

# Otomatisasi Wiper Mobil Saat Hujan Berbasis Raspberry Pi Menggunakan Metode *Fuzzy Logic*

## *Automation Of The Car Wiper When Rain Based On Raspberry Pi Using Fuzzy Logic Method*

1<sup>st</sup> Alenka Kejar Adicandra  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
alenkaadicandra@student.telkomuniversity.ac.id

2<sup>nd</sup> Porman Pangaribuan  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
porman@telkomuniversity.ac.id

3<sup>rd</sup> Muhammad Hablul Barri  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
mhbarri@telkomuniversity.ac.id

**Abstrak**—Wiper merupakan alat pembersih kaca mobil depan dan belakang. Wiper membantu pengguna mobil saat keadaan hujan maupun keadaan kaca mobil kotor agar penglihatannya terfokus ke jalan dan tidak terganggu oleh air hujan atau kotoran yang terdapat dikaca mobil. Hal tersebut dapat mengganggu pengemudi yang harus mengatur gerak wiper sesuai kebutuhan. Wiper dirancang bekerja secara otomatis apabila hujan mengenai kaca mobil. Intensitas hujan dideteksi melalui adanya getaran yang timbul dikaca mobil. Untuk itu pada kaca mobil ditempelkan sensor pizeoelektrik yang mendeteksi getaran dengan keluaran sebuah sinyal analog yang diubah menjadi sinyal digital. Dalam perancangan wiper otomatis dengan metode fuzzy logic yang mengklasifikasikan kecepatan gerak wiper secara otomatis dari tingkat getaran pada kaca mobil. Sistem yang dirancang membantu pengguna kendaraan bermobil agar tetap terfokus dalam berkendara tanpa mengatur wiper ketika terjadinya hujan gerimis, sedang, dan deras, dengan ambang batas ADC ketiga sensor masing – masing kondisi hujan adalah 500 – 1300 untuk hujan gerimis, 1300 – 3100 untuk hujan sedang, dan >3100 untuk hujan deras. Percobaan kecepatan dalam RPM masing – masing untuk hujan gerimis adalah 411 rpm, hujan sedang adalah 840 rpm, dan hujan deras adalah 2600 rpm. Selain itu jarak

kabel tidak berpengaruh pada hasil nilai ADC sensor dan jarak sensor terbaik adalah 32cm.

**Kata Kunci**—wiper, fuzzy logic, raspberry pi, intensitas hujan, kecepatan gerak, kaca mobil.

*Abstract*—Wiper is a tool for cleaning the front and rear windshield. The wiper helps car users when it is raining or when the windshield is dirty so that their vision is focused on the road and is not disturbed by rainwater or dirt on the windshield. This can interfere with the driver who has to adjust the wiper motion as needed. The wiper is designed to work automatically when rain hits the windshield. The intensity of the rain is detected through the vibrations that arise in the car glass. For this reason, a pizeoelectric sensor is attached to the windshield which detects vibrations with the output of an analog signal which is converted into a digital signal. In the design of automatic wipers with fuzzy logic method which classifies the speed of movement of the wiper automatically from the level of vibration on the windshield. The system is designed to help car users stay focused on driving without adjusting the wipers when drizzling, moderate, and heavy rain occurs, with the ADC thresholds of the three sensors for each rain condition of 500 – 1300 for drizzling rain, 1300 – 3100 for moderate rain. , and >3100 for heavy rain. Experimental speed in RPM for drizzling rain is 411 rpm, moderate rain is 840 rpm, and heavy rain is 2600 rpm. In addition, the

*cable distance has no effect on the results of the ADC sensor value and the best sensor distance is 32cm.*

**Keywords**—*wiper, fuzzy logic, raspberry pi, rain intensity, movement speed, windshield.*

## I. PENDAHULUAN

*Wiper* adalah komponen bawaan wajib dari setiap mobil yang terpasang di depan dan belakang mobil memiliki bagian yang berfungsi sedemikian rupa. Pada saat musim hujan *wiper* berfungsi membersihkan kaca mobil dari air hujan. Dengan begitu kaca mobil menjadi bersih dan jarak pandang tidak terganggu. Kemudian, pada saat musim kemarau *wiper* dilengkapi dengan *washer pump* yang dapat membantu membersihkan kaca mobil dari debu.

Pengembangan teknologi yang sudah ada untuk sistem *wiper* yang bekerja secara otomatis masih banyak menggunakan sensor hujan dengan sistem *on* dan *off* saja. Hal tersebut tentunya kurang efektif dikarenakan *wiper* akan bergerak dengan satu kondisi percepatan. *Wiper* otomatis yang bekerja dengan sistem *on* dan *off* kurang efektif karena percepatannya tidak memiliki fitur untuk mengatur kecepatan sesuai intensitas hujan dan sistem hanya akan bekerja pada saat hujan saja (kondisi *on*) [9]. Penelitian lain melakukan modifikasi untuk keperluan otomatis *wiper* dan *washer*, dengan memanfaatkan sensor hujan dan debu. namun penempatan dan besar volume dari sensor debu yang dinilai kurang efektif untuk kenyamanan berkendara [10]. Jika *wiper* diatur dengan kecepatan gerak cepat dan volume sensor yang besar, tentunya akan mempengaruhi daya tahan catu daya (*accu*) pada mobil, dan mengganggu penglihatan pengendara ketika terjadinya hujan gerimis, yang dimana sistem tersebut hanya menggunakan sensor hujan, sensor hujan sendiri hanya mampu mendeteksi adanya air atau tidak pada penampang sensor, makadari itu dibuat lah sistem *wiper* yang lebih efektif.

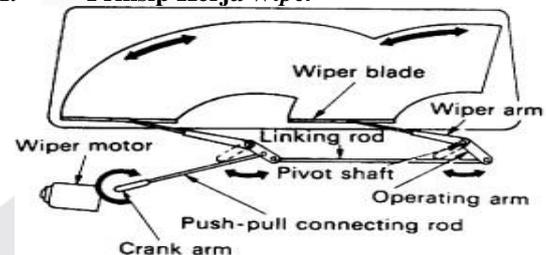
Sistem *wiper* yang digunakan pada Tugas Akhir ini menggunakan metode *fuzzy logic* yang akan membedakan intensitas hujan sehingga mampu menggerakkan motor *wiper* sesuai PWM yang akan diatur. Sistem yang dibuat dengan pengatur kecepatan gerak *wiper* yang sesuai dengan intensitas hujan

mampu membedakan antara hujan gerimis, hujan sedang maupun hujan deras, dikarenakan menggunakan sensor getar yang dimana sensor tersebut akan menerima getaran pada kaca mobil lalu akan dilakukan proses *fuzzy logic* pada mikrokomputer. Sistem tersebut akan membantu pengemudi lebih fokus ketika berkendara disaat terjadinya hujan.

Dalam Tugas Akhir ini dilakukan pengembangan dalam mengatasi masalah tersebut yaitu membuat suatu sistem *wiper* otomatis. Sistem *wiper* otomatis ini terdiri dari prototipe power window (terdiri dari *wiper* dan motor DC), *raspberry pi* (mikrokomputer), sensor getar (pizeoelektrik), dan catu daya. Alat ini akan bekerja saat sensor getar membaca getaran pada kaca mobil bergantung curah hujan yang diterima, besar nilai getaran yang terbaca akan menentukan kecepatan *wiper* saat terjadinya hujan baik curah hujan dengan intensitas kecil, intensitas sedang, maupun hujan dengan intensitas tinggi.

## II. KAJIAN TEORI

### A. Prinsip Kerja *Wiper*



GAMBAR 2.1 Prinsip kerja sistem *wiper* dan *washer*

*Wiper* adalah alat penting untuk menyapu air hujan, salju dan barang yang mengendap pada permukaan kaca depan dan belakang agar menjaga penglihatan pengemudi. *Wiper* terdiri dari *wiper blade*, *wiper arm*, *wiper motor*, dan *wiperlink* yang saling berhubungan. Sebagai alat bantu, *wiper* dilengkapi dengan *washer* (pencuci) yang memancarkan cairan (*washer liquid*) ke kaca. *Washer* terdiri dari *washer tank*, *washer motor*, *washer tube*, dan *washer nozzle*.

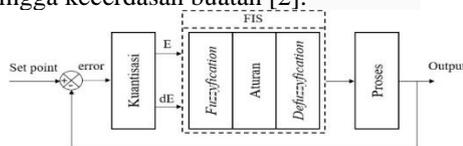
Prinsip kerja sistem *wiper* dan *washer* adalah ketika switch diposisi *mist*, *low*, *INT* atau *high* maka motor *wiper* berputar sesuai

posisi switchnya. Motor mulai memutar crank arm, Batang penghubung tarik-dorong dihubungkan dengan crank arm, menyebabkan arm bekerja untuk membuat gerak penghapusan setengah lingkaran mengelilingi poros pivot. Linking rod lain yang terpasang pada kerja arm selalu membuat gerak penghapusan setengah lingkaran secara parallel [1].

**B. Fuzzy Logic (Logika Samar)**

*Fuzzy Logic* pertama kali dikembangkan oleh Lotfi A. Zadeh melalui tulisannya pada tahun 1965 tentang teori himpunan *fuzzy*. Lotfi Asker Zadeh adalah seorang ilmuwan Amerika Serikat berkebangsaan Iran dari Universitas California di Berkeley, Meskipun logika *fuzzy* dikembangkan di Amerika, namun ia lebih populer dan banyak diaplikasikan secara luas oleh praktisi Jepang dengan mengadaptasikannya ke bidang kendali [2].

Saat ini banyak dijual produk elektronik elektronik buatan Jepang yang menerapkan prinsip logika *fuzzy*, seperti mesin cuci, AC, dan lain-lain. *Fuzzy logic* sudah diterapkan pada banyak bidang, mulai dari teori kendali hingga kecerdasan buatan [2].



GAMBAR 2.2 Struktur Logika Fuzzy

Struktur pengendalian logika *fuzzy* dapat dilihat pada Gambar 2.7. Terdapat tiga proses yang dilakukan dalam pengambilan keputusan menggunakan sistem kendali *fuzzy* yang sering disebut dengan FIS (*Fuzzy Inference System*) antara lain *fuzzyfication*, *inference*, dan *defuzzification* [3].

$$\text{trapezoid}(x; a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x - a}{b - a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d - x}{d - c}, & d \leq x \end{cases} \tag{2.1}$$

Parameter {a, b, c, d} (dengan a < b <= c < d) menentukan koordinat x dari empat sudut MF trapesium [5].

**1. Fuzzifikasi**

Fuzzifikasi merupakan tahap pembagian nilai masuk dan keluar kedalam bentuk himpunan *fuzzy*. Himpunan *fuzzy* didapat dari data masuk yang berupa himpunan *crisp* dengan *range* untuk setiap variable masuknya. Untuk menentukan himpunan *fuzzy* dari himpunan *crisp* membutuhkan dua proses fuzzifikasi yaitu memperhatikan nilai masuk dan nilai keluar serta fungsi keanggotaan (*membership function*) atau disingkat dengan MF [4].

Untuk proses fuzzifikasi digunakan bentuk fungsi keanggotaan gaussian, dipilihnya fungsi gaussian karena mempunyai tingkat keakurasian tinggi dalam membaca data. Selanjutnya proses iterasi digunakan dalam prses fuzzifikasi dengan merubah nilai *range* dan parameter untuk membangun fungsi keanggotaan serta merubah jenis fungsi keanggotaan. Gambar 2.8 adalah ilustrasi dari MF *fuzzy*.

**2. Inferensi Fuzzy**

Inferensi *fuzzy* merupakan proses pemetaan ruang *input* kedalam ruang *output* dengan menggunakan *IF-THEN rules*. Sistem yang dapat mengevaluasi semua *rule* secara simultan untuk menghasilkan kesimpulan dan urutan *rule*.

Inferensi *fuzzy* memiliki banyak metode seperti metode Mamdani, Sugeno, Tsukamoto, dan masih abnyak lagi. Metode yang digunakan adlaah metode Sugeno, Metode ini diajukan oleh Takagi, Sugeno, dan Kang pada tahun 1985 dala upaya pendekatan sistematis untuk membangkitkan atran aturan *fuzzy* dari himpunan data *input* sampai *output* yang diberikan [4]. Aturan *fuzzy* yang digunakan pada model *fuzzy* Sugeno dalam bentuk:

**3. Defuzzifikasi**

Defuzzifikasi merupakan proses terakhir dari sitem *fuzzy logic* dengan mengkonversikan setiap hasil dari *inference engine* yang ditampilkan dalam bentuk *fuzzy*

set kesuatu bilangan real, hasil dari proses defuzzifikasi merupakan aksi yang diambil dari kendali *fuzzy logic*.

Proses *input* defuzzifikasi adalah himpunan *fuzzy* yang diperoleh dari aturan *fuzzy*, sedangkan *output* yang dihasilkan berupa suatu bilangan pada domain himpunan *fuzzy* tersebut, untuk keluaran dari defuzzifikasi harus diberikan himpunan *fuzzy* dalam range tertentu dari suatu nilai *crisp* [6].

**4. Fuzzy Inference System**

Terdapat tiga metode yang populer untuk digunakan pada FIS (*Fuzzy Inference System*) seperti Metode Tsukamoto, Metode Mamdani, dan Metode Sugeno. Metode Sugeno adalah metode yang populer untuk diterapkan pada logika *fuzzy* yang mudah di implementasikan dan keluaran berupa nilai linier. Persamaan 2.2 adalah persamaan matematika dari Metode Sugeno [8].

$$x = \frac{\mu_1 z_1 + \dots + \mu_n z_n}{\mu_1 + \dots + \mu_n}$$

dimana,

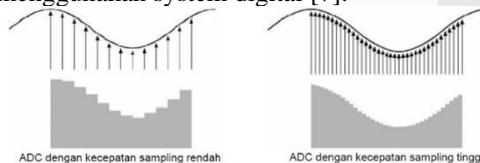
$\mu$  = MF hasil *inference*

$z$  = nilai aturan *fuzzy*

$n$  = data ke- $n$ .

**C. Analog to Digital Conversion (ADC)**

*Analog to Digital Conversion* merupakan pengubah input sinyal analog menjadi sinyal digital, *Analog to Digital Conversion* banyak digunakan di bidang komunikasi digital dan perangkat antar sensor analog dengan system computer seperti contohnya sensor cahaya, sensor tekanan, sensor cahaya dan diukur menggunakan system digital [7].



**GAMBAR 2.3** Tampilan ADC dengan kecepatan rendah dan kecepatan tinggi

*Analog to Digital Conversion* memiliki dua prinsip kerja, yaitu kecepatan sampling resolusi. Kecepatan Sampling bisa dinyatakan dalam *Sample per Second* (SPS), kecepatan sampling dalam ADC dinyatakan ketika seberapa sering sinyal analog dikonversikan dalam bentuk sinyal digital dalam selang waktu tertentu [7].

**D. Raspberry Pi**

Mikrokontroler merupakan system mikroprosesor dimana didalamnya terdapat *ROM, RAM, CPU, I/O* dan *Clock* yang berfungsi untuk mengolah dan menyimpan data. Mikrokontroler yang digunakan berupa *Raspberry Pi* yang biasa dikenal juga dengan *Raspi* merupakan komputer kecil yang sama fungsinya dengan komputer pada umumnya dan mampu menjalankan program, video dengan resolusi tinggi, dan permainan *computer* [8].



**GAMBAR 2.4** Raspberry Pi

**E. Sensor Piezoelectric**

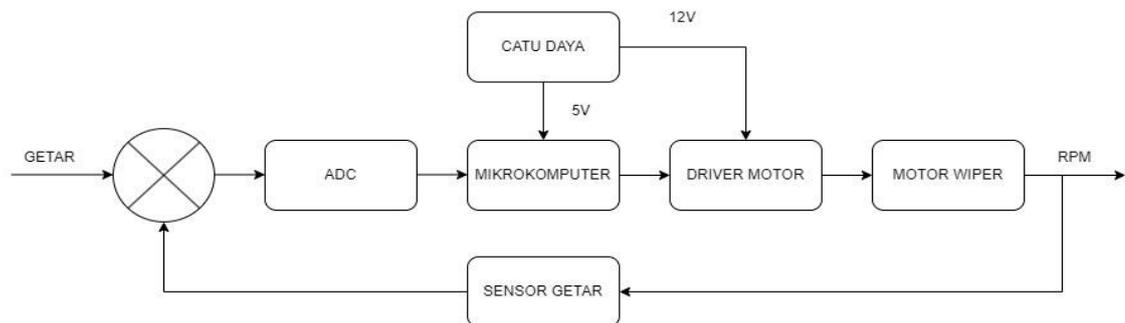
Sensor piezoelektrik adalah sensor tekanan yang memanfaatkan efek gaya tekanan untuk mengukur tegangan melintasi elemen piezoelektrik yang dihasilkan. Ketika gaya diterapkan pada bahan piezoelektrik, muatan listrik dihasilkan di seluruh permukaan kristal. Ini dapat diukur sebagai tegangan yang sebanding dengan tekanan. Sensor piezoelektrik biasanya tidak cocok untuk mengukur tekanan statis. Sinyal output secara bertahap akan turun ke nol, bahkan dengan adanya tekanan konstan. Gambar 2.11 adalah ilustrasi dari sensor piezoelektrik [11].



**GAMBAR 2.5** Sensor Piezoelektrik

**III. METODE**

**A. Desain Sistem**



GAMBAR 3.1 Diagram Blok Keseluruhan.

Pada Gambar 3.1 sistem dirancang menggunakan sensor getaran yang dipasangkan pada tiga titik pada kaca depan mobil yang dibuat menjadi prototipe seperti pada Gambar 3.2. Peletakan sensor pada Gambar 3.2 adalah pada bagian dalam. Nilai sensor dibaca oleh modul ADC yang di konfigurasi dengan mikrokomputer. Mikrokomputer mengolah nilai ADC dari sensor dengan menggunakan metode *fuzzy logic*. Kaluaran dari *fuzzy logic* adalah PWM yang diatur dengan bantuan *driver* motor yang kemudian dari *driver* motor memberikan nilai konversi PWM menjadi tegangan akan diberikan ke aktuator untuk mengatur kecepatan aktuator dalam RPM (*Revolutions Per Minute*). Komponen – komponen yang digunakan pada Gambar 3.1 memiliki fungsi sebagai berikut:

1. Sensor Getar berfungsi untuk membaca getaran kaca akibat dari kontak kaca dengan hujan. Sensor yang digunakan pada sistem ini adalah sensor piezoelektrik.
2. Modul ADC memiliki fungsi untuk melakukan konversi dari nilai analog menjadi digital. Modul ini digunakan karena pada mikrokomputer tidak tersedia ADC.
3. Mikrokomputer memiliki fungsi untuk mengolah sinyal informasi dari sensor kedalam *fuzzy logic* dan memberikan keluaran berupa PWM ke *driver* motor. *Board* mikrokomputer yang

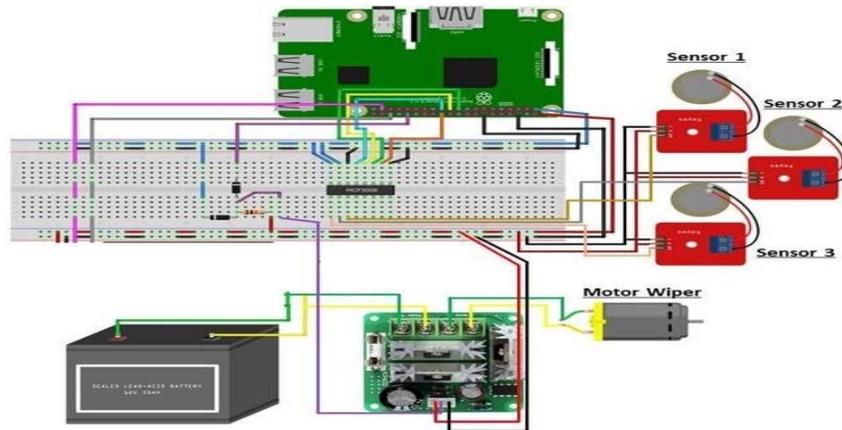
digunakan pada sistem ini adalah *Raspberry Pi*.

4. *Driver* motor memiliki fungsi untuk mengatur PWM menjadi tegangan yang berguna pada kecepatan motor.
5. Motor *wiper* digunakan sebagai aktuator yang memiliki fungsi sebagai penggerak yang dapat membersihkan kaca.
6. Catu daya memiliki fungsi untuk memberikan daya untuk mikrokomputer bekerja dan memberikan daya pada *driver* motor untuk bekerja.



GAMBAR 3.2 Prototipe Kaca Mobil

Sesuai dari penjelasan pada Gambar 3.1, sistem otomatis ini memiliki desain skematik yang diilustrasikan pada Gambar 3.3.



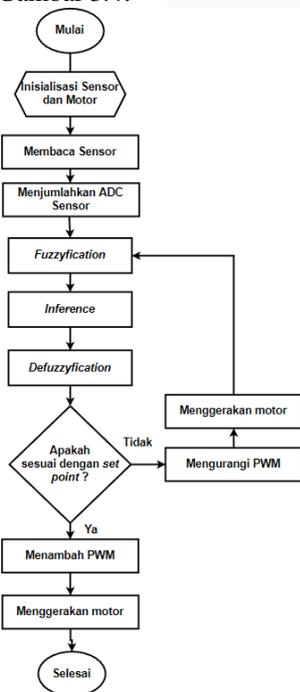
GAMBAR 3.3 Skematika Rangkaian

Pada Gambar 3.3 diatas memiliki spesifikasi komponen sebagai berikut:

1. Mikrokomputer : Raspberry Pi 3B
2. Driver Motor : DC 12V 10A
3. ADC Modul : MCP3008
4. Catu Daya : SPMS 12V 2A
5. Sensor : Sensor Piezoelektrik
6. Aktuator : Motor DC 12 power window max 100 rpm
7. Frimware : Python Versi 3.7
8. Software : Tonny Python

**B. Desain Perangkat Lunak**

Sistem ini memiliki diagram alir (*Flowchart*) sebagai Gambar 3.4.



GAMBAR 3.5 Diagram Alir Sistem

Pada Gambar 3.4 dimulai dari inialisasi sensor dan motor. Sensor yang digunakan berjumlah tiga buah. Nilai setiap sensor akan dibaca dan diakumulasi yang kemudian akan dicari nilai *setpoint*. Jika nilai *setpoint* sesuai maka akan terjadi penambahan PWM yang bertujuan untuk memberikan *trigger* pada motor agar dapat bergerak dan begitupun sebaliknya. Kemudian nilai akumulasi sensor akan masuk kedalam proses *fuzzy logic* yang terdiri dari *fuzzyfication*, *inference*, dan *defuzzyfication* untuk mengatur kecepatan motor sesuai dengan kondisi hujan gerimis, sedang, dan deras.

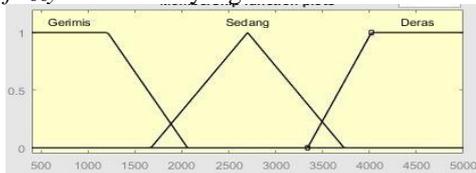
**C. Fuzzy Logic**

Fuzzy logic akan terdiri dari beberapa proses seperti *fuzzyfication*, *inference*, dan *defuzzyfication*. Setiap proses *fuzzy* akan dijelaskan pada *subsection* berikutnya.

**1. Fuzzyfication**

*Fuzzyfication* merupakan proses untuk mengubah nilai data menjadi sebuah himpunan *fuzzy*, mengubah himpunan *fuzzy* menjadi MF *fuzzy*, dan menentukan jumlah variabel linguistik yang akan digunakan. Tujuan dari proses ini adalah untuk mendapatkan MF *fuzzy* yang akan diproses

pada tahap *inference*. Gambar 3.5 adalah MF fuzzy dari sistem yang akan dibuat.



GAMBAR 3.6 MF Fuzzy Sistem

2. **Inference**

*Inference* adalah proses transformasi dari nilai-nilai masukan yang merupakan hasil dari proses *fuzzyfication* menjadi nilai-nilai keluaran dalam domain *fuzzy*. Proses transformasi ini membutuhkan aturan-aturan yang dirancang untuk memenuhi kebutuhan sistem. Aturan-aturan yang digunakan padaperancangan sistem ini dapat dilihat pada Tabel 3.1.

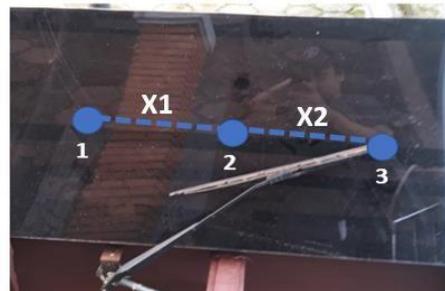
TABEL 3.1 Aturan Fuzzy Sistem

Kondisi Hujan	Kecepatan Motor		
	ADC Kecil	ADC Sedang	ADC Besar
Gerimis	Pelan	-	-
Sedang	-	Agak Cepat	-
Deras	-	-	Cepat

3. **Defuzzification**

*Defuzzification* merupakan tahap akhir. Proses ini akan memetakan nilai-nilai hasil proses *inference* menjadi suatu nilai keluaran berupa konstanta. Pada perancangan sistem ini, keluaran berupa nilai PWM.

tiga yaitu jarak 32cm, 22cm, dan 12cm dengan kondisi hujan deras saja.



GAMBAR 4.1 Jarak Sensor 1 dan 3 Terhadap Sensor 2

IV. HASIL DAN PEMBAHASAN

A. **Pengujian Pengaruh Jarak Sensor**

Pengujian ini dilakukan untuk mendapatkan posisi ketiga sensor agar sistem dapat bekerja dengan baik. Pengujian ini dilakukan dengan menempatkan ketiga sensor pada prototipe alat pada jarak yang berbeda – beda. Posisi acuan adalah posisi tengah atau sensor nomor 2 dan posisi sensor 1 dan 3 yang diubah – ubah untuk mendapatkan jarak yang terbaik seperti pada Gambar 4.1. Jarak pengujian dibagi menjadi

Pada Gambar 4.1 Jarak sensor 1 ke sensor 2 dilambangkan dengan X1 dan jarak sensor 3 ke sensor 2 dilambangkan dengan X2. Variabel X1 dan X2 adalah variabel jarak yang akan diubah nilainya pada jarak 32cm, 22cm, dan 12cm. Data pengujian disajikan pada Tabel 4.1.

TABEL 4.1 Pengujian Jarak Sensor Pada Hujan Deras

No.	Jarak Sensor								
	X1 & X2 = 32cm			X1 & X2 = 22cm			X1 & X2 = 12cm		
	S1	S2	S3	S1	S2	S3	S1	S2	S3
1.	3390	4660	3430	3809	4680	3901	4321	4510	4432
2.	3370	4530	3160	3908	5310	3801	4322	5260	4452
3.	3760	5200	3360	3890	4940	3990	4366	4940	4732
4.	3530	4980	3500	3900	4330	3901	4567	4330	4632
5.	3180	4380	3320	3899	4690	3890	4327	4660	4532
6.	3430	4700	3280	3809	5110	3801	4221	5250	4432
7.	3810	5310	3640	3998	4650	3890	4563	4880	4321

8.	3490	4940	3420	3980	4500	3999	4456	4500	4213
.....									
100	3180	4380	3640	3910	4600	3981	4489	4631	4331

Data pada Tabel 4.1 adalah nilai ADC pada setiap sensor. Nilai ADC yang dihasilkan oleh sensor dikalikan dengan 10 untuk memperbesar nilai ADC. Pada Tabel 4.1 terlihat semakin menjauh dari titik tengah getaran yang diterima oleh sensor akan semakin mengecil, yang artinya nilai ADC yang dihasilkan akan semakin kecil.

Setiap jarak sensor memiliki standar deviasi masing – masing. Pada jarak sensor 32 cm memiliki nilai standar deviasi masing – masing sensor, pada sensor 1 adalah 206,9, sensor 2 adalah 320,3, sensor 3 adalah 158,2. Pada jarak sensor 22 cm memiliki nilai standar deviasi masing – masing sensor, pada sensor 1 adalah 77,3, sensor 2 adalah 282,2, dan sensor 3 adalah 61,1. Selain itu pada jarak sensor 12 cm memiliki nilai standar deviasi masing – masing sensor, pada sensor 1 adalah 146,8, sensor 2 adalah 316,9, dan sensor 3 adalah 123,8.

Semakin besar nilai standar deviasi, maka akan semakin lebar rentang variasi datanya. Sehingga penempatan terbaik adalah pada jarak 32 cm, hal ini dikarenakan nilai ADC yang dihasilkan memiliki variasi data yang lebih banyak dan agar lebih merata dalam pembacaan nilai getaran hujan pada kaca prototipe.

### B. Pengujian Panjang Kabel Terhadap Kinerja Sensor

Pengujian panjang kabel digunakan untuk melihat pengaruh panjang kabel terhadap hasil sensor yang dihasilkan. Pengujian ini dilakukan dengan menggunakan panjang kabel pada sensor yang berbeda – beda yaitu 1 m, 1,5 m, dan 2 m dengan menggunakan satu kondisi hujan yaitu hujan deras. Pada Tabel 4.2 menyajikan hasil nilai ADC pada pengujian panjang kabel.

TABEL 4.2 Pengujian Panjang Kabel Pada Hujan Deras

No	Jarak Kabel								
	P = 1m			P= 1,5m			P=2m		
	S1	S2	S3	S1	S2	S3	S1	S2	S3
1.	3390	4760	3430	3390	4660	3430	3350	4560	3330
2.	3270	4630	3260	3370	4530	3160	3170	5230	3260
3.	3560	5210	3340	3260	5200	3360	3530	4200	3150
4.	3520	4920	3300	3530	4980	3500	3460	5380	3430
5.	3120	4350	3350	3180	4380	3320	3480	4380	3440
6.	3410	4670	3290	3430	4700	3280	3560	4800	4280
7.	3340	5430	3340	3810	5310	3140	3530	5410	3650
8.	3467	4900	3310	3490	4740	3420	3400	4360	3330
9.	3391	4210	3530	3390	4660	3430	3400	4160	3560
10.	3388	4330	3180	3370	4530	3160	3180	4170	3280

Keterangan: S1 (Sensor 1); S2 (Sensor 2); S3 (Sensor 3); P (Panjang Kabel)

Pada Tabel 4.2 nilai ADC yang dihasilkan ketiga sensor dengan panjang kabel 1m, 1,5m, dan 2m terlihat perbedaan yang dihasilkan. Hal itu dapat dilihat rata – rata nilai untuk masing masing pengujian adalah 38.198,67 untuk panjang kabel 1m, 38.370 untuk panjang kabel 1,5m, dan 38.473,33 untuk panjang kabel 2m. Namun

selisih rata – rata nilai ADC pada Tabel 4.2 memiliki selisih tidak lebih dari 400 nilai ADC, yang artinya panjang kabel tidak berpengaruh pada nilai ADC sensor. Hal itu karena noise yang dihasilkan memiliki rentang hingga 400 pada nilai ADC yang dikalikan 10. Hal tersebut dilakukan karena nilai sensor yang kecil makadari itu dilakukan pengalihan 10 kali pada ADC di mikrokomputer agar menampilkan nilai sensor yang besar dan dapat dihitung nilainya

dengan mudah. Penggunaan jarak kabel 1,5m bertujuan agar dapat aman dari sebaran air.

### C. Pengujian Ambang Batas Masing – Masing Kondisi Hujan

Pengujian ini dilakukan dengan menggunakan penyemprot air yang dikondisikan seperti hujan deras, sedang, dan

gerimis. Punguan ambang batas ini digunakan untuk mengambil nilai ADC rata – rata ketiga sensor terbaik untuk kondisi hujan gerimis, sedang, dan deras yang digunakan untuk MF *fuzzy*. Tabel 4.3 sampai dengan Tabel 4.5 adalah data nilai ADC sensor ketiga kondisi.

TABEL 4.3 Pengujian Curah Hujan Gerimis

No	Sensor 1	Sensor 2	Sensor 3
1.	1070	1390	690
2.	750	830	1320
3.	520	690	900
4.	1170	1200	690
5.	790	1100	1320
6.	800	690	900
7.	1010	1320	690
8.	700	900	1320
9.	500	761	900
.....			
100	678	571	851

TABEL 4.4 Pengujian Curah Hujan Sedang

No	Sensor 1	Sensor 2	Sensor 3
1.	2456	2546	3210
2.	1853	2808	2368
3.	1573	2978	2346
4.	2001	2769	2790
5.	1456	2534	2865
6.	1874	2684	3213
7.	2452	3012	3211
8.	2684	2351	2697
9.	2354	2786	2007
.....			
100	1547	2689	2790

TABEL 4.5 Pengujian Curah Hujan Deras

No	Sensor 1	Sensor 2	Sensor 3
1.	3390	4660	3430
2.	3370	4530	3160
3.	3760	5200	3360
4.	3530	4980	3500
5.	3180	4380	3320
6.	3430	4700	3280
7.	3810	5310	3640
8.	3490	4940	3420
9.	3390	4660	3430

100	3388	4330	3180
-----	------	------	------

Jika nilai pecilan diabaikan maka akan didapat nilai ambang batas untuk masing – masing kondisi hujan adalah 500 – 1300 untuk hujan gerimis, 1300 – 3100 untuk hujan sedang, dan >3100 untuk hujan deras. Nilai ambang batas pada masing – masing kondisi hujan diperoleh dengan mencari nilai minimum dan maksimum pembacaan sensor.

#### D. Pengujian Curah Hujan Terhadap RPM dan PWM Motor

Pengujian kecepatan *power window* dalam RPM dan hubungannya dengan PWM berdasarkan curah hujan gerimis, sedang, deras dengan tatacara mensimulasikan terjadinya hujan menggunakan pompa air yang diatur *muzzleselang* agar mendapatkan curah hujan yang diinginkan disajikan pada Tabel 4.6 sampai 4.8.

**TABEL 4.6** Pengujian PWM dan RPM Pada Hujan Gerimis

No	Rata – Rata ADC	PWM (%)	Rotasi motor
1.	1070	50	650,7 RPM
2.	750	50	592,7 RPM
3.	520	50	468,7 RPM

**TABEL 4.7** Pengujian PWM dan RPM Pada Hujan Sedang

No	Rata – Rata ADC	PWM (%)	Rotasi motor
1.	2315	70	840,2 RPM
2.	2356	70	1028 RPM
3.	2586	70	1269 RPM

**TABEL 4.8** Pengujian PWM dan RPM Pada Hujan Deras

No	Rata – Rata ADC	PWM (%)	Rotasi motor
1.	3635	90	2063 RPM
2.	4246	90	2096 RPM
3.	4537	90	2607 RPM

Dari Tabel 4.5 sampai dengan Tabel 4.8 diatas terlihat pada hujan gerimis dengan intensitas hujan rendah menghasilkan ADC dari 520 – 1070 dengan PWM 50% menghasilkan RPM 468,7 – 650,7 untuk hujan gerimis. Pada hujan sedang dengan intensitas hujan sedang menghasilkan ADC dari 2315 – 2586 dengan PWM 70% menghasilkan RPM 840,2– 1269. Dan pada hujan deras dengan kondisi intensitas hujan deras menghasilkan ADC dari 3635 – 4537 dengan PWM 90% menghasilkan RPM 2063– 2607.

#### V. KESIMPULAN

Berdasarkan hasil pengujian dan analisis, kesimpulan dari Tugas Akhir ini seagai berikut:

1. Jarak pada sensor berpengaruh terhadap getaran yang diterima oleh sensor pizeoelektrik, seperti halnya semakin jauh jarak sensor atau penempatan sensor pada kaca yang digunakan, getaran akan semakin kecil nilainya.
2. Penggunaan panjang kabel tidak berpengaruh terhadap kinerja sensor, dikarenakan noise yang dihasilkan rentangnya tidak lebih dari 400 pada nilai ADC.
3. Masing – masing nilai ADC yang dihasilkan dari pengujian ambang batas kondisi hujan gerimis menghasilkan rentang 500 – 1300, hujan sedang menghasilkan 1300 – 3100 dan hujan deras menghasilkan >3100, jika nilai ADC nya rendah

maka getaran yang dihasilkan juga sedikit atau intensitas hujannya juga rendah yang menandai hujan gerimis, jika nilai ADC nya meningkat maka getaran yang diterimapun akan semakin banyak menandai intensitas hujan meningkat sehingga terjadinya hujan sedang, dan jika nilai ADC nya sangat besar maka getaran yang diterimapun akan semakin banyak yang menandai intensitas hujan yang diterimapun semakin besar yang menandai terjadinya hujan deras.

4. Nilai RPM yang dihasilkan dari masing masing kondisi dipengaruhi oleh nilai PWM dan ADC yang dihasilkan, jika nilai ADC nya rendah maka nilai PWM nya pun akan rendah dan menghasilkan nilai RPM yang rendah seperti hujan gerimis yang menghasilkan ADC dari 520 – 1070 dengan PWM 50% menghasilkan RPM 468,7 – 650,7. Sebaliknya jika nilai ADC nya tinggi maka nilai PWM nya pun akan tinggi dan menghasilkan nilai RPM yang tinggi seperti hujan deras yang menghasilkan ADC dari 3635 – 4537 dengan PWM 90% menghasilkan RPM 2063– 2607.

#### REFERENSI

- [1] A. J. Nathan and A. Scobell, *Wiper dan Wahser Pump*, vol. 91, no. 5. 2012.
- [2] R. Munir, “Pengantar Logika Fuzzy,” *Tek. Inform. - STEI ITB*, p. 95, 2007.
- [3] B. Siswoyo and A. Zaenal, “Model Peramalan Fuzzy Logic,” *J. Manaj. Inform.*, vol. 8, no. 1, pp. 1–14, 2018, doi: 10.34010/jamika.v8i1.897.
- [4] D. Pemilihan, P. Mahasiswa, and U. Tugas, “Penggunaan Metode Fuzzy Inference System (Fis) Mamdani Dalam Pemilihan Peminatan Mahasiswa Untuk Tugas Akhir,” *J. Inform. Darmajaya*, vol. 15, no. 1, pp. 10–23, 2015.
- [5] Leyva-Bravo, Jansel; Chiñas-Sanchez, Pamela; Hernandez-Rodriguez, Alejandra; Hernandez-Alba, Gerardo G. (2020). Electrochemical discharge machining modeling through different soft computing approaches. *The International Journal of Advanced Manufacturing Technology*.
- [6] S. Sutikno and I. Waspada, “Perbandingan Metode Defuzzifikasi Sistem Kendali Logika Fuzzy Model Mamdani Pada Motor Dc,” *J. Masy. Inform.*, vol. 2, no. 3, pp. 27–38, 2012, doi: 10.14710/jmasif.2.3.27-38.
- [7] E. B. Loewenstein, “Analog-To-Digital converters,” *Electr. Meas. Signal Process. Displays*, vol. 153, pp. 25-1-25–22, 2003, doi: 10.1201/9780203009406.
- [8] E. S. Han and A. goleman, daniel; boyatzis, Richard; Mckee, “濟無NoTitle No Title,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2019.
- [9] Vaibhav Ashok kharche, & Sheetal Hemant Sawkare. (2022). Automatic Rain Sensor Wiper System. *Research and Reviews on Experimental and Applied Mechanics*, 5(1), 1–4.
- [10] Rahmadewi, R., Adzillah, W. N., Purwanti, E., Efelina, V. & Utama, R. W. Pengendali Wiper dan Washer Otomatis Menggunakan Sensor Hujan dan Sensor Debu Berbasis Arduino. *J. Teknol. (Jurnak Tek.* 10, 1–4 (2020).