

Pengembangan Sistem Prediksi Risiko Banjir Berdasarkan Ketinggian Air Menggunakan Logika *Fuzzy*

Mohammad Hazel Dimas Prayogi
Fakultas Teknik Elektro, Program Studi
Teknik Elektro
Universitas Telkom, Kampus Surabaya
Samarinda, Indonesia
mhazeldimas99@gmail.com

Dimas Adiputra, B.Sc., M.Phil., Ph.D.
Fakultas Teknik Elektro, Program Studi
Teknik Elektro
Universitas Telkom, Kampus Surabaya
Surabaya, Indonesia
dimasze@telkomuniversity.ac.id

Dr. Susijanto Tri Rasmana, S.Kom., M.T.
Fakultas Teknik Elektro, Program Studi
Teknik Komputer
Universitas Telkom, Kampus Surabaya
Surabaya, Indonesia
susijanto@telkomuniversity.ac.id

Abstrak — Sistem prediksi risiko banjir berbasis ketinggian air telah dirancang menggunakan logika *Fuzzy* untuk memprediksi risiko banjir berdasarkan perubahan ketinggian air. Sistem ini menggunakan Arduino MKR Wi-Fi 1010 dan sensor ultrasonik HC-SR04 sebagai perangkat utama. Data ketinggian air diolah menggunakan logika *Fuzzy* untuk mengklasifikasikan risiko banjir menjadi "aman", "siaga", dan "bahaya". Hasil prediksi ditampilkan melalui LCD dan disimpan dalam Firebase Database secara real-time. Hasil pengujian menunjukkan sensor ultrasonik memiliki rata-rata error 2,37%, sedangkan prediksi logika *Fuzzy* memiliki akurasi tinggi dengan error 0,01%. Sistem ini dapat memberikan informasi risiko banjir secara akurat, meskipun pengiriman data ke Firebase masih tergantung pada kestabilan koneksi internet.

Kata kunci— banjir, logika *Fuzzy*, Arduino, prediksi risiko, sensor ultrasonik.

I. PENDAHULUAN

Banjir merupakan salah satu bencana alam yang sering terjadi di berbagai wilayah Indonesia. Penyebab banjir cukup beragam, mulai dari curah hujan tinggi, luapan sungai, hingga pasang air laut. Dampaknya sangat merugikan, baik dari segi lingkungan, sosial, maupun ekonomi masyarakat terdampak. Oleh karena itu, upaya mitigasi dan pencegahan sangat penting dilakukan secara sistematis dan berkelanjutan. Langkah-langkah seperti peningkatan sistem drainase, penerapan sistem peringatan dini, serta edukasi masyarakat mengenai pengelolaan air yang baik menjadi bagian dari solusi dalam menghadapi risiko banjir.

Seiring kemajuan teknologi, pemanfaatan perangkat berbasis mikrokontroler seperti Arduino, serta teknologi *Internet of Things* (IoT) dan *Artificial Intelligence* (AI), semakin relevan dalam pengembangan sistem pemantauan banjir. Beberapa penelitian sebelumnya [1], [2] telah mengembangkan sistem deteksi banjir menggunakan Arduino dan sensor ultrasonik untuk memantau ketinggian air. Sistem-sistem tersebut mengirimkan informasi

peringatan melalui SMS atau menampilkan data secara *realtime* melalui *website*. Meski demikian, sistem tersebut hanya berfungsi sebagai alat *monitoring*, belum memiliki kemampuan untuk memprediksi risiko banjir berdasarkan perubahan data yang diperoleh. Hal ini menunjukkan perlunya inovasi lebih lanjut dalam pengembangan sistem prediktif yang lebih cerdas dan responsif.

Berdasarkan hal tersebut, penelitian ini mengusulkan pengembangan sistem prediksi risiko banjir berbasis logika *Fuzzy*. Sistem ini menggunakan mikrokontroler Arduino MKR *Wi-Fi* 1010 yang terhubung dengan sensor ultrasonik HC-SR04 untuk membaca ketinggian air. Logika *Fuzzy* diterapkan untuk mengolah data ketinggian air dan kecepatan perubahannya, sehingga sistem dapat memprediksi status risiko banjir seperti aman, siaga, atau bahaya. Pengukuran dan prediksi dilakukan setiap lima detik secara otomatis dan hasilnya ditampilkan melalui LCD serta diperkuat dengan *buzzer* sebagai indikator peringatan. Diharapkan sistem ini mampu memberikan informasi prediktif secara cepat dan akurat, serta menjadi solusi teknologi yang membantu upaya mitigasi bencana banjir di Indonesia.

II. KAJIAN TEORI

A. Banjir

Salah satu bencana yang sering melanda kota-kota di Indonesia ialah banjir. Banjir merupakan suatu kondisi daratan yang tergenang air dalam jumlah yang cukup besar. Banjir dapat terjadi karena beberapa hal seperti hujan lebat, sungai meluap, air laut dalam kondisi pasang tinggi dan lain sebagainya. Banjir juga menimbulkan dampak yang cukup serius terhadap lingkungan, masyarakat, ataupun ekonomi daerah yang terdampak. Saat ini, perhatian masyarakat Indonesia terpusat pada bencana banjir, terutama di wilayah-wilayah yang sering mengalami banjir, sehingga banjir dianggap sebagai kejadian yang umum terjadi [3]. Upaya pencegahan dan mitigasi banjir menjadi krusial untuk meminimalisir kerugian dan melindungi masyarakat serta aset-aset penting di wilayah yang rentan terhadap bencana ini. Peningkatan kapasitas saluran drainase, implementasi

sistem peringatan dini, dan kesadaran masyarakat terhadap tata kelola air yang baik menjadi langkah-langkah penting dalam menghadapi risiko banjir di Indonesia.

B. Arduino IDE dan Arduino MKR *Wi-Fi* 1010

Arduino merupakan sebuah platform perangkat keras (*hardware*) dan perangkat lunak (*software*) yang cukup populer digunakan untuk membuat *prototype* peralatan elektronik. Arduino IDE sangat mudah digunakan karena bahasa pemrograman yang digunakan adalah mirip dengan bahasa C/C++. Penelitian ini menggunakan Arduino MKR *Wi-Fi* 1010 sebagai *microcontroller* utama untuk mengintegrasikan setiap komponen yang ada. Arduino ini juga dilengkapi dengan modul Wi-Fi yang membuat pengiriman data ke *cloud* menjadi lebih mudah. Salah satu komponen yang akan terintegrasi ke Arduino ini adalah sensor ultrasonik HC-SR04. Sensor ultrasonik ini dapat mendeteksi ketinggian air secara akurat. Data yang didapat dari sensor tersebut kemudian diolah oleh Arduino dan memicu respon atau pemberitahuan ketika ketinggian air mencapai batas yang ditentukan. Hal ini menjadikan Arduino sebagai solusi yang efektif dalam mengembangkan sistem pendeteksi ketinggian banjir.

C. Sensor Ultrasonik HC-SR04

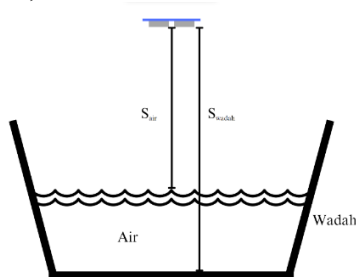
Sensor ultrasonik HC-SR04 merupakan sensor pengukur jarak yang menggunakan gelombang ultrasonik. Sensor ultrasonik terdiri dari dua bagian, yaitu pemancar gelombang (*transmitter*) dan penerima gelombang (*receiver*) [4]. Cara kerja dari sensor ultrasonik yaitu dengan memantulkan gelombang ultrasonik ke objek didepannya kemudian diterima balik oleh *receiver* ultrasonik [5]. Jarak dari objek dapat diukur melalui jarak antara waktu tempuh pemancar ke objek dan waktu dari objek ke penerima. Hal ini menunjukkan bahwa waktu yang dibutuhkan dari pemancar hingga penerima sama dengan dua kali jarak dari sensor dengan objek, sehingga jarak antara sensor dengan objek dapat ditentukan dengan persamaan (1):

$$S = \frac{v \times t}{2} \quad (1)$$

Dimana:

S = Jarak (m)
v = Kecepatan suara (340 m/s atau 0,034 cm/μs)
t = Waktu (s)

Dalam penelitian ini, sensor ini digunakan untuk mengukur ketinggian air [3], [5], [6] dengan cara memantulkan gelombang ultrasonik ke permukaan air seperti gambar 1 berikut,



GAMBAR 1

Jarak antara ultrasonik dengan air dan wadah

Dari gambar tersebut dapat dirumuskan rumus untuk mencari ketinggian air seperti persamaan 2 berikut ini,

$$h_{\text{air}} = S_{\text{wadah}} - S_{\text{air}} \quad (2)$$

Dimana,

h_{air} = Ketinggian Air (cm)
 S_{wadah} = Jarak antara ultrasonik dengan wadah (cm)
 S_{air} = Jarak antara ultrasonik dengan air (cm)

Terhubungnya sensor ultrasonik ke Arduino dapat memberikan informasi ketinggian air secara *realtime*. Hal tersebut juga berpengaruh pada pengambilan keputusan yang cepat dan tepat.

D. *Liquid Crystal Display* (LCD)

Liquid Crystal Display (LCD) adalah komponen tampilan elektronik yang digunakan untuk menampilkan karakter seperti huruf, simbol, dan angka. Menurut Sarmidi & Rahmat S [7], LCD bekerja tanpa menghasilkan cahaya sendiri, melainkan memantulkan atau mentransmisikan cahaya dari sumber eksternal. Teknologi ini umum digunakan pada berbagai perangkat elektronik seperti televisi, laptop, *handphone*, dan kalkulator karena efisiensinya yang tinggi dan desainnya yang ringkas. Dalam pengembangan sistem berbasis mikrokontroler, LCD berfungsi sebagai antarmuka untuk menampilkan data atau output dari program. Pada sistem ini, LCD digunakan untuk menampilkan hasil pembacaan sensor ultrasonik HC-SR04 serta prediksi risiko banjir dari logika *Fuzzy*.

E. *Buzzer*

Buzzer merupakan suatu komponen elektronika yang memiliki fungsi mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja *Buzzer* hampir sama dengan *loudspeaker*, jadi *Buzzer* juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara [8]. Dalam penelitian ini, *Buzzer* digunakan sebagai penampil indikator secara suara.

F. *Firestore Database*

Firestore Database merupakan layanan penyimpanan *cloud* yang ditawarkan oleh Google melalui platform *Firestore*, dirancang khusus untuk mendukung pengembangan aplikasi *mobile* dan *web* [9]. *Database* ini menggunakan struktur data JSON dan memungkinkan sinkronisasi data secara *realtime* antar perangkat, memastikan bahwa perubahan yang terjadi pada satu perangkat segera tercermin pada perangkat lain. *Firestore Database* juga menyajikan fitur keamanan tingkat tinggi, menyertakan pendengar (*listener*) untuk memonitor perubahan data, fasilitas kueri, dan integrasi yang lancar dengan berbagai platform pengembangan seperti Android, iOS, dan *web*. Oleh karena itu, *Firestore Database* seringkali

dipilih untuk membangun aplikasi yang membutuhkan penyimpanan dan sinkronisasi data secara langsung.

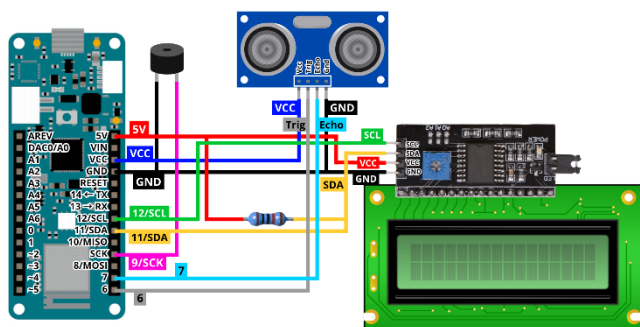
G. Logika Fuzzy

Logika *Fuzzy* adalah metode kecerdasan buatan yang digunakan untuk menangani ketidakpastian dalam pengambilan keputusan, terutama ketika informasi bersifat ambigu atau tidak pasti [10]. Berbeda dengan logika biner yang hanya mengenal nilai benar atau salah, logika Fuzzy memungkinkan adanya derajat kebenaran di antara keduanya [10]. Metode ini terdiri dari beberapa tahapan, yaitu fuzzifikasi, *rule-based* (aturan jika-maka), *inference machine* (penalaran), dan defuzzifikasi yang menghasilkan nilai output dalam bentuk angka pasti. Logika *Fuzzy* banyak diterapkan dalam sistem kontrol, pengambilan keputusan, dan kecerdasan buatan karena kemampuannya menangani data dunia nyata yang kompleks. Dalam penelitian ini, logika *Fuzzy* digunakan untuk memprediksi risiko banjir berdasarkan ketinggian dan kecepatan perubahan air pada periode waktu tertentu.

III. METODE

A. Pembuatan Prototype

Pada tahap ini, alat dan bahan mulai digunakan untuk membuat *prototype* sistem prediksi risiko banjir. Mulai dari perakitan Arduino MKR *Wi-Fi* 1010 dengan sensor ultrasonik HC-SR04, LCD dan *Buzzer* menjadi sistem pendeteksi ketinggian air hingga pembuatan simulasi banjir. Gambaran dari *wiring hardware* sistem ini dapat dilihat pada Gambar 2,



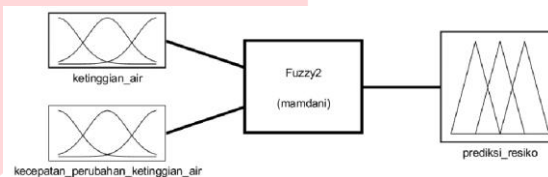
GAMBAR 2
Rancangan Alat

Pada sistem ini, Arduino MKR *Wi-Fi* 1010 berperan sebagai pusat kontrol dan komunikasi antar komponen seperti sensor ultrasonik HC-SR04, LCD, dan *buzzer*. Sensor HC-SR04 memiliki empat pin yang terhubung ke Arduino, yaitu VCC, GND, Trig pada pin 6, dan Echo pada pin 7, di mana Trig mengirim sinyal dan Echo menerima pantulan sinyal tersebut. LCD menggunakan modul I2C untuk menghemat penggunaan pin, sehingga hanya membutuhkan empat pin yaitu VCC, GND, SDA, dan SCL beserta dengan pull-up resistor 10k Ω digunakan pada pin SDA dan SCL untuk menjaga kestabilan sinyal. Pin VCC dan GND LCD masing-masing terhubung ke VCC dan GND Arduino, sedangkan SDA dan SCL terhubung ke pin SDA dan SCL Arduino. Buzzer terhubung ke Arduino menggunakan dua pin, dengan

pin positif (+) dihubungkan ke pin digital 9 dan pin negatif (-) ke GND.

B. Pembuatan Logika Fuzzy

Pada tahap ini, penulis menggunakan metode *Fuzzy* Mamdani karena hasil *output*nya lebih mudah dipahami, seperti “aman”, “siaga”, atau “bahaya”. Kategori banjir ditentukan berdasarkan ketinggian air, yaitu rendah (10–35 cm), sedang (35–80 cm), dan tinggi (lebih dari 80 cm) [11]. Berdasarkan klasifikasi tersebut, sistem dirancang dengan dua variabel *input* dan satu variabel *output*. Variabel *input* yang digunakan adalah “ketinggian air” dan “kecepatan perubahan ketinggian air”. Kedua input ini akan dianalisis menggunakan logika *Fuzzy* untuk memprediksi tingkat risiko banjir. Pada variabel *output*, terdapat “prediksi risiko” seperti pada Gambar 5 dibawah ini,

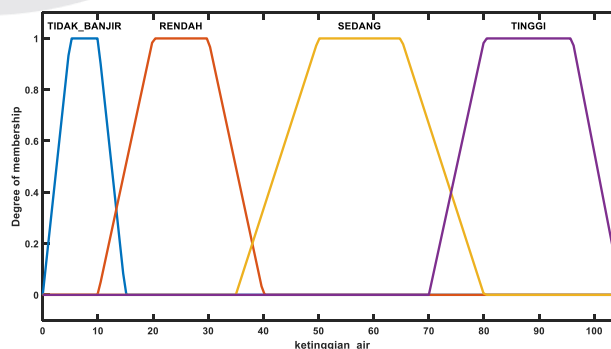


GAMBAR 3
Desain Logika Fuzzy Sistem

Variabel “ketinggian air” menggunakan *membership function* untuk mengklasifikasikan tingkat bahaya banjir, di mana semakin tinggi air, maka risikonya semakin besar. Pengukuran dilakukan berdasarkan jarak antara sensor ultrasonik dan permukaan air, dengan titik nol berada pada posisi sensor. Jarak minimum yang dapat dibaca sensor adalah 3 cm, sehingga nilai pembacaan dikurangi 3 cm untuk mendapatkan ketinggian air yang sebenarnya. Jika tinggi maksimal wadah adalah 104 cm, maka ketinggian air tertinggi yang dapat terdeteksi adalah 101 cm. Nilai-nilai ini kemudian digunakan untuk mengkonversi data sensor menjadi ketinggian air sebenarnya sesuai persamaan yang digunakan. Nilai yang telah dikonversi inilah yang digunakan dalam sistem ini yang dapat dilihat pada tabel (1) berikut,

TABEL 1
Membership function dari ketinggian air

Klasifikasi	Swadah	Sair	h _{air}
TINGGI	104 cm	3 – 10 cm	80 – 101 cm
SEDANG		10 – 35 cm	35 – 80 cm
RENDAH		80 – 35 cm	10 – 35 cm
TIDAK ADA		80 – 101 cm	0 – 10 cm



GAMBAR 4
Membership function dari “ketinggian air” pada Matlab

Selain variabel ketinggian air, terdapat juga variabel "kecepatan perubahan ketinggian air", variabel ini bergantung pada selisih ketinggian air saat ini dan sebelumnya. Variabel ini berisi mengenai kecepatan perubahan ketinggian air. Apabila kecepatan perubahan tersebut "bertambah cepat" dan ketinggian air itu "tinggi" maka keputusan akan ditentukan melalui variabel *output* yaitu variabel "prediksi". Adapun rumus untuk menghitung kecepatan perubahan ketinggian air seperti persamaan 3 berikut ini,

$$v = \frac{h_t - h_{t-1}}{\Delta t} \quad (3)$$

Dimana,

v = Kecepatan perubahan ketinggian air (cm/s)

h_t = Ketinggian saat ini (cm)

h_{t-1} = Ketinggian sebelumnya (cm)

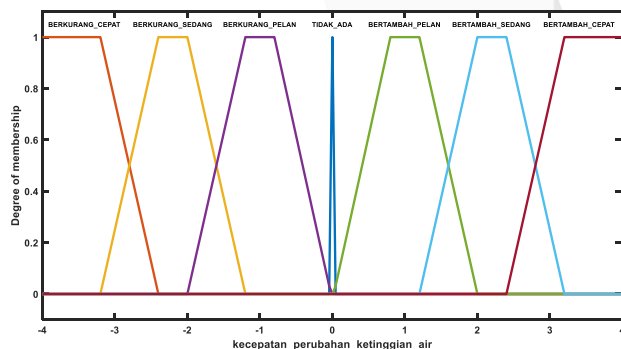
Δt = Selisih waktu (diketahui per 5 detik) (s)

Berdasarkan pada persamaan tersebut, penulis membuat rentang selisih ketinggian beserta konversi selisih tersebut dalam satuan kecepatan perubahan ketinggian air (cm/s). Rentang nilai yang telah dikonversi inilah yang digunakan dalam sistem ini yang dapat dilihat pada tabel 2 yang berisi *Membership function* dari "kecepatan perubahan ketinggian air" beserta gambar *Membership function* tersebut dalam aplikasi Matlab pada gambar 5 berikut ini,

TABEL 2

Membership function dari kecepatan perubahan ketinggian air

Status	Selisih Ketinggian Setiap 5 detik (cm)	Kecepatan Perubahan Ketinggian Air (cm/s)
TIDAK ADA	0	0
BERTAMBAH PELAN	0.1 – 8	0.02 – 1.6
BERTAMBAH SEDANG	8 – 14	1.6 – 2.8
BERTAMBAH CEPAT	> 14	> 2.8
BERKURANG PELAN	-0.1 – -8	-0.02 – -1.6
BERKURANG SEDANG	-8 – -14	-1.6 – -2.8
BERKURANG CEPAT	< -14	< -2.8



GAMBAR 5

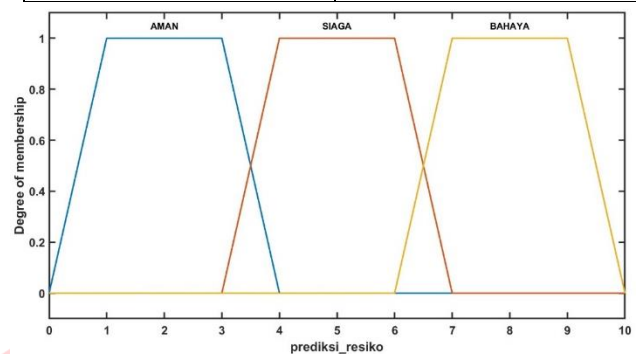
Membership function dari "ketinggian air" pada Matlab

Selanjutnya terdapat variabel "prediksi", variabel ini berisi mengenai prediksi tingkat risiko banjir, risiko tersebut terdiri dari "aman", "siaga" dan "bahaya". Kedua *input* tersebut menentukan *output* yang akan dikeluarkan oleh logika *Fuzzy*. Apabila kecepatan perubahan tersebut bertambah cepat dan ketinggian air itu tinggi maka keputusan yang diambil oleh variabel "prediksi risiko" adalah "bahaya". *Membership function* dari "prediksi" dapat dilihat dari tabel 3 dan gambar 6 berikut ini,

TABEL 3

Membership function prediksi risiko

Status	Skala Risiko
AMAN	0 – 4
SIAGA	3 – 7
BAHAYA	6 – 10



GAMBAR 6

Membership function dari "kecepatan perubahan ketinggian air" pada Matlab

Selain dari ketiga variabel tersebut, penulis juga membuat *rule* atau aturan logika *Fuzzy* agar sistem dapat memprediksi risiko dengan tepat yang dilihat pada tabel 4 berikut ini,

TABEL 4

Rules dari logika *Fuzzy*

	TIDAK BANJIR	RENDAH	SEDANG	TINGGI
TIDAK ADA	AMAN	AMAN	SIAGA	BAHAYA
BERTAMBAH PELAN	AMAN	AMAN	SIAGA	BAHAYA
BERTAMBAH SEDANG	AMAN	AMAN	SIAGA	BAHAYA
BERTAMBAH CEPAT	AMAN	SIAGA	BAHAYA	BAHAYA
BERKURANG PELAN	AMAN	AMAN	SIAGA	BAHAYA
BERKURANG SEDANG	AMAN	AMAN	SIAGA	BAHAYA
BERKURANG CEPAT	AMAN	AMAN	AMAN	SIAGA

C. Pembuatan Program Arduino

Program Arduino dibuat menggunakan aplikasi Arduino IDE yang dapat diunduh dari situs resmi Arduino. Arduino IDE menggunakan bahasa pemrograman mirip C/C++ dan digunakan untuk menghubungkan Arduino MKR Wi-Fi 1010 dengan komponen lain dalam satu sistem. Program terdiri dari dua fungsi utama, yaitu *setup()* dan *loop()*. Fungsi *setup()* dijalankan sekali saat awal program untuk menginisialisasi library sensor HC-SR04 dan LCD. Sementara itu, fungsi *loop()* dijalankan terus-menerus untuk membaca data dari sensor secara berulang.

D. Uji Coba dan Evaluasi

Pada tahap ini, semua komponen dan logika *Fuzzy* dilakukan uji coba untuk memastikan sistem benar-benar bekerja dengan baik sebelum pengambilan data dilakukan. Terdapat beberapa parameter yang digunakan untuk mengevaluasi kinerja sistem. Adapun parameternya sebagai berikut,

a. Sistem hardware dapat bekerja dengan baik,

- Memastikan Arduino MKR *Wi-Fi* 1010 dapat mengirimkan data ketinggian dan status risiko ke

database pada *Firestore* dengan cara melakukan pemantauan pada database *Firestore*.

- Memastikan sensor ultrasonik dapat melakukan pengukuran secara akurat dengan cara melakukan perbandingan pengukuran antara sensor ultrasonik HC-SR04 dengan alat ukur penggaris dan dengan presentase error dibawah 10%, rumus mencari presentase tersebut dapat dilihat melalui persamaan 3 berikut,

$$Error = \frac{[hP - hU]}{hP} \times 100\% \quad (3)$$

Dimana,

hP = Nilai ketinggian air menggunakan penggaris (cm)

hU = Nilai ketinggian air menggunakan ultrasonik (cm)

- Memastikan LCD dapat menampilkan hasil dari sistem dengan baik.
 - Memastikan *buzzer* berbunyi sesuai dengan kondisi yang ditetapkan.
- b. Logika *Fuzzy* melakukan prediksi dengan benar. Pengujian logika *Fuzzy* dilakukan untuk mengetahui bentuk logika manakah yang cocok untuk sistem ini.
- c. Logika *Fuzzy* dapat diintegrasikan dengan Arduino MKR *Wi-Fi* 1010 agar menjadi satu kesatuan sistem.

Apabila sistem belum bekerja dengan baik maka perlu dilakukan evaluasi mengenai kendala yang terjadi.

IV. HASIL DAN PEMBAHASAN

A. Pengujian Sensor Ultrasonik HC-SR04

Pengujian dilakukan dengan cara membandingkan pengukuran dari sensor ultrasonik HC-SR04 dengan pengukuran manual menggunakan penggaris. Berdasarkan persamaan 3 dapat dilihat perbandingan uji pengukuran yang dapat dilihat pada tabel 5 berikut,

TABEL 5
Perbandingan Uji Pengukuran Ultrasonik HC-SR04 dengan Penggaris

No	Ultrasonik HC-SR04 (cm)	Penggaris (cm)	Error (%)
1	27	28	3,57%
2	23	24	4,17%
3	17	17	0,00%
4	15	15	0,00%
5	12	12	0,00%
6	19	19	0,00%
7	27	28	3,57%
8	19	20	5,00%
9	15	15	0,00%
10	13	14	7,14%
Rata-Rata Error (%)			2,37%

Percobaan dilakukan sebanyak 20 kali untuk melihat perbandingan pengukuran menggunakan sensor ultrasonik HC-SR04 dan diambil 10 data sebagai sampel perbandingan. Data yang didapat dari proses pengujian pada tabel 5 menunjukkan bahwa perbandingan pengukuran antara sensor ultrasonik HC-SR04 dengan penggaris memiliki nilai rata-rata *error* dibawah 10% yaitu sebesar 2,37%. Hal ini menunjukkan bahwa sensor ultrasonik HC-SR04 mampu membaca ketinggian banjir dengan cukup akurat. Dari data tersebut juga dapat terlihat nilai *error* tertinggi adalah pada percobaan ke-10 yaitu sebesar 7,14% dan terdapat beberapa

percobaan yang memiliki nilai *error* terendah yaitu sebesar 0,00%. Besarnya nilai *error* mungkin dapat terjadi karena sinyal gelombang ultrasonik terpantul ke arah lain yang mengakibatkan kesalahan pembacaan pada sensor. Dapat dilihat pada tabel tersebut bahwa sebenarnya selisih ketinggian antara sensor ultrasonik dengan penggaris hanya selisih 1 cm. Nilai *error* yang berbeda-beda juga dipengaruhi oleh ketinggian yang terukur dari penggaris sebagai nilai pembagi dalam perhitungan mencari persentase *error*.

B. Pengujian Logika *Fuzzy*

Pengujian dilakukan dengan cara melihat kesesuaian kondisi yang logika *Fuzzy* yang ada pada alat dengan *Rules Viewer*. Terdapat juga *Surface Viewer* yang digunakan untuk melihat kesesuaian kondisi desain aturan logika *Fuzzy*. Aturan *Fuzzy* dalam penelitian ini berdasarkan pada hubungan antara variabel *input*, yaitu ketinggian air dan kecepatan perubahan ketinggian air untuk menghasilkan variabel *output* berupa prediksi risiko.

a. Pengujian *Rules Viewer*

Rules Viewer merupakan visualisasi dari aturan-aturan logika *Fuzzy* yang diterapkan dalam sistem prediksi risiko. Berikut merupakan hasil perbandingan kesesuaian antara *rules* yang dibuat di Matlab dengan implementasi logika *Fuzzy* pada alat,

TABEL 6
Perbandingan nilai prediksi pada *Rules Viewer* dengan implementasi pada alat

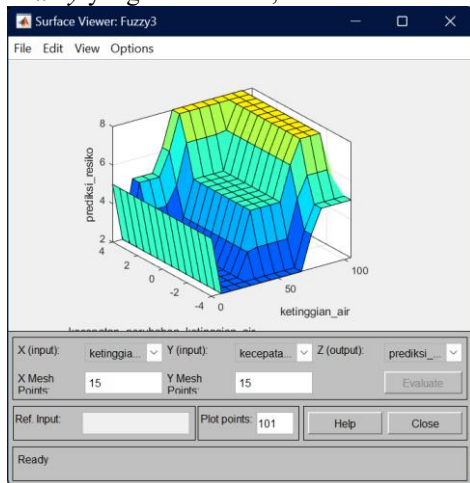
No	Ketinggian (cm)	Kecepatan Perubahan Ketinggian Air (cm/s)	Prediksi		Error (%)
			Alat	Rules Viewer Fuzzy	
1	77	0,40	7,12	7,12	0,00%
2	79	0,40	7,63	7,64	0,13%
3	77	-0,40	7,12	7,12	0,00%
4	73	-0,80	6,18	6,18	0,00%
5	71	-0,40	5,51	5,51	0,00%
6	69	-0,40	5,00	5,00	0,00%
7	67	-0,40	5,00	5,00	0,00%
8	64	-0,60	5,00	5,00	0,00%
9	63	-0,20	5,00	5,00	0,00%
10	61	-0,40	5,00	5,00	0,00%
Rata-Rata Error (%)					0,01%

Pada tabel 6 terdapat percobaan pengujian dengan membandingkan nilai prediksi pada alat dengan nilai prediksi yang seharusnya pada logika *Fuzzy*. Pengujian dilakukan sebanyak 55 data dan diambil 10 data sebagai sampel perbandingan. Dapat dilihat dari tabel 6, rata-rata nilai *error* sebesar 0,01% yang artinya nilai prediksi yang seharusnya pada logika *Fuzzy* dengan nilai prediksi pada alat menghasilkan nilai yang sama. Terlihat nilai *error* terbesar terdapat pada percobaan ke-2 yaitu sebesar 0,13% dan nilai terendah terdapat pada beberapa percobaan yaitu sebesar 0,00%. Hal ini menunjukkan bahwa kesesuaian antara logika *Fuzzy* seharusnya dengan logika *Fuzzy* pada alat. Walaupun nilai *error*nya berbeda-beda, selisih nilai prediksi antara logika *Fuzzy* seharusnya dengan logika *Fuzzy* yang digunakan pada alat memiliki nilai selisih sebesar 0,01 seperti percobaan ke-2. Pada percobaan tersebut alat memiliki nilai skala prediksi sebesar 7,63 sedangkan pada logika *Fuzzy* seharusnya memiliki nilai skala

sebesar 7,64. Hal ini bisa terjadi karena logika *Fuzzy* pada Matlab melakukan pembulatan nilai skala prediksi Seperti 7,63 dibulatkan menjadi 7,64. Perbedaan nilai *error* juga dipengaruhi oleh nilai skala prediksi logika *Fuzzy* sebagai pembagi dalam perhitungan persentase nilai *error*.

b. Surface View

Selain *Rules Viewer*, terdapat pula *Surface Viewer* yang diamati pada logika *Fuzzy*. *Surface Viewer* merupakan representasi grafis dalam tiga dimensi yang menggambarkan hubungan antara *input* (ketinggian air dan kecepatan perubahan ketinggian air) dan *output* (prediksi risiko). Berikut merupakan *Surface View* dari logika *Fuzzy* yang telah dibuat,



GAMBAR 7
Surface View

Visualisasi ini dapat mempermudah pemahaman mengenai pola keputusan sistem logika *Fuzzy*, serta melakukan validasi terhadap aturan-aturan yang telah diterapkan. Pada Gambar 7, dapat terlihat bahwa pada *Surface View* memiliki dua *input* yaitu ketinggian air dan kecepatan perubahan ketinggian air yang dapat mempengaruhi *output* yaitu prediksi risiko. Misalkan pada titik ketinggian 50 cm yaitu dalam kondisi ketinggian tergolong sedang dan terjadi pengurangan ketinggian air secara cepat maka kondisi prediksi risiko aman.

C. Pengujian Firebase Database

Pengujian dilakukan dengan cara melihat kesesuaian data yang telah dikirim oleh Arduino melalui modul *Wi-Fi* ke *Firestore Database*. Berikut merupakan tabel 7 yang menunjukkan percobaan kesesuaian data ke *Firestore Database*,

TABEL 7

Percobaan kesesuaian data ke *Firestore Database*

No	Kecepatan	Ketinggian	Prediksi	Kesesuaian Data
1	Berkurang Pelan	93	BAHAYA	Sesuai
2	Berkurang Pelan	91	BAHAYA	Sesuai
3	Berkurang Pelan	87	BAHAYA	Sesuai
4	Berkurang Pelan	85	BAHAYA	Sesuai
5	Berkurang Pelan	83	BAHAYA	Sesuai

No	Kecepatan	Ketinggian	Prediksi	Kesesuaian Data
6	Berkurang Pelan	80	BAHAYA	Sesuai
7	Berkurang Pelan	77	BAHAYA	Sesuai
8	Berkurang Pelan	75	SIAGA	Sesuai
9	Berkurang Pelan	72	SIAGA	Sesuai
10	Berkurang Pelan	69	SIAGA	Sesuai
Akurasi Kesesuaian				100%

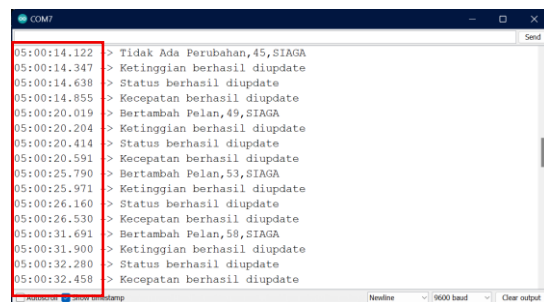
Berdasarkan tabel 7 mengenai percobaan kesesuaian data pada *Firestore Database* dan *Serial Monitor* yang dilakukan sebanyak 30 kali pengiriman data oleh Arduino ke *Firestore Database* dan diambil 10 data untuk dijadikan sebagai sampel. Pada saat pengujian pengiriman data, penulis melihat langsung antara *Firestore Database* dengan *Serial Monitor* dari Arduino agar dapat mengetahui kesesuaian data yang terkirim. Dapat dilihat bahwa dari 20 data yang diambil, semua data yang terkirim sesuai dengan *Serial Monitor* pada Arduino. Maka dari itu dapat terlihat bahwa pengiriman berhasil dengan presentase 100% data yang terkirim ke *Firestore Database* merupakan data yang sesuai dengan *Serial Monitor*.

Adapun percobaan untuk mengetahui perbandingan waktu *update* data pada *Serial Monitor* dan *Firestore Database* beserta selisih waktu keduanya. Percobaan dengan cara melihat *Firestore Database* dan *Serial Monitor* Arduino. Waktu *update* data ke *Firestore Database* dapat dilihat dengan cara menambahkan *serial print* mengenai ketinggian, status dan kecepatan apabila sudah melakukan *update* data. Berikut merupakan tabel 8 mengenai perbandingan waktu *update* data pada *Serial Monitor* dan *Firestore Database* beserta selisih waktu keduanya.

TABEL 8

Perbandingan waktu *update* data pada *Serial Monitor* dan *Firestore Database* beserta selisih waktu keduanya.

No	Waktu Data realtime (25/04/2025)		Selisih Waktu (s)
	Serial Monitor	Firestore	
1	22:28:56,148	22:28:57,206	1,058
2	22:29:02,260	22:29:02,918	0,658
3	22:29:07,971	22:29:08,933	0,962
4	22:29:13,978	22:29:15,291	1,313
5	22:29:20,321	22:29:21,172	0,851
6	22:29:26,210	22:29:27,359	1,149
7	22:29:32,413	22:29:33,227	0,814
8	22:29:38,248	22:29:39,035	0,787
9	22:29:44,238	22:29:44,881	0,643
10	22:29:50,086	22:29:50,919	0,833
Selisih Rata-Rata			0,907



GAMBAR 8

Pengambilan data waktu dari *Serial Monitor*

Dapat dilihat dari Tabel 8 bahwa waktu yang dibutuhkan Arduino untuk melakukan *update* data pada *Firestore Database* apabila mengacu pada data rata-rata selisih waktu antara Serial Monitor dengan *Firestore Database* adalah selama 0,907 detik. Terlihat pula waktu tercepat untuk *update* data pada *Firestore Database* adalah selama 0,643 detik dan waktu terlamanya adalah selama 1,313 detik.

D. Efektivitas Peringatan

a. Interface LCD

Interface LCD dapat menampilkan informasi ketinggian banjir dan status risiko secara jelas seperti Gambar 9 dibawah sehingga pengguna dengan mudah memahami informasi yang didapat terkait kondisi banjir secara langsung pada LCD.



GAMBAR 9
Tampilan LCD pada alat

b. Buzzer

Buzzer memberikan peringatan berupa suara yang bergantung pada status risiko banjir. Apabila aman maka tidak ada suara yang dihasilkan. Jika sudah memasuki kondisi siaga maka *Buzzer* akan berbunyi dengan frekuensi 5000 Hz. Apabila sudah memasuki kondisi bahaya maka *Buzzer* akan mengeluarkan suara dengan frekuensi 1000 Hz.

E. Kelebihan dan Kekurangan Alat

Terdapat beberapa kelebihan serta kelemahan pada alat ini antara lain,

a. Kelebihan

- Dapat mengukur ketinggian banjir secara akurat dan *realtime*.
- Logika *fuzzy* dapat menghasilkan prediksi risiko yang tepat.
- *Firestore* membuat pemantauan risiko banjir dapat dilakukan dari jarak jauh.

b. Kekurangan

- Pengiriman data ke *Firestore* bergantung pada koneksi internet yang stabil.
- Tidak adanya sistem penyimpanan data sementara ketika terputus dari koneksi internet yang membuat data saat koneksi terputus berpotensi hilang.
- Berpotensi adanya gangguan eksternal (adanya benda yang menghalangi sensor ultrasonik HC-SR04) yang mengakibatkan terganggunya pengukuran ketinggian banjir.

F. Simulasi Banjir

Simulasi banjir menggunakan wadah yang berisi air untuk melihat secara langsung bagaimana sistem dan logika *Fuzzy* bekerja sesuai dengan keadaan yang sebenarnya.



GAMBAR 10

Wadah yang digunakan untuk simulasi banjir

V. KESIMPULAN

Sistem prediksi risiko banjir berbasis logika *Fuzzy* berhasil dirancang menggunakan Arduino MKR Wi-Fi 1010 dan sensor ultrasonik HC-SR04, dengan akurasi pengukuran tinggi dan tingkat error prediksi hanya 0,01%. Sistem ini mengklasifikasikan risiko banjir menjadi “aman”, “siaga”, dan “bahaya”, serta menampilkan hasil secara *real-time* melalui LCD dan *Firestore Database*. Hasil pengujian menunjukkan sistem mampu memberikan informasi relevan untuk mendukung langkah mitigasi banjir. Namun, sistem masih memiliki kendala seperti ketergantungan pada koneksi internet dan potensi gangguan pengukuran akibat benda penghalang. Pengembangan lanjutan disarankan, seperti penambahan penyimpanan data sementara dan integrasi dengan aplikasi mobile agar sistem lebih andal dan mudah diakses.

REFERENSI

- [1] Suradi, A. Hanafie, dan S. Leko, “RANCANG BANGUN SISTEM ALAM PENDETEKSI BANJIR BERBASIS ARDUINO UNO,” *ILTEK*, vol. 14, hlm. 2039–2043, Apr 2019.
- [2] N. Pratama, U. Darusalam, dan N. D. Nathasia, “Perancangan Sistem Monitoring Ketinggian Air Sebagai Pendeteksi Banjir Berbasis IoT Menggunakan Sensor Ultrasonik,” *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 4, no. 1, hlm. 117, Jan 2020, doi: 10.30865/mib.v4i1.1905.
- [3] C. Fotian, F. A. Haqqi, dan Yohandri, “A flood detection system and flood door control based on Arduino,” dalam *Journal of Physics: Conference Series*, Institute of Physics, 2023. doi: 10.1088/1742-6596/2582/1/012026.
- [4] Arasada Bakhtiyar dan Suprianto Bambang, “Aplikasi Sensor Ultrasonik Untuk Deteksi Posisi Jarak Pada Ruang Menggunakan Arduino Uno,” *Jurnal Teknik Elektro*, vol. 06, hlm. 137–145, 2017.
- [5] R. Risdiandi, “Analisis Cara Kerja Sensor Ultrasonik Menggunakan Mikrokontroler Arduino Uno Untuk Merancang Alat Deteksi Banjir Secara Otomatis,” *Open Science Framework*, 2020.
- [6] R. Dias Valentin, M. Ayu Desmita, dan A. Alawiyah, “Implementasi Sensor Ultrasonik Berbasis Mikrokontroler Untuk Sistem Peringatan Dini Banjir,” *Jimel*, vol. 2, no. 2, hlm. 2723–598, 2021, doi: 10.33365/jimel.v1i1.
- [7] Sarmidi dan S. I. Rahmat, “SISTEM PERINGATAN DINI BANJIR MENGGUNAKAN SENSOR ULTRASONIK BERBASIS ARDUINO UNO,”

- JURNAL MANAJEMEN DAN TEKNIK INFORMATIKA*, vol. 03, hlm. 31–41, 2019.
- [8] R. Mardiaty, F. Ashadi, dan G. F. Sugihara, “Rancang Bangun Prototipe Sistem Peringatan Jarak Aman pada Kendaraan Roda Empat Berbasis Mikrokontroler ATMEGA32,” *TELKA: Jurnal Telekomunikasi, Elektronika, Komputasi, dan Kontrol*, vol. 2, no. 1, hlm. 53–61, 2016.
- [9] I. F. Maulana, “Penerapan Firebase Realtime Database pada Aplikasi E-Tilang Smartphone berbasis Mobile Android,” *JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 5, hlm. 854–863, 2020.
- [10] S. Nduru, A. Alhafiz, dan D. H. Pane, “Implementasi Metode Fuzzy Berbasis Internet Of Things (IoT) Untuk Peringatan Dini Banjir,” *JURNAL SISTEM KOMPUTER TGD*, vol. 1, hlm. 26–33, 2022.
- [11] F. Seftiani, M. Muzani, dan R. Handawati, “Pemetaan Tingkat Bahaya Banjir Menggunakan Metode Penginderaan Jauh Di Kecamatan Jatinegara Kota Administrasi Jakarta Timur,” *Geosfera: Jurnal Penelitian Geografi*, vol. 2, no. 2, hlm. 48–55, Des 2023, doi: 10.37905/geojpg.v2i2.22485.

