatur suhu *oil* agar stabil sesuai *set point* dapat dilakukan dengan mengatur kecepatan putaran kipas pada tingkat *Fast* dengan suhu udara normal dan hasil yang diperoleh respons sistem dengan *delay time* 20 menit dan *steady state error* sebesar 0,61 pada *set point* 40°C dengan persen *error steady state* sebesar 1,52%. Serta untuk pengendalian level *oil* menunjukkan sesuai *setting*, pompa akan mengisi jika level *oil* di tangki transformator berkurang hingga  $\leq 26$  cm dan pompa berhenti pada level *oil* 28 cm dalam keadaan *full* [11].

Penelitian yang dilakukan oleh Muhtarom dan Sujono mengenai “Perbandingan Sistem Kendali PID Dan Kendali Logika *Fuzzy* Pada Pengendalian Suhu Sistem Pemanas Induksi”. Penelitian tersebut Pengendali PID menggunakan struktur paralel dengan *tunning* parameter Kp, Ki dan Kd yang diperoleh dengan metode heuristik (*trial and error*). Sedangkan logika *fuzzy* yang dirancang adalah *fuzzy* mamdani. Parameter yang dibandingkan adalah respons transien dan respons *steady state* sehingga dapat diketahui pengendali yang lebih tepat untuk sistem pemanas induksi. Dari hasil pengujian pada penelitian ini sistem kendali PID dengan nilai *set point* 72°C s/d 89°C didapatkan nilai *rise time* 354 s/d 465 detik, *peak time* 433 s/d 818 detik dan *error steady state* 2,08% s/d 2,52%. Sedangkan hasil pengujian menggunakan kendali logika *fuzzy* dengan nilai *set point* 72°C s/d 89°C didapatkan nilai *rise time* 444 s/d 523 detik, *peak time* 974 s/d 818 detik dan *error steady state* 0,024% s/d 0,008% [12].

Penelitian yang dilakukan oleh Irfan Naufal Ardil mengenai “Kendali *Bang-bang* Untuk Pengendalian Suhu Pada Mesin *Roaster* Kopi Otomatis. Pada penelitian tersebut menggunakan pengendali P, PI, PID, dan *bang-bang control* dimana keempat kendali tersebut diimplementasikan dengan menggunakan *software* MATLAB Sistem kendali diperlukan untuk mengendalikan suhu biji kopi. Suhu biji kopi saat proses penyangraian menjadi variabel yang dapat menentukan progres dari proses *roasting* yang sedang dilakukan. Panas dari proses penyangraian dapat dihasilkan dari dua sumber, yakni kompor dan juga blower. Sumber panas untuk mesin *roaster* sederhana dapat hanya menggunakan satu jenis sumber panas, dalam hal ini kompor. Pada penelitian tersebut hasil Pengendali P setelah dilakukan *tuning* didapatkan nilai  $K_p = 16$  dan sistem belum dapat mengikuti suhu yang diinginkan. Pengendali PI didapatkan nilai  $K_p$  dan  $K_i$  adalah 11,4 dan 0,0897 dan sistem tersebut juga masih belum dapat mendekati nilai yang diinginkan secara cepat dan juga akurat. Pengendali PID dengan menggunakan *Simulink* didapatkan nilai  $K_p=4,6$ ,  $K_i=0,0427$ ,  $K_d=66,6$  dan sistem dengan pengendali PID juga masih belum dapat mendekati suhu yang diinginkan. Pada percobaan yang telah dilakukan sebenarnya kendali PI mendapatkan nilai yang lebih baik dibandingkan dengan pengendali lainnya yang telah dilakukan. Pemilihan metode kendali juga perlu untuk memerhatikan hal-hal lain seperti harga perangkat keras yang ada di pasaran agar tetap bisa diterima oleh target pasar yaitu UMKM. Memerhatikan hal tersebut maka kendali yang mungkin untuk diimplementasikan adalah kendali *bang-bang* dengan kondisi *on-off* yang dimodifikasi. Modifikasi kondisi *on-off* diperlukan untuk mengurangi osilasi yang terjadi pada saat proses penyesuaian suhu, dan juga mengurangi efek dari *deadtime* yang ada pada mesin. Setelah melakukan modifikasi didapatkan kesimpulan dengan memanfaatkan delta suhu sebelumnya dapat mengurangi osilasi menjadi maksimal  $2,4^{\circ}\text{C}$  dari suhu referensi. Kendali *bang-bang* untuk pengendalian suhu biji kopi juga dapat digunakan pada mesin *roaster* kopi karena memiliki hasil penyangraian yang cenderung sama [13].

Penelitian yang dilakukan oleh Bagus Prihantoro, Hari Kurnia Safitri, dan Tundung Subali mengenai “Implementasi Kontrol Suhu pada Proses Penyulingan Pembuatan *Bioetanol* Berbahan *Eceng gondok* Menggunakan *Fuzzy Logic*”. Sistem ini dibuat dengan menggunakan metode pengendali *Fuzzy Logic* yang di gunakan

untuk mengontrol *heater* yang dikombinasikan dengan sensor MQ3 dan LM35. Saat suhu di atas *set point*, maka *pwm* pada *driver heater* akan turun begitupun sebaliknya. Sistem kontrol suhu ini mendapatkan hasil yakni pada suhu stabil pada angka 75- 85°C. Dengan suhu yang sesuai titik didih *bioethanol* mendapatkan hasil *Bioetanol* maksimal dengan waktu yang cepat. *Bioethanol* yang di hasilkan dengan suhu 80°C adalah 90%. Pada pengujian menggunakan metode *fuzzy* untuk mengatur *heater*. Kontrol *fuzzy* menggunakan *error range* = -60 – 60 dan *delta error* = -5 – 5. Dengan menggunakan *error* dan *delta error* tersebut didapatkan grafik respon sistem yang di harapkan yaitu grafik yang tidak memiliki *overshoot* yang tinggi walaupun respon untuk menuju *stady state* membutuhkan waktu yang lama [14].

Penelitian yang dilakukan oleh Fajri Hidayat mengenai “Implementasi *Fuzzy* Pada Sistem Pengidentifikasi Cuaca di Tempat Wisata Berbasis Arduino Uno dan *LabVIEW*”. Penelitian tersebut menggunakan metode logika *fuzzy* untuk pengambilan keputusan menentukan cuaca yang sedang terjadi yang dirancang menggunakan *LabVIEW*. Sensor DHT11 digunakan untuk membaca suhu dan kelembapan udara, BMP180 untuk membaca tekanan udara, *Arduino UNO* sebagai *mikrokontroler*. Pembuatan sistem pengidentifikasi cuaca berbasis *Arduino UNO* dan *LabVIEW* tersebut dilakukan dengan membuat perancangan sistem, perakitan sensor, pembuatan program pada *Arduino UNO* dan pengujian program. Perancangan *LabVIEW* berupa logika *fuzzy* yang digunakan, pengujian alat, dan pengambilan data. *Output* dari penelitian tersebut berupa informasi mengenai kondisi cuaca yang terjadi pada daerah tertentu seperti hujan, mendung atau cerah yang dikirimkan melalui email. Dari hasil pengujian didapatkan persentase keakuratan sistem sebesar 60% [15].

**Tabel 2.1 Tinjauan Pustaka**

| Tahun | Judul   | Penulis  | Perbedaan   |
|-------|---|--|---|
| 2020  | Perancangan Sistem Kendali <i>Temperature</i> Suhu dan Level Oil Pada <i>Prototype</i> Transformator Tenaga | Fajar Ramadhan Pratama dan Peby Wahyu Purnawan | Pada penelitian ini integrasi antara sistem kendali <i>fuzzy</i> dan sistem kendali <i>on/off</i> |
| 2019  | Perbandingan Sistem Kendali PID dan Kendali   | Muhtarom dan Sujono                            | Pada penelitian ini menganalisis perbedaan  |

|      |   |   |   |
|------|---|---|---|
|      | Logika <i>Fuzzy</i> pada Pengendalian Suhu Sistem Pemanas Induksi   |   | sistem kendali fuzzy dan PID  |
| 2021 | Kendali <i>Bang-bang</i> Untuk Pengendalian Suhu Pada Mesin <i>Roaster</i> Kopi Otomatis  | Irfan Naufal Ardila                                       | Pada penelitian ini membandingkan sistem kendali P, PI, PID dan <i>Bang-bang</i>                                |
| 2020 | Implementasi Kontrol Suhu pada Proses Penyulingan Pembuatan <i>Bioetanol</i> Berbahan Eceng gondok Menggunakan <i>Fuzzy Logic</i> | Bagus Prihantoro, Hari Kurnia Safitri, dan Tundung Subali | Pada penelitian ini menggunakan kendali <i>fuzzy</i> dan <i>mikrokontroller Arduino</i> Uno                     |
| 2018 | Implementasi <i>Fuzzy</i> Pada Sistem Pengidentifikasi Cuaca di Tempat Wisata Berbasis <i>Arduino Uno</i> dan <i>Labview</i>      | Fajri Hidayat   | Pada penelitian ini menggunakan kendali <i>fuzzy</i> dengan <i>mikrokontroller Arduino</i> Uno dan Sensor DHT11 |

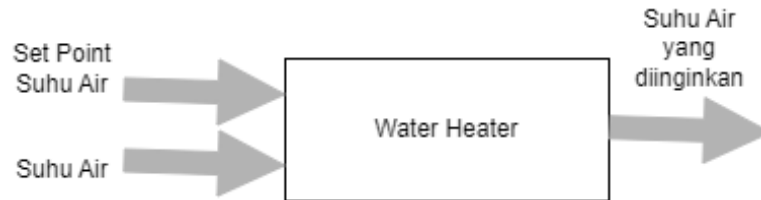
## 2.2 DASAR TEORI

Penelitian yang akan dilakukan memiliki beberapa dasar teori yang digunakan sebagai pedoman dalam melakukan penelitian. Di bawah ini terdapat beberapa dasar teori yang relevan dengan konteks penelitian yang dilakukan.

### 2.2.1 Sistem Kendali

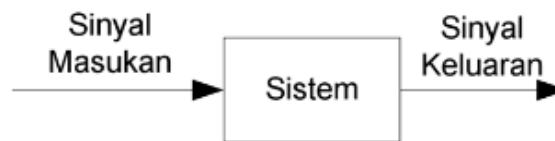
Sistem kendali merupakan suatu kumpulan dari beberapa komponen yang tersusun sedemikian rupa sehingga dapat mengatur dan menghasilkan keluaran (*output*) yang diinginkan. Sistem kendali membutuhkan sinyal masukan yang nantinya akan diproses oleh sistem sehingga menghasilkan keluaran yang diinginkan. Sistem kendali dapat memiliki lebih dari satu sinyal masukan dan sinyal keluaran.

Salah satu contoh alat yang mengaplikasikan sistem kendali adalah *Air Conditioning* yang merupakan sebuah alat yang dapat mengendalikan suhu di sekitarnya dengan cara menentukan berapa suhu yang diinginkan dan setelah beberapa saat maka suhu yang diinginkan dapat tercapai.

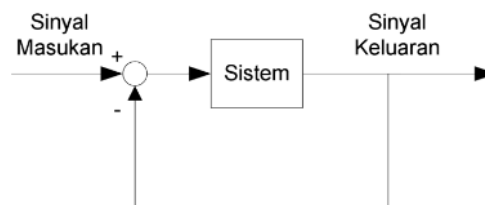


**Gambar 2.1 Sistem Kendali *Water Heater* [16]**

Dari gambar 2.1 mengenai sistem kendali *Water Heater* di atas dapat dilihat sistem membutuhkan sinyal masukan dari pengguna yang merupakan keluaran yang diharapkan. Sinyal masukan ini disebut sebagai *set point*. Sebuah sistem kendali diharapkan dapat mengendalikan keluaran sehingga nilai *set point* dapat tercapai.



**Gambar 2.2 Sistem Kendali *Open Loop* [16]**



**Gambar 2.3 Sistem Kendali *Close Loop* [16]**

Terdapat 2 jenis sistem kendali yaitu sistem kendali *open loop* dan sistem kendali *close loop* dimana keduanya dibedakan dari sinyal *input*-nya. Untuk sistem kendali *open loop* sinyal keluaran tidak digabung dengan sinyal masukan menjadi sinyal *input*. Sedangkan pada sistem kendali *close loop*, sinyal keluaran digabung dengan sinyal masukan sehingga sistem kendali *close loop* mempunyai *input* gabungan dari sinyal masukan dan sinyal keluaran.

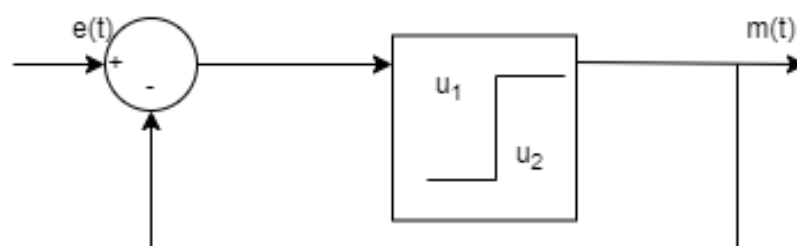
Sistem kendali *open loop* mempunyai kelebihan yaitu lebih sederhana dibandingkan dengan sistem lainnya. Kebanyakan sistem kendali *open loop*

menggunakan *timer* atau pengatur waktu untuk mengatur sistemnya. Namun, sistem ini memiliki kelemahan, yaitu karena tidak adanya informasi dari keluaran maka sistem tidak mengetahui apakah keluarannya sudah sesuai dengan yang dikehendaki.

Sistem kendali *close loop* mempunyai kelebihan, yaitu keluaran yang mendekati keluaran yang diharapkan, sistem lebih stabil meski ada perubahan masukan. Hal ini karena sistem kendali *close loop* menerima informasi dari sinyal keluaran. Akan tetapi sistem kendali *close loop* memiliki kekurangan dimana sistem ini lebih rumit dibandingkan sistem kendali *open loop* [16].

### 2.2.2 Sistem Kendali *Bang-bang*

Sistem kendali *Bang-bang* adalah sejenis pengontrol umpan balik, yang artinya sistem ini termasuk ke dalam sistem *close loop* sebab-akibat. Sistem kendali *bang-bang* merupakan sistem kendali yang hanya memeriksa variabel proses yang melebihi atau turun di bawah *set point* yang telah ditentukan, oleh karena itu sistem kendali ini juga dikenal dengan sistem kendali *on/off* karena *output* variabel yang dimanipulasi dari pengontrol dengan cepat beralih antara sepenuhnya "aktif" dan sepenuhnya "mati" tanpa keadaan perantara. Sistem kendali kasar ini biasanya memberikan kendali yang sangat tidak tepat terhadap variabel proses. Sistem kendali *bang-bang* akan mengontrol variabel proses "bersiklus" antara *set point* atas (*Upper Set Point*) dan bawah (*Lower Set Point*) tanpa pernah stabil pada *set point* yang ditentukan [17].



**Gambar 2.4 Diagram Sistem Kendali *Bang-bang* [18]**

Sistem kendali *bang-bang* merupakan sistem kendali yang sangat sederhana dan biayanya tidak mahal, sehingga sangat banyak digunakan pada sistem kendali yang ada di industri maupun di kehidupan sehari-hari. Pada gambar 2.4 dapat dilihat bahwa :

$$m(t) = u_1 \text{ untuk } e(t) > 0 \text{ atau posisi "on" atau}$$

$$m(t) = u_2 \text{ untuk } e(t) < 0 \text{ atau posisi "off"}$$

Dimana :

$e(t)$  = masukan/*input* pada saat  $t$

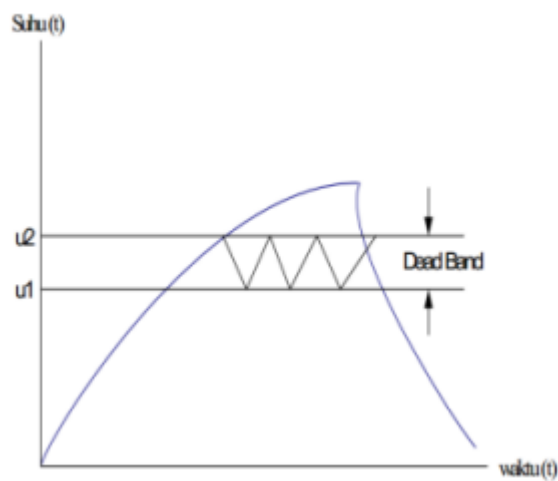
$m(t)$  = keluaran/*output* pada saat  $t$  (kondisi *on/off*)

$u_1$  = *upper set point*

$u_2$  = *lower set point*

Dengan  $u_1$  dan  $u_2$  konstan.

Salah satu alat yang menggunakan prinsip sistem kendali *bang-bang* adalah Termostat rumah versi dulu. Ketika suhu turun ke titik setel rendah yang telah ditentukan sebelumnya, termostat mengaktifkan sistem pemanas. Ketika suhu mencapai titik setel tinggi yang telah ditentukan sebelumnya, termostat mematikan sistem pemanas. Bergantung pada titik setel, keadaan ini dapat menghasilkan kisaran suhu yang dapat diterima yang cukup luas dan menyebabkan sistem pemanas memiliki waktu respons yang cukup lama. Inilah sebabnya mengapa pengontrol *bang-bang* kadang-kadang disebut sebagai pengontrol histeresis, kata histeresis menjelaskan kelambatan dalam menanggapi perubahan.



**Gambar 2.5 Kurva Sistem Kendali Histeresis [19]**

Sistem kendali histeresis adalah sistem kendali *on-off* yang mempunyai *dead band* pada suatu rentang nilai batas referensi (*set point*). Contohnya ketika nilai *set point* ditentukan pada suhu tertentu kemudian dilakukan kendali *on-off* yang akan membuat nilai *set point* berada diantara nilai  $u_1$  (batas bawah) dan  $u_2$

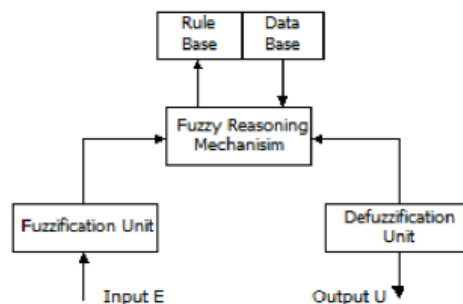
(batas atas). Suhu akan berada di dalam *dead band* karena ketika suhu melewati *set point* dan melewati nilai  $u_2$  maka sistem akan mematikan pemanas, kondisi tersebut akan membuat suhu berubah sampai pada nilai  $u_1$ . Selanjutnya ketika suhu dibawah  $u_1$  maka sistem akan menghidupkan pemanas [19].

Sistem kendali *bang-bang* mungkin cukup untuk mengontrol suhu di rumah, tetapi tidak untuk proses kimiawi yang sensitif. Tidak hanya kurangnya presisi yang menjadi masalah, tetapi seringnya siklus elemen kontrol akhir dapat menyebabkan kegagalan prematur akibat keausan mekanis. Dalam skenario penukar panas, siklus termal (panas-dingin-panas-dingin) akan menyebabkan kelelahan logam pada tabung, sehingga mempersingkat masa pakai. Selain itu, setiap ekskursi dari variabel proses di atas *set point* adalah pemborosan energi, karena fluida proses dipanaskan pada suhu yang lebih tinggi dari yang diperlukan [18].

### 2.2.3 Sistem Kendali Fuzzy

Sistem Kendali Logika Fuzzy adalah sistem kendali yang menggunakan himpunan aturan kontrol dan mekanisme pengambilan keputusan (*inference*) untuk menentukan tindakan kontrol pada sebuah *plant*. Aturan kontrol merupakan representasi yang bersifat *fuzzy*, yang menghubungkan masukan dari sistem kendali ke nilai keluaran sistem kendali. Mekanisme pengambil keputusan mengevaluasi aturan-aturan pengendali, yang kemudian menentukan aksi kontrol yang sesuai berdasarkan evaluasi tersebut.

Sistem kendali *fuzzy* merupakan jenis sistem kendali logika *close loop*. Elemen utama dari sistem kendali *fuzzy* terdiri dari fuzzifikasi, penalaran logika *fuzzy*, pengetahuan dasar *fuzzy* dan defuzzifikasi.



**Gambar 2.6 Struktur Dasar Sistem Kendali Logika Fuzzy [20]**



Pengetahuan dasar *fuzzy* terdiri dari 2 tipe informasi, yaitu data *base* yang membagi fungsi keanggotaan dari masing-masing *variable* yang digunakan, dan *rule base* yang terdiri dari sejumlah aturan *fuzzy*, aturan ini dinyatakan dalam bentuk *IF-THEN*. Variabel sistem mempunyai, yaitu variabel masukan/*input* (E) yang diukur dari proses yang dikontrol, dan variabel keluaran/*output* (U) yang digunakan oleh sistem kendali logika *fuzzy* untuk mengontrol proses. Sedangkan penalaran logika *fuzzy* sendiri menggunakan logika *fuzzy* yang sama dengan cara pengambilan keputusan dari sisi manusia. Nilai *input* ke sistem kendali dan nilai keluaran dari sistem kendali berbentuk himpunan *crisp*, proses fuzzifikasi mengubah himpunan *crisp* kedalam himpunan *fuzzy*, sedangkan proses defuzzifikasi memetakan himpunan *fuzzy* kedalam himpunan *crisp*.

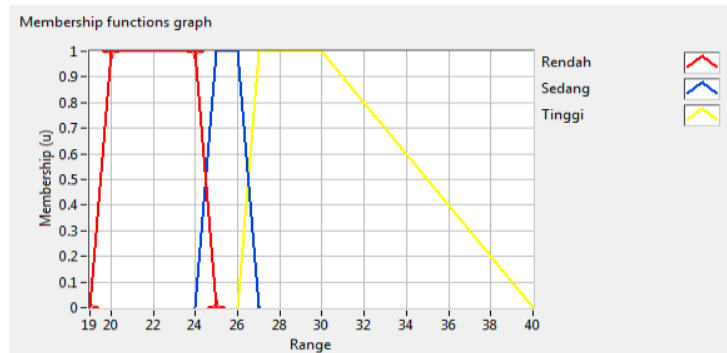
Himpunan *crisp* sendiri merupakan himpunan yang membedakan elemen yang termasuk ke dalam himpunan dan elemen yang tidak termasuk ke dalam anggota dengan batas yang jelas. Himpunan *crisp* merupakan nilai keanggotaan suatu elemen  $y$  dalam suatu himpunan  $B$ , yang sering ditulis dengan  $\mu_B(y)$  dan hanya ada 2 nilai keanggotaan ( $\mu$ ) yaitu nilai keanggotaan “1” jika elemen  $y$  adalah anggota himpunan  $B$ , ditulis  $y \in B$ , atau nilai keanggotaan “0” jika elemen  $y$  adalah bukan anggota himpunan  $B$ , ditulis  $x \notin A$ . Himpunan *fuzzy* adalah himpunan-himpunan yang terdapat pada suatu variabel sistem *fuzzy* yang mewakili sebuah nilai misalnya temperatur : sangat dingin, dingin, normal, hangat, panas dan sangat panas [20].

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menggambarkan pemetaan titik-titik data baik itu *input* atau *output* ke dalam nilai-nilai keanggotaannya. Untuk mendapatkan nilai keanggotaan, dapat dilakukan dengan menggunakan metode pendekatan fungsi.

Dalam sistem ini, pembagian *rules* logika *fuzzy* terdiri dari aturan-aturan yang berperan sebagai fungsi implikasi. Aturan-aturan ini dibuat dengan tujuan untuk menghubungkan antara nilai *input* dan nilai *output* dari sistem kendali logika *fuzzy* [15].

Salah satu contoh penggunaan *fuzzy* adalah untuk membuat sistem prediksi kondisi cuaca dengan menggunakan *fuzzy*. Sistem tersebut memiliki 3 *input* yaitu

suhu, kelembapan, dan tekanan udara, serta 1 *output* yaitu kondisi cuaca yang diprediksi berdasarkan kombinasi *rule* dari ke-3 *input*-nya



**Gambar 2.7 Contoh Fungsi Keanggotaan *Fuzzy LabVIEW* [15]**

Gambar 2.7 merupakan contoh fungsi keanggotaan *fuzzy* pada *software LabVIEW*, dimana pada gambar 2.6 adalah fungsi keanggotaan dari suhu yang memiliki 3 fungsi keanggotaan, yaitu fungsi keanggotaan suhu rendah dengan *range* dari 19 sampai dengan 25°C, fungsi keanggotaan suhu sedang dengan *range* dari 24 sampai dengan 27°C, dan fungsi keanggotaan suhu tinggi dengan *range* dari 26 sampai dengan 40°C.



**Gambar 2.8 Pembuatan *Rule Logika Fuzzy* [15]**

Gambar 2.8 merupakan contoh *rule logika fuzzy* yang dibuat pada *software LabVIEW*, dimana terdapat 3 *input* yaitu suhu, kelembapan, dan tekanan udara yang dihubungkan dengan operator “AND”, sedangkan untuk *output*-nya sendiri terdapat 1 *output* yaitu kondisi cuaca[15].

#### 2.2.4 NI-DAQ USB 6008

National Instruments USB-6008 merupakan perangkat penerima dan pengolah data digital to analog dan analog to digital yang memiliki 8 saluran input analog (AI), 2 saluran output analog (AO), 12 saluran digital input/output (DIO), dan counter 32-bit dengan interface USB kecepatan penuh.



**Gambar 2.9 NI-DAQ USB 6008 [21]**

*National Instruments* USB-6008 merupakan perangkat penerima dan pengolah data *digital to analog* dan *analog to digital* yang memiliki 8 saluran *input* analog (AI), 2 saluran *output* analog (AO), 12 saluran *digital input/output* (DIO), dan *counter* 32-bit dengan *interface* USB kecepatan penuh. NI-DAQ USB-6008 menyediakan fungsionalitas dasar untuk aplikasi seperti pencatatan data sederhana, pengukuran portabel, dan eksperimen lab akademik. Perangkat ini dilengkapi penutup mekanis yang ringan dan bertenaga bus untuk portabilitas yang mudah. Anda dapat dengan mudah menyambungkan sensor dan sinyal ke USB-6008 dengan konektivitas terminal sekrup. *Driver NI-DAQmx* dan utilitas konfigurasi yang disertakan menyederhanakan konfigurasi dan pengukuran. Penggunaan perangkat ini terbilang mudah karena menggunakan pemrograman berbasis grafis dan dikombinasikan dengan *software* NI *LabVIEW* [21].

Jenis *analog to digital converter* (ADC) yang digunakan pada NI-DAQ USB 6008 adalah *Successive Approximation Register* (SAR) dengan resolusi 12bits. *Successive-Approximation-Register* (SAR) *Analog-to-Digital Converter* (ADC) mengambil sinyal analog kontinu ke dalam format diskrit menggunakan register aproksimasi. *Analog to Digital Converter* (ADC) jenis *Successive Approximation Register* (SAR) menggunakan algoritma biner untuk mengubah bentuk gelombang analog kontinu menjadi keluaran diskrit yang jauh lebih kecil. Ini berarti setiap titik sinyal masukan analog diperiksa untuk melihat apakah lebih besar atau lebih kecil dari titik referensi digital. Jika lebih besar, bit digital disetel ke *high*, atau 1 dalam biner, kemudian dipindahkan ke bit berikutnya. Jika rendah,

bit digital disetel ke *low*, atau 0 dalam biner, kemudian dipindahkan ke bit berikutnya. Ini berlanjut hingga konversi titik sinyal masukan analog selesai. [21]

### 2.2.5 Pemanas DC 12V

Pemanas DC 12V adalah suatu alat yang memiliki elemen pemanas listrik yang dapat digunakan untuk memanaskan air. Sumber panas yang dihasilkan pada pemanas DC 12V didapat dari belitan kawat yang memiliki ketahanan listrik yang tinggi sehingga kawat tersebut tidak terbakar meskipun terus menghasilkan panas ketika diberi tegangan.



**Gambar 2.10 Pemanas DC 12V**

Dalam penelitian ini, elemen pemanas yang digunakan adalah pemanas air *portable* yang dapat dikendalikan oleh rangkaian pengendali suhu dengan sumber tegangan DC sebesar 12 Volt dengan spesifikasi tegangan *input*: 12V, dimensi pemanas :12,6 cm kabel 80 cm, dan daya 120 W[22].

### 2.2.6 Modul Driver BTS7960

Modul *driver* BTS7960 merupakan modul *driver* dengan *chip driver full H-bridge* BTS7960 ganda yang memiliki perlindungan untuk arus berlebih yang dapat mengakibatkan *overheating* sehingga modul ini dapat digunakan untuk arus sampai 43A.



nilai temperatur di rentang  $-55^{\circ}$  hingga  $150^{\circ}$ . Sumber tegangan LM35 menggunakan catu daya tunggal dengan tegangan kerja antara 4V sampai 20V dan konsumsi arus sebesar  $60\mu\text{A}$  [24].



**Gambar 2.13 Sensor LM35 Waterproof [25]**

Sensor Suhu LM35 memiliki 3 *pin* yang masing-masing mempunyai fungsi tertentu, yaitu *pin* 1 (VCC) berfungsi untuk mensupply catu daya untuk tegangan kerja dari sensor LM35, *pin* 2 atau *pin* tengah (*Vout*) berfungsi sebagai tegangan keluaran dengan jangkauan kerja dari 0V sampai dengan 1,5V, *pin* 3 berfungsi sebagai *ground*.

Berikut ini merupakan beberapa karakteristik dari sensor LM35 :

1. Sensor ini dikalibrasi langsung dalam satuan  $^{\circ}\text{C}$ .
2. Memiliki skala linier dengan perubahan tegangan sebesar  $+10\text{mV}/^{\circ}\text{C}$ .
3. Memiliki keakurasian sekitar  $0,5^{\circ}\text{C}$  pada suhu  $+25^{\circ}\text{C}$ .
4. Dapat membaca rentang suhu dari  $-55^{\circ}\text{C}$  hingga  $+150^{\circ}\text{C}$ .
5. Cocok digunakan untuk aplikasi sistem jarak jauh (*remote*).
6. Tegangan Operasi 4V sampai 20V
7. Konsumsi arus kurang dari  $60\mu\text{A}$ .
8. *Self-heating* yang rendah.
9. Ketidaklinearitasan sekitar  $\pm \frac{1}{4}^{\circ}\text{C}$ .
10. Memiliki impedansi keluaran yang rendah, sekitar 0,1 W untuk 1 mA [25].

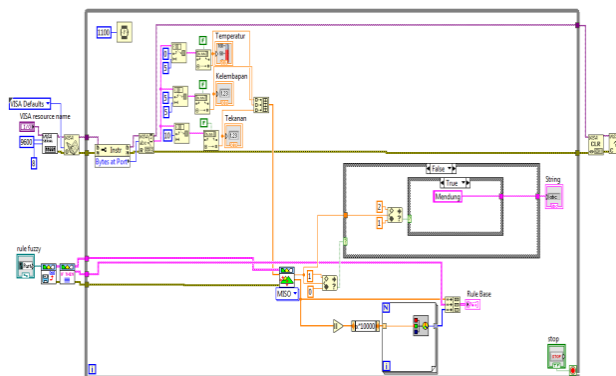
### 2.2.8 Software LabVIEW

Laboratory Virtual Instrument Engineering Workbench atau disingkat LabVIEW adalah software pemrograman yang dirancang oleh perusahaan National Instrument. Software ini difungsikan untuk pengolahan dan visualisasi data dalam berbagai bidang seperti kendali instrumentasi, akuisisi data, dan automasi industry.



**Gambar 2.14 Software LabVIEW**

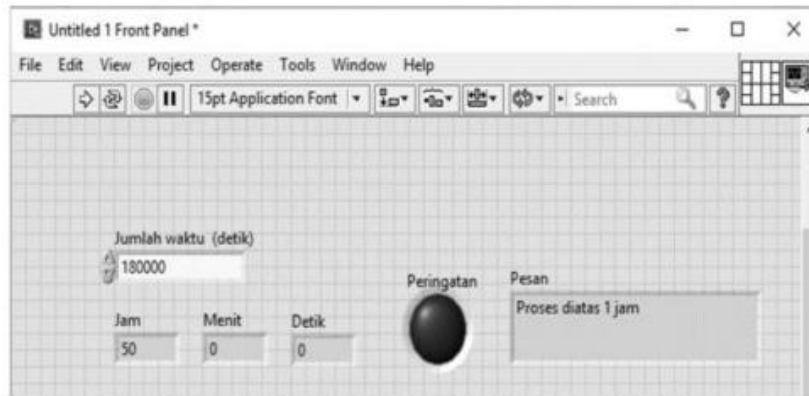
Pemrograman pada *LabVIEW* berdasarkan pada penggunaan *graphical* atau gambar atau dikenal sebagai Bahasa pemrograman G untuk perintah-perintahnya.



**Gambar 2.15 Blok Diagram Rule Base Fuzzy Pada LabVIEW [15]**

Penggunaan bahasa pemrograman G tergolong Bahasa pemrograman yang mudah sehingga dapat dipelajari oleh seseorang yang tidak mempunyai pengetahuan dasar pemrograman. *LabVIEW* menyediakan banyak sekali *library* dari fungsi atau *virtual instrument*. Selain itu *LabVIEW* juga menyediakan *library* yang khusus untuk aplikasi-aplikasi tertentu seperti untuk akuisisi data, simulasi dari sistem kendali, dan lain sebagainya [16]. Program pada *LabVIEW* disebut sebagai *Virtual Instrument (VI)* yang terdiri dari 3 bagian yaitu *front panel*, *block diagram* dan *function dan control pallete*.

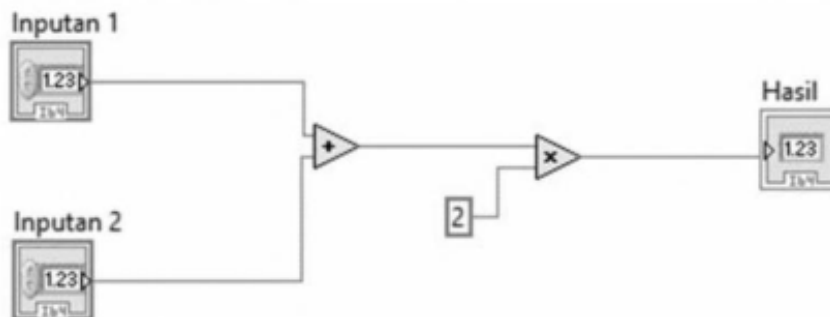
## 1. *Front Panel*



**Gambar 2.16 *Front Panel*** [26]

*Front panel* merupakan sebuah *form* di mana pengguna merancang tampilan antar muka dari proyek yang akan dibuat. *Front panel* menyediakan berbagai objek seperti *knob*, *button*, *graph*, indikator LED dan yang lainnya. Objek-objek tersebut dibagi ke dalam dua jenis yaitu kontrol yang merupakan objek *input*/masukan dan indikator yang merupakan objek *output*/keluaran.

## 2. *Block Diagram*

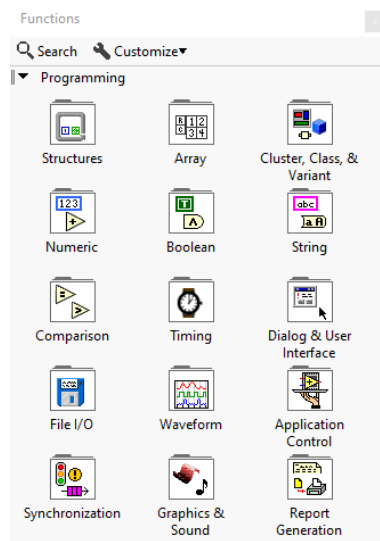


**Gambar 2.17 *Block Diagram*** [26]

*Source code* dalam bentuk gambar dibuat pada sebuah *form* yang disebut *block diagram*. *Block diagram* berisi berbagai objek seperti terminal, *subVI* (*Virtual Instrument*), struktur data, konstanta, fungsi, dan kabel penghubung. Objek-objek yang terdapat pada *front panel* akan otomatis muncul sebagai terminal pada *block diagram*. Terminal merupakan titik masuk dan keluar yang menjadi tempat pertukaran informasi antara *block diagram* dan *front panel* [26].

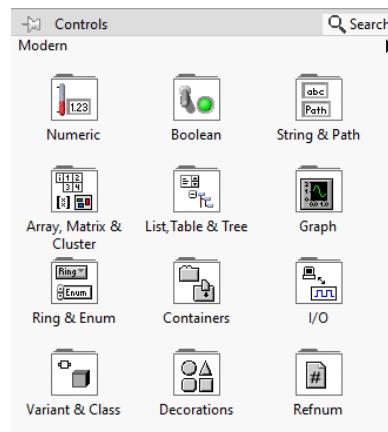


### 3. *Function dan Controls Pallete*



**Gambar 2.18 Fuctions Pallete**

*Function Pallete* merupakan sebuah *pallet* yang berada pada *block diagram* dan berfungsi untuk membangun sebuah *block diagram*.



**Gambar 2.19 Controls Pallete**

*Control Pallete* merupakan tempat beberapa berbagai objek yaitu *control* dan indikator pada *front panel*.

#### 2.2.9 Akurasi, Error dan Mean Square Error

*Error* persen dapat digunakan untuk mengecek kesalahan dari hasil percobaan yang dilakukan, *error* persen memiliki rumus sebagai berikut :

$$Error\% = \left| \frac{\text{Hasil Percobaan} - \text{Nilai Seharusnya}}{\text{Nilai Seharusnya}} \right| \times 100\%$$

$$\text{Akurasi} = 100\% - \text{Rata-rata error \%}$$

*Mean Square Error* atau disingkat MSE adalah jumlah kuadrat *error* (selisih nilai aktual pembacaan sensor dengan nilai *set point*) dibagi dengan jumlah periode. MSE merupakan metrik yang digunakan untuk mengukur baik tidaknya performa sebuah sistem dalam beroperasi terhadap nilai *set point* yang telah ditentukan. Semakin kecil nilai MSE maka semakin baik sebuah sistem karena penyimpangan atau kesalahannya semakin kecil [27].

Rumus MSE adalah sebagai berikut :

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)$$

Dimana :

$y_i$  = nilai aktual pembacaan sensor

$\hat{y}_i$  = nilai *set point*

N = Jumlah data percobaan

#### **2.2.10 Pulse Width Modulation (PWM)**

*Pulse Width Modulation* (PWM) merupakan sebuah teknik untuk memanipulasi lebar sinyal dalam bentuk pulsa selama suatu periode sehingga menghasilkan tegangan rata-rata yang berbeda. PWM digunakan untuk mendapatkan hasil sinyal analog dari sebuah sinyal digital. Kendali digital digunakan untuk membangkitkan gelombang kotak, sinyal yang berpindah antara kondisi tinggi (*high/on*) dan rendah (*low/off*), karena pada dasarnya PWM hanya memiliki dua kondisi, yaitu sinyal *high* (1) dan sinyal *low* (0). Salah satu fungsi PWM sendiri adalah untuk mengontrol tegangan atau daya yang masuk ke beban. Pada PWM terdapat beberapa istilah yaitu *Duty Cycle* yang merupakan perbandingan antara lebar sinyal *high* dengan lebar keseluruhan siklus (*Cycle*), Amplitudo yang merupakan besar nilai sinyal saat mencapai keadaan *high* [28].

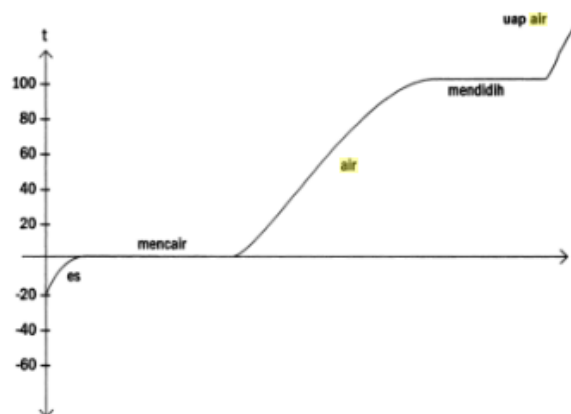


**Gambar 2.20 PWM dengan *Duty Cycle* 50% [29]**

Pada Gambar 2.20 terlihat perbandingan antara sinyal *high* dan sinyal *low* pada suatu siklus sama maka gelombang di atas memiliki *duty cycle* 50%. Apabila nilai amplitudo gelombang PWM tersebut adalah 12V, maka tegangan rata-rata yang dihasilkan adalah 6V ( $12 \times 50\%$ ) [28].

### 2.2.11 Suhu Panas Air

Air merupakan senyawa yang dapat mengalami perubahan wujud dari beku (es), mencair (air), dan mendidih (uap air) dimana hal tersebut dipengaruhi oleh berbagai hal.



**Gambar 2.21 Grafik Perubahan Wujud Air Murni [30]**

Pada gambar 2.21 terlihat titik didih normal air murni adalah pada temperatur 100°C, yang mana proses pemanasan air dalam keadaan cair sampai pada keadaan mendidih berawal dari temperatur 0-100°C. Berdasarkan catatan dari Organisasi Kesehatan Dunia (WHO) bakteri mati dengan cepat pada suhu di atas 149°F atau 65°C, yang mana suhu ini di bawah suhu air mendidih. [30].