

BAB 2

DASAR TEORI

2.1 KAJIAN PUSTAKA

Penelitian Mohd Iqbal, Dr.K.A.Khan yang dilakukan pada tahun 2017 dengan judul “ *Simulation of Water level Control in a Tank Using Fuzzy Logic in Matlab*” melakukan perancangan simulasi *control* ketinggian air menggunakan *Fuzzy Logic* Matlab yang mengusulkan menggunakan pengontrol logika *fuzzy* dengan menggunakan metode mamdani [1]. Pada penelitian ini logika *fuzzy* sebagai fungsi akurat waktu dengan mendeteksi ketinggian air yang mencapai tingkat ketinggian sesuai yang diinginkan. *Input* yang digunakan pada penelitian ini hanya dua, *input* pertama dengan nama *level*, *input* kedua dengan nama *rate* dan *output* yang digunakan empat *output*, *output* pertama rendah, kedua lambat terbuka, ketiga tidak berubah, dan ke empat terbuka cepat. *Control* ketinggian air menggunakan lima *rule* pada *rule editor* matlab, *rule* pertama *If level* okay maka katup tidak ada perubahan, *rule* kedua *If level* positif maka *valve* akan terbuka cepat, *rule* ketiga *If level* negative maka *valve* akan tertutup cepat, *rule* keempat *If level* okay dan rata-rata positif maka katupnya akan tertutup lambat, dan *rule* ke lima *If level* okay dan rata-rata negative maka *valve* akan terbuka pelan. Hasil penelitian ini membuktikan bahwa pengontrol logika *fuzzy* melakukan pekerjaannya dengan cukup baik dibandingkan mengendalikan sistem lainnya.

Penelitian yang dilakukan oleh Bourdilon Odianonsen Omijen, Promise Elechi, dan Mathew Enikhamenle pada tahun 2015 dengan judul penelitian “ *Simulated Design of Water level Control System* “ meneliti tentang bagaimana sistem *control* air digunakan untuk mengontrol *level* dan sirkulasi air untuk berbagai jenis proses, proses ini biasanya dipakai di bendungan air, pabrik petrokimia, pabrik pengolahan air, pabrik olahan makanan, dan pabrik farmasi [2]. Penelitian ini menggunakan *software* Simulink , sistem desainnya terdiri dari tangki, katup, pengontrol logika *fuzzy*, *input* dan *output*. Pengontrol logika *fuzzy* di dalam *penulisan* ini bekerja dengan *level input*nya yang akan menghasilkan *output* atau *level* air yang diinginkan, penambahan *input* pada pengontrol logika *fuzzy* bertujuan untuk membedakan *level* tangki. *Rule* yang digunakan oleh logika *fuzzy*

akan memberikan keuntungan dalam penulisan ini karena mengetahui berapa jumlah *rule* yang digunakan pada sistem ini untuk mendapatkan *level* ketinggian air yang bagus. Untuk sistem *control level* ketinggian air berisi kan blok diagram simulasi. Penelitian ini yang mencakup keterkaitan satu sama lain antara sistem, sistem *control level* tangki air memiliki dua *input* ke multiplexer yang memiliki fungsi laju aliran keluar yang diperoleh dengan membedakan tingkat *output*, *output* dari multiplexer diumpankan ke pengontrol logika *fuzzy rule* viewer. Pada *penulisan* ini *rule* yang diuji coba menggunakan 3 *rule*, 4 *rule*, 5 *rule* hingga 6 *rule*, tetapi untuk hasil *rule* yang dipakai pada *penulisan* ini yang digunakan 5 *rule* dikarenakan bentuk respon controlnya tidak menghasilkan overshoot.

Penelitian Dendi Supriadi pada tahun 2015 dengan judul penelitian “Rancang Bangun Sistem Pengendalian Ketinggian Air Menggunakan Sensor Ultrasonic Berbasis PLC (*Programmable Logic Controller*)” meneliti bahwa sensor ultrasonic memiliki keuntungan yang lebih baik dibandingkan dengan menggunakan sensor lain untuk pengendalian ketinggian air, dikarenakan pada hasil pengukuran lebih presisi, kemudian pada saat pengukuran level ketinggian air sensor tanpa harus bersentuhan dengan airnya, Sedangkan tegangan analog yang dihasilkan digunakan oleh PLC untuk mengelola data dan membaca ketinggian air [3].

Penulisan Apip Pudir pada tahun 2013 dengan judul penelitian “Pengendalian *Level* Cairan Tangki Dengan Menggunakan Simulasi *Fuzzy* Berbasis Matlab” meneliti bahwa pengontrol logika *fuzzy* untuk pengendalian *level* cairan pada tangki lebih baik dibandingkan menggunakan pengontrol lainnya seperti pengontrol PID [4]. Metode pada penelitian ini menggunakan metode sugeno yang keluaran sistem berupa konstanta konstanta yang berbeda. Sensor *level* akan mendeteksi ketinggian air yang akan di konversi ke dalam sinyal yang dapat diterima oleh controller. Controller akan membandingkan antara sinyal yang terukur dengan nilai set point atau yang mendekati yang akan mendapatkan nilai kesalahan (*error*) dan perubahan kesalahan (*derror*) yang dipakai untuk *input fuzzy*. Dalam pengontrol *fuzzy* terdapat fungsi keanggotaan untuk fungsi keanggotaan *error* menggunakan variable linguistik yaitu negative, nol, dan positif dengan nilai data tegas (*crisp*) antara 1 dan -1 sedangkan untuk perubahan *error* fungsi

keanggotannya negatif, nol dan positif dengan data tegas antara -0.1 dan 0.1. Model Simulink pengendali sistem tangki memiliki tinggi 2 m dan luas alas 1 m², luas penampang outlet 0.05 m² integritas satuan 0 untuk menyatakan *valve* dalam keadaan tertutup dan integritas 1 dalam keadaan terbuka *valvenya*. Model Simulink pengendalian PID rentang waktu yang dipakai t=0 sampai t =10 detik dengan kondisi ini mengasumsikan kecepatan dalam pengisian air tidak memperhitungkan volume tangki.

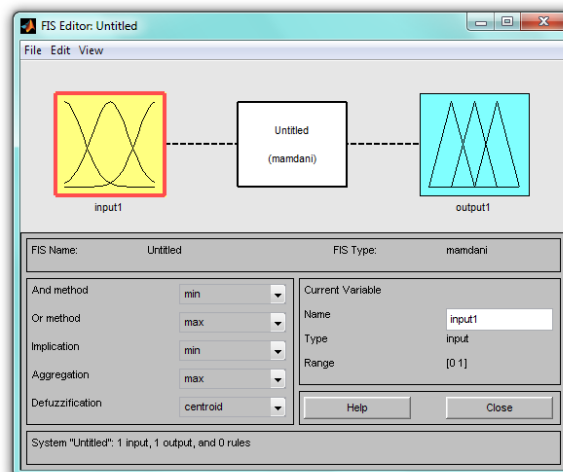
Pada tahun 2012 penelitian yang dilakukan oleh Disha, Mr. Pawan Kumar Pandey dan Rajeev Chugh yang berjudul “*Simulation of Water level Control in a Tank Using Fuzzy Logic*” melakukan perancangan simulasi *fuzzy logic controller* untuk *level* ketinggian air dengan menggunakan *fuzzy logic toolbox* dan Simulink pada matlab [5]. Dalam menentukan desain sistem yang digunakan benar-benar berhasil hal yang harus diperhatikan tentang *rule* (aturan) yang digunakan, *fuzzy* yang digunakan pada penelitian ini menggunakan *fuzzy mamdani*. Perancangan sistem ini menggunakan 3 variabel *input* dan 3 variabel *output*, variabel *input* pertama dinamakan dengan *level*, *input* kedua nilai dan *input* ketiga tekanan, *output* pertama yang diberi nama katup, *output* kedua api, dan ketiga katup uap, *input* pertama untuk *level* menggunakan fungsi keanggotaan Gaussian yang di *editor* bernama *gaussmf* dengan memiliki derajat keanggotaan 3 yaitu negatif, sedang dan positif, variabel *input* kedua untuk nilai fungsi keanggotaan menggunakan *gaussmf* dengan derajat keanggotannya negatif, nol dan positif, variabel ketiga tekanan menggunakan fungsi keanggotaan *trimf* dengan derajat keanggotaan nol, kecil, dan besar. Variabel *output* pertama katup fungsi keanggotanya menggunakan *trimf* dengan derajat keanggotannya penutupan cepat, penutupan lambat, tidak berubah, pembukaan lambat dan pembukaan cepat, variabel *output* kedua api yang memakai fungsi keanggotannya *trimf* dengan derajat keanggotaan *on* dan *off*, variabel *output* ketiga katup uap dengan jenis fungsi keanggotaan *trimf* dan derajat keanggotaan ditutup dan dibuka. *Rule* yang dipakai pada sistem ini menggunakan 5 *rule* dengan menyesuaikan jumlah *outputnya*.

2.2 DASAR TEORI

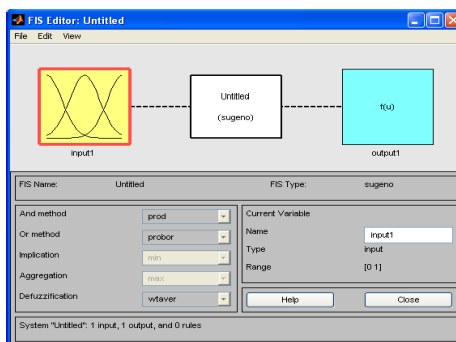
2.2.1 Control Logika Fuzzy

Sistem *fuzzy* merupakan sistem pengontrol yang berdasar pengetahuan (*Knowledge Base System*) atau berdasar aturan (*Base Rule System*). Bagian utama sistem *fuzzy* yaitu dasar pengetahuan yang terdiri atas aturan-aturan JIKA-MAKA. Dasar *control* logika *fuzzy* terdiri dari *Fuzzyfikasi* yang merupakan pemetaan ruang *input* dan ruang *output*. Basis pengetahuan, yang berisi aturan-aturan *fuzzy* yang digunakan untuk pengendalian scalar, berdasarkan logika dan intuisi manusia, Mekanisme *Fuzzy Reasoning*, decision making *logic*, inferensi *fuzzy* yang menggunakan berbagai logika *fuzzy* seperti logika AND, OR, dan NOT, *Defuzzyfikasi* merupakan proses perubahan nilai *fuzzy* ke bentuk nilai *crisp* [6].

Di dalam logika *fuzzy* terdapat 2 metode yaitu metode mamdani dan metode sugeno, metode mamdani merupakan salah satu bagian dari *fuzzy* inference sistem yang digunakan untuk penarikan kesimpulan atau suatu keputusan terbaik dalam permasalahan yang tidak pasti, sedangkan metode sugeno, metode inferensi *fuzzy* untuk aturan yang direpresentasikan dalam bentuk *IF-THEN* di mana *output* sisten tidak berupa himpunan *fuzzy* melainkan berupa konstanta atau linier.



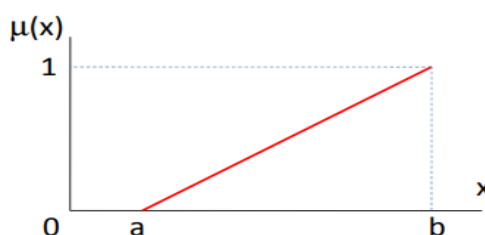
Gambar 2.1 FIS Editor Mamdani[6]



Gambar 2.2 FIS Editor Sugeno[6]

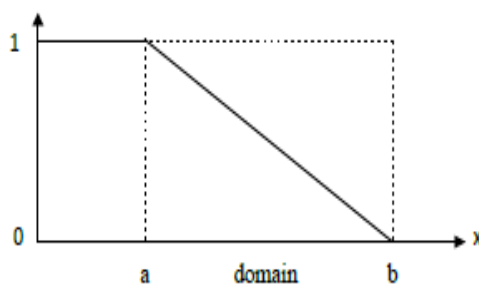
2.2.2 Fungsi Keanggotaan

Fungsi keanggotaan kesepakatan yang digunakan untuk dasar himpunan *fuzzy* pada universal U dari bilangan riil. Fungsi keanggotaan yang sering digunakan yaitu *S-function*, π -*function*, triangular, trapezoid, exponential, klir dan folger [7].



Gambar 2.3 Representasi Linear Naik[8]

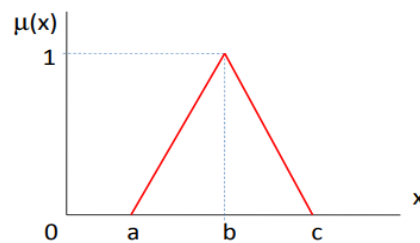
Pada gambar 2.3 kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotannya nol bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotannya lebih tinggi.



Gambar 2.4 Representasi Linear Turun[8]

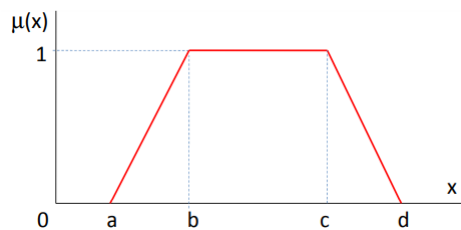
Pada gambar 2.4 garis lurus dimulai dari domain dengan derajat keanggotannya tertinggi pada sisi kiri, dan bergerak menurun ke nilai domain yang memiliki derajat keanggotannya lebih rendah.

Fungsi kedua Representasi Kurva Segitiga yang di tunjukan pada gambar 2.5, yang merupakan representasi gabungan antara dua garis *linear* pada *membership function* kurva segitiga berdasarkan jenis *type* tirmf.



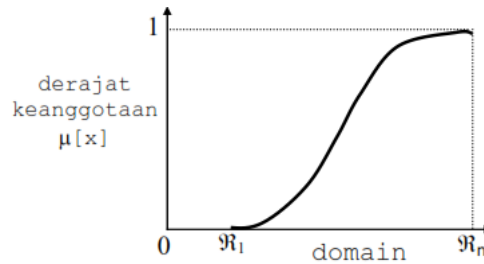
Gambar 2.5 Representasi Kurva Segitiga[8]

Fungsi ketiga Representasi Kurva Trapesium pada gambar 2.6 yang merupakan bentuk representasi kurva segitiga hanya saja ada beberapa titik yang memiliki nilai keanggotannya 1 dan pada *membership function* yang digunakan trapesium dengan *type* trapmf.



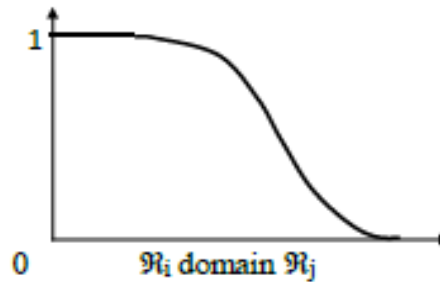
Gambar 2.6 Representasi Kurva Trapesium[8]

Fungsi keempat Representasi Kurva Bentuk S adalah kurva pertumbuhan dan penyusutan merupakan kurva-S yang berhubungan dengan kenaikan dan penurunan permukaan secara tidak *linear*.



Gambar 2.7 Himpunan *Fuzzy* dengan kurva-S Pertumbuhan[8]

Pada gambar 2.7 kurva-S bergerak dari sisi paling kiri dengan nilai keanggotanya nol ke sisi paling kanan dengan nilai keanggotannya 1 merupakan kurva-S Pertumbuhan.



Gambar 2.8 Himpunan *Fuzzy* dengan Kurva-S Penyusutan [8]

Pada gambar 2.8 kurva-S yang bergerak dari sisi paling kanan dengan nilai keanggotannya 1 ke sisi paling kiri nilai keanggotanya nol merupakan kurva penyusutan.

2.2.3 Simulink

Simulink adalah merupakan *software* matlab yang berbasis notasi diagram blok. Menggunakan Simulink untuk membuat model sistem *nonlinear* yang kompleks, model Simulink dapat menyertakan komponen kontinyu dan diskrit. Model Simulink dapat menghasilkan animasi grafis yang menunjukkan kemajuan simulasi secara visual, secara signifikan meningkatkan pemahaman tentang perilaku sistem. Di masa lalu untuk mengembangkan model computer sistem dinamik dimulai dari diagram blok diterjemahkan ke dalam bahasa pemrograman. Praktek ini melibatkan duplikasi usaha, karena

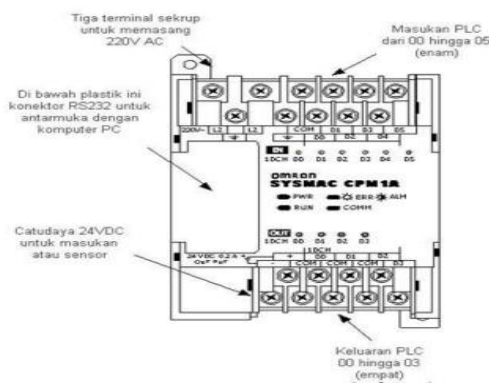
sistem dan pengontrol harus dijelaskan dua kali sekali dalam bentuk diagram blok dan dalam bahasa pemrograman [8].

Simulink memiliki resiko dalam pembuatan blok diagram yang akan dihubungkan dengan pemrograman atau sebaliknya dari pemrograman ke blok diagram, sebenarnya yang membuat *error* pada Simulink itu sendiri dari programnya sendiri karena blok diagram blok menghilangkan resiko bahwa program tidak secara akurat mengimplementasikan diagram bloknya. Di dalam Simulink memiliki banyak model sistem yang dapat digunakan sesuai yang akan digunakan misalnya, *fuzzy logic controller*, *transfer function* dan masih banyak lagi.

2.2.4 PLC (Programmable Logic Controller)

PLC merupakan suatu rangkaian elektronik yang bisa bekerja berbagai fungsi kontrol pada suatu *level-level* yang rumit. PLC juga dapat diprogram, dicontrol, dan dioperasikan oleh *user* atau operator. Pada program PLC digambarkan dengan garis pada diagram ladder. Pada PLC memiliki perintah dasar seperti, LD, LD NOT, AND, AND NOT, OR, OR NOT, OUT, OUT NOT, *timer*, dan *counter* [9].

Cara kerja PLC dengan cara menghubungkan sensor pada *input* PLC dan alat-alat apa saja yang akan dicontrol pada bagian *output* PLC. Program pada PLC akan memproses data dari *input* dan *output* akan bekerja sesuai dengan program yang telah tersimpan pada memori PLC [10].



Gambar 2.9 PLC Omron CPM1A[10]

2.2.5 OPC (OLE For Process Control)

OLE merupakan kepanjangan dari Object Linking and Embedding. OLE sendiri merupakan teknologi yang dapat menampilkan suatu objek dari program aplikasi lain, OLE sendiri dibagi 2 jenis yaitu linking sebagai hubungan antara kedua aplikasi dan Embedding sebagai memperbaharui data jika ada perubahan data.

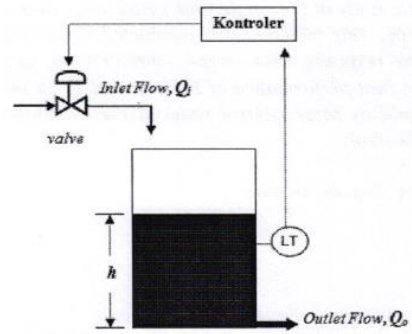
OPC merupakan suatu konektivitas yang digunakan untuk memberikan informasi secara bergantian antara *hardware* pada suatu plant dengan *software* yang digunakan untuk memantau [11]. OPC juga dirangkai untuk menjembatani berbagai *software* yang berbasis *windows* dengan perangkat maupun aplikasi dengan tujuan untuk kontrol proses .

Penggunaan OPC kebanyakan pada dunia industri otomasi yang digunakan untuk menjelaskan tentang komunikasi data real time dari suatu plant antar *device* kontrol yang berbeda-beda [11].

2.2.6 Sistem Tangki

Sistem tangki yang dilakukan pemantauan seperti gambar 2.10 dengan sistem pengendaliannya memenuhi proses *nonselv regulating* di mana persamaan untuk mencari *level* ketinggian dalam tangki ditentukan dengan persamaan:

Sistem terdiri dari katup air, tangki , sensor *level*, dan controller [4] . Ketinggian air dalam tangki dideteksi dengan sensor *level* dan dikonversi ke dalam sinyal yang dapat diterima oleh controller. Untuk mencari nilai *error* dan perubahan kesalahan (*deerror*) dengan membandingkan sinyal *level* dengan nilai *level* yang dikehendaki. Untuk mengendalikan *level* cairan dalam tangki bisa dengan cara mengatur bukaan *valve* yang dikonfigurasi sesuai dengan aturan *fuzzy sugeno*.



Gambar 2.10 Diagram Skema Pengontrol *Level* Cairan Dalam Tangki[4]

2.2.7 Arduino Uno

Arduino Uno yaitu sebuah papan board yang menggunakan mikrokontroler ATmega328. Pada Arduino Uno memiliki 14 pin *input* atau *output*, di mana 6 pin diantaranya dapat digunakan sebagai keluaran *Pulse Widht Modulation* (PWM), 6 pin *input* analog, satu akses koneksi *Universal Serial Bus* (USB), kabel power, dan tombol reset [12].

Spesifikasi Arduino Uno:

1. Mikrokontroler : ATmega328
2. Tegangan Operasi : 5V
3. Tegangan Input: 7 - 12 V
4. Pin digital I/O : 14 (6 diantaranya pin PWM)
5. Pin Analog input : 6 – Arus DC per pin I/O : 40 mA
6. Arus DC untuk pin 3.3 V : 150 mA
7. Flash Memory : 32 KB dengan 0.5 KB digunakan untuk bootloader
8. EEPROM : 1 KB
9. Kecepatan Pewaktuan : 16 Mhz [12].



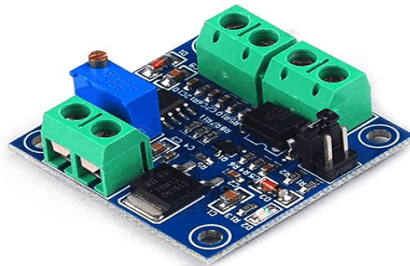
Gambar 2.11 Arduino Uno[12]

2.2.8 PWM To Voltage

PWM To Voltage adalah sistem mengubah *input* sinyal PWM 0-5V menjadi *output* analog 0-10V. Seluruh sistem dapat dialiri daya dari catu daya dc 15-24V (>500mA). Frekuensi modulasi lebar pulsa *input* yang direcomendasikan adalah 500Hz +/- 2% [13].

Spesifikasi PWM To Voltage:

1. Tegangan Kerja: DC 12V-30V; (> 100MA)
2. Frekuensi Penerima PWM: 1KHZ-3KHZ
3. Rentang level input sinyal PWM
4. Puncak level 4.5V hingga 10V
5. Puncak level 12V sampai 24V
6. Rentang konversi: 0%-100% PWM ke 0-10V
7. Kesalahan yang diizinkan: 5% [13].



Gambar 2.12 PWM To Voltage[13]

2.2.9 Sensor Ultrasonic

Sensor Ultrasonic adalah jenis sensor yang memanfaatkan sebuah pancaran gelombang. Sensor Ultrasonic terdiri dari berbagai rangkaian pemancar ultrasonic yang disebut juga transmitter dan rangkaian penerima ultrasonic atau yang sering disebut receiver *output*. Sensor Ultrasonic merupakan jenis sensor *outputnya* berupa pulsa yang mempersentasikan jarak. Cara kerja sensor ultrasonic yaitu dari sebuah pantulan suatu gelombang sehingga dapat digunakan untuk menafsirkan jarak suatu benda dengan frekuensi tertentu [3].



Gambar 2.13 Sensor Ultrasonic[3]