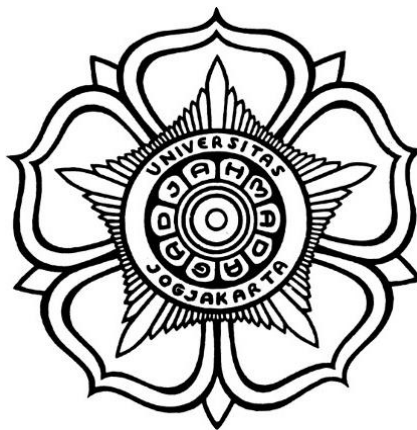


SKRIPSI

**OTOMATISASI SISTEM DETEKSI TINGKAT KEKAKUAN DINDING
BETON BERBASIS SENSOR IMU MENGGUNAKAN METODE *FUZZY***

***AUTOMATICATION OF CONCRETE WALL STIFFNESS LEVEL
DETECTION SYSTEM BASED ON IMU SENSOR USING FUZZY METHOD***



HANIF MALIKI DEWANTO
16/398408/PA/17369

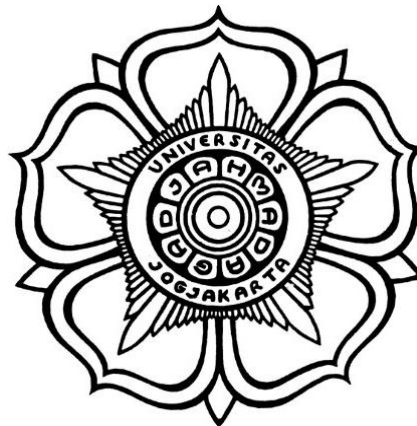
**PROGRAM STUDI ELEKTRONIKA DAN INSTRUMENTASI
DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA
2020**

SKRIPSI

OTOMATISASI SISTEM DETEKSI TINGKAT KEKAKUAN DINDING BETON BERBASIS SENSOR IMU MENGGUNAKAN METODE *FUZZY*

AUTOMATICATION OF CONCRETE WALL STIFFNESS LEVEL DETECTION SYSTEM BASED ON IMU SENSOR USING FUZZY METHOD

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Sarjana Sains
Program Studi Elektronika dan Instrumentasi



HANIF MALIKI DEWANTO
16/398408/PA/17369

**PROGRAM STUDI ELEKTRONIKA DAN INSTRUMENTASI
DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA
2020**



UNIVERSITAS GADJAH MADA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

Sekip Utara BLS 21 Yogyakarta 55281 Telp. (0274) 513339 Fax. (0274) 513339
http://mipa.ugm.ac.id, E-mail:arsiparis_mipa@ugm.ac.id

SURAT KETERANGAN

NOMOR: 130/J01.1.28/PP.03.07/2020

Tanggal : 27 Juli 2020

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Gadjah Mada menyetujui bahwa mahasiswa di bawah ini:

nama : Hanif Maliki Dewanto
nomor induk mahasiswa : 16/398408/PA/17369
program studi : S1-Elektronika dan Instrumentasi
departemen : Ilmu Komputer dan Elektronika
judul karya akhir : Otomatisasi Sistem Deteksi Tingkat Kekakuan Dinding Beton Berbasis Sensor IMU menggunakan Metode *Fuzzy*
tanggal ujian : 10 Juli 2020
pembimbing : 1. Lukman Awaludin, S.Si., M.Cs.
2. –
penguji : 1. Dr. Danang Lelono, S.Si., M.T.
2. M. Idham Ananta Timur, S.T., M.Kom.

telah mendapatkan persetujuan dari para pembimbing/promotor dan penguji (TA/skripsi) sehingga dinyatakan telah menyelesaikan revisi final pada tanggal 24 Juli 2020

Surat Keterangan ini dibuat dan berlaku pada masa tanggap darurat covid 19 dan dapat dipergunakan sebagai pengganti lembar pengesahan dan persetujuan karya tulis akhir sebagai syarat yudisium atau wisuda pada program Sarjana.

Demikian surat keterangan ini dikeluarkan untuk dipergunakan sebagaimana mestinya.

a.n. Dekan
Wakil Dekan Bidang Akademik dan
Kemahasiswaan



Dr.rer.nat. Nurul Hidayat A., M.Si.

Handwritten mark

PERNYATAAN BEBAS PLAGIARISASI

Saya yang bertanda tangan di bawah ini :

Nama : Hanif Maliki Dewanto
NIM : 16/398408/PA/17369
Tahun terdaftar : 2016
Program Studi : Elektronika dan Instrumentasi
Fakultas/Sekolah : Fakultas Matematika dan Ilmu Pengetahuan Alam

Menyatakan bahwa dalam dokumen ilmiah Skripsi ini tidak terdapat bagian dari karya ilmiah lain yang telah diajukan untuk memperoleh gelar akademik di suatu lembaga Pendidikan Tinggi, dan juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang/ lembaga lain, kecuali yang secara tertulis disitasi dalam dokumen ini dan disebutkan sumbernya secara lengkap dalam daftar pustaka.

Dengan demikian saya menyatakan bahwa dokumen ilmiah ini bebas dari unsur-unsur plagiasi dan apabila dokumen ilmiah Skripsi ini di kemudian hari terbukti merupakan plagiasi dari hasil karya penulis lain dan/atau dengan sengaja mengajukan karya atau pendapat yang merupakan hasil karya penulis lain, maka penulis bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Yogyakarta, 24 Juli 2020



Hanif Maliki Dewanto

16/398408/PA/17369

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas segala rahmat, anugrah, dan kesempatan yang telah diberikan sehingga penulis dapat menyelesaikan skripsi ini yang berjudul “Otomatisasi Sistem Deteksi Tingkat Kekakuan Dinding Beton Berbasis Sensor IMU menggunakan Metode *Fuzzy*”. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana S1 Program Studi Elektronika dan Instrumentasi, Departemen Ilmu Komputer dan Elektronika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada.

Pembuatan dan penyusunan skripsi ini tidak terlepas dari bantuan serta dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terimakasih yang sebesar-besarnya kepada:

1. Kedua orang tua dan keluarga tercinta, Bapak Prabowo Sakti, Ibu Amik Sularmi, Ramadhan Galih Wibisana, dan Muh. Dafi Ul Haq yang selalu mendoakan dan memberikan motivasi untuk kelancaran dalam menyelesaikan skripsi ini.
2. Bapak Lukman Awaludin, S.Si., M.Cs. yang telah bersabar dan meluangkan waktu untuk membimbing, memberikan ide dan pemikiran serta saran kepada penulis sehingga dapat menyelesaikan skripsi ini dengan baik.
3. Dosen-dosen Program Studi Elektronika dan Instrumentasi yang telah membimbing dan memberikan ilmunya kepada penulis.
4. Luthfiana Khairun Annisa yang selalu memberikan dukungan moril dan dukungan non teknis yang membantu kelancaran penulis dalam menyelesaikan skripsi ini.
5. Teman-teman satu bimbingan, Shidqi, Jorgi, Dodi, dan Kresna yang membantu bertukar pikiran dalam masalah perskripsian.
6. Seluruh keluarga ELINS angkatan 2016 yang tidak bisa disebutkan satu persatu, terimakasih atas dukungan, bantuan, dan kebersamaan dalam proses penyelesaian skripsi ini.
7. Semua pihak yang memberikan bantuan, semangat dan motivasi dalam proses penyelesaian skripsi ini yang tidak dapat disebutkan satu per satu.

Penulis menyadari masih terdapat kekurangan dan kesalahan dalam penyusunan skripsi ini dikarenakan keterbatasan kemampuan dan ilmu pengetahuan yang penulis miliki. Kritik dan saran yang bersifat membangun sangat penulis harapkan demi perbaikan di masa yang akan datang. Semoga skripsi ini dapat bermanfaat untuk penelitian selanjutnya.

Yogyakarta, 24 Juli 2020

Hanif Maliki Dewanto

DAFTAR ISI

HALAMAN PENGESAHAN.....	iii
PERNYATAAN BEBAS PLAGIARISASI	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	xi
INTISARI	xii
ABSTRACT.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	2
1.6 Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA.....	4
BAB III LANDASAN TEORI.....	8
3.1 Gempa Bumi.....	8
3.2 Skala MMI (Modified Mercalli Intensity).....	8
3.3 Skala Intensitas Gempabumi (SIG)	10
3.4 Peak Ground Acceleration (PGA)	11
3.5 Pengaruh Beban Lateral.....	12
3.6 Inertial Measurement Unit (IMU)	13
3.7 Sensor Fusion	14
3.8 Digital Motion Processor (DMP)	15
3.9 Sistem Fuzzy.....	16
3.10 Fuzzy Inference System Sugeno.....	17
3.11 Pengertian Node	18
3.12 Confusion Matrix.....	18
BAB IV ANALISIS DAN PERANCANGAN SISTEM.....	18
4.1 Analisis Sistem	18
4.2 Rancangan Perangkat Keras	19
4.3 Rancangan Perangkat Lunak	21
4.4 Rancangan Akuisisi Data.....	25
4.5 Rancangan Fuzzy Logic	27
4.6 Rancangan Komunikasi antara Node dengan Server	29
4.7 Rancangan Graphical User Interface	30
4.8 Rancangan Data Logger	30
4.9 Rencana Pengujian Sistem.....	30
BAB V IMPLEMENTASI.....	33
5.1 Implementasi Perangkat Keras	33
5.2 Implementasi Perangkat Lunak	34
5.3 Implementasi Akuisisi Data.....	35
5.4 Implementasi Fuzzy Logic	39

5.5	Implementasi Komunikasi antara Node dengan Server	40
5.6	Implementasi Graphical User Interface	46
5.7	Implementasi Data Logger	48
5.8	Pengujian Fungsional	51
5.9	Pengujian Penentuan Tingkat Kekakuan Dinding Rumah Berbahan Beton terhadap Getaran	52
BAB VI HASIL DAN PEMBAHASAN		55
6.1	Hasil Pengujian Fungsional	55
6.2	Hasil Pengujian Penentuan Tingkat Kekakuan Dinding Rumah Berbahan Beton terhadap Getaran	58
BAB VII KESIMPULAN DAN SARAN		60
7.1	Kesimpulan	60
7.2	Saran	60
DAFTAR PUSTAKA		62
LAMPIRAN		

DAFTAR GAMBAR

Gambar 3.1 Pengaruh Beban Lateral	12
Gambar 3.2 Diagram Blok Linear Accel in World	15
Gambar 3.3 Diagram Blok Sistem Fuzzy	16
Gambar 3.4 Confusion Matrix (Anonim, 2018c).....	18
Gambar 4.1 Diagram Blok Rancangan Perangkat Keras	20
Gambar 4.2 Diagram Blok Rancangan Perangkat Lunak	21
Gambar 4.3 Flowchart Core 1 Node Bawah	22
Gambar 4.4 Flowchart Core 1 Node Atas	23
Gambar 4.5 Flowchart Core 0 Node Bawah & Atas.....	24
Gambar 4.6 Rancangan Penempatan Node	25
Gambar 4.7 Grafik Keanggotaan PGA_{DRI}	27
Gambar 4.8 Rancangan Komunikasi antara Node dengan Server	30
Gambar 4.9 Rancangan Pengujian pada Dinding	32
Gambar 5.1 Skematik Papan Elektronik	33
Gambar 5.2 Implementasi Papan Elektronik	34
Gambar 5.3 Implementasi Pemempatan Node	35
Gambar 5.4 Kode Program Pemanggilan Library I2Cdevlib-MPU6050.....	35
Gambar 5.5 Kode Program Kalibrasi Sensor Node Bawah	36
Gambar 5.6 Kode Program Kalibrasi Sensor Node Atas	36
Gambar 5.7 Kode Program Akuisisi Ground Acceleration	37
Gambar 5.8 Kode Program Akuisisi Inclination Angle	37
Gambar 5.9 Kode Program Akuisisi Displacement	38
Gambar 5.10 Kode Program Akuisisi Drift Ratio.....	38
Gambar 5.11 Kode Program Nilai Puncak Node Bawah	38
Gambar 5.12 Kode Program Nilai Puncak Node Atas.....	39
Gambar 5.13 Kode Program Fuzzy Logic	40
Gambar 5.14 Kode Program Pembuatan Task Node Bawah	41
Gambar 5.15 Kode Program Komunikasi HTTP Node Bawah	42
Gambar 5.16 Kode Program Komunikasi HTTP Node Atas	43
Gambar 5.17 Kode Penerima HTTP Request nodebawah.php	44
Gambar 5.18 Kode Penerima HTTP Request nodeatas.php	44
Gambar 5.19 Web Server ke Database nodebawah.php	45
Gambar 5.20 Web Server ke Database nodeatas.php	45
Gambar 5.21 Kirim HTTP Response nodebawah.php	46
Gambar 5.22 Kirim HTTP Response nodeatas.php	46
Gambar 5.23 Potongan Kode Penampil Data gui.php	47
Gambar 5.24 Potongan Kode Pengirim Perintah gui.php	48
Gambar 5.25 Potongan Kode Pengirim Perintah fungsi.php	48
Gambar 5.26 Kode Program Micro SD Card Data Logger.....	49
Gambar 5.27 Kode Program Database MySQL Data Logger gui.php	50
Gambar 5.28 Kode Program Database MySQL Data Logger fungsi.php.....	50
Gambar 5.29 Kode Pengujian Fuzzy Logic	52
Gambar 5.30 Penempelan Magnet Neodymium ke Plat Besi	53

Gambar 5.31 Penempelan Plat Besi Ke Papan Elektronik.....	53
Gambar 5.32 Penempelan Node ke Dinding.....	53
Gambar 5.33 Lokasi Pengujian pada Dinding	54
Gambar 5.34 Beban Pengujian pada Dinding	54

DAFTAR TABEL

Tabel 2.1 Perbandingan penelitian yang pernah dilakukan	6
Tabel 3.1 Skala MMI BMKG	9
Tabel 3.2 Skala Intensitas Gempabumi BMKG.....	10
Tabel 4.1 Komposisi Aturan Fuzzy	28
Tabel 4.2 Rencana Pengujian Sistem	32
Tabel 6.1 Pengukuran Ground Acceleration Titik Nol (1×10^{-4})g	55
Tabel 6.2 Pengukuran Inclination Angle Titik Nol (1×10^{-3})°	56
Tabel 6.3 Perbandingan Fuzzy Logic pada Sistem dengan MATLAB.....	57
Tabel 6.4 Hasil Pengujian pada Dinding	58

INTISARI

Otomatisasi Sistem Deteksi Tingkat Kekakuan Dinding Beton Berbasis Sensor IMU menggunakan Metode *Fuzzy*

Oleh
Hanif Maliki Dewanto
16/398408/PA/17369

Kurangnya kekakuan dinding dapat mengakibatkan bangunan rusak saat terjadi gempa bumi. Selama ini, tingkat kekakuan (*stiffness*) dinding diukur secara *manual* dengan seismograf melalui perbandingan antara beban lateral dengan *displacement* pada dinding, sehingga kemungkinan *human error* tinggi. Disamping itu, keberadaan *noise* kadang menutup sinyal. Otomatisasi sistem berbasis *fuzzy logic* didalam pengambilan data diharapkan dapat mengurangi kesalahan yang disebabkan oleh faktor manusia dan dapat mengurangi *noise*.

Metode yang digunakan pada penelitian ini adalah *fuzzy logic* dengan akuisisi data berbasis sensor IMU. Pada tahap akuisisi data, ada beberapa data yang dicari yaitu *ground acceleration*, *inclination angle*, *displacement*, *drift ratio*, dan nilai puncak (*peak*) dari keempat data tersebut. Masukan *fuzzy* yang digunakan adalah nilai *peak ground acceleration* yang dibutuhkan untuk menyebabkan *drift ratio* sebesar 1%. Keluaran yang dihasilkan adalah *score* tingkat kekakuan dinding dengan rentang antara 0 sampai 4.

Secara fungsional, proses otomasi dapat bekerja sesuai dengan yang diharapkan. Dengan *filter* pada sensor, *noise* yang dihasilkan kecil, $(4,3 \pm 2,65) \times 10^{-4}$ g untuk *ground acceleration* dan $(4,2 \pm 3,33) \times 10^{-3}$ ° untuk *inclination angle*. Pemakaian *fuzzy* pada pemrosesan menghasilkan *score* yang 100% akurat, sesuai dengan SIG BMKG. Tapi, pengujian pada dinding mendapatkan hasil standar deviasi yang cukup tinggi yang berarti tingkat presisi kurang baik, yang mungkin dikarenakan adanya faktor selain PGA yang dapat mempengaruhi *drift ratio* pada dinding, yang belum dipertimbangkan dalam penelitian ini. Secara umum, sistem ini sudah layak dipakai, tetapi presisi perlu ditingkatkan lagi kedepannya.

Kata kunci: kekakuan dinding, *fuzzy logic*, sensor IMU, getaran

ABSTRACT

Automation of Concrete Wall Stiffness Level Detection System Based on IMU Sensor using Fuzzy Method

By

Hanif Maliki Dewanto

16/398408/PA/17369

Lack of wall stiffness can result in buildings being damaged during an earthquake. During this time, wall stiffness is measured manually with a seismograph through a comparison between lateral load and displacement in the wall, so the possibility of high human error. Besides that, the presence of noise sometimes closes the signal. Automation of fuzzy logic-based systems in data retrieval is expected to reduce errors caused by human factors and can reduce noise.

The method used in this study is fuzzy logic with IMU sensor-based data acquisition. At the data acquisition stage, there are some data that are sought, namely ground acceleration, inclination angle, displacement, drift ratio, and peak values of the four data. Fuzzy input is the peak ground acceleration value needed to cause a drift ratio of 1%. The resulting output is a score of the level of wall stiffness with a range between 0 to 4.

Functionally, the automation process can work as expected. With the filter on the sensor, the resulting noise is small, $(4,3 \pm 2,65) \times 10^{-4}$ g for ground acceleration and $(4,2 \pm 3,33) \times 10^{-3}$ ° for inclination angle. The use of fuzzy in processing produces 100% accurate scores, according to the SIG BMKG. But, testing on the wall gets a fairly high standard deviation which means the level of precision is not good, which may be due to factors other than PGA that can affect the drift ratio on the wall, which has not been considered in this study. In general, this system is feasible to use, but the precision needs to be increased in the future.

Keywords: wall stiffness, fuzzy logic, IMU sensor, vibration

BAB I

PENDAHULUAN

1.1 Latar Belakang

Gempa bumi dengan skala yang cukup besar dapat mengakibatkan banyak bangunan rusak serta ambruk, salah satu penyebabnya adalah karena dinding bangunan kurang kaku terhadap getaran yang besar seperti gempa bumi (Murty, Goswami, Vijayanarayanan, & Mehta, 2012). Setiap dinding bangunan mempunyai tingkat kekakuan terhadap getaran yang berbeda-beda, tingkat kekakuan ini dapat ditentukan melalui perbandingan antara besar beban lateral yang diterima dengan *displacement* yang dialami oleh dinding bangunan (Weng, Li, Tu, & Hwang, 2017). Nilai tersebut biasanya diukur dengan memanfaatkan seismograf, tetapi ada masalah yaitu perhitungan tingkat kekakuan dinding yang masih dilakukan secara *manual* sehingga kemungkinan *human error* tinggi, serta adanya *noise* yang kadang menutupi sinyal (Bormann & Wielandt, 2013) sehingga terjadi simpangan akurasi yang besar. Di Indonesia sendiri masih sedikit penelitian yang menguji tentang tingkat kekakuan suatu dinding bangunan terhadap getaran khususnya gempa bumi. Hal ini menjadi penting untuk diteliti, mengingat bencana alam gempa bumi merupakan bencana alam yang cukup sering terjadi, dan terjadi pada kurun waktu yang begitu cepat, sehingga menganalisis dampak gempa bumi terhadap dinding bangunan dapat meminimalisir akibat.

Oleh karena itu, pada penelitian ini dibuatlah sistem yang mampu menentukan tingkat kekakuan dinding terhadap getaran dengan proses perhitungan yang otomatis sehingga dapat mengurangi kesalahan yang disebabkan oleh faktor manusia serta *noise* yang sedikit, besar getaran yang digunakan berdasarkan Skala Intensitas Gempabumi (SIG) BMKG (Anonim, n.d.-a). Objek yang digunakan dalam penelitian ini adalah dinding rumah berbahan beton yang dipasangkan sebuah alat. Sistem alat ini menggunakan sensor IMU untuk mendeteksi data sinyal yang dihasilkan oleh getaran serta mikrokontroler untuk mengotomatisasi perhitungan yang dilakukan. Untuk mengurangi *noise*, diterapkan *filter* menggunakan *Digital Motion Processor* (DMP). Data yang terdeteksi diolah menggunakan *fuzzy logic* untuk menentukan tingkat kekakuan dinding rumah

terhadap getaran, penggunaan *fuzzy logic* dikarenakan besaran SIG BMKG tidak linier dengan PGA-nya.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, rumusan masalah pada penelitian ini adalah pengukuran tingkat kekakuan dinding yang masih *manual* sehingga kemungkinan *human error* tinggi serta *noise* pada seismograf yang kadang menutupi sinyal menyebabkan simpangan akurasi yang besar. Besaran SIG BMKG yang tidak linier dengan PGA-nya mengharuskan penggunaan *fuzzy logic* dalam proses pengotomatisan.

1.3 Batasan Masalah

Beberapa batasan digunakan dalam penelitian ini untuk memudahkan fokus penelitian. Batasan-batasan tersebut adalah sebagai berikut:

1. Hasil akhir perhitungan merupakan tingkat kekakuan dinding dengan nilai antara 0 sampai dengan 4
2. Data yang digunakan dalam penelitian adalah data pengujian pada dinding rumah berbahan beton.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah membuat sistem yang dapat mengukur tingkat kekakuan dinding dengan *noise* yang sedikit menggunakan *filter* pada sensor IMU, serta proses perhitungan yang otomatis menggunakan *fuzzy logic*.

1.5 Manfaat Penelitian

Penelitian ini diharapkan mampu menghasilkan sistem penentuan tingkat kekakuan dinding yang baik, sehingga dapat digunakan untuk menganalisis dampak gempa bumi terhadap dinding bangunan serta meminimalisir akibat melalui pencegahan dini.

1.6 Sistematika Penulisan

Garis besar dari penulisan laporan penelitian skripsi ini, yaitu:

BAB I: PENDAHULUAN

Bab ini berisi tentang latar belakang masalah yang dikaji, rumusan masalah, batasan masalah pada penelitian, tujuan penelitian, dan manfaat penelitian yang dilakukan serta sistematika penulisan laporan penelitian.

BAB II: TINJAUAN PUSTAKA

Bab ini berisi tentang penelitian-penelitian terdahulu yang berhubungan dengan perancangan otomatisasi sistem deteksi tingkat kekakuan dinding beton berbasis sensor IMU menggunakan *Fuzzy Logic*.

BAB III: LANDASAN TEORI

Bab ini berisi tentang dasar teori yang mencakup teori-teori mengenai gempa bumi, skala MMI, SIG, PGA, pengaruh beban lateral, IMU, *sensor fusion*, DMP, sistem *fuzzy*, FIS Sugeno, *node*, serta *confusion matrix*.

BAB IV: ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisi tentang tahapan perancangan sistem mulai dari analisis sistem, rancangan perangkat keras dan lunak, rancangan sistem, serta rencana pengujian.

BAB V: IMPLEMENTASI

Bab ini berisi implementasi sistem berdasarkan analisis dan perancangan sistem yang ada pada BAB IV, serta pengujian sistem.

BAB VI: HASIL DAN PEMBAHASAN

Bab ini berisi hasil penelitian yang dilakukan berupa hasil pengujian fungsional dan hasil pengujian sistem pada objek, beserta pembahasan dari data-data yang didapatkan dari pengamatan yang telah dilakukan.

BAB VII: KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari hasil penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.

BAB II TINJAUAN PUSTAKA

Untuk mendukung penelitian “Otomatisasi Sistem Deteksi Tingkat Kekakuan Dinding Beton Berbasis Sensor IMU menggunakan Metode *Fuzzy*”, penulis melakukan studi literatur dari berbagai penelitian terdahulu sebagai salah satu acuan dalam melakukan penelitian sehingga penulis dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Dari penelitian terdahulu, penulis tidak menemukan penelitian dengan judul yang sama seperti judul penelitian penulis. Namun penulis mengangkat beberapa penelitian sebagai referensi dalam memperkaya bahan kajian pada penelitian penulis. Berikut merupakan penelitian terdahulu berupa beberapa jurnal terkait dengan penelitian yang dilakukan penulis.

Weng et al. (2017) dalam penelitiannya mencoba memprediksikan kurva beban lateral terhadap *displacement* pada dinding beton bertulang yang gagal dalam bergeser. Paper ini mengusulkan kurva dengan tiga titik balik yaitu: titik retak, titik kekuatan, dan titik keruntuhan. Dengan proses perhitungan yang disederhanakan, model yang diusulkan terbukti cocok untuk aplikasi dalam praktik teknik dan sukses memprediksikan kurva beban lateral terhadap *displacement* pada dinding beton bertulang.

Kumar & Venkat (2016) melakukan penelitian terhadap efek gaya lateral pada dinding geser pracetak. Penelitian ini dilakukan pada empat zona seismik dan empat zona angin yang berbeda. Fokus penelitian adalah efek beban lateral pada momen di luar dinding, gaya aksial, gaya geser, pergeseran lantai, *drift* maksimum, dan gaya tarik pada dinding geser. Hasil penelitian di plot pada grafik.

Saputri, Ramadhani, & Adhitama (2019) dalam penelitiannya menggunakan *fuzzy* Sugeno untuk pengambilan keputusan dalam penjadwalan dan pengingat *service* sepeda motor. Solusi yang ditawarkan pada penelitian ini adalah aplikasi *mobile* yang dikhususkan untuk pengguna sepeda motor dalam melakukan perawatan rutin sebagai penjadwal dan pengingat. Presentase keakuratan *system* dengan perhitungan *manual* 100% sama dengan perhitungan *system*. Kesamaan dengan penelitian ini adalah metode Sugeno dan *output* akhir yang juga

diklasifikasikan. Hal yang ditiru dalam penelitian ini adalah proses untuk pembuatan sistem *fuzzynya*.

Simanjuntak, Suharyanto, & Khairiyah (2018) dalam penelitiannya menggunakan logika *fuzzy* dengan metode Sugeno orde nol untuk membuat sistem pengambilan keputusan guna melakukan penilaian terhadap kompetensi karyawan di PT Schneider Batam. Penilaian yang dilakukan meliputi motif, pengetahuan, keterampilan, dan konsep diri. Langkah pertama penyelesaian penilaian kompetensi karyawan dengan menggunakan metode Sugeno yaitu menentukan variabel *input* dan variabel *output* yang merupakan himpunan tegas, langkah kedua yaitu mengubah variabel input menjadi himpunan *fuzzy* dengan proses fuzzifikasi. Kesamaan dengan penelitian ini adalah metode Sugeno dan metode pengujian akurasi. Hal yang ditiru dalam penelitian ini adalah proses pembuatan sistem *fuzzynya* dan metode pengujian akurasi yang menggunakan *confusion matrix*.

Setiawati, Budiman, & Soesanto (2016) dalam penelitiannya menggunakan *Fuzzy Inference System* Takagi-Sugeno-Kang (Metode Sugeno) untuk diterapkan pada sistem pakar dalam memberikan diagnosa penyakit gigi. Kesimpulan yang didapatkan dari penelitian ini yaitu metode *fuzzy inference system* metode Takagi-Sugeno-Kang dapat diterapkan pada sistem pakar gigi. Kesamaan dengan penelitian ini adalah penggunaan metode Sugeno Orde Nol. Hal yang ditiru dalam penelitian ini adalah proses pembuatan sistem *fuzzynya*.

Musaab, Sulle, & Minarso (2017) dalam penelitiannya menggunakan *Digital Motion Processor* (DMP) untuk melakukan proses *filter* dan perhitungan AHRS yang akurat secara independen dengan mengurangi proses perhitungan pada mikrokontroler. Dalam paper ini, diskusi berfokus pada teknik DMP dan perbandingan AHRS. Hasil penelitian menunjukkan bahwa pengukuran posisi sudut menggunakan DMP memiliki *noise* lebih sedikit daripada pengukuran langsung sensor *accelerometer* dan *gyroscope*. AHRS yang diperoleh dari DMP memiliki hasil yang serupa dengan hasil perhitungan algoritma AHRS Mahony dan algoritma AHRS Madgwick. Desain AHRS menggunakan DMP mampu mengendalikan *quadrotor* dengan baik.

Nirmal et al. (2016) pada penelitiannya melakukan pemodelan dan analisis *noise* pada sensor IMU, lalu menerapkan *filter* dan *sensor fusion* untuk meningkatkan kinerjanya. Ditemukan bahwa kinerja IMU menurun seiring waktu karena akumulasi kesalahan yang berbeda. Filter penghalusan seperti *moving average filter* dan *Savitzky-Golay filter* digunakan untuk mencoba menghilangkan *noise* frekuensi tinggi, hasil dari *filter* ini dapat memfilter beberapa *noise* frekuensi tinggi tetapi tidak memuaskan. Distribusi *noise* acak juga ditemukan pada IMU, dengan menggunakan *kalman filter*, *noise* dapat dihapus secara *realtime*.

Yuliani & Saputra (2016) melakukan penelitian terkait kolaborasi *kalman filter* dengan *complementary filter* untuk mengoptimasi hasil sensor *gyroscope* dan *accelerometer*. Paper ini membahas tiga model kolaborasi, model 1 diperoleh dari nilai rata-rata kedua *filter*. Model 2 menggunakan nilai yang diperoleh dari hasil *kalman filter* yang akan difilter kembali oleh *complementary filter*. Model 3 menggunakan cara yang sama dengan model 2, perbedaannya pada model 3 hasil dari *complementary filter* yang akan difilter kembali oleh *kalman filter*. Hasil yang terbaik pada ketiga model ini adalah pada model 3.

Rangkuman perbandingan penelitian terdahulu yang digunakan sebagai acuan dalam penelitian ini ditunjukkan pada Tabel 2.1.

Tabel 2.1 Perbandingan penelitian yang pernah dilakukan

No.	Penulis	Judul	Keterangan
1.	(Weng et al., 2017)	<i>Prediction of the Lateral Load-Displacement Curves for Reinforced Concrete Squat Walls Failing in Shear</i>	Memprediksikan kurva beban lateral terhadap <i>displacement</i> pada dinding beton bertulang yang gagal dalam bergeser
2.	(Kumar & Venkat, 2016)	<i>Effect of Lateral Forces on Precast Shear Wall</i>	Melakukan penelitian terhadap efek gaya lateral pada momen di luar dinding, gaya aksial, gaya geser, pergeseran lantai, <i>drift</i> maksimum, dan gaya tarik pada dinding geser
3.	(Saputri et al., 2019)	Logika <i>Fuzzy</i> Sugeno untuk Pengambilan Keputusan dalam Penjadwalan dan	Menggunakan <i>fuzzy</i> Sugeno untuk pengambilan keputusan dalam penjadwalan dan

		Pengingat <i>Service Sepeda Motor</i>	pengingat <i>service</i> sepeda motor
4.	(Simanjuntak et al., 2018)	<i>Fuzzy Sugeno</i> untuk menentukan Penilaian Kompetensi Karyawan PT. Schneider Batam	Menggunakan logika <i>fuzzy</i> dengan metode Sugeno orde nol untuk membuat sistem pengambilan keputusan
5.	(Setiawati et al., 2016)	Penerapan <i>Fuzzy Inference System</i> Takagi-Sugeno-Kang pada Sistem Pakar Diagnosa Penyakit Gigi	Menggunakan Fuzzy Inference System Metode Sugeno untuk diterapkan pada sistem pakar dalam memberikan diagnosa penyakit gigi
6.	(Musaab et al., 2017)	<i>Design of AHRS for Quadrotor Control using Digital Motion Processor</i>	Menggunakan <i>Digital Motion Processor</i> (DMP) untuk melakukan proses <i>filter</i> dan perhitungan AHRS yang akurat secara independen dengan mengurangi proses perhitungan pada mikrokontroler
7.	(Nirmal et al., 2016)	<i>Noise modeling and analysis of an IMU-based attitude sensor: improvement of performance by filtering and sensor fusion</i>	Melakukan pemodelan dan analisis <i>noise</i> pada sensor IMU, lalu menerapkan <i>filter</i> dan <i>sensor fusion</i> untuk meningkatkan kinerjanya
8.	(Yuliani & Saputra, 2016)	Kolaborasi <i>Kalman Filter</i> dengan <i>Complementary Filter</i> untuk Mengoptimasi Hasil Sensor <i>Gyroscope</i> dan <i>Accelerometer</i>	Melakukan penelitian terkait kolaborasi <i>kalman filter</i> dengan <i>complementary filter</i> untuk mengoptimasi hasil sensor <i>gyroscope</i> dan <i>accelerometer</i>

BAB III

LANDASAN TEORI

3.1 Gempa Bumi

Gempa bumi adalah suatu peristiwa alam dimana terjadi getaran pada permukaan bumi akibat adanya pelepasan energi secara tiba-tiba dari pusat gempa. Energi yang dilepaskan tersebut merambat melalui tanah dalam bentuk gelombang getaran. Gelombang getaran yang sampai ke permukaan bumi disebut gempa bumi (Mulia, 2015).

Di bawah kerak bumi terdapat lapisan lunak terbentuk dari sebuah batuan panas yang lumer. Kerak bumi terbentuk dari nikel dan besi dengan bagian yang padat ditengahnya. Kerak tersebut bisa mencapai ketebalan 70 km di bawah barisan pengunungan terbesar di dunia. Kebanyakan gempa bumi berasal dari kerak bumi. Berdasarkan kedalaman dari posisi gempa, gempa dapat dikategorikan atas tiga kategori (Anonim, 2018b):

1. Gempa dangkal, (hiposenternya terletak pada kedalaman 0 – 65 km)
2. Gempa sedang, (hiposenternya terletak pada kedalaman 65 – 200 km)
3. Gempa dalam, (hiposenternya terletak pada kedalaman > 200 km).

Sedangkan menurut sumber terjadinya gempa, gempa dikelompokkan menjadi:

1. Gempa tektonik yaitu gempa bumi yang berasal dari pergeseran lapisan-lapisan batuan sepanjang bidang patahan di dalam bumi.
2. Gempa vulkanik yaitu gempa bumi yang berasal dari pergerakan magma karena aktifitas gunung api.
3. Gempa longsor atau runtuh yaitu gempa bumi yang terjadi karena aktifitas runtuh pada daerah pertambangan atau daerah tanah longsor.
4. Gempa buatan yaitu getaran gempa bumi yang terjadi karena adanya aktivitas manusia di kulit bumi yang menyebabkan getaran yang cukup kuat.

3.2 Skala MMI (Modified Mercalli Intensity)

Skala Mercalli adalah satuan untuk mengukur kekuatan gempa bumi. Satuan ini diciptakan oleh seorang vulkanologis dari Italia yang bernama Giuseppe Mercalli pada tahun 1902. Skala Mercalli terbagi menjadi dua belas pecahan

berdasarkan informasi dari orang-orang yang selamat dari gempa tersebut dan juga dengan melihat serta membandingkan tingkat kerusakan akibat gempa bumi tersebut. Oleh itu skala Mercalli adalah sangat subjektif dan kurang tepat dibanding dengan perhitungan magnitudo gempa yang lain. Oleh karena itu, saat ini penggunaan Skala Richter lebih luas digunakan untuk mengukur kekuatan gempa bumi. Tetapi skala Mercalli yang dimodifikasi, pada tahun 1931 oleh ahli seismologi Harry Wood dan Frank Neumann masih sering digunakan terutama apabila tidak terdapat peralatan *seismometer* yang dapat mengukur kekuatan gempa bumi di tempat kejadian (Anonim, n.d.-b). Skala MMI yang digunakan oleh BMKG ditunjukkan pada Tabel 3.1.

Tabel 3.1 Skala MMI BMKG

I. Tidak dirasakan	Getaran tidak dirasakan kecuali dalam keadaan luarbiasa oleh beberapa orang.
II. Lemah	Getaran dirasakan oleh beberapa orang, benda-benda ringan yang digantung bergoyang.
III. Lemah	Getaran dirasakan nyata dalam rumah. Terasa getaran seakan-akan ada truk berlalu.
IV. Ringan	Pada siang hari dirasakan oleh orang banyak dalam rumah, di luar oleh beberapa orang, gerabah pecah, jendela/pintu berderik dan dinding berbunyi.
V. Sedang	Getaran dirasakan oleh hampir semua penduduk, orang banyak terbangun, gerabah pecah, barang-barang terpelanting, tiang-tiang dan barang besar tampak bergoyang, bandul lonceng dapat berhenti.
VI. Kuat	Getaran dirasakan oleh semua penduduk. Kebanyakan semua terkejut dan lari keluar, plester dinding jatuh dan cerobong asap pada pabrik rusak, kerusakan ringan.
VII. Sangat kuat	Tiap-tiap orang keluar rumah. Kerusakan ringan pada rumah-rumah dengan bangunan dan konstruksi yang baik. Sedangkan pada bangunan yang konstruksinya kurang baik terjadi retak-retak bahkan hancur, cerobong asap pecah. Terasa oleh orang yang naik kendaraan.
VIII. Parah	Kerusakan ringan pada bangunan dengan konstruksi yang kuat. Retak-retak pada bangunan dengan konstruksi kurang baik, dinding dapat lepas dari rangka rumah, cerobong asap pabrik dan monumen-monumen roboh, air menjadi keruh.
IX. Hebat	Kerusakan pada bangunan yang kuat, rangka-rangka rumah menjadi tidak lurus, banyak retak. Rumah tampak agak

	berpindah dari pondamennya. Pipa-pipa dalam rumah putus.
X. Ekstrem	Bangunan dari kayu yang kuat rusak, rangka rumah lepas dari pondamennya, tanah terbelah rel melengkung, tanah longsor di tiap-tiap sungai dan di tanah-tanah yang curam.
XI. Ekstrem	Bangunan-bangunan hanya sedikit yang tetap berdiri. Jembatan rusak, terjadi lembah. Pipa dalam tanah tidak dapat dipakai sama sekali, tanah terbelah, rel melengkung sekali.
XII. Ekstrem	Hancur sama sekali, Gelombang tampak pada permukaan tanah. Pemandangan menjadi gelap. Benda-benda terlempar ke udara.

3.3 Skala Intensitas Gempabumi (SIG)

SIG adalah Skala Intensitas Gempabumi. Skala ini menyatakan dampak yang ditimbulkan akibat terjadinya gempabumi. Skala Intensitas Gempabumi (SIG-BMKG) digagas dan disusun dengan mengakomodir keterangan dampak gempabumi berdasarkan tipikal budaya atau bangunan di Indonesia. Skala ini disusun lebih sederhana dengan hanya memiliki lima tingkatan yaitu I-V.

SIG-BMKG diharapkan bermanfaat untuk digunakan dalam penyampaian informasi terkait mitigasi gempabumi dan atau respon cepat pada kejadian gempabumi merusak. Skala ini dapat memberikan kemudahan kepada masyarakat untuk dapat memahami tingkatan dampak yang terjadi akibat gempabumi dengan lebih baik dan akurat (Anonim, n.d.-a). Tabel 3.2 menunjukkan skala intensitas gempabumi yang dibuat oleh BMKG.

Tabel 3.2 Skala Intensitas Gempabumi BMKG

Skala SIG BMKG	Warna	Deskripsi Sederhana	Deskripsi Rinci	Skala MMI	PGA (gal)
I	Putih	TIDAK DIRASAKAN (Not Felt)	Tidak dirasakan atau dirasakan hanya oleh beberapa orang tetapi terekam oleh alat.	I-II	< 2.9
II	Hijau	DIRASAKAN (Felt)	Dirasakan oleh orang banyak tetapi tidak menimbulkan kerusakan. Benda-benda ringan yang	III-V	2.9-88

			digantung bergoyang dan jendela kaca bergetar.		
III	Kuning	KERUSAKAN RINGAN (Slight Damage)	Bagian non struktur bangunan mengalami kerusakan ringan, seperti retak rambut pada dinding, genteng bergeser ke bawah dan sebagian berjatuhan.	VI	89-167
IV	Jingga	KERUSAKAN SEDANG (Moderate Damage)	Banyak Retakan terjadi pada dinding bangunan sederhana, sebagian roboh, kaca pecah. Sebagian plester dinding lepas. Hampir sebagian besar genteng bergeser ke bawah atau jatuh. Struktur bangunan mengalami kerusakan ringan sampai sedang.	VII-VIII	168-564
V	Merah	KERUSAKAN BERAT (Heavy Damage)	Sebagian besar dinding bangunan permanen roboh. Struktur bangunan mengalami kerusakan berat. Rel kereta api melengkung.	IX-XII	> 564

3.4 Peak Ground Acceleration (PGA)

Peak ground acceleration (PGA) merupakan percepatan tanah maksimum yang terjadi selama gempa bumi di suatu lokasi. PGA sama dengan amplitudo percepatan absolut terbesar yang tercatat pada *accelerogram* di suatu lokasi saat terjadi gempa bumi tertentu (Douglas, 2003). Gempa bumi umumnya terjadi di ketiga arah (x, y, dan z). Oleh karena itu, PGA sering dibagi menjadi komponen horisontal dan vertikal (Xie, 2015).

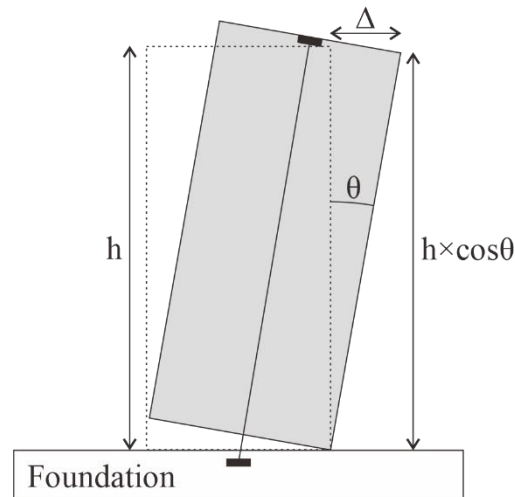
Peak horizontal acceleration (PHA) adalah jenis percepatan tanah yang paling umum digunakan dalam aplikasi teknik. PGA horisontal umumnya lebih besar daripada yang berada di arah vertikal tetapi ini tidak selalu benar, terutama yang dekat dengan gempa bumi besar. Dalam gempa bumi, kerusakan bangunan dan infrastruktur lebih terkait dengan gerakan tanah, yang ukurannya merupakan

ukuran PGA, dan bukan besarnya gempa itu sendiri. Untuk gempa bumi sedang, PGA merupakan penentu kerusakan yang cukup baik (Anonim, 2014).

PGA diukur dalam satuan Galileo - Gal (cm/s^2) atau g - gaya gravitasi. $1g = 9,81 \text{ m/s}^2$; $1g = 981 \text{ Gal}$; $1 \text{ Gal} = 0,01 \text{ m/s}^2$ (Anonim, 2018a).

3.5 Pengaruh Beban Lateral

Ketika mengalami beban lateral, seperti gempa bumi, deformasi lentur dari tulang dinding akan terkonsentrasi pada celah tunggal yang terbuka di dasar dinding, seperti yang ditunjukkan pada Gambar 3.1. Pengangkatan di dasar dinding menyebabkan perpindahan vertikal dan rotasi di lokasi koneksi dinding ke lantai (Henry, Ingham, & Sritharan, 2012).



Gambar 3.1 Pengaruh Beban Lateral

Jika diasumsikan tidak ada retakan (*crack*) pada dinding, maka *displacement* (Δ) dapat dihitung dengan memanfaatkan rumus Pythagoras seperti berikut:

$$\Delta = \sqrt{(h)^2 - (h \times \cos\theta)^2} \quad (3.1)$$

Drift ratio merupakan persentase dari perpindahan lateral dinding bagian atas dibanding tinggi bangunan. *Drift ratio* merupakan indikator yang berguna untuk menentukan besar kerusakan pada dinding. Rumus untuk menghitung *drift ratio* adalah sebagai berikut (Guitierrez & Alpizar, 2004):

$$DR = \frac{\Delta}{h} \times 100\% \quad (3.2)$$

Dalam *Eurocode 8: Design of structures for earthquake resistance* (Lubkowski & Duan, 2001), disebutkan bahwa batas terbesar *drift ratio* pada dinding saat diberi getaran seismik adalah $0,010h$ atau 1% agar tidak runtuh.

Nilai *drift ratio* berbanding lurus dengan nilai PGA, dikarenakan *drift ratio* berbanding lurus dengan gaya lateral (Kumar & Venkat, 2016) dan gaya lateral berbanding lurus dengan percepatan lateral dalam rumus: $F = ma$ (Roselli, 2011). Karena itu, besar PGA untuk menyebabkan *drift ratio* sebesar 1% dapat dihitung dengan persamaan berikut:

$$PGA_{DR1} = PGA \times \frac{1\%}{DR_{peak}} \quad (3.3)$$

Keterangan:

PGA = Percepatan tanah maksimum saat getaran seismik

PGA_{DR1} = Besar nilai PGA untuk menyebabkan *drift ratio* sebesar 1%

DR_{peak} = *Drift ratio* maksimum pada dinding saat diberi getaran seismik.

Lorant (2016) menyebutkan bahwa bangunan yang dapat menahan akselerasi sebesar 0,50g merupakan bangunan yang baik.

3.6 Inertial Measurement Unit (IMU)

Sensor *Inertial Measurement Unit* (IMU) adalah sebuah perangkat elektronik yang terdiri dari *accelerometer*, *gyroscope*, dan *magnetometer*. Dengan mengkombinasikan ketiga sensor tersebut, orientasi, gaya gravitasi, dan percepatan linier juga bisa diukur (Christiansen & Shalamov, 2017).

Sensor *accelerometer* akan mengukur perubahan akselerasi pada tiga arah yang berbeda (x, y, dan z), gaya gravitasi juga akan mempengaruhi *output* sensor ini. Nilai vektor gravitasi dapat diukur dengan menghilangkan pembacaan akselerasi pada sensor. Sebaliknya, nilai percepatan linier dapat diukur dengan menghilangkan pembacaan gaya gravitasi pada sensor (Kostiainen & Shalamov, 2019).

Gyroscope akan mengukur kecepatan sudut relatif terhadap dirinya sendiri. Nilai posisi sudut dapat diukur dengan mengintegalkan *raw output* dari *gyroscope* yang berupa kecepatan sudut, tetapi integral akan menyebabkan *drift* pada nilai *output* (Kostiainen & Pozdnyakov, 2019).

Magnetometer adalah sensor medan magnet, yang berarti bahwa tanpa pengaruh magnet yang kuat di dekatnya, ia akan merasakan medan magnet bumi, yang lebih atau kurang mengarah ke utara, tetapi bukan utara yang sebenarnya. Selama tidak ada medan magnet yang bergerak disekitarnya, pembacaan *magnetometer* cukup stabil (Kostiainen & Bhaumik, 2019).

3.7 Sensor Fusion

Sensor fusion adalah penggabungan data sensorik atau data yang berasal dari sumber yang berbeda sehingga informasi yang dihasilkan memiliki ketidakpastian yang lebih sedikit daripada yang mungkin terjadi ketika sumber-sumber ini digunakan secara individual. Istilah pengurangan ketidakpastian dalam hal ini dapat berarti lebih akurat, lebih lengkap, atau lebih dapat diandalkan (Elmenreich, 2002). *Sensor fusion* dapat diaplikasikan pada sensor IMU untuk mengukur orientasi, gaya gravitasi, dan percepatan linier (Christiansen & Shalamov, 2017).

Quaternion adalah bilangan kompleks empat dimensi (qw , qx , qy , dan qz) yang dapat digunakan untuk mewakili orientasi kerangka koordinat dalam ruang tiga dimensi. Nilai ini dihitung dengan melakukan *fusion* pada *accelerometer* dan *gyroscope* (Madgwick, 2010).

Gaya gravitasi dapat diukur dengan melakukan *fusion* pada *accelerometer* dan *gyroscope* menggunakan *complementary filter* atau perhitungan *quaternion*. *Fusion* dilakukan untuk mengurangi *noise* dan menghilangkan pengukuran akselerasi pada *accelerometer* agar hanya menyisakan data vektor gravitasi (Christiansen & Shalamov, 2017).

Euler angles adalah tiga sudut untuk menggambarkan orientasi benda kaku pada sistem koordinat tetap. Nilai ini dapat dicari dengan perhitungan *quaternion* atau vektor gravitasi. *Euler angles* biasanya dilambangkan sebagai ψ , θ , ϕ atau *yaw*, *pitch*, *roll* (Weisstein, n.d.). Berikut adalah rumus untuk menghitung *yaw*, *pitch*, dan *roll* dari perhitungan *quaternion* (Madgwick, 2010):

$$\psi = \arctan\left(\frac{2qxqy - 2qwqz}{2qw^2 + 2qx^2 - 1}\right) \quad (3.4)$$

$$\theta = -\sin^{-1}(2qxqz + 2qwqy) \quad (3.5)$$

$$\varphi = \arctan\left(\frac{2qyqz - 2qwqx}{2qw^2 + 2qz^2 - 1}\right) \quad (3.6)$$

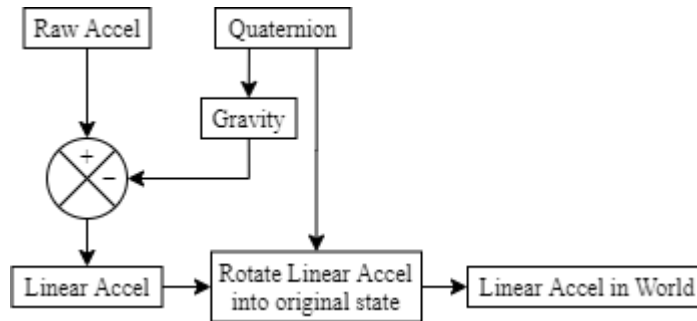
Berikut adalah rumus untuk menghitung *pitch* dan *roll* dari nilai vektor gravitasi (Pedley, 2013):

$$\theta = \arctan\left(\frac{-GravityX}{\sqrt{GravityY^2 + GravityZ^2}}\right) \quad (3.7)$$

$$\varphi = \arctan\left(\frac{GravityY}{GravityZ}\right) \quad (3.8)$$

Linear acceleration diukur dengan cara mengurangi data *raw output* dari *accelerometer* dengan data vektor gravitasi agar hanya terbaca nilai akselerasi tanpa pengaruh gaya gravitasi (Christiansen & Shalamov, 2017).

Linear accel in world adalah *linear acceleration* yang menggunakan koordinat bumi sebagai kerangka acuan. Nilai ini didapatkan dengan memposisikan sensor sejajar dengan tanah, lalu merotasikan *linear acceleration* ke *state* awal setiap terjadi perubahan orientasi pada sensor (Baker, 2007). Diagram *blok linear accel in world* ditunjukkan pada Gambar 3.2.



Gambar 3.2 Diagram Blok Linear Accel in World

3.8 Digital Motion Processor (DMP)

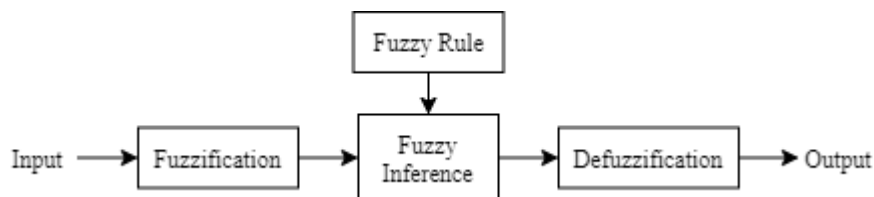
Digital Motion Processor (DMP) adalah unit pemrosesan internal yang ada dalam perangkat MPU. InvenSense menjelaskan, DMP seperti semacam ALU (*Arithmetic Logic Unit*) yang dibangun dengan set instruksi yang dirancang untuk operasi matematika 3D yang diperlukan untuk perhitungan orientasi (Rowberg, 2012).

DMP dapat memproses *3-axis* (*gyro*), *6-axis* (*gyro* dan *accel*), atau *9-axis* (*gyro*, *accel*, dan *compass*) *sensor fusion*. *Sensor fusion* akan menghasilkan *output*

berupa *quaternion* (qw , qx , qy , dan qz). Banyak data lain yang dapat diturunkan dari data *quaternion*, diantaranya adalah *heading*, *rotational matrix*, *euler angles*, *linear acceleration*, dan *gravity vector* (Anonim, 2017).

3.9 Sistem Fuzzy

Sistem *fuzzy* dibangun dengan beberapa tahapan tergantung metode yang digunakan, karena banyak metode untuk membangun sistem *fuzzy*. Namun, secara garis besar dapat disimpulkan seperti ditunjukkan pada Gambar 3.3 (Wang, 1997):



Gambar 3.3 Diagram Blok Sistem Fuzzy

1. Fuzzifikasi

Fuzzifikasi didefinisikan sebagai pemetaan dari himpunan tegas ke himpunan *fuzzy*. Kriteria yang harus dipenuhi pada proses fuzzifikasi adalah semua anggota pada himpunan tegas harus termuat dalam himpunan *fuzzy*, tidak terdapat gangguan pada *input* sistem *fuzzy* yang digunakan, dan harus bisa mempermudah perhitungan pada sistem *fuzzy*.

2. Aturan Fuzzy

Aturan yang digunakan pada himpunan *fuzzy* adalah aturan *if-then*. Aturan *fuzzy IF-THEN* merupakan pernyataan yang direpresentasikan dengan:

$$IF < \text{proposisi fuzzy} > THEN < \text{proposisi fuzzy} >$$

3. Inferensi Fuzzy

Inferensi *fuzzy* merupakan tahap evaluasi pada aturan *fuzzy*. Tahap evaluasi dilakukan berdasarkan penalaran dengan menggunakan *input fuzzy* dan aturan *fuzzy* sehingga diperoleh *output* berupa himpunan *fuzzy*.

4. Defuzzifikasi

Defuzzifikasi merupakan proses yang berkebalikan dengan proses pada fuzzifikasi. Defuzzifikasi didefinisikan sebagai pemetaan dari himpunan *fuzzy* ke himpunan tegas. Himpunan *fuzzy* yang dimaksud disini adalah hasil *output* yang

diperoleh dari hasil inferensi. Pada proses defuzzifikasi ada tiga kriteria yang harus dipenuhi yaitu masuk akal, perhitungannya sederhana, dan kontinu.

3.10 Fuzzy Inference System Sugeno

Fuzzy Inference System (FIS) Sugeno pertama kali dikenalkan oleh Takagi-Sugeno-Kang pada tahun 1985. *Output* dari *fuzzy* Sugeno adalah hasil defuzzifikasi yang berupa konstanta atau persamaan linier. Pada metode Sugeno, agregasi berupa singleton-singleton. Sistem *fuzzy* Sugeno memperbaiki kelemahan yang dimiliki oleh sistem *fuzzy* murni untuk menambah suatu perhitungan matematika sederhana sebagai bagian *THEN* (Kusumadewi, 2002).

Menurut Wang (1997), keuntungan menggunakan metode Sugeno ini diantaranya adalah:

- a) Lebih efisien dalam masalah komputasi
- b) Bekerja paling baik untuk teknik-teknik linear
- c) Bekerja paling baik untuk teknik optimasi dan adaptif
- d) Menjamin kontinuitas permukaan *output*.

Ada dua model metode Sugeno yaitu model *fuzzy* sugeno orde nol dan model *fuzzy* sugeno orde satu (Abdussaid, 2018).

Bentuk umum model *fuzzy* Sugeno orde nol adalah:

$$IF\ x\ is\ A\ AND\ y\ is\ B\ THEN\ z\ is\ k$$

Bentuk umum model *fuzzy* Sugeno orde satu adalah:

$$IF\ x\ is\ A\ AND\ y\ is\ B\ THEN\ z = f(x, y)$$

Pada *fuzzy* model Sugeno, proses defuzzifikasi menggunakan rata-rata tertimbang (*weighted average*). Metode ini mengambil nilai rata-rata dengan menggunakan pembobotan berupa derajat keanggotaan. Berikut merupakan persamaan matematikanya (Abdussaid, 2018):

$$y^* = \sum \frac{\mu(y)y}{\mu(y)'} \quad (3.9)$$

dimana y merupakan nilai *crisp* dan $\mu(y)$ merupakan derajat keanggotaan dari nilai *crisp* y .

3.11 Pengertian Node

Node (Latin *nodus*, ‘simpul’) adalah salah satu titik sambungan, titik redistribusi, atau titik akhir komunikasi (beberapa terminal peralatan). Definisi *node* tergantung kepada jaringan dan protokol lapisan tersebut. *Node* jaringan fisik adalah perangkat elektronik aktif yang terpasang kedalam jaringan, dan mampu membuat, menerima, atau mengirimkan informasi melalui saluran komunikasi (Amaliyah, 2019).

3.12 Confusion Matrix

Confusion matrix adalah suatu metode yang biasanya digunakan untuk melakukan perhitungan akurasi pada konsep *data mining* atau sistem pendukung keputusan. Pada pengukuran kinerja menggunakan *confusion matrix*, terdapat empat istilah sebagai representasi hasil proses klasifikasi. Keempat istilah tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN) (Anonim, 2018c). Tabel *confusion matrix* ditunjukkan pada Gambar 3.4.

		True Values	
		True	False
Prediction	True	TP Correct result	FP Unexpected result
	False	FN Missing result	TN Correct absence of result

Gambar 3.4 Confusion Matrix (Anonim, 2018c)

Akurasi adalah ukuran seberapa dekat suatu pengukuran dengan nilai yang sebenarnya dari kuantitas yang diukur. Berikut adalah rumus menghitung akurasi berdasarkan *confusion matrix*:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (3.10)$$

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1 Analisis Sistem

Pengukuran tingkat kekakuan dinding terhadap getaran biasanya dilakukan dengan memanfaatkan seismograf untuk mencari parameter-parameter yang dibutuhkan. Tetapi, ada masalah yaitu proses akuisisi data dan perhitungan masih dilakukan secara *manual* serta adanya *noise* pada seismograf yang kadang menyebabkan sinyal yang diinginkan tertutupi olehnya. Oleh karena itu, diperlukan instrument lain yang dapat mengatasi masalah pada penggunaan seismograf dalam pengukuran tingkat kekakuan dinding.

Pada penelitian skripsi ini, akan buat sistem yang mampu mengatasi masalah pada penggunaan seismograf dalam pengukuran tingkat kekakuan dinding. Parameter utama yang dibutuhkan dalam proses penentuan tingkat kekakuan dinding adalah *displacement* ujung atas dinding dan *ground acceleration*, jadi dibutuhkan dua *node*, yaitu pada ujung atas dan ujung bawah dinding. Seismograf memiliki luaran berupa kecepatan linier, jadi *displacement* dan *ground acceleration* dicari secara manual menggunakan diferensial dan integral pada keluaran *node* bagian atas dan bawah. Instrumen yang dibuat juga harus dapat melakukan perhitungan *displacement* dan *ground acceleration*, maka dari itu dibutuhkan sensor yang dapat mencari parameter tersebut. Luaran dari sensor juga perlu ditapis terlebih dahulu untuk mendapatkan sinyal yang rendah *noise*. Besar kerusakan pada dinding bergantung pada *drift ratio* yang merupakan persentase *displacement* dibanding tinggi bangunan, menurut *handbook Eurocode 8* (Lubkowski & Duan, 2001) batas *drift* terbesar saat mengalami getaran seismik adalah 1% agar bangunan tidak runtuh, maka perhitungan tingkat kekakuan dinding membutuhkan besar nilai PGA untuk menyebabkan *drift ratio* sebesar 1% (PGA_{DR1}). Nilai ini yang akan diproses untuk menghasilkan *score* tingkat kekakuan dinding dengan nilai antara 0 sampai dengan 4 berdasarkan Skala Intensitas Gempabumi BMKG (Anonim, n.d.-a). Dikarenakan besaran SIG BMKG tidak linier dengan PGA-nya, perhitungan *score* tidak akan akurat jika hanya menggunakan persamaan regresi, maka diperlukan metode lain untuk perhitungan *score* yang akurat. Semua proses tersebut

diatur menggunakan mikrokontroler untuk mengurangi campur tangan manusia dalam perhitungan.

Sistem ini mempunyai beberapa keunggulan dari pengukuran menggunakan seismograf, yaitu mempunyai cara kerja yang sederhana sehingga orang yang awam pada perhitungan tingkat kekakuan dinding tetap bisa menjalankan sistem ini, sinyal yang masuk sudah ditapis sehingga lebih akurat, dan proses perhitungan yang tidak membutuhkan banyak campur tangan manusia sehingga kemungkinan *human error* semakin kecil.

4.2 Rancangan Perangkat Keras

Rancangan perangkat keras yang dibuat disesuaikan dengan fungsi yang dibutuhkan oleh sistem. Perangkat keras utama yang dibutuhkan adalah mikrokontroler, *powerbank*, sensor IMU, *micro SD card*, *micro SD card module*, dan PC. Diagram blok rancangan perangkat keras dapat dilihat pada Gambar 4.1.

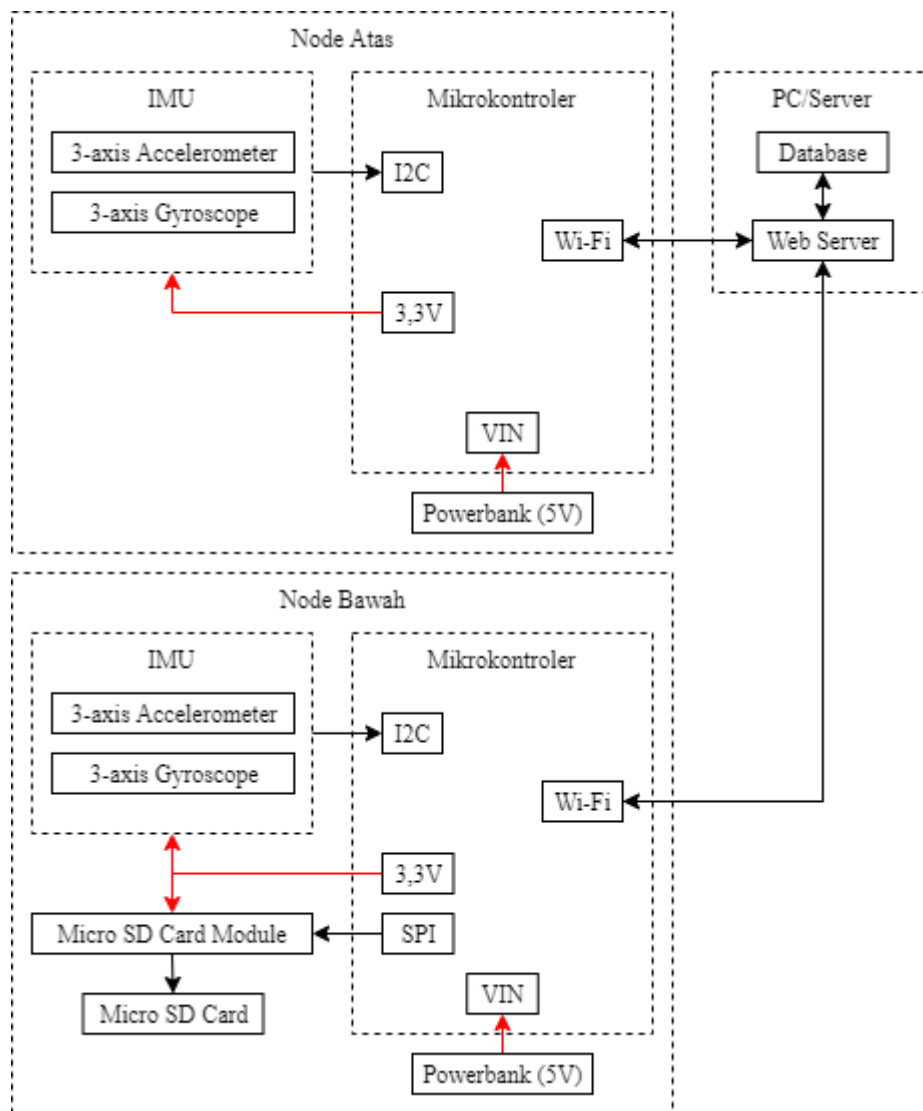
Mikrokontroler yang digunakan pada penelitian skripsi ini adalah ESP32 DEV KIT DOIT dikarenakan mempunyai ukuran yang cukup kecil, spesifikasi prosesor yang memadai (*dual-core* dan *clock frequency* mencapai 240MHz) serta sudah memiliki *wireless connectivity* internal berupa Wi-Fi dan Bluetooth. Sumber daya yang digunakan adalah *powerbank* karena praktis dan tegangannya sesuai dengan mikrokontroler yaitu 5V.

Sensor IMU yang digunakan adalah MPU6050 dengan mempertimbangkan MPU6050 sudah memiliki *3-axis accelerometer*, *3-axis gyroscope*, serta fitur *Digital Motion Processor* (DMP). Sensor IMU dan mikrokontroler dihubungkan menggunakan komunikasi *Inter-Integrated Circuit* (I²C).

Micro SD card digunakan sebagai *data logger* yang dipasang pada *node* bawah, untuk menghubungkannya dengan mikrokontroler dibutuhkan *micro SD card module*. Komunikasi yang digunakan adalah *Serial Peripheral Interface* (SPI).

PC digunakan sebagai *server* yang mempunyai *web server* dan *database*. *Web server* digunakan untuk berkomunikasi dengan mikrokontroler dan membuat GUI. Mikrokontroler dikomunikasikan dengan *web server* melalui Wi-Fi dengan

access point menggunakan *mobile hotspot*. *Database* digunakan untuk menyimpan data dari *web server*.

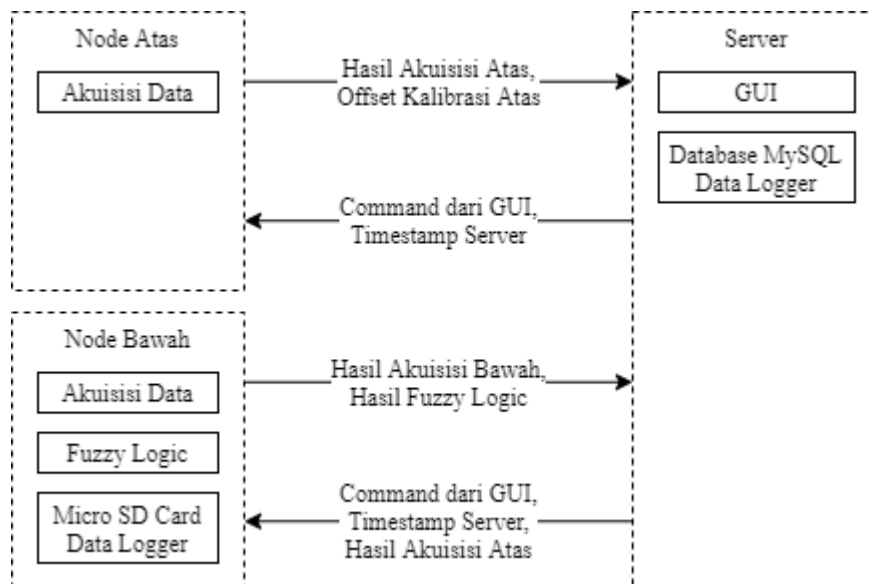


Gambar 4.1 Diagram Blok Rancangan Perangkat Keras

4.3 Rancangan Perangkat Lunak

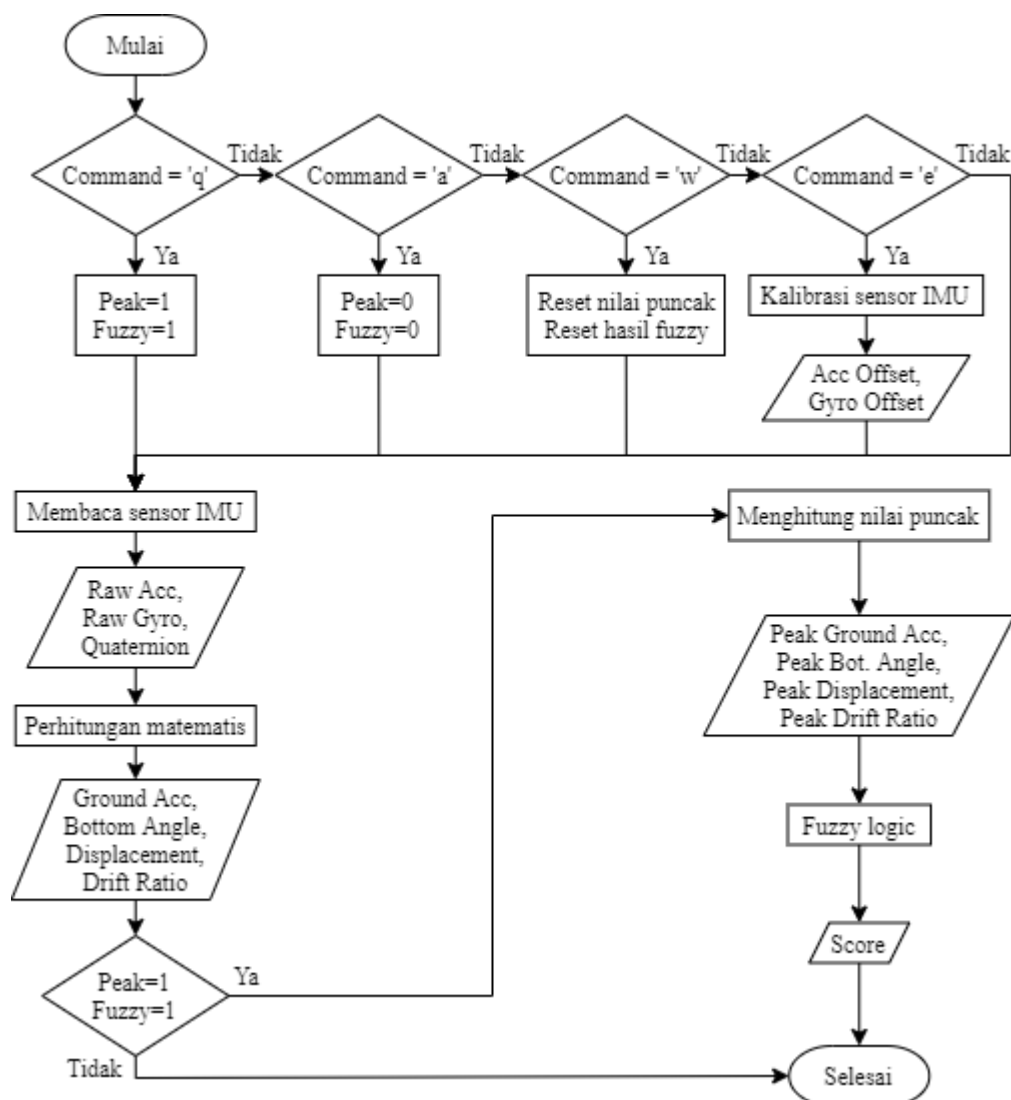
Rancangan perangkat lunak yang dibuat sesuai dengan fungsi yang diharapkan pada sistem. *Node* bawah digunakan untuk melakukan proses akuisisi data, *fuzzy logic*, serta menyimpan data pada *micro SD card*. *Node* atas hanya digunakan untuk melakukan akuisisi data *top inclination angle*, yang nantinya akan dibandingkan dengan *bottom inclination angle* untuk mengetahui ada atau tidaknya retakan (*crack*) pada dinding. *Server* digunakan untuk membuat GUI (penampil data dan pengontrol *node*), menyimpan data pada *database MySQL*, serta menghubungkan *node* bawah dengan *node* atas. Komunikasi antara *node* dengan *server* menggunakan WiFi. Diagram blok rancangan perangkat lunak dapat dilihat pada Gambar 4.2.

Aplikasi pendukung yang dibutuhkan adalah Arduino IDE dan XAMPP (Apache & MySQL). Arduino IDE digunakan untuk memprogram mikrokontroler yang ada pada *node* bawah dan *node* atas. Setiap *node* memiliki dua *core* yang berjalan terpisah, yaitu *core* 1 dan *core* 0. *Core* 1 digunakan untuk proses pengolahan data, sedangkan *core* 0 digunakan untuk berkomunikasi dengan *server*. XAMPP digunakan untuk membuat serta mengatur *web server* dan *database*.



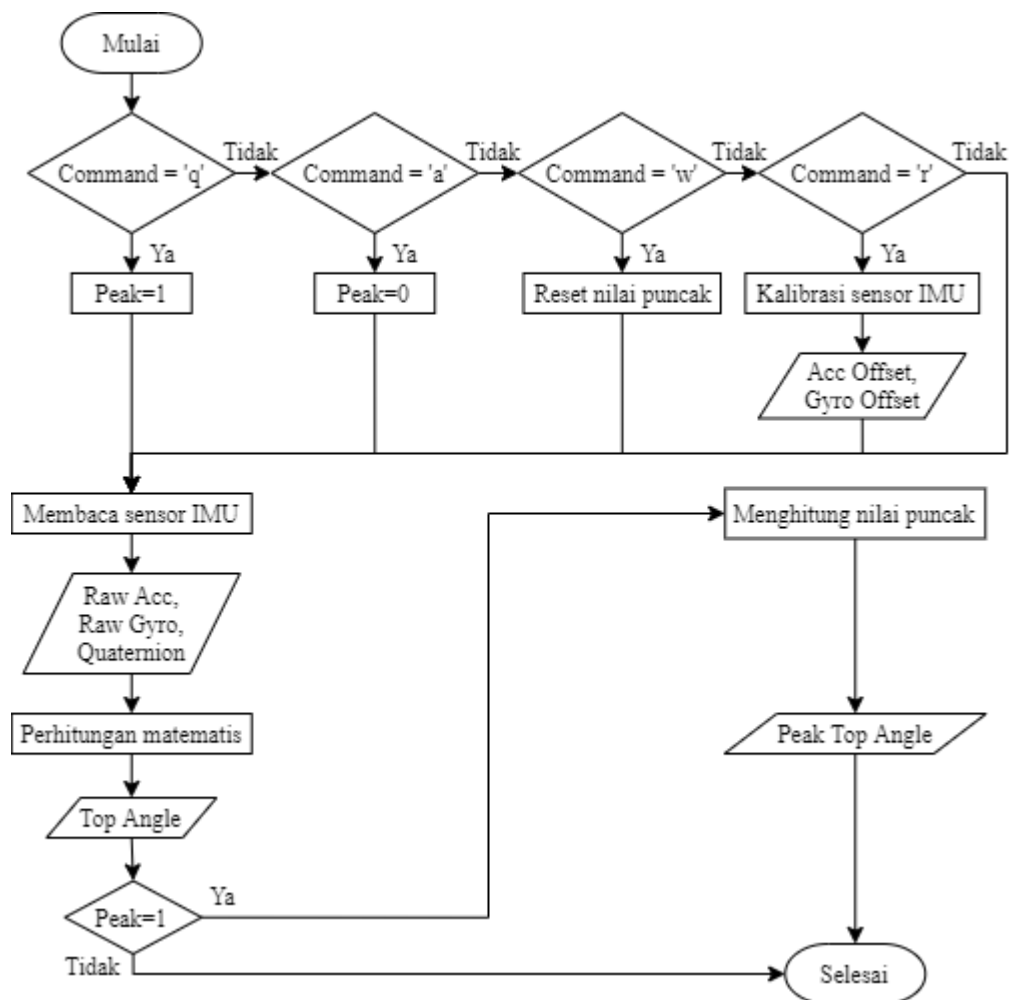
Gambar 4.2 Diagram Blok Rancangan Perangkat Lunak

Core 1 pada *node* bawah digunakan untuk melakukan proses akuisisi data dan *fuzzy logic*. Akuisisi data akan menghasilkan nilai *ground acceleration*, *bottom inclination angle*, *displacement*, *drift ratio*, dan puncak dari data tersebut. *Fuzzy logic* akan menghasilkan nilai *score* tingkat kekakuan dinding terhadap getaran. Perhitungan nilai puncak dan *fuzzy logic* dapat berjalan jika *command* bernilai 'q', dan berhenti saat *command* bernilai 'a'. Proses kalibrasi sensor pada *node* bawah berjalan saat *command* bernilai 'e'. Flowchart program pada core 1 *node* bawah ditunjukkan pada Gambar 4.3.



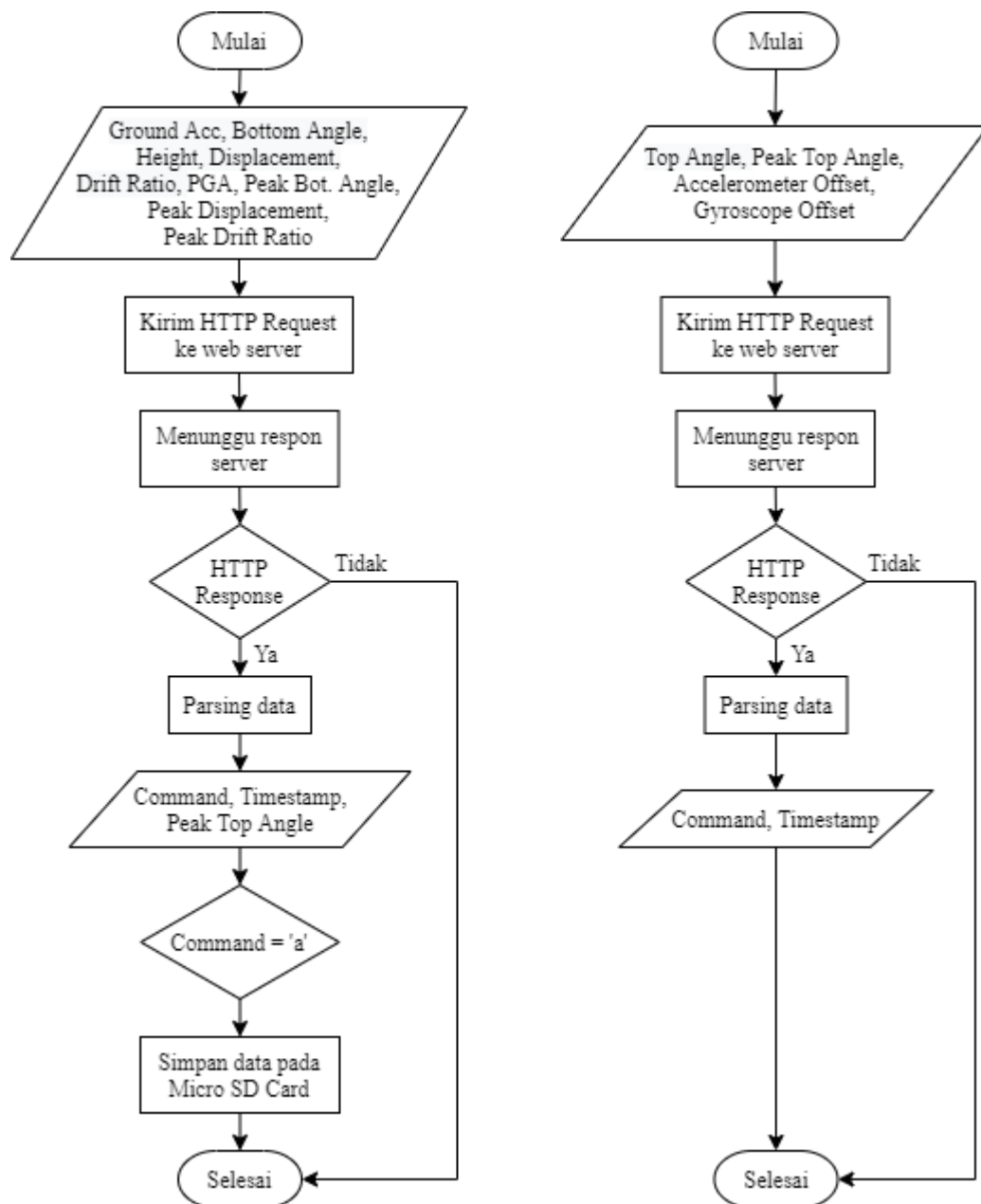
Gambar 4.3 Flowchart Core 1 Node Bawah

Core 1 pada *node* atas hanya digunakan untuk melakukan proses akuisisi *top inclination angle* dan nilai puncaknya, yang nantinya akan dibandingkan dengan *bottom inclination angle* untuk mengetahui ada atau tidaknya retakan pada dinding. Perhitungan nilai puncak dapat berjalan jika *command* bernilai 'q', dan berhenti saat *command* bernilai 'a'. Proses kalibrasi sensor pada *node* atas berjalan saat *command* bernilai 'r'. Flowchart program pada core 1 *node* atas ditunjukkan pada Gambar 4.4.



Gambar 4.4 Flowchart Core 1 Node Atas

Core 0 pada *node* bawah dan atas digunakan untuk berkomunikasi dengan *server*. *Node* bawah juga menggunakan *core 0* untuk proses *data logger* pada *micro SD card*. Data dikirimkan dari *node* ke *server* melalui *HTTP Request*, lalu *node* akan menunggu respon dari *server*. *HTTP Response* yang diterima oleh *node* harus diparse agar data yang dikirim dari *server* tidak bercampur. Jika *command* bernilai 'a' maka *micro SD card data logger* pada *node* bawah akan berjalan. *Flowchart* program pada *core 0 node* bawah dan atas ditunjukkan pada Gambar 4.5.

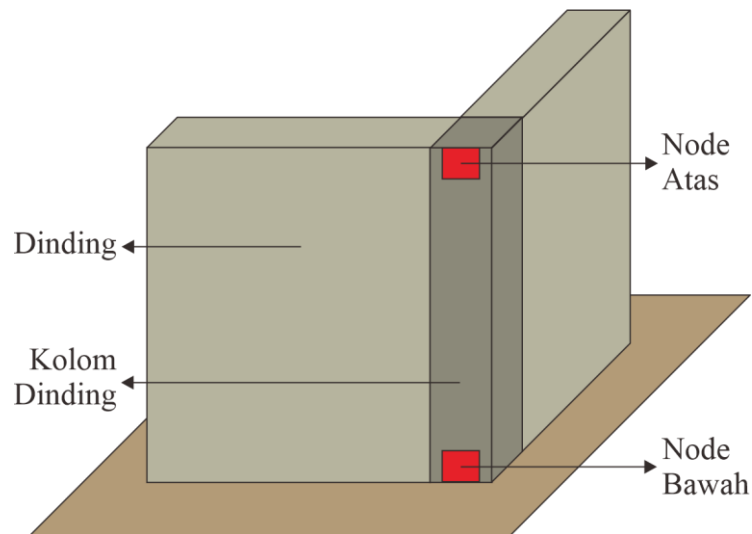


Gambar 4.5 Flowchart Core 0 Node Bawah & Atas

4.4 Rancangan Akuisisi Data

Pada tahap ini data *raw* dari sensor IMU diproses untuk menghasilkan data yang di perlukan. *Output* sensor dapat berupa nilai *accelerometer* (ax, ay, az), nilai *gyroscope* (gx, gy, gz), dan hasil *sensor fusion* yaitu *quaternion* (qw, qx, qy, qz). *Sensor fusion* menggunakan *Digital Motion Processor* (DMP) yang merupakan fitur dari MPU6050, dikarenakan hasil yang sudah baik dan pengkodean yang lebih ringkas.

Terdapat dua *node* yang digunakan pada penelitian ini, yaitu *node* yang berada di ujung bawah kolom dinding (*node* bawah) dan *node* yang berada di ujung atas kolom dinding (*node* atas). Pada tahap akuisisi data, *node* bawah digunakan untuk menghitung nilai *ground acceleration*, *bottom inclination angle*, *displacement*, dan *drift ratio*, sedangkan *node* atas hanya digunakan untuk menghitung *top inclination angle*. Rancangan penempatan *node* ditunjukkan pada Gambar 4.6.



Gambar 4.6 Rancangan Penempatan Node

4.4.1 Kalibrasi sensor

Kalibrasi bertujuan untuk mendapatkan nilai *offset* sensor, sehingga *output* sensor bernilai sama seperti nilai aslinya. Pada sistem ini kalibrasi dilakukan pada nilai *accelerometer* dan *gyroscope*, yang juga akan mempengaruhi *quaternion* karena merupakan hasil *fusion* dari kedua data. Kalibrasi dilakukan dengan cara

menjalankan program kalibrasi saat kondisi sensor diam, posisi sensor saat kalibrasi akan menjadi posisi awal sensor.

4.4.2 Ground acceleration

Ground acceleration (percepatan tanah) dapat dihitung dengan mencari *linear accel in world* pada *node* bawah, yang merupakan titik yang paling dekat dengan tanah. Karena sensor IMU dihadapkan ke depan dinding, maka sumbu yang dipakai adalah sumbu-z. Diagram blok untuk mendapatkanya nilai *linear accel in world* ada pada Gambar 3.2. Perhitungan dengan pendekatan ini diasumsikan dapat menghasilkan nilai percepatan tanah yang mendekati nilai aslinya, karena peletakan sensor pada tanah tidak dimungkinkan.

4.4.3 Inclination angle

Inclination angle (sudut kemiringan) pada dinding merupakan *pitch* atau *roll* yang terukur oleh sensor IMU. Jika sensor dipasang secara vertikal maka *inclination angle* adalah *roll*, jika sensor dipasang secara horisontal maka *inclination angle* adalah *pitch*. Rumus untuk menghitung *pitch* dan *roll* dapat dilihat pada persamaan (3.7) dan persamaan (3.8).

4.4.4 Displacement

Displacement (Δ) dapat dihitung menggunakan rumus pada persamaan (3.1). Tinggi dinding (h) diukur menggunakan meteran dan di deklarasikan secara manual pada program mikrokontroler.

4.4.5 Drift Ratio

Drift ratio (DR) dapat dihitung menggunakan rumus pada persamaan (3.2).

4.4.6 Nilai puncak

Nilai puncak (*peak*) merupakan nilai mutlak terbesar yang terukur oleh *node* bawah maupun *node* atas. Pada *node* bawah akan dicari nilai *peak ground acceleration* (PGA), *peak bottom inclination angle*, *peak displacement*, dan *peak drift ratio*. Sedangkan pada *node* atas hanya akan dicari nilai *peak top inclination angle* yang akan dibandingkan dengan *peak bottom inclination angle* untuk mengetahui ada atau tidaknya retakan (*crack*).

4.5 Rancangan Fuzzy Logic

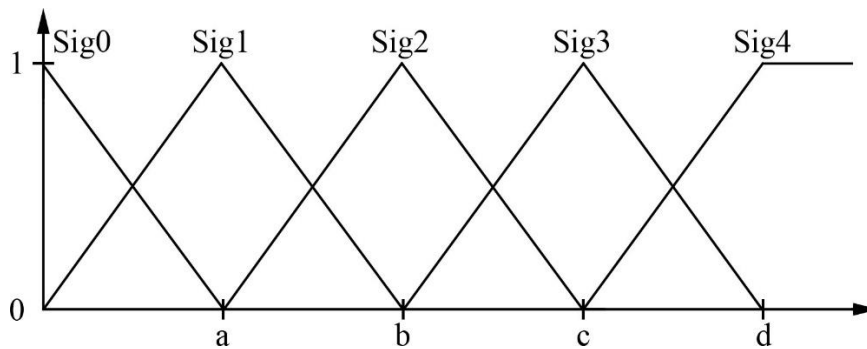
Pada sub bab ini akan dibahas rancangan *fuzzy logic* yang digunakan untuk menentukan tingkat kekakuan dinding terhadap getaran. *Fuzzy logic* digunakan karena *input* dan *output* tidak linier. Model *fuzzy* yang digunakan adalah Sugeno Orde Nol dikarenakan *outputnya* sudah berupa konstanta.

4.5.1 Masukan fuzzy

Masukan yang digunakan pada *fuzzy logic* hanya berjumlah satu, yaitu besar nilai PGA yang dibutuhkan untuk menyebabkan *drift ratio* sebesar 1% (PGA_{DR1}). Nilai tersebut dapat dicari menggunakan rumus pada persamaan (3.3). Perhitungan dengan pendekatan ini diasumsikan dapat menghasilkan nilai PGA_{DR1} yang mendekati nilai sebenarnya, dikarenakan sangat sulit untuk mensimulasikan gempa besar yang dapat menyebabkan *drift ratio* sebesar 1%.

4.5.2 Fuzzifikasi

Dengan berdasarkan Skala Intensitas Gempabumi BMKG (Anonim, n.d.-a), variabel *fuzzy* PGA_{DR1} dibuat memiliki lima himpunan yaitu Sig0, Sig1, Sig2, Sig3, dan Sig4, serta dapat dimodelkan menjadi grafik keanggotaan. Grafik keanggotaan PGA_{DR1} ditunjukkan pada Gambar 4.7.



Gambar 4.7 Grafik Keanggotaan PGA_{DR1}

Fungsi Keanggotaan:

$$\mu_{Sig0}(x) = \begin{cases} \frac{a-x}{a}, & 0 \leq x \leq a \\ 0, & x \geq a \end{cases}$$

$$\mu_{Sig1}(x) = \begin{cases} \frac{x}{a}, & 0 \leq x \leq a \\ \frac{b-x}{b-a}, & a \leq x \leq b \\ 0, & x \geq b \end{cases}$$

$$\mu_{Sig2}(x) = \begin{cases} 0, & x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \end{cases}$$

$$\mu_{Sig3}(x) = \begin{cases} 0, & x \leq b \text{ atau } x \geq d \\ \frac{x-b}{c-b}, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \end{cases}$$

$$\mu_{Sig4}(x) = \begin{cases} 0, & x \leq c \\ \frac{x-c}{d-c}, & c \leq x \leq d \\ 1, & x \geq d \end{cases}$$

dimana $a = 0.0030\text{g}$, $b = 0.0897\text{g}$, $c = 0.1703\text{g}$, $d = 0.5751\text{g}$

4.5.3 Aturan fuzzy

Dari himpunan *fuzzy* yang sudah ada, agar *output* sesuai yang kita harapkan maka ditentukanlah aturan *fuzzy* yang ditunjukkan pada Tabel 4.1.

Tabel 4.1 Komposisi Aturan Fuzzy

PGA _{DR1}	Sig0	Sig1	Sig2	Sig3	Sig4
Score	0	1	2	3	4

4.5.4 Inferensi fuzzy

Proses inferensi *fuzzy* menggunakan metode Sugeno Orde Nol. Karena masukan *fuzzy* hanya berjumlah satu, maka α -predikat dan z dapat langsung ditentukan seperti berikut:

$$\begin{aligned} \alpha_0 &= \mu_{Sig0}(x) & z_1 &= 0 \\ \alpha_1 &= \mu_{Sig1}(x) & z_1 &= 1 \\ \alpha_2 &= \mu_{Sig2}(x) & z_2 &= 2 \\ \alpha_3 &= \mu_{Sig3}(x) & z_3 &= 3 \\ \alpha_4 &= \mu_{Sig4}(x) & z_4 &= 4 \end{aligned}$$

4.5.5 Defuzzifikasi

Proses defuzzifikasi dilakukan menggunakan metode *weighted average* untuk mendapatkan *output* berupa nilai *crisp*. Berikut penerapan rumus *weighted average* dari persamaan (3.9) untuk mendapatkan nilai *score (output)*:

$$Score = \frac{\sum_i^n \alpha_i z_i}{\sum_i^n \alpha_i}$$

$$Score = \frac{\alpha_0(0) + \alpha_1(2) + \alpha_2(5) + \alpha_3(6) + \alpha_4(8)}{\alpha_0 + \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4}$$

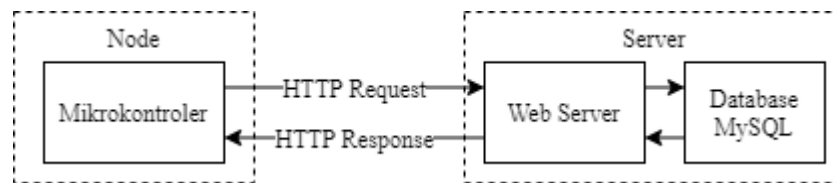
Pada sistem ini hasil akhir dari *fuzzy logic* adalah *score* yang merupakan nilai antara 0 sampai 4. *Score* melambangkan tingkat ketahanan dinding terhadap getaran, besar getaran yang digunakan berdasarkan SIG (Anonim, n.d.-a). Dari *score* yang diperoleh maka dapat diasumsikan jika:

- a) $Score \geq 1$, dinding dapat menahan hingga SIG I
- b) $Score \geq 2$, dinding dapat menahan hingga SIG II
- c) $Score \geq 3$, dinding dapat menahan hingga SIG III
- d) $Score = 4$, dinding dapat menahan hingga SIG IV

Score maksimal adalah 4. Menurut Lorant (2016) bangunan yang dapat menahan akselerasi sebesar 0,50g (SIG IV) merupakan bangunan yang baik.

4.6 Rancangan Komunikasi antara Node dengan Server

Komunikasi antara *node* dengan *server* dilakukan dengan menggunakan WiFi melalui *HTTP Request* dan *HTTP Response*. WiFi dipilih untuk mengurangi *noise* getaran yang tidak diinginkan saat menggunakan kabel dan pengontrolan dengan cara menyentuh *node*. Bagian yang melakukan komunikasi adalah mikrokontroler dan *web server*. Data dikirimkan dari mikrokontroler ke *web server* melalui *HTTP Request*, setelah itu data akan diterima *web server* dan disimpan ke *database MySQL* menggunakan perintah dalam kode PHP. Untuk mengirimkan data dari *server* ke *node*, data yang ingin dikirim harus dipanggil dari *database* ke *web server* terlebih dahulu, baru setelah itu data dikirim ke mikrokontroler melalui *HTTP Response*. Diagram blok rancangan komunikasi antara *node* dengan *server* ditunjukkan pada Gambar 4.8.



Gambar 4.8 Rancangan Komunikasi antara Node dengan Server

4.7 Rancangan Graphical User Interface

Graphical user interface (GUI) digunakan untuk menampilkan data dari *node* dan mengirimkan perintah ke *node*. Pada penelitian skripsi ini, GUI yang akan digunakan berbasis *localhost web server*. Penggunaan *web server* untuk GUI dikarenakan dapat dibuat antarmuka yang menyediakan informasi dan alat kontrol yang lengkap serta dapat diakses oleh banyak *device* secara bersamaan. Penampilan data pada GUI akan mengambil data terbaru dari *database MySQL*, data yang ditampilkan akan *terupdate* secara otomatis jika ada data baru yang diterima *database*. Untuk pengiriman perintah ke *node*, dibuat antarmuka tombol yang jika ditekan akan mengirimkan *char* ke *database*, yang setelahnya *char* akan dikirimkan ke *node* oleh *web server* melalui *HTTP Response*. GUI yang dibuat akan dapat diakses melalui *browser PC* ataupun *mobile* yang tersambung dengan *access point* yang sama dengan *server*.

4.8 Rancangan Data Logger

Data logger dibuat agar proses pencatatan data tidak perlu dilakukan secara *manual* dan nantinya proses analisis data dapat berjalan dengan mudah. Pada penelitian skripsi ini, *data logger* yang akan digunakan berjumlah dua, agar ada *backup* saat salah satu *data logger* mengalami *error*, yaitu *micro SD card* dan *database MySQL*. Data yang disimpan adalah hasil akuisisi data (termasuk *offset kalibrasi*) dan hasil *fuzzy logic*. *Micro SD card* akan menyimpan data dalam bentuk *file .txt* sedangkan *database MySQL* akan menyimpan data dalam tabel.

4.9 Rencana Pengujian Sistem

Rencana pengujian sistem dibuat untuk mengetahui kinerja sistem secara keseluruhan. Rencana pengujian sistem secara keseluruhan ditunjukkan pada Tabel 4.2.

4.9.1 Rancangan pengujian fungsional

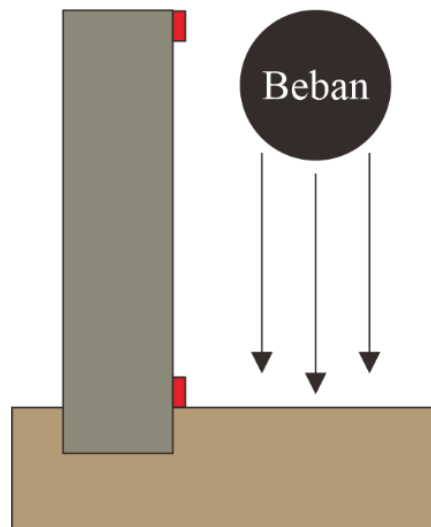
Pengujian fungsional dilakukan untuk mengetahui kinerja sistem per bagian dan memastikan setiap bagian berjalan sesuai dengan yang diharapkan. Pengujian ini terdiri dari pengujian akuisisi data dan pengujian *fuzzy logic*. Setiap bagian akan diuji untuk memastikan sistem berjalan dengan baik.

4.9.2 Rancangan pengujian penentuan tingkat kekakuan dinding rumah berbahan beton terhadap getaran

Pengujian ini dilakukan untuk mengetahui *score* tingkat kekakuan dinding yang diuji terhadap getaran, sekaligus mengetahui ada atau tidaknya retakan pada dinding. *Score* dicari untuk mengetahui batas SIG yang dapat ditahan oleh dinding. *Score* yang diharapkan adalah 4, yang berarti dinding dapat menahan getaran dengan SIG IV, dikarenakan menurut Lorant (2016) bangunan yang dapat menahan akselerasi sebesar 0,50g (sekitar SIG IV) merupakan bangunan yang baik.

Dari pengujian ini juga akan diketahui bagaimana kinerja sistem jika diujikan pada objek yang sesungguhnya, yaitu dinding beton. Presisi dari sistem akan dihitung dan dianalisis untuk menentukan apakah kualitas dari sistem sudah baik atau belum, presisi akan dianggap baik jika standar deviasi dari *output* sistem bernilai kecil. Jika kualitas dari sistem belum baik, maka diperlukan penelitian lanjutan.

Pengujian akan melibatkan dinding rumah satu tingkat dengan bahan beton, *node* dipasang pada kolom dinding bagian atas dan bagian bawah. Untuk membuat getaran pada tanah, dijatuhkan beban dengan berat tertentu dan dari ketinggian tertentu hingga menimbulkan getaran yang akan diukur oleh sensor. Semakin besar getaran yang ditimbulkan oleh beban akan semakin baik, karena perbandingan *noise* terhadap getaran asli akan semakin kecil. Rancangan pengujian pada dinding dapat dilihat pada Gambar 4.9.



Gambar 4.9 Rancangan Pengujian pada Dinding

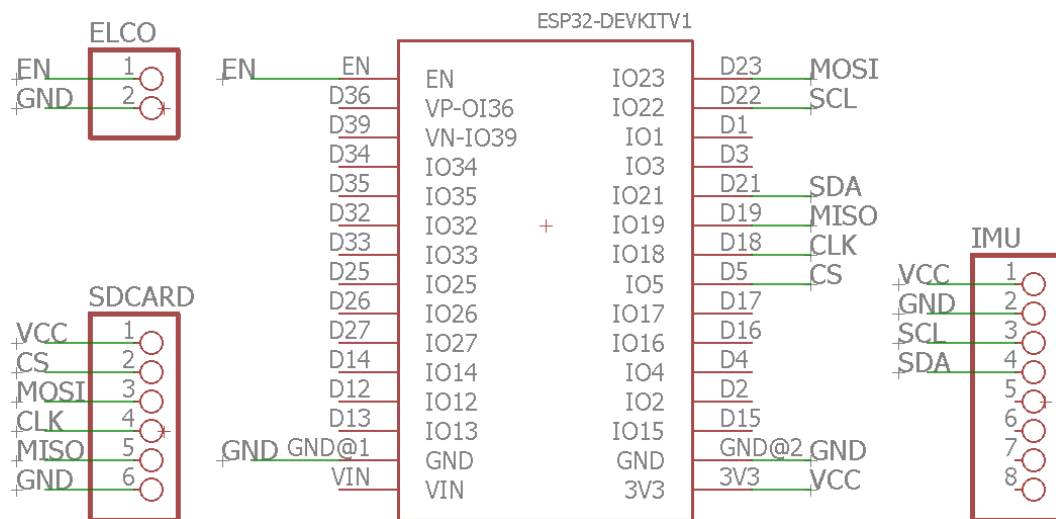
Tabel 4.2 Rencana Pengujian Sistem

No.	Rencana Pengujian	Tujuan Pengujian
1	Pengujian akuisisi data	Mengetahui <i>error</i> dari akuisisi data Mengetahui kecepatan akuisisi data
2	Pengujian <i>fuzzy logic</i>	Menguji akurasi <i>fuzzy logic</i>
3	Pengujian penentuan tingkat kekakuan dinding rumah berbahan beton terhadap getaran	Mengetahui tingkat kekakuan dinding terhadap getaran Mengetahui ada atau tidaknya retakan pada dinding Mengetahui kinerja sistem saat diujikan pada objek sesungguhnya.

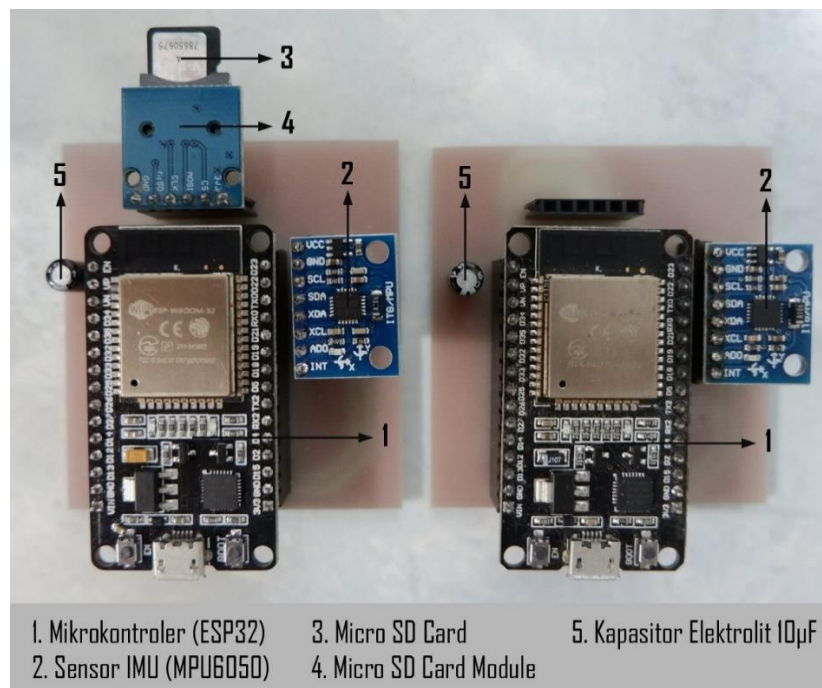
BAB V IMPLEMENTASI

5.1 Implementasi Perangkat Keras

Pada kedua *node*, komponen diletakkan pada sebuah papan elektronik. Papan elektronik didesain menggunakan perangkat lunak Eagle 7.7.0. Papan elektronik yang dibuat berisi soket mikrokontroler (ESP32), soket sensor IMU (MPU6050), dan soket *micro SD card module*. Sumber daya mikrokontroler menggunakan *powerbank* dengan kapasitas 5000 mAh, tegangan DC sebesar 5V, dan arus maksimal 2,4A. Sedangkan sensor IMU dan *micro SD card module* mendapatkan tegangan sebesar 3,3V dari *pinout* mikrokontroler. Kapasitor elektrolit 10 μ F disambungkan diantara pin EN dan GND agar saat mengupload program ke ESP32 tidak perlu menahan tombol BOOT. Skematik dan implementasi papan elektronik ditunjukkan pada Gambar 5.1 dan Gambar 5.2.



Gambar 5.1 Skematik Papan Elektronik



Gambar 5.2 Implementasi Papan Elektronik

Server yang digunakan adalah laptop. Pembuatan *web server* dan *database* menggunakan aplikasi XAMPP. *Server* dan *node* dikomunikasikan melalui Wi-Fi dengan *access point* berupa *mobile hotspot*.

5.2 Implementasi Perangkat Lunak

Program pada *node* bawah dan *node* atas dibuat menggunakan Arduino IDE. *Library* tambahan yang digunakan adalah *arduino-esp32* (Grokhov, Sovani, Jain, Gratton, & Domburg, 2019) dan *I2Cdevlib-MPU6050* (Rowberg, 2020). *Listing* program *node* bawah dapat dilihat pada Lampiran 1, sedangkan *listing* program *node* atas dapat dilihat pada Lampiran 2.

Web server dan *database* dibuat dan diatur pada PC menggunakan aplikasi XAMPP (Apache dan MySQL). *Database* diatur melalui phpMyAdmin dengan cara mengetikkan alamat **localhost/phpmyadmin** pada *browser*. *Web server* diatur dengan membuat file PHP pada folder **\xampp\htdocs**. Kode program dibuat dengan memodifikasi kode program pada artikel *DIY Cloud Weather Station* (Santos, 2019). Struktur *database* pada MySQL dapat dilihat pada Lampiran 3, *listing* program komunikasi *server* dengan *node* ada pada Lampiran 4, *listing*

program GUI ada pada Lampiran 5, dan *listing* fungsi-fungsi yang dipanggil pada *file* .php ada pada Lampiran 6.

5.3 Implementasi Akuisisi Data

Data didapatkan dari sensor IMU yang terpasang pada kedua *node*, yaitu *node* atas dan *node* bawah. *Node* bawah digunakan untuk menghitung nilai *ground acceleration*, *bottom inclination angle*, *displacement*, dan *drift ratio*, sedangkan *node* atas hanya digunakan untuk menghitung nilai *top inclination angle*. Selain itu, kedua *node* juga menghitung nilai puncak (*peak*) dari setiap data yang didapatkan. Implementasi penempatan *node* ditunjukkan pada Gambar 5.3.



Gambar 5.3 Implementasi Pemempatan Node

Program akuisisi data dibuat menggunakan Arduino IDE dengan didasarkan pada *library* I2Cdevlib-MPU6050 (Rowberg, 2020). Program pemanggilan *library* pada bagian *header* ditunjukkan pada Gambar 5.4.

8	#include "I2Cdev.h"
9	#include "MPU6050_6Axis_MotionApps20.h"

16	MPU6050 mpu;

Gambar 5.4 Kode Program Pemanggilan Library I2Cdevlib-MPU6050

5.3.1 Kalibrasi sensor

Kalibrasi sensor dilakukan menggunakan program kalibrasi dari *library* I2Cdevlib-MPU6050 (Rowberg, 2020). Untuk melakukan kalibrasi, sensor harus berada pada posisi diam saat program kalibrasi dijalankan. Kode program kalibrasi sensor ditunjukkan pada Gambar 5.5 dan 5.6.

```

121     if(command=='e') imu_calibration();
      .....
172 void imu_calibration(void)
173 {
174     imuCal = 1;
175     mpu.CalibrateAccel(20);
176     mpu.CalibrateGyro(20);
177     logSDCard();
178     imuCal = 0;
179 }
```

Gambar 5.5 Kode Program Kalibrasi Sensor Node Bawah

```

90     if(command=='r') imu_calibration();
      .....
130 void imu_calibration(void)
131 {
132     mpu.CalibrateAccel(20);
133     mpu.CalibrateGyro(20);
134
135     axOff=mpu.getXAccelOffset();
136     ayOff=mpu.getYAccelOffset();
137     azOff=mpu.getZAccelOffset();
138     gxOff=mpu.getXGyroOffset();
139     gyOff=mpu.getYGyroOffset();
140     gzOff=mpu.getZGyroOffset();
141
142     imuCal_http = 1;
143 }
```

Gambar 5.6 Kode Program Kalibrasi Sensor Node Atas

Pada *node* bawah, agar program kalibrasi berjalan, variabel *command* harus bernilai 'e', agar variabel *command* bernilai 'e', GUI harus mengirimkan 'e' dari *form* "Node command". *Offset* hasil kalibrasi *node* bawah akan disimpan pada *micro SD card*.

Pada *node* atas, agar program kalibrasi berjalan, variabel *command* harus bernilai 'r'. Sama seperti *node* bawah, agar variabel *command* bernilai 'r', GUI harus mengirimkan 'r' dari *form* "Node command". *Offset* hasil kalibrasi *node* atas akan dikirim ke *server* dan disimpan pada *database*.

5.3.2 Ground acceleration

Ground acceleration dapat dihitung dengan mencari *linear accel in world* pada *node* bawah, karena merupakan titik yang paling dekat dengan tanah. Karena sensor IMU dihadapkan ke depan dinding, maka sumbu yang dipakai adalah sumbu-z. *Linear accel in world* dicari menggunakan fitur DMP pada *library* I2Cdevlib-MPU6050 (Rowberg, 2020). Program untuk menghitung *ground acceleration* ditunjukkan pada Gambar 5.7.

```

184     if (mpu.dmpGetCurrentFIFOPacket(fifoBuffer)) {
185         mpu.dmpGetQuaternion(&q, fifoBuffer);
186         mpu.dmpGetAccel(&aa, fifoBuffer);
187         .....
188         mpu.dmpGetGravity(&gravity, &q);
189         mpu.dmpGetLinearAccel(&aaReal, &aa, &gravity);
190         mpu.dmpGetLinearAccelInWorld(&aaWorld, &aaReal, &q);
191         .....
193         accWorldX = float(aaWorld.x)/8192;
194         accWorldY = float(aaWorld.y)/8192;
195         accWorldZ = float(aaWorld.z)/8192;
196         .....
199         ground_acc = accWorldZ;
200         .....
210     }

```

Gambar 5.7 Kode Program Akuisisi Ground Acceleration

5.3.3 Inclination angle

Program akuisisi *inclination angle* dibuat menggunakan fitur DMP pada *library* I2Cdevlib (Rowberg, 2020). Karena sensor dipasang secara vertikal, maka *inclination angle* merupakan nilai *roll* yang dibaca oleh sensor. Kode program untuk menghitung *inclination angle* ditunjukkan pada Gambar 5.8.

```

184     if (mpu.dmpGetCurrentFIFOPacket(fifoBuffer)) {
185         mpu.dmpGetQuaternion(&q, fifoBuffer);
186         .....
188         mpu.dmpGetGravity(&gravity, &q);
189         .....
191         mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
192         .....
196         pitch = -ypr[1]*180/M_PI; // AngleY
197         roll = ypr[2]*180/M_PI; // AngleX
198         .....
200         inc_angle = roll;
201         .....
210     }

```

Gambar 5.8 Kode Program Akuisisi Inclination Angle

5.3.4 Displacement

Kode program perhitungan *displacement* ditunjukkan pada Gambar 5.9. Sebelum dimasukkan ke rumus *displacement*, *inclination angle* dikonversikan ke radian terlebih dahulu pada baris ke 202.

```

202     inc_angle_rad = inc_angle/(180/M_PI);
203     displacement = sqrt(pow(h,2)
204                     -pow(h*cos(inc_angle_rad),2));
205     if (inc_angle_rad<0) displacement=displacement*(-1);

```

Gambar 5.9 Kode Program Akuisisi Displacement

5.3.5 Drift ratio

Kode program untuk menghitung *drift ratio* ditunjukkan pada Gambar 5.10.

```

206     drift_ratio = displacement/h*100;

```

Gambar 5.10 Kode Program Akuisisi Drift Ratio

5.3.6 Nilai puncak

Kode program perhitungan nilai puncak ditunjukkan pada Gambar 5.11 dan Gambar 5.12. Perhitungan nilai puncak dapat dimulai dengan mengirimkan *command* 'q' dengan cara menekan tombol "Start" pada GUI. Selain itu, perhitungan nilai puncak dapat dihentikan dengan mengirim *command* 'a', serta dapat di *reset* dengan mengirimkan *command* 'w'.

```

118     if(command=='q') start = true;
119     if(command=='a') start = false;
120     if(command=='w') {peak_reset(); fuzzy_reset();}
121     .....
135     if (start==true) {peak_calc(); fuzzy_start();}
136     .....
217 void peak_calc(void)
218 {
219     if (abs(ground_acc)>pga) pga=abs(ground_acc);
220     if (abs(inc_angle)>inc_angle_peak)
221         {inc_angle_peak=abs(inc_angle);}
222     if (abs(displacement)>displacement_peak)
223         {displacement_peak=abs(displacement);}
224     if (abs(drift_ratio)>drift_ratio_peak)
225         {drift_ratio_peak=abs(drift_ratio);}
226 }
227
228 void peak_reset(void)
229 {
230     pga = 0;
231     inc_angle_peak = 0;
232     displacement_peak = 0;
233     drift_ratio_peak = 0;
234 }

```

Gambar 5.11 Kode Program Nilai Puncak Node Bawah

```

87   if(command=='q') start = true;
88   if(command=='a') start = false;
89   if(command=='w') peak_reset();
.....
94   if (start==true) {peak_calc();}
.....
145 void peak_calc(void)
146 {
147     if (abs(inc_angle)>inc_angle_peak)
148         {inc_angle_peak=abs(inc_angle);}
149 }
150
151 void peak_reset(void)
152 {
153     inc_angle_peak = 0;
154 }

```

Gambar 5.12 Kode Program Nilai Puncak Node Atas

5.4 Implementasi Fuzzy Logic

Program *fuzzy logic* menggunakan metode Sugeno orde nol, kode program merupakan modifikasi dari kode program pada artikel “Implementasi Fuzzy Logic Controller untuk Kontrol Kecepatan Motor DC” (Wijaya, 2015). Penerapan *fuzzy logic* metode Sugeno orde nol untuk menentukan tingkat kekakuan dinding terhadap getaran pada program Arduino IDE ditunjukkan pada Gambar 5.13.

```

118   if(command=='q') start = true;
119   if(command=='a') start = false;
120   if(command=='w') {peak_reset(); fuzzy_reset();}
.....
135   if (start==true) {peak_calc(); fuzzy_start();}
.....
236 void fuzzy_start(void)
237 {
238     pga_dr1 = pga/drift_ratio_peak;
239
240     x=pga_dr1; a=0.0030; b=0.0897; c=0.1703; d=0.5751;
241
242     if (x>=0&& x<=a) {sig[0] = (a-x)/(a);} // SIG 0
243     else if (x>=a) {sig[0] = 0;}
244
245     if (x>=0&& x<=a) {sig[1] = x/a;} // SIG I
246     else if (x>=a&& x<=b) {sig[1] = (b-x)/(b-a);}
247     else if (x>=b) {sig[1] = 0;}
248
249     if (x<=a||x>=c) {sig[2] = 0;} // SIG II
250     else if (x>=a&& x<=b) {sig[2] = (x-a)/(b-a);}
251     else if (x>=b&& x<=c) {sig[2] = (c-x)/(c-b);}
252
253     if (x<=b||x>=d) {sig[3] = 0;} // SIG III
254     else if (x>=b&& x<=c) {sig[3] = (x-b)/(c-b);}
255     else if (x>=c&& x<=d) {sig[3] = (d-x)/(d-c);}
256

```

```

257  if (x<=c) {sig[4] = 0;} // SIG IV
258  else if (x>=c&&x<=d) {sig[4] = (x-c)/(d-c);}
259  else if (x>=d) {sig[4] = 1;}
260
261  z0=0; z1=1; z2=2; z3=3; z4=4;
262
263  score = (sig[0]*z0+sig[1]*z1+sig[2]*z2
264          +sig[3]*z3+sig[4]*z4)
265          /(sig[0]+sig[1]+sig[2]+sig[3]+sig[4]);
266  }
267
268  void fuzzy_reset(void)
269  {
270      pga_dr1 = 0;
271      score = 0;
272  }

```

Gambar 5.13 Kode Program Fuzzy Logic

Sama seperti perhitungan nilai puncak, untuk menjalankan fungsi **fuzzy_start** tombol “Start” pada GUI harus ditekan agar nilai *command* menjadi ‘q’. Masukan *fuzzy* adalah nilai PGA saat *drift ratio* sebesar 1%, untuk mendapatkan nilai tersebut diterapkan rumus yang ada pada baris 238. Fuzzyfikasi untuk menentukan derajat keanggotaan himpunan *fuzzy* ada pada baris 240-259. Defuzzifikasi untuk mendapatkan nilai *score* ada pada baris 263-265. Pada baris 268-272 terdapat fungsi **fuzzy_reset** yang dapat dipanggil dengan menekan tombol “Reset” pada GUI agar *command* bernilai ‘w’.

5.5 Implementasi Komunikasi antara Node dengan Server

Node dan *server* dikomunikasikan melalui Wi-Fi dengan *access point* berupa *mobile hotspot*. Bagian yang melakukan komunikasi adalah mikrokontroler dan *web server*. Mikrokontroler akan mengirimkan *HTTP Request* dan menerima *HTTP Response*, sedangkan *web server* akan menerima *HTTP Request* dan mengembalikan *HTTP Response*.

5.5.1 Komunikasi HTTP pada mikrokontroler

Pada mikrokontroler, komunikasi dengan *server* akan dijalankan pada *core* 0 supaya tidak mengganggu proses pengolahan data yang berjalan pada *core* 1 (dalam fungsi *loop()*). Untuk menjalankan program secara *dual-core* digunakan fitur FreeRTOS pada ESP32, yang memungkinkan untuk menangani beberapa task secara paralel pada *core* yang berbeda. Kode program yang digunakan bersumber dari *library* *arduino-esp32* (Grokhotkov et al., 2019). Kode untuk pembuatan *task*

dengan *core* 0 pada *node* bawah ditunjukkan pada Gambar 5.14, pada *node* atas pembuatan task dilakukan pada kode baris ke 14 dan 68-76.

```

17 TaskHandle_t Task1;
   .....
96 //create a task
97 xTaskCreatePinnedToCore(
98     Task1code,      /* Task function. */
99     "Task1",        /* name of task. */
100    10000,           /* Stack size of task */
101    NULL,            /* parameter of the task */
102    1,               /* priority of the task */
103    &Task1,          /* Task handle */
104    0);              /* pin task to core 0 */

```

Gambar 5.14 Kode Program Pembuatan Task Node Bawah

Program komunikasi akan ditulis pada fungsi Task1code agar berjalan pada *core* 0. Pengiriman *request* pada *web server* menggunakan metode HTTP GET. Data yang dikirimkan melalui *HTTP Request* ke *server* adalah hasil akuisisi data dan *fuzzy logic*. Pada *node* atas, data *offset* hasil kalibrasi juga dikirimkan ke *server* untuk disimpan pada *database* MySQL. Selain mengirimkan data ke *server*, mikrokontroler juga menerima data yang dikirimkan dari *server* melalui *HTTP Response*, data yang diterima harus melalui proses *parsing* terlebih dahulu agar tidak bercampur. Kode program komunikasi HTTP pada mikrokontroler *node* bawah dan *node* atas ditunjukkan pada Gambar 5.15 dan Gambar 5.16, kode program merupakan modifikasi dari kode program “BasicHttpClient” pada *library* *arduino-esp32* (Grokhotkov et al., 2019).

```

108 void Task1code( void * pvParameters ) {
109     for(;;) {
110         komunikasi_http();
111         if(command=='a') logSDCard();
112     }
113 }
   .....
274 void komunikasi_http(void) {
275     if((wifiMulti.run() == WL_CONNECTED)) {
276
277         HTTPClient http;
278
279         if(command=='x') command_send='0';
280         else command_send='x';
281
282         http.begin("http://192.168.43.148/nodebawah.php?gacc="
283             + String(ground_acc,4)
284             + "&angle=" + String(inc_angle,3)
285             + "&drift=" + String(drift_ratio,3)

```

```

286     + "&pga=" + String(pga,4)
287     + "&anglepeak=" + String(inc_angle_peak,3)
288     + "&height=" + String(h)
289     + "&displacepeak=" + String(displacement_peak)
290     + "&driftpeak=" + String(drift_ratio_peak,3)
291     + "&pgadr1=" + String(pga_dr1,4)
292     + "&score=" + String(score)
293     + "&command=" + String(command_send));
294
295     int httpCode = http.GET();
296     if(httpCode > 0) {
297         if(httpCode == HTTP_CODE_OK) {
298             String payload = http.getString();
299
300             // Parsing data
301             int j = 0;
302             dt[j] = "";
303             for (int k = 58; k < payload.length()-1; k++) {
304                 if ((payload[k] == '#' ) || (payload[k] == '*')) {
305                     j++;
306                     dt[j] = "";
307                 }
308                 else {
309                     dt[j] = dt[j] + payload[k];
310                 }
311             }
312             command = dt[1][0];
313             server_time = dt[2];
314             top_angle_peak = dt[3].toFloat();
315
316             .....
321         }
322     }
323     http.end();
324 }
325 }

```

Gambar 5.15 Kode Program Komunikasi HTTP Node Bawah

```

80 void Task1code( void * pvParameters ) {
81     for(;;){
82         komunikasi_http();
83     }
84 }
85
86 .....
156 void komunikasi_http(void) {
157     if((wifiMulti.run() == WL_CONNECTED)) {
158
159         HTTPClient http;
160
161         if(command=='x') command_send='0';
162         else command_send='x';
163
164         if (imuCal_http==1) {imuCal_http=0; imuCal_send=1;}
165         else if (imuCal_http==0) {imuCal_send=0;}
166
167         http.begin("http://192.168.43.148/nodeatas.php?angle="
168             + String(inc_angle,3)

```



```

169 + "&anglepeak=" + String(inc_angle_peak,3)
170 + "&command=" + String(command_send)
171 + "&imuCal=" + String(imuCal_send)
172 + "&axOff=" + String(axOff)
173 + "&ayOff=" + String(ayOff)
174 + "&azOff=" + String(azOff)
175 + "&gxOff=" + String(gxOff)
176 + "&gyOff=" + String(gyOff)
177 + "&gzOff=" + String(gzOff));
178
179 int httpCode = http.GET();
180 if(httpCode > 0) {
181     if(httpCode == HTTP_CODE_OK) {
182         String payload = http.getString();
183
184         // Parsing data
185         int j = 0;
186         dt[j] = "";
187         for (int k = 49; k < payload.length()-1; k++) {
188             if ((payload[k] == '#' || (payload[k] == '*')) {
189                 j++;
190                 dt[j] = "";
191             }
192             else {
193                 dt[j] = dt[j] + payload[k];
194             }
195         }
196         command = dt[1][0];
197         server_time = dt[2];
198
199         .....
202     }
203 }
204 http.end();
205 }
206 }

```

Gambar 5.16 Kode Program Komunikasi HTTP Node Atas

5.5.2 Komunikasi HTTP pada web server

Komunikasi HTTP pada *web server* diatur dalam *file nodebawah.php* dan *nodeatas.php*, kode ini merupakan modifikasi dari kode pada artikel *DIY Cloud Weather Station* (Santos, 2019).

Data yang berupa *request* HTTP GET dari *node* akan diterima dan disimpan dalam bentuk variabel pada *web server*. Kode program penerima *HTTP Request* dari *node* ditunjukkan pada Gambar 5.17 dan 5.18.

```

4  if($_GET['gacc'] != '' and $_GET['angle'] != '' and
5      $_GET['drift'] != '' and $_GET['pga'] != '' and
6      $_GET['anglepeak'] != '' and $_GET['height'] != '' and
7      $_GET['displacepeak'] != '' and
8      $_GET['driftpeak'] != '' and $_GET['pgadr1'] != '' and
9      $_GET['score'] != '' and $_GET['command'] != '') {

```

```

10 $gndacc = $_GET['gacc'];
11 $angle = $_GET['angle'];
12 $drift = $_GET['drift'];
13 $pga = $_GET['pga'];
14 $anglepeak = $_GET['anglepeak'];
15 $height = $_GET['height'];
16 $displacepeak = $_GET['displacepeak'];
17 $driftpeak = $_GET['driftpeak'];
18 $pgadr1 = $_GET['pgadr1'];
19 $score = $_GET['score'];
20 $command = $_GET['command'];

```

Gambar 5.17 Kode Penerima HTTP Request nodebawah.php

```

4  if($_GET['pidCal'] != '' and
5      $_GET['axOff'] != '' and $_GET['ayOff'] != '' and
6      $_GET['azOff'] != '' and $_GET['gxOff'] != '' and
7      $_GET['gyOff'] != '' and $_GET['gzOff'] != '') {
8      $pidCal = $_GET['pidCal'];
9      $axOff = $_GET['axOff'];
10     $ayOff = $_GET['ayOff'];
11     $azOff = $_GET['azOff'];
12     $gxOff = $_GET['gxOff'];
13     $gyOff = $_GET['gyOff'];
14     $gzOff = $_GET['gzOff'];
15     .....
34    if($_GET['angle'] != '' and $_GET['anglepeak'] != '' and
35        $_GET['command'] != '') {
36        $angle = $_GET['angle'];
37        $anglepeak = $_GET['anglepeak'];
38        $command = $_GET['command'];

```

Gambar 5.18 Kode Penerima HTTP Request nodeatas.php

Data yang sudah menjadi variabel pada *web server* akan dimasukkan ke *database* MySQL dengan menggunakan kode perintah PHP. Kode program untuk menyimpan variable pada *web server* ke *database* ditunjukkan pada Gambar 5.19 dan Gambar 5.20.

```

22 // Create connection
23 $conn = new mysqli($servername, $username,
24     $password, $dbname);
25 .....
30 $sql = "INSERT INTO botdata (gndacc, angle, drift, pga,
31     anglepeak, height,
32     displacepeak, driftpeak,
33     pgadr1, score)
34     VALUES ('".$gndacc."', '".$angle."', '".$drift."',
35         '".$pga."', '".$anglepeak."', '".$height."',
36         '".$displacepeak."', '".$driftpeak."',
37         '".$pgadr1."', '".$score."')";
38
39 $conn->query($sql);
40
41 $last_reading = getLastCommand();

```

```

42 $last_reading_command = $last_reading["command"];
43
44 if ($last_reading_command!='x') {
45     if ($command=='x') {
46         $sql = "INSERT INTO botcommand (command)
47             VALUES ('" . $command . "')";
48
49         $conn->query($sql);
50         $last_reading_command='x';
51     }
52 }

```

Gambar 5.19 Web Server ke Database nodebawah.php

```

16 if ($pidCal=='1') {
17     $conn = new mysqli($servername, $username,
18         $password, $dbname);
19     if ($conn->connect_error) {
20         die("Connection failed: " . $conn->connect_error);
21     }
22
23     $sql= "INSERT INTO topcalib (pidCal, axOff, ayOff,
24         azOff, gxOff, gyOff, gzOff)
25     VALUES ('".$pidCal."', '".$axOff."', '".$ayOff."',
26         '".$azOff."', '".$gxOff."',
27         '".$gyOff."', '".$gzOff."')";
28
29     $conn->query($sql);
30     $conn->close();
31 }
32
33 .....
40 $conn = new mysqli($servername, $username,
41     $password, $dbname);
42 if ($conn->connect_error) {
43     die("Connection failed: " . $conn->connect_error);
44 }
45
46 $sql = "INSERT INTO topdata (angle, anglepeak)
47 VALUES ('" . $angle . "', '" . $anglepeak . "')";
48
49 $conn->query($sql);
50
51 $last_reading = getLastCommand2();
52 $last_reading_command = $last_reading["command"];
53
54 if ($last_reading_command!='x') {
55     if ($command=='x') {
56         $sql = "INSERT INTO topcommand (command)
57         VALUES ('" . $command . "')";
58
59         $conn->query($sql);
60         $last_reading_command='x';
61     }
62 }

```

Gambar 5.20 Web Server ke Database nodeatas.php

Pada akhir program, *web server* mengembalikan *HTTP Response* yang nantinya akan diterima oleh *node*. Kode program untuk mengirimkan *HTTP Response* dapat dilihat pada Gambar 5.21 dan Gambar 5.22.

```

54  $last_reading = getLastReadings();
55  $last_reading_time = $last_reading["time"];
56
57  $last_reading = getLastReadings2();
58  $last_reading_anglepeak2 = $last_reading["anglepeak"];
59
60  echo "<br>Command&Time&TopAngle: *";
61  echo $last_reading_command;
62  echo "*";
63  echo $last_reading_time;
64  echo "*";
65  echo $last_reading_anglepeak2;
66  echo "#";

```

Gambar 5.21 Kirim HTTP Response nodebawah.php

```

64  $last_reading = getLastReadings2();
65  $last_reading_time = $last_reading["time"];
66
67  echo "<br>Command&Time: *";
68  echo $last_reading_command;
69  echo "*";
70  echo $last_reading_time;
71  echo "#";

```

Gambar 5.22 Kirim HTTP Response nodeatas.php

5.6 Implementasi Graphical User Interface

GUI pada sistem ini berfungsi untuk menampilkan data dari *node* dan mengirimkan perintah ke *node*. Program utama pengatur GUI adalah **gui.php** dan program pengatur *style* dari HTML-nya adalah **style.css**. Program ini merupakan modifikasi dari program pada artikel *DIY Cloud Weather Station* (Santos, 2019). GUI dapat diakses dengan menyambungkan *device* ke *access point* yang sama dengan *server*, pada penelitian ini menggunakan *mobile hotspot*, lalu mengetikkan **192.168.43.148/gui.php** pada browser (192.168.43.148 adalah *IP adress server*).

Data yang akan di tampilkan pada GUI adalah waktu pembacaan terakhir, *score* beserta *gauge*-nya, data pembacaan terbaru, data puncak, dan tabel *logs*. Data tersebut ditampilkan menggunakan *HTML element* pada kode baris ke 50, 53, 66, 78, dan 83 file **gui.php**, *content* pada *HTML element* berasal dari file .php lain yang akan di *refresh* secara *realtime* setiap 100 *millisecond* menggunakan javascript pada

baris 17-28, isi dari *file* .php yang akan di *refresh* dapat dilihat pada Gambar 3-7 lampiran 5. Potongan kode penampil data pada GUI ditunjukkan pada Gambar 5.23.

```

17 <script>
18     setInterval(function () {$('#latest-time')
19         .load('read-time.php');}, 100);
20     setInterval(function () {$('#latest-data')
21         .load('read-latest.php');}, 100);
22     setInterval(function () {$('#peak-data')
23         .load('read-peak.php');}, 100);
24     setInterval(function () {$('#score-gauge')
25         .load('read-score.php');}, 100);
26     setInterval(function () {$('#logs-table')
27         .load('read-logs.php');}, 100);
28 </script>
29 .....
50 <p id="latest-time">
51 .....
53 <center id="score-gauge">
54 .....
66 <tr id="latest-data">
67 .....
78 <tr id="peak-data">
79 .....
83 <table cellpadding="5" cellspacing="5" id="logs-table">

```

Gambar 5.23 Potongan Kode Penampil Data gui.php

Untuk mengirimkan perintah ke *node*, dibuat tombol yang akan mengirim *value* ke *database* dengan metode HTTP GET, nantinya *value* tersebut akan dikirimkan ke *node* oleh *web server* melalui *HTTP Response*. Tombol tersebut dibuat dengan kode pada baris ke 32-47 *file* **gui.php**. Tombol “Start” akan mengirimkan ‘q’, tombol “Stop” akan mengirimkan ‘a’, dan tombol “Reset” akan mengirimkan ‘w’ jika ditekan. *Value* lain juga dapat dikirimkan melalui *form* “Node command”. Potongan kode pengirim perintah pada GUI ditunjukkan pada Gambar 5.24 dan Gambar 5.25.

```

1 <?php
2     include_once('fungsi.php');
3
4     $command="x";
5
6     if (isset($_GET["command"])) {$command = $_GET["command"];}
7     if ($command!="x") {writeCommand($command);}
8     .....
9     ?>
10    .....
32    <form method="get" style="margin-bottom: 13px">
33        <input type="text" name="command" placeholder="Node
34        command">
35        <input type="submit" value="Send">

```

```

35 </form>
36 <form class="button" method="get">
37   <input type="hidden" name="command" value="q">
38   <input type="submit" class="buttonGreen" value="Start">
39 </form>
40 <form class="button" method="get">
41   <input type="hidden" name="command" value="a">
42   <input type="submit" class="buttonGreen" value="Stop">
43 </form>
44 <form class="button" method="get">
45   <input type="hidden" name="command" value="w">
46   <input type="submit" class="buttonGreen" value="Reset">
47 </form>

```

Gambar 5.24 Potongan Kode Pengirim Perintah gui.php

```

105 function writeCommand($command) {
106   global $servername, $username, $password, $dbname;
107
108   $conn = new mysqli($servername, $username,
109                     $password, $dbname);
110   if ($conn->connect_error) {
111     die("Connection failed: " . $conn->connect_error);
112   }
113
114   $sql = "INSERT INTO botcommand (command)
115   VALUES ('" . $command . "')";
116
117   $conn->query($sql);
118
119   $sql = "INSERT INTO topcommand (command)
120   VALUES ('" . $command . "')";
121
122   $conn->query($sql);
123   $conn->close();
124 }

```

Gambar 5.25 Potongan Kode Pengirim Perintah fungsi.php

5.7 Implementasi Data Logger

Sistem ini menggunakan dua *data logger*, yaitu *micro SD card* dan *database MySQL*. Pada *micro SD card*, data akan disimpan dalam *file .txt*, sedangkan pada *database MySQL*, data akan disimpan dalam tabel.

5.7.1 Micro SD card data logger

Prosedur *micro SD card data logger* diatur pada program mikrokontroler *node* bawah. Data hasil pengukuran akan disimpan dalam *file data.txt* saat *command* bernilai 'a'. *Offset* hasil kalibrasi *node* bawah juga akan otomatis tersimpan dalam *file calib.txt* setelah program kalibrasi dijalankan. Kode program merupakan modifikasi dari kode program "SD_Test" pada *library arduino-esp32*

(Grokhotkov et al., 2019). Kode program *micro SD card data logger* ditunjukkan pada Gambar 5.26.

```

108 void Task1code( void * pvParameters ) {
109     for(;;) {
110         komunikasi_http();
111         if(command=='a') logSDCard();
112     }
113 }
114 .....
172 void imu_calibration(void)
173 {
174     imuCal = 1;
175     mpu.CalibrateAccel(20);
176     mpu.CalibrateGyro(20);
177     logSDCard();
178     imuCal = 0;
179 }
180 .....
327 void logSDCard() {
328     if (imuCal==0) {
329         dataMessage = server_time + "\tPGA=" + String(pga,4)
330             + "g\tBotAngle=" + String(inc_angle_peak,3)
331             + "°\tTopAngle=" + String(top_angle_peak,3)
332             + "°\tH=" + String(h)
333             + " cm\tDisp=" + String(displacement_peak)
334             + " cm\tDR=" + String(drift_ratio_peak,3)
335             + "%\tPGA-DR1=" + String(pga_dr1,4)
336             + "g\tScore=" + String(score) + "\r\n";
337     }
338     if (imuCal==1) {
339         dataMessage = server_time + "\tBOT CAL\tAccOffsetX="
340             + String(mpu.getXAccelOffset())
341             + "\tAccOffsetY=" + String(mpu.getYAccelOffset())
342             + "\tAccOffsetZ=" + String(mpu.getZAccelOffset())
343             + "\tGyroOffsetX=" + String(mpu.getXGyroOffset())
344             + "\tGyroOffsetY=" + String(mpu.getYGyroOffset())
345             + "\tGyroOffsetZ=" + String(mpu.getZGyroOffset())
346             + "\r\n";
347     }
348     Serial.print("Save data: ");
349     Serial.println(dataMessage);
350     if (imuCal==0)
351         {appendFile(SD, "/data.txt", dataMessage.c_str());}
352     if (imuCal==1)
353         {appendFile(SD, "/calib.txt", dataMessage.c_str());}
354 }

```

Gambar 5.26 Kode Program Micro SD Card Data Logger

5.7.2 Database MySQL data logger

Prosedur *database MySQL data logger* diatur pada program **gui.php** dan **fungsi.php**. Program akan berjalan saat tombol “Stop” ditekan dan *command* bernilai ‘a’. Data akan tersimpan pada tabel

“datalogger” database “skripsi”. Kode program database MySQL data logger ditunjukkan pada Gambar 5.27 dan Gambar 5.28.

```

1  <?php
2  include_once('fungsi.php');
3
4  $command="x";
5
6  if (isset($_GET["command"])){$command = $_GET["command"];}
7  .....
8  if ($command=="a"){writeLog();}
9  ?>
10 .....
40 <form class="button" method="get">
41     <input type="hidden" name="command" value="a">
42     <input type="submit" class="buttonGreen" value="Stop">
43 </form>

```

Gambar 5.27 Kode Program Database MySQL Data Logger gui.php

```

126 function writeLog() {
127     $last_reading = getLastReadings();
128     $pga = $last_reading["pga"];
129     $anglepeak = $last_reading["anglepeak"];
130     $height = $last_reading["height"];
131     $displacepeak = $last_reading["displacepeak"];
132     $driftpeak = $last_reading["driftpeak"];
133     $pgadr1 = $last_reading["pgadr1"];
134     $score = $last_reading["score"];
135
136     $last_reading = getLastReadings2();
137     $anglepeak2 = $last_reading["anglepeak"];
138
139     global $servername, $username, $password, $dbname;
140
141     $conn = new mysqli($servername, $username,
142                       $password, $dbname);
143     if ($conn->connect_error) {
144         die("Connection failed: " . $conn->connect_error);
145     }
146
147     $sql = "INSERT INTO datalogger (pga, botanglepeak,
148                                   topanglepeak, height, displacepeak,
149                                   driftpeak, pgadr1, score)
150     VALUES ('".$pga."', '".$anglepeak."', '".$anglepeak2."',
151             '".$height."', '".$displacepeak."',
152             '".$driftpeak."', '".$pgadr1."', '".$score."')";
153
154     $conn->query($sql);
155     $conn->close();
156 }

```

Gambar 5.28 Kode Program Database MySQL Data Logger fungsi.php

5.8 Pengujian Fungsional

Pengujian fungsional dilakukan untuk mengetahui kinerja sistem per bagian dan memastikan setiap bagian berjalan sesuai dengan yang diharapkan. Pengujian ini dilakukan terpisah dengan pengujian pada objek (dinding).

5.8.1 Pengujian akuisisi data

1. *Error* pada akuisisi data

Error pada akuisisi data dihitung dengan cara melakukan pengukuran saat posisi sensor diam pada keadaan awal (titik nol), lalu dihitung rata-rata dan standar deviasi dari harga mutlak hasil pengukuran.

2. Kecepatan pada akuisisi data

Kecepatan pada akuisisi data dihitung dengan cara melakukan pengamatan pada jumlah pembacaan paket FIFO sensor IMU per detik. Pengamatan dilakukan dengan menggunakan *serial monitor*.

5.8.2 Pengujian fuzzy logic

Pengujian *fuzzy logic* dilakukan dengan cara memvariasikan masukan *fuzzy* yaitu PGA_{DR1} menggunakan data *dummy*, lalu membandingkan hasil *outputnya* dengan MATLAB untuk mengetahui akurasi *fuzzy logic*. Data *dummy* divariasikan melalui *serial monitor* dengan mengedit beberapa kode pada program *node* bawah. Beberapa kode yang harus diedit ditunjukkan pada Gambar 5.29.

```

116   command = Serial.read();
.....
123   if(command=='v') pga_dr1 += 0.0001;
124   if(command=='b') pga_dr1 += 0.001;
125   if(command=='n') pga_dr1 += 0.01;
126   if(command=='m') pga_dr1 += 0.1;
127
128   if(command=='h') pga_dr1 -= 0.0001;
129   if(command=='j') pga_dr1 -= 0.001;
130   if(command=='k') pga_dr1 -= 0.01;
131   if(command=='l') pga_dr1 -= 0.1;
.....
137   //print_Data();
.....
139   if (pga_dr1!=pga_dr1_lama){
140       Serial.print(pga_dr1,4);
141       Serial.print("\t");
142       Serial.println(score);
143   }
144   pga_dr1_lama = pga_dr1;
.....

```

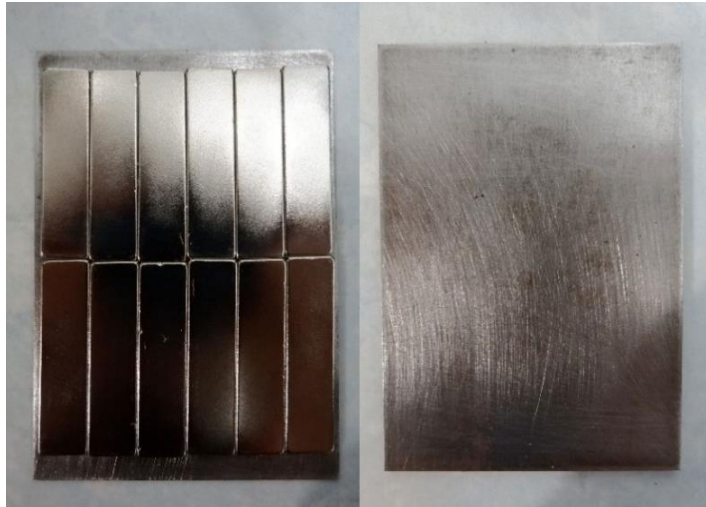
238	<code>//pga_dr1 = pga/drift_ratio_peak;</code>
312	<code>..... //command = dt[1][0];</code>

Gambar 5.29 Kode Pengujian Fuzzy Logic

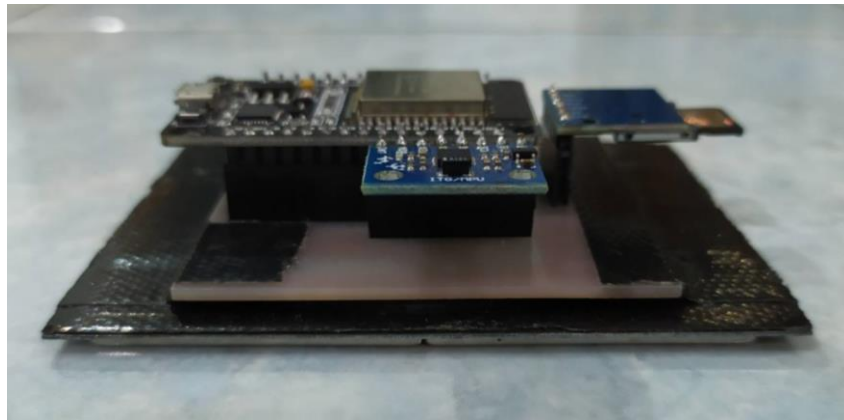
5.9 Pengujian Penentuan Tingkat Kekakuan Dinding Rumah Berbahan Beton terhadap Getaran

Pengujian pada objek dilakukan dengan menempelkan kedua *node* pada kolom dinding. Penempelan *node* ke dinding membutuhkan beberapa perangkat tambahan yaitu magnet neodmium dan plat besi. Magnet neodmium bisa langsung menempel ke plat besi, karena besi bersifat feromagnetik. Penempelan magnet neodmium ke plat besi ditunjukkan pada Gambar 5.30. Penempelan papan elektronik ke plat besi memerlukan bahan tambahan yaitu lakban dan *super glue*. Lakban digunakan untuk mencegah plat besi bersentuhan dengan papan elektronik agar tidak terjadi *short circuit*. *Super glue* digunakan untuk menempelkan papan elektronik ke plat besi yang sudah dilapisi dengan lakban. Penempelan plat besi ke papan elektronik ditunjukkan pada Gambar 5.31. Penempelan *node* ke dinding membutuhkan tambahan lakban agar *node* tertempel lebih kuat dan tidak rawan jatuh, dikarenakan jika hanya dengan magnet, *node* tidak dapat menempel kuat. Untuk menempelkan *powerbank* juga digunakan lakban. Penempelan *node* ke dinding ditunjukkan pada Gambar 5.32.

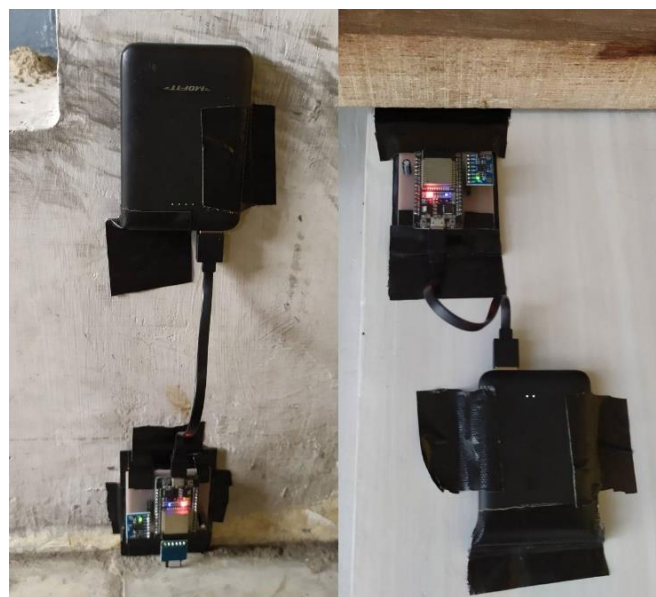
Pengujian dilakukan pada tiga dinding dengan lokasi yang berbeda. Tinggi dinding berturut-turut adalah 284 cm, 268 cm, dan 248 cm. Pengujian dilakukan dengan menjatuhkan beban yaitu semen seberat 28 kg di depan dinding dari ketinggian 1,6 meter, jarak dinding dengan tempat terjatuhnya beban sekitar 18 cm. Lokasi dan beban yang digunakan untuk pengujian pada dinding ditunjukkan pada Gambar 5.33 dan Gambar 5.34.



Gambar 5.30 Penempelan Magnet Neodymium ke Plat Besi



Gambar 5.31 Penempelan Plat Besi Ke Papan Elektronik



Gambar 5.32 Penempelan Node ke Dinding



Gambar 5.33 Lokasi Pengujian pada Dinding



Gambar 5.34 Beban Pengujian pada Dinding

BAB VI HASIL DAN PEMBAHASAN

6.1 Hasil Pengujian Fungsional

6.1.1 Hasil pengujian akuisisi data

1. Error pada akuisisi data

Untuk mengetahui besar *error* pada akuisisi data, dilakukan pengukuran *ground acceleration* dan *inclination angle* saat posisi *node* diam pada keadaan awal (titik nol). Jika tidak ada *error* pada akuisisi data, seharusnya nilai *ground acceleration* dan *inclination angle* adalah 0g dan 0°. Berikut merupakan data hasil dari pengukuran saat posisi sensor berapa pada titik nol, data yang diambil adalah hasil pengukuran dari 160 kali *looping* akuisisi data.

Tabel 6.1 Pengukuran Ground Acceleration Titik Nol (1×10^{-4})g

Pembacaan Data (1-160)							
7	3	-1	9	-6	-3	3	-7
3	-6	-6	5	-9	-2	-2	-6
2	-1	-2	9	-6	-6	4	-6
-4	7	-6	5	-2	-6	0	-6
-3	4	-4	9	-2	-2	-2	-9
-6	-1	-6	3	0	0	-1	-9
-6	3	-6	0	-2	2	-4	-2
-4	4	-6	5	-4	3	-8	5
4	2	-1	-4	2	-1	-8	8
9	7	-6	-3	-3	0	-7	5
4	9	-9	-1	-6	4	-6	4
7	7	-6	-3	3	5	-3	8
-1	-1	-6	-2	-6	0	0	8
-3	-4	-1	-3	-8	0	-2	5
-3	-9	2	-6	-8	-3	0	5
-2	-9	-6	-3	-6	-8	2	9
0	-6	-9	-1	0	-6	-6	3
-3	-6	-3	-6	7	-4	0	-1
-1	-2	-3	-7	8	-6	-6	2
2	-1	4	-6	8	-3	-7	3

Dari pengukuran *ground acceleration* pada Tabel 6.1, dapat dihitung rata-rata beserta standar deviasi dari harga mutlak pada *ground acceleration* saat posisi sensor diam pada keadaan awal adalah:

$$(4,3 \pm 2,65) \times 10^{-4} g$$

Tabel 6.2 Pengukuran Inclination Angle Titik Nol (1×10^{-3})°

Pembacaan Data (1-160)							
1	8	8	1	1	1	1	1
8	8	8	1	1	1	1	1
8	8	8	8	1	1	1	-6
8	8	8	8	1	8	1	-6
8	8	8	8	1	8	1	-6
8	8	8	8	1	8	1	-6
1	8	8	8	1	8	1	-6
1	1	8	8	1	1	1	-6
1	8	8	8	1	1	1	-6
1	8	8	8	1	1	1	-6
1	8	8	8	1	1	1	-6
1	8	8	8	1	1	1	-6
1	8	8	8	1	1	1	-6
1	8	8	8	1	1	1	-6
1	8	8	8	1	1	1	-6
1	8	8	8	1	1	1	-6
8	8	8	1	1	1	1	-6
8	8	8	1	1	1	1	-6
8	8	8	1	1	1	1	-6
8	8	8	1	1	1	1	-6
8	8	1	1	1	1	1	-6

Dari pengukuran *inclination angle* pada Tabel 6.2, dapat dihitung rata-rata beserta standar deviasi dari harga mutlak pada *inclination angle* saat posisi sensor diam pada keadaan awal adalah:

$$(4,2 \pm 3,33) \times 10^{-3} ^\circ$$

Dari kedua perhitungan di atas, dapat disimpulkan bahwa *error* pada akuisisi data menggunakan *sensor fusion* pada DMP tergolong sangat kecil.

2. Kecepatan pada akuisisi data

Untuk mengetahui kecepatan akuisisi data, dilakukan pengamatan pada jumlah pembacaan paket FIFO sensor IMU per detik menggunakan *serial monitor*. Data pembacaan paket FIFO sensor IMU dapat dilihat pada Lampiran 7. Dari pembacaan data, dapat diketahui bahwa kecepatan akuisisi data pada sistem adalah 100 Hz.

6.1.2 Hasil pengujian fuzzy logic

Pengujian *fuzzy logic* dilakukan dengan cara memvariasikan masukan *fuzzy* yaitu PGA_{DR1} menggunakan data *dummy*. Hasil dari *fuzzy logic* pada sistem dibandingkan dengan *fuzzy logic* yang dibuat menggunakan MATLAB untuk mengetahui akurasi *fuzzy logic* pada sistem. Data pengujian *fuzzy logic* dapat dilihat pada Lampiran 8. Dari data yang telah didapatkan, dapat dibuat tabel perbandingan yang ditunjukkan pada Tabel 6.3.

Tabel 6.3 Perbandingan Fuzzy Logic pada Sistem dengan MATLAB

No	Input	Output Sistem	Output Matlab	No	Input	Output Sistem	Output Matlab
1	0.0000	0.00	0.00	33	0.1697	2.99	2.99
2	0.0010	0.33	0.33	34	0.1698	2.99	2.99
3	0.0020	0.67	0.67	35	0.1699	3.00	3.00
4	0.0030	1.00	1.00	36	0.1700	3.00	3.00
5	0.0130	1.12	1.12	37	0.1701	3.00	3.00
6	0.0230	1.23	1.23	38	0.1702	3.00	3.00
7	0.0330	1.35	1.35	39	0.1703	3.00	3.00
8	0.0430	1.46	1.46	40	0.2703	3.25	3.25
9	0.0530	1.58	1.58	41	0.3703	3.49	3.49
10	0.0630	1.69	1.69	42	0.4703	3.74	3.74
11	0.0730	1.81	1.81	43	0.5703	3.99	3.99
12	0.0830	1.92	1.92	44	0.5713	3.99	3.99
13	0.0840	1.93	1.93	45	0.5723	3.99	3.99
14	0.0850	1.95	1.95	46	0.5733	4.00	4.00
15	0.0860	1.96	1.96	47	0.5743	4.00	4.00
16	0.0870	1.97	1.97	48	0.5744	4.00	4.00
17	0.0880	1.98	1.98	49	0.5745	4.00	4.00
18	0.0890	1.99	1.99	50	0.5746	4.00	4.00
19	0.0891	1.99	1.99	51	0.5747	4.00	4.00
20	0.0892	1.99	1.99	52	0.5748	4.00	4.00
21	0.0893	2.00	2.00	53	0.5749	4.00	4.00
22	0.0894	2.00	2.00	54	0.5750	4.00	4.00
23	0.0895	2.00	2.00	55	0.5751	4.00	4.00
24	0.0896	2.00	2.00	56	0.6751	4.00	4.00
25	0.0897	2.00	2.00	57	0.7751	4.00	4.00
26	0.0997	2.12	2.12	58	0.8751	4.00	4.00
27	0.1097	2.25	2.25	59	0.9751	4.00	4.00
28	0.1197	2.37	2.37	60	1.0751	4.00	4.00

29	0.1297	2.50	2.50	61	1.1751	4.00	4.00
30	0.1397	2.62	2.62	62	1.2751	4.00	4.00
31	0.1497	2.74	2.74	63	1.3751	4.00	4.00
32	0.1597	2.87	2.87	64	1.4751	4.00	4.00

Akurasi dari *fuzzy logic* dapat dihitung berdasarkan *confusion matrix* menggunakan rumus pada persamaan (3.10). Data bernilai *true* jika *output* pada sistem sama dengan *output* pada MATLAB. Berikut adalah penerapan persamaan (3.10) untuk menghitung akurasi *fuzzy logic*:

$$Akurasi = \frac{64}{64} \times 100\%$$

$$Akurasi = 100\%$$

Jika hasil dari *fuzzy logic* dibandingkan dengan SIG BMKG (Anonim, n.d.-a), hasil yang didapatkan juga sudah sesuai.

6.2 Hasil Pengujian Penentuan Tingkat Kekakuan Dinding Rumah Berbahan Beton terhadap Getaran

Dari data hasil pengujian pada dinding yang ada pada Lampiran 9, dibuatlah ringkasan hasil pada Tabel 6.4. Data yang ditampilkan adalah rata-rata dan standar deviasi dari PGA_{DR1} dan *score*, serta selisih sudut dinding bagian bawah dan bagian atas.

Tabel 6.4 Hasil Pengujian pada Dinding

Objek Pengujian	Rata-Rata PGA_{DR1}	Rata-Rata Score	STDEV PGA_{DR1}	STDEV Score	Selisih Sudut
Dinding 1	0.1684g	2,90	0.0189g	0.15	$\leq 0,007^\circ$
Dinding 2	0.1826g	3.00	0.0185g	0.12	$\leq 0,008^\circ$
Dinding 3	0.1440g	2.67	0.0140g	0.17	$\leq 0,008^\circ$

Dinding 1 mendapatkan *score* rata-rata sebesar 2,90 yang artinya dapat diasumsikan dinding dapat menahan getaran yang setara dengan SIG II, dan hampir kuat menahan getaran dengan SIG III. Selisih dari *peak bottom inclination angle* dan *peak top inclination angle* tidak terlalu besar, dari 10 kali percobaan selisih paling besar hanya $0,007^\circ$, oleh karena itu dapat diasumsikan juga bahwa tidak ada *crack* pada dinding.

Dinding 2 mendapatkan *score* rata-rata sebesar 3,00 yang artinya dapat diasumsikan dinding dapat menahan getaran yang setara dengan SIG III. Selisih

dari *peak bottom inclination angle* dan *peak top inclination angle* tidak terlalu besar, dari 11 kali percobaan selisih paling besar hanya $0,008^\circ$, oleh karena itu dapat diasumsikan juga bahwa tidak ada *crack* pada dinding.

Dinding 3 mendapatkan *score* rata-rata sebesar 2,67 yang artinya dapat diasumsikan dinding dapat menahan getaran yang setara dengan SIG II. Selisih dari *peak bottom inclination angle* dan *peak top inclination angle* tidak terlalu besar, dari 12 kali percobaan selisih paling besar hanya $0,008^\circ$, oleh karena itu dapat diasumsikan juga bahwa tidak ada *crack* pada dinding.

Dari seluruh pengujian pada dinding, standar deviasi rata-rata yang didapatkan cukup tinggi yaitu $\pm 0,0171g$ untuk PGA_{DRI} dan $\pm 0,15$ untuk *score*, yang artinya presisi sistem kurang baik. Presisi yang kurang baik ini terjadi karena PGA tidak selalu berbanding lurus dengan *drift ratio*. Menurut teori yang dipakai, seharusnya *drift ratio* berbanding lurus dengan gaya lateral (Kumar & Venkat, 2016) dan gaya lateral berbanding lurus dengan percepatan lateral dalam rumus: $F = ma$ (Roselli, 2011). Masalah ini mungkin dikarenakan adanya faktor selain PGA yang dapat mempengaruhi *drift ratio* pada dinding, yang belum dipertimbangkan dalam penelitian ini.

BAB VII

KESIMPULAN DAN SARAN

7.1 Kesimpulan

Berdasarkan hasil penelitian, implementasi sistem dan pengujian, dapat diambil kesimpulan sebagai berikut:

1. Proses akuisisi data dapat berjalan dengan baik dengan kecepatan sekitar 100 Hz. *Noise* yang dihasilkan pada proses akuisisi data sangat kecil, yaitu $(4,3 \pm 2,65) \times 10^{-4}$ g untuk *ground acceleration* dan $(4,2 \pm 3,33) \times 10^{-3}^\circ$ untuk *inclination angle*.
2. Pemakaian *fuzzy* pada pemrosesan menghasilkan *score* yang 100% akurat, hasil sudah sesuai dengan SIG BMKG dan luaran *fuzzy* pada mikrokontroler sudah sesuai dengan luaran *fuzzy* pada MATLAB.
3. Pengujian pada dinding mendapatkan tingkat presisi yang kurang baik, yang mungkin dikarenakan adanya faktor selain PGA yang dapat mempengaruhi *drift ratio* pada dinding, yang belum dipertimbangkan dalam penelitian ini.
4. Secara umum, sistem ini dapat berjalan sesuai dengan yang diharapkan dan sudah layak dipakai, tetapi presisi perlu ditingkatkan lagi kedepannya.

7.2 Saran

Pada penelitian ini terdapat beberapa hal yang perlu disempurnakan. Saran-saran yang dapat digunakan untuk menyempurnakan penelitian selanjutnya adalah sebagai berikut:

1. Pada penelitian ini, arah getaran yang diukur hanya depan dan belakang, coba untuk mengukur arah kanan dan kiri juga.
2. Berat dan ketinggian beban yang dijatuhkan diperbesar, agar gaya lateral yang dihasilkan lebih besar.
3. Tingkatkan presisi sistem, dengan cara mencari parameter lain yang kemungkinan juga mempengaruhi *drift ratio*.
4. Beberapa hal pada sistem ini masih dilakukan secara manual, seperti memasukkan tinggi dinding dan konfigurasi komunikasi *node* dengan *server*, untuk kedepannya bisa dibuat sistem yang sepenuhnya otomatis.

5. PGA yang diukur belum divalidasi apakah hasilnya sama dengan PGA yang digunakan pada skala SIG BMKG, kedepannya perlu validasi.

DAFTAR PUSTAKA

- Abdussaid, O. (2018). FUZZY LOGIC – METODE SUGENO. Retrieved March 14, 2020, from Elektronika Portal website: <https://elektronika-portal.com/2018/12/24/fuzzy-logic-metode-sugeno/>
- Amaliyah, R. (2019). Pengertian Node dan Fungsi Node pada Jaringan Komputer. Retrieved June 17, 2020, from Nesaba Media website: <https://www.nesabamedia.com/pengertian-node/>
- Anonim. (n.d.-a). Skala Intensitas Gempabumi (SIG) BMKG. Retrieved May 28, 2020, from Badan Meteorologi, Klimatologi dan Geofisika website: <https://www.bmkg.go.id/gempabumi/skala-intensitas-gempabumi.bmkg>
- Anonim. (n.d.-b). Skala MMI (Modified Mercalli Intensity). Retrieved May 28, 2020, from Badan Meteorologi, Klimatologi dan Geofisika website: <https://www.bmkg.go.id/gempabumi/skala-mmi.bmkg>
- Anonim. (2014). ShakeMap Scientific Background. Retrieved May 28, 2020, from USGS website: <https://earthquake.usgs.gov/data/shakemap/background.php>
- Anonim. (2017). Embedded Motion Driver 20x48 – Users Guide. *InvenSense Inc.*, 1, 1–14. Retrieved from <https://invensense.tdk.com/wp-content/uploads/2015/12/App-Note-eMD-20x48-User-Guide.pdf>
- Anonim. (2018a). Nuclear Power Plants and Earthquakes. Retrieved June 10, 2020, from World Nuclear Association website: <https://www.world-nuclear.org/information-library/safety-and-security/safety-of-plants/nuclear-power-plants-and-earthquakes.aspx>
- Anonim. (2018b). Pengertian Gempa Bumi, Jenis-Jenis, Penyebab, Akibat, dan Cara Menghadapi Gempa Bumi. Retrieved November 10, 2019, from BPBD Banda Aceh website: <http://bpbd.bandaacehkota.go.id/2018/08/05/pengertian-gempa-bumi-jenis-jenis-penyebab-akibat-dan-cara-menghadapi-gempa-bumi/>
- Anonim. (2018c). Pengujian Dengan Confusion Matrix. Retrieved June 16, 2020, from Kuliah Komputer Blogger website: <http://www.kuliahkomputer.com/2018/07/pengujian-dengan-confusion-matrix.html>
- Baker, M. J. (2007). Transformations using Quaternions. Retrieved May 21, 2020, from Euclidean Space website: <http://www.euclideanspace.com/maths/algebra/realNormedAlgebra/quaternions/transforms/index.htm>
- Bormann, P., & Wielandt, E. (2013). *Seismic Signals and Noise*. June, 3. https://doi.org/10.2312/GFZ.NMSOP-2_ch4
- Christiansen, K. R., & Shalamov, A. (2017). Motion Sensors Explainer. Retrieved April 10, 2020, from W3C website: <https://www.w3.org/TR/motion-sensors/>
- Douglas, J. (2003). Earthquake ground motion estimation using strong-motion

- records: A review of equations for the estimation of peak ground acceleration and response spectral ordinates. *Earth-Science Reviews*, 61(1–2), 43–104. [https://doi.org/10.1016/S0012-8252\(02\)00112-5](https://doi.org/10.1016/S0012-8252(02)00112-5)
- Elmenreich, W. (2002). *Sensor Fusion in Time-Triggered Systems*. (9226605). Retrieved from http://www.vmars.tuwien.ac.at/~wilfried/papers/elmenreich_Dissertation_sensorFusionInTimeTriggeredSystems.pdf
- Grokhotkov, I., Sovani, K., Jain, M., Gratton, A., & Domburg, J. (2019). *Arduino core for the ESP32*. Retrieved from <https://github.com/espressif/arduino-esp32>
- Guitierrez, J., & Alpizar, M. (2004). An Effective Method for Displacement-Based Earthquake Design of Buildings. *13th World Conference on Earthquake Engineering*, (1512).
- Henry, R. S., Ingham, J. M., & Sritharan, S. (2012). Wall-to-floor interaction in concrete buildings with rocking wall systems. *2012 NZSEE Conference*, (072), 1–8.
- Kostiainen, A., & Bhaumik, R. (2019). Magnetometer. Retrieved May 30, 2020, from W3C website: <https://www.w3.org/TR/magnetometer/>
- Kostiainen, A., & Pozdnyakov, M. (2019). Gyroscope. Retrieved May 30, 2020, from W3C website: <https://www.w3.org/TR/gyroscope/>
- Kostiainen, A., & Shalamov, A. (2019). Accelerometer. Retrieved May 30, 2020, from W3C website: <https://www.w3.org/TR/accelerometer/>
- Kumar, J. D. C., & Venkat, L. (2016). Effect of Lateral Forces on Precast Shear Wall. *International Journal of Civil and Structural Engineering Research*, 3(2), 74–84. Retrieved from www.researchpublish.com
- Kusumadewi, S. (2002). *Analisis Desain Sistem Fuzzy Menggunakan Tool Box Matlab*. Yogyakarta: Graha Ilmu.
- Lorant, G. (2016). Seismic Design Principles. Retrieved May 10, 2020, from Whole Building Design Guide website: <https://www.wbdg.org/resources/seismic-design-principles>
- Lubkowski, Z. A., & Duan, X. (2001). EN1998 Eurocode 8: Design of structures for earthquake resistance. *Proceedings of the Institution of Civil Engineers - Civil Engineering*, 144(6), 55–60. <https://doi.org/10.1680/cien.2001.144.6.55>
- Madgwick, S. O. H. (2010). *An efficient orientation filter for inertial and inertial / magnetic sensor arrays*.
- Mulia, A. (2015). MONITORING PENGUKURAN GETARAN GEMPA MENGGUNAKAN MIKROKONTROLLER 8535. *E-Proceeding of Applied Science*, 1(2), 1276–1282.
- Murty, C. V. R., Goswami, R., Vijayanarayanan, A. R., & Mehta, V. V. (2012). *Earthquake Behaviour of Buildings*. Retrieved from

- https://www.iitk.ac.in/nicee/IITK-GSDMA/EBB_001_30May2013.pdf
- Musaab, A. A., Sulle, B., & Minarso, A. (2017). Design of AHRS for Quadrotor Control using Digital Motion Processor. *Journal of Telecommunication, Electronic and Computer Engineering*, 9(1–5), 77–82.
- Nirmal, K., Sreejith, A. G., Mathew, J., Sarpotdar, M., Suresh, A., Prakash, A., ... Murthy, J. (2016). Noise modeling and analysis of an IMU-based attitude sensor: improvement of performance by filtering and sensor fusion. In R. Navarro & J. H. Burge (Eds.), *Advances in Optical and Mechanical Technologies for Telescopes and Instrumentation II* (Vol. 9912, p. 99126W). <https://doi.org/10.1117/12.2234255>
- Pedley, M. (2013). Tilt Sensing Using a Three-Axis Accelerometer. *Freescale Semiconductor Application Note*, Rev. 6, 1–22.
- Roselli, M. (2011). THE LATERAL FORCES OF EARTHQUAKES. Retrieved April 20, 2020, from Mike Roselli, PE website: <http://www.mikeroselli.net/the-lateral-forces-of-earthquakes/>
- Rowberg, J. (2012). MPU-6050 6-axis accelerometer/gyroscope. Retrieved April 13, 2020, from I2C Device Library website: <https://www.i2cdevlib.com/devices/mpu6050#help>
- Rowberg, J. (2020). *The I2C Device Library (i2cdevlib)*. Retrieved from <https://github.com/jrowberg/i2cdevlib>
- Santos, R. (2019). *DIY Cloud Weather Station with ESP32/ESP8266 (MySQL Database and PHP)*. Retrieved from <https://randomnerdtutorials.com/cloud-weather-station-esp32-esp8266/>
- Saputri, A. D., Ramadhani, R. D., & Adhitama, R. (2019). Logika Fuzzy Sugeno untuk Pengambilan Keputusan Dalam Penjadwalan dan Peningkat Service Sepeda Motor. *Journal of Informatics, Information System, Software Engineering and Applications*, 2, 49–55.
- Setiawati, L. S., Budiman, I., & Soesanto, O. (2016). Penerapan Fuzzy Inference System Takagi-Sugeno-Kang pada Sistem Pakar Diagnosa Penyakit Gigi. *Kumpulan Jurnal Ilmu Komputer (KLIK)*, 04(01), 1–10.
- Simanjuntak, P., Suharyanto, C., & Khairiyah, R. (2018). Fuzzy Sugeno Untuk Menentukan Penilaian Kompetensi Karyawan PT. Schneider Batam. *Information System Development (ISD)*, 3(2), 97–103.
- Wang, L. X. (1997). *A Course in Fuzzy Systems and Control* (Internatio). Hong Kong: Prentice-Hall International, Inc.
- Weisstein, E. W. (n.d.). Euler Angles. Retrieved from MathWorld--A Wolfram Web Resource website: <https://mathworld.wolfram.com/EulerAngles.html>
- Weng, P. W., Li, Y. A., Tu, Y. S., & Hwang, S. J. (2017). Prediction of the Lateral Load-Displacement Curves for Reinforced Concrete Squat Walls Failing in Shear. *Journal of Structural Engineering (United States)*, 143(10).

[https://doi.org/10.1061/\(ASCE\)ST.1943-541X.0001872](https://doi.org/10.1061/(ASCE)ST.1943-541X.0001872)

- Wijaya, B. (2015). Implementasi Fuzzy Logic Controller untuk Kontrol Kecepatan Motor DC pada Prototype Kipas Angin. Retrieved February 25, 2020, from sonoku.com website: <https://sonoku.com/?p=4007>
- Xie, L. (2015). *Resources, Environment and Engineering II* (CREE 2015; L. Xie, Ed.). Retrieved from <https://books.google.co.id/books?id=-DU0CwAAQBAJ>
- Yuliani, S., & Saputra, H. M. (2016). Kolaborasi Kalman Filter dengan Complementary Filter untuk Mengoptimasi Hasil Sensor Gyroscope dan Accelerometer. *Seminar Nasional ITENAS*, (Mesin), 63–68.

LAMPIRAN

Lampiran 1 Listing program node bawah

Gambar lampiran 1 Listing program mikrokontroler node bawah

```
1  #include <Arduino.h>
2  #include <WiFi.h>
3  #include <WiFiMulti.h>
4  #include <HTTPClient.h>
5  #include "FS.h"
6  #include "SD.h"
7  #include <SPI.h>
8  #include "I2Cdev.h"
9  #include "MPU6050_6Axis_MotionApps20.h"
10
11 #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
12     #include "Wire.h"
13 #endif
14
15 WiFiMulti wifiMulti;
16 MPU6050 mpu;
17 TaskHandle_t Task1;
18
19 // Wifi
20 String dt[5];
21 char command;
22 char command_send;
23 String server_time;
24 float top_angle_peak;
25
26 // SD Card
27 #define SD_CS 5
28 String dataMessage;
29
30 // LED biru ESP32
31 const int ledPin = 2;
32 const int freq = 5000;
33 const int ledChannel = 0;
34 const int resolution = 8;
35 int dutyCycle = 0;
36
37 // MPU control/status vars
38 uint8_t devStatus;
39 uint8_t fifoBuffer[64];
40
41 // orientation/motion vars
42 Quaternion q;
43 VectorInt16 aa;
44 VectorInt16 gyro;
45 VectorInt16 aaReal;
46 VectorInt16 aaWorld;
47 VectorFloat gravity;
48 float ypr[3];
49
50 float accWorldX;
```



```

51 float accWorldY;
52 float accWorldZ;
53 float pitch;
54 float roll;
55 float ground_acc;
56 float inc_angle;
57
58 // for displacement & drift ratio
59 float inc_angle_rad;
60 float h = 248; // tinggi dinding (cm)
61 float displacement;
62 float drift_ratio;
63
64 bool start = false;
65
66 // Peak calc
67 float pga;
68 float inc_angle_peak;
69 float displacement_peak;
70 float drift_ratio_peak;
71
72 // Fuzzy Logic
73 float pga_dr1;
74 float pga_dr1_lama;
75 float x,a,b,c,d;
76 float sig[5];
77 float z0,z1,z2,z3,z4;
78 float score;
79
80 // IMU Calibration
81 int imuCal;
82
83 void setup() {
84     Serial.begin(115200);
85
86     wifiMulti.addAP("Hanif Maliki", "hanifmlk");
87
88     SD_Setup();
89
90     IMU_Setup();
91
92     // LED biru ESP32
93     ledcSetup(ledChannel, freq, resolution);
94     ledcAttachPin(ledPin, ledChannel);
95
96     //create a task
97     xTaskCreatePinnedToCore(
98         Task1code, /* Task function. */
99         "Task1", /* name of task. */
100         10000, /* Stack size of task */
101         NULL, /* parameter of the task */
102         1, /* priority of the task */
103         &Task1, /* Task handle */
104         0); /* pin task to core 0 */
105     delay(500);
106 }

```

```

107
108 void Task1code( void * pvParameters ) {
109     for(;;){
110         komunikasi_http();
111         if(command=='a') logSDCard();
112     }
113 }
114
115 void loop() {
116     // command = Serial.read();
117
118     if(command=='q') start = true;
119     if(command=='a') start = false;
120     if(command=='w') {peak_reset(); fuzzy_reset();}
121     if(command=='e') imu_calibration();
122
123     // if(command=='v') pga_dr1 += 0.0001;
124     // if(command=='b') pga_dr1 += 0.001;
125     // if(command=='n') pga_dr1 += 0.01;
126     // if(command=='m') pga_dr1 += 0.1;
127     //
128     // if(command=='h') pga_dr1 -= 0.0001;
129     // if(command=='j') pga_dr1 -= 0.001;
130     // if(command=='k') pga_dr1 -= 0.01;
131     // if(command=='l') pga_dr1 -= 0.1;
132
133     IMU_Calc();
134
135     if (start==true) {peak_calc(); fuzzy_start();}
136
137     print_Data();
138
139     // if (pga_dr1!=pga_dr1_lama){
140     //     Serial.print(pga_dr1,4);
141     //     Serial.print("\t");
142     //     Serial.println(score);
143     // }
144     // pga_dr1_lama = pga_dr1;
145 }
146
147 void print_Data(void)
148 {
149     Serial.print(ground_acc,4);
150     Serial.print("\t");
151     Serial.print(inc_angle,3);
152     Serial.print("\t");
153     Serial.print(displacement);
154     Serial.print("\t");
155     Serial.print(drift_ratio,3);
156     Serial.print("\t");
157     Serial.print(pga,4);
158     Serial.print("\t");
159     Serial.print(inc_angle_peak,3);
160     Serial.print("\t");
161     Serial.print(displacement_peak);
162     Serial.print("\t");

```

```

163 Serial.print(drift_ratio_peak,3);
164 Serial.print("\t");
165 Serial.print(pga_dr1,4);
166 Serial.print("\t");
167 Serial.print(score);
168 Serial.print("\t");
169 Serial.println();
170 }
171
172 void imu_calibration(void)
173 {
174     imuCal = 1;
175     mpu.CalibrateAccel(20);
176     mpu.CalibrateGyro(20);
177     logSDCard();
178     imuCal = 0;
179 }
180
181 void IMU_Calc(void)
182 {
183     // read a packet from FIFO
184     if (mpu.dmpGetCurrentFIFOPacket(fifoBuffer)) {
185         mpu.dmpGetQuaternion(&q, fifoBuffer);
186         mpu.dmpGetAccel(&aa, fifoBuffer);
187         mpu.dmpGetGyro(&gyro, fifoBuffer);
188         mpu.dmpGetGravity(&gravity, &q);
189         mpu.dmpGetLinearAccel(&aaReal, &aa, &gravity);
190         mpu.dmpGetLinearAccelInWorld(&aaWorld, &aaReal, &q);
191         mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
192
193         accWorldX = float(aaWorld.x)/8192;
194         accWorldY = float(aaWorld.y)/8192;
195         accWorldZ = float(aaWorld.z)/8192;
196         pitch = -ypr[1]*180/M_PI; // AngleY
197         roll = ypr[2]*180/M_PI; // AngleX
198
199         ground_acc = accWorldZ;
200         inc_angle = roll;
201
202         inc_angle_rad = inc_angle/(180/M_PI);
203         displacement = sqrt(pow(h,2)
204                             -pow(h*cos(inc_angle_rad),2));
205         if (inc_angle_rad<0) displacement=displacement*(-1);
206         drift_ratio = displacement/h*100;
207
208         dutyCycle = 10; // terang LED biru 0-255
209         ledcWrite(ledChannel, dutyCycle);
210     }
211     else {
212         if (dutyCycle>0) dutyCycle--;
213         ledcWrite(ledChannel, dutyCycle);
214     }
215 }
216
217 void peak_calc(void)
218 {

```

```

219     if (abs(ground_acc)>pga) pga=abs(ground_acc);
220     if (abs(inc_angle)>inc_angle_peak)
221         {inc_angle_peak=abs(inc_angle);}
222     if (abs(displacement)>displacement_peak)
223         {displacement_peak=abs(displacement);}
224     if (abs(drift_ratio)>drift_ratio_peak)
225         {drift_ratio_peak=abs(drift_ratio);}
226 }
227
228 void peak_reset(void)
229 {
230     pga = 0;
231     inc_angle_peak = 0;
232     displacement_peak = 0;
233     drift_ratio_peak = 0;
234 }
235
236 void fuzzy_start(void)
237 {
238     pga_dr1 = pga/drift_ratio_peak;
239
240     x=pga_dr1; a=0.0030; b=0.0897; c=0.1703; d=0.5751;
241
242     if (x>=0&&x<=a) {sig[0] = (a-x)/(a);} // SIG 0
243     else if (x>=a) {sig[0] = 0;}
244
245     if (x>=0&&x<=a) {sig[1] = x/a;} // SIG I
246     else if (x>=a&&x<=b) {sig[1] = (b-x)/(b-a);}
247     else if (x>=b) {sig[1] = 0;}
248
249     if (x<=a||x>=c) {sig[2] = 0;} // SIG II
250     else if (x>=a&&x<=b) {sig[2] = (x-a)/(b-a);}
251     else if (x>=b&&x<=c) {sig[2] = (c-x)/(c-b);}
252
253     if (x<=b||x>=d) {sig[3] = 0;} // SIG III
254     else if (x>=b&&x<=c) {sig[3] = (x-b)/(c-b);}
255     else if (x>=c&&x<=d) {sig[3] = (d-x)/(d-c);}
256
257     if (x<=c) {sig[4] = 0;} // SIG IV
258     else if (x>=c&&x<=d) {sig[4] = (x-c)/(d-c);}
259     else if (x>=d) {sig[4] = 1;}
260
261     z0=0; z1=2; z2=5; z3=6; z4=8;
262
263     score = (sig[0]*z0+sig[1]*z1+sig[2]*z2
264             +sig[3]*z3+sig[4]*z4)
265             /(sig[0]+sig[1]+sig[2]+sig[3]+sig[4]);
266 }
267
268 void fuzzy_reset(void)
269 {
270     pga_dr1 = 0;
271     score = 0;
272 }
273
274 void komunikasi_http(void) {

```

```

275     if((wifiMulti.run() == WL_CONNECTED)) {
276
277         HTTPClient http;
278
279         if(command=='x') command_send='0';
280         else command_send='x';
281
282         http.begin("http://192.168.43.148/nodebawah.php?gacc="
283             + String(ground_acc,4)
284             + "&angle=" + String(inc_angle,3)
285             + "&drift=" + String(drift_ratio,3)
286             + "&pga=" + String(pga,4)
287             + "&anglepeak=" + String(inc_angle_peak,3)
288             + "&height=" + String(h)
289             + "&displacepeak=" + String(displacement_peak)
290             + "&driftpeak=" + String(drift_ratio_peak,3)
291             + "&pgadr1=" + String(pga_dr1,4)
292             + "&score=" + String(score)
293             + "&command=" + String(command_send));
294
295         int httpCode = http.GET();
296         if(httpCode > 0) {
297             if(httpCode == HTTP_CODE_OK) {
298                 String payload = http.getString();
299
300                 // Parsing data
301                 int j = 0;
302                 dt[j] = "";
303                 for (int k = 58; k < payload.length()-1; k++) {
304                     if ((payload[k] == '#' || (payload[k] == '*')) {
305                         j++;
306                         dt[j] = "";
307                     }
308                     else {
309                         dt[j] = dt[j] + payload[k];
310                     }
311                 }
312                 command = dt[1][0];
313                 server_time = dt[2];
314                 top_angle_peak = dt[3].toFloat();
315
316                 // Serial.print(command);
317                 // Serial.print(" || ");
318                 // Serial.print(server_time);
319                 // Serial.print(" || ");
320                 // Serial.println(top_angle_peak);
321             }
322         }
323         http.end();
324     }
325 }
326
327 void logSDCard() {
328     if (imuCal==0) {
329         dataMessage = server_time + "\tPGA=" + String(pga,4)
330             + "g\tBotAngle=" + String(inc_angle_peak,3)

```

```

331         + "°\tTopAngle=" + String(top_angle_peak,3)
332         + "°\tH=" + String(h)
333         + " cm\tDisp=" + String(displacement_peak)
334         + " cm\tDR=" + String(drift_ratio_peak,3)
335         + "%\tPGA-DR1=" + String(pga_dr1,4)
336         + "g\tScore=" + String(score) + "\r\n";
337     }
338     if (imuCal==1) {
339         dataMessage = server_time + "\tBOT CAL\tAccOffsetX="
340             + String(mpu.getXAccelOffset())
341             + "\tAccOffsetY=" + String(mpu.getYAccelOffset())
342             + "\tAccOffsetZ=" + String(mpu.getZAccelOffset())
343             + "\tGyroOffsetX=" + String(mpu.getXGyroOffset())
344             + "\tGyroOffsetY=" + String(mpu.getYGyroOffset())
345             + "\tGyroOffsetZ=" + String(mpu.getZGyroOffset())
346             + "\r\n";
347     }
348     Serial.print("Save data: ");
349     Serial.println(dataMessage);
350     if (imuCal==0)
351         {appendFile(SD, "/data.txt", dataMessage.c_str());}
352     if (imuCal==1)
353         {appendFile(SD, "/calib.txt", dataMessage.c_str());}
354 }

```

Lampiran 2 Listing program node atas

Gambar lampiran 2 Listing program mikrokontroler node atas

```

1  #include <Arduino.h>
2  #include <WiFi.h>
3  #include <WiFiMulti.h>
4  #include <HTTPClient.h>
5  #include "I2Cdev.h"
6  #include "MPU6050_6Axis_MotionApps20.h"
7
8  #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
9      #include "Wire.h"
10 #endif
11
12 WiFiMulti wifiMulti;
13 MPU6050 mpu;
14 TaskHandle_t Task1;
15
16 // Wifi
17 String dt[4];
18 char command;
19 char command_send;
20 String server_time;
21
22 // LED biru ESP32
23 const int ledPin = 2;
24 const int freq = 5000;
25 const int ledChannel = 0;
26 const int resolution = 8;
27 int dutyCycle = 0;
28
29 // MPU control/status vars
30 uint8_t devStatus;
31 uint8_t fifoBuffer[64];
32
33 // orientation/motion vars
34 Quaternion q;
35 VectorFloat gravity;
36 float ypr[3];
37
38 float pitch;
39 float roll;
40 float inc_angle;
41
42 bool start = false;
43
44 // Peak calc
45 float inc_angle_peak;
46
47 // IMU Calibration
48 int imuCal_http;
49 int imuCal_send;
50 float axOff;
51 float ayOff;
52 float azOff;
53 float gxOff;

```

```

54 float gyOff;
55 float gzOff;
56
57 void setup() {
58     Serial.begin(115200);
59
60     wifiMulti.addAP("Hanif Maliki", "hanifmlk");
61
62     IMU_Setup();
63
64     // LED biru ESP32
65     ledcSetup(ledChannel, freq, resolution);
66     ledcAttachPin(ledPin, ledChannel);
67
68     //create a task
69     xTaskCreatePinnedToCore(
70         Task1code,      /* Task function. */
71         "Task1",        /* name of task. */
72         10000,          /* Stack size of task */
73         NULL,           /* parameter of the task */
74         1,              /* priority of the task */
75         &Task1,         /* Task handle */
76         0);             /* pin task to core 0 */
77     delay(500);
78 }
79
80 void Task1code( void * pvParameters ) {
81     for(;;){
82         komunikasi_http();
83     }
84 }
85
86 void loop() {
87     if(command=='q') start = true;
88     if(command=='a') start = false;
89     if(command=='w') peak_reset();
90     if(command=='g') imu_calibration();
91
92     IMU_Calc();
93
94     if (start==true) {peak_calc();}
95
96     print_Data();
97 }
98
99 void print_Data(void)
100 {
101     Serial.print(inc_angle,3);
102     Serial.print("\t");
103     Serial.print(inc_angle_peak,3);
104     Serial.print("\t");
105     Serial.println();
106 }
107
108 void IMU_Calc(void)
109 {

```



```

110 // read a packet from FIFO
111 if (mpu.dmpGetCurrentFIFOPacket(fifoBuffer)) {
112     mpu.dmpGetQuaternion(&q, fifoBuffer);
113     mpu.dmpGetGravity(&gravity, &q);
114     mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
115
116     pitch = -ypr[1]*180/M_PI; // AngleY
117     roll = ypr[2]*180/M_PI; // AngleX
118
119     inc_angle = roll;
120
121     dutyCycle = 10; // terang LED biru 0-255
122     ledcWrite(ledChannel, dutyCycle);
123 }
124 else {
125     if (dutyCycle>0) dutyCycle--;
126     ledcWrite(ledChannel, dutyCycle);
127 }
128 }
129
130 void imu_calibration(void)
131 {
132     mpu.CalibrateAccel(20);
133     mpu.CalibrateGyro(20);
134
135     axOff=mpu.getXAccelOffset();
136     ayOff=mpu.getYAccelOffset();
137     azOff=mpu.getZAccelOffset();
138     gxOff=mpu.getXGyroOffset();
139     gyOff=mpu.getYGyroOffset();
140     gzOff=mpu.getZGyroOffset();
141
142     imuCal_http = 1;
143 }
144
145 void peak_calc(void)
146 {
147     if (abs(inc_angle)>inc_angle_peak)
148         {inc_angle_peak=abs(inc_angle);}
149 }
150
151 void peak_reset(void)
152 {
153     inc_angle_peak = 0;
154 }
155
156 void komunikasi_http(void) {
157     if((wifiMulti.run() == WL_CONNECTED)) {
158
159         HTTPClient http;
160
161         if(command=='x') command_send='0';
162         else command_send='x';
163
164         if (imuCal_http==1) {imuCal_http=0; imuCal_send=1;}
165         else if (imuCal_http==0) {imuCal_send=0;}

```

```

166     http.begin("http://192.168.43.148/nodeatas.php?angle="
167         + String(inc_angle,3)
168         + "&anglepeak=" + String(inc_angle_peak,3)
169         + "&command=" + String(command_send)
170         + "&imuCal=" + String(imuCal_send)
171         + "&axOff=" + String(axOff)
172         + "&ayOff=" + String(ayOff)
173         + "&azOff=" + String(azOff)
174         + "&gxOff=" + String(gxOff)
175         + "&gyOff=" + String(gyOff)
176         + "&gzOff=" + String(gzOff));
177
178
179     int httpCode = http.GET();
180     if(httpCode > 0) {
181         if(httpCode == HTTP_CODE_OK) {
182             String payload = http.getString();
183
184             // Parsing data
185             int j = 0;
186             dt[j] = "";
187             for (int k = 49; k < payload.length()-1; k++) {
188                 if ((payload[k] == '#' || (payload[k] == '*')) {
189                     j++;
190                     dt[j] = "";
191                 }
192                 else {
193                     dt[j] = dt[j] + payload[k];
194                 }
195             }
196             command = dt[1][0];
197             server_time = dt[2];
198
199             // Serial.print(command);
200             // Serial.print(" || ");
201             // Serial.println(server_time);
202         }
203     }
204     http.end();
205 }
206 }

```

Lampiran 3 Struktur database pada MySQL

Gambar 1 lampiran 3 Struktur database skripsi

Table	Action	Rows	Type	Collation
botcommand	Browse Structure Search Insert Empty Drop	151	InnoDB	utf8mb4_general_ci
botdata	Browse Structure Search Insert Empty Drop	32,172	InnoDB	utf8mb4_general_ci
datalogger	Browse Structure Search Insert Empty Drop	14	InnoDB	utf8mb4_general_ci
topcalib	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci
topcommand	Browse Structure Search Insert Empty Drop	151	InnoDB	utf8mb4_general_ci
topdata	Browse Structure Search Insert Empty Drop	37,198	InnoDB	utf8mb4_general_ci
6 tables	Sum	69,691	InnoDB	utf8mb4_general_ci

Gambar 2 lampiran 3 Struktur tabel botcommand

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	bigint(20)			No	None		AUTO_INCREMENT
2	command	char(1)	utf8mb4_general_ci		Yes	NULL		
3	time	timestamp			No	current_timestamp()		ON UPDATE CURRE


Gambar 3 lampiran 3 Struktur tabel botdata

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	bigint(20)			No	None		AUTO_INCREMENT
2	gndacc	float			Yes	NULL		
3	angle	float			Yes	NULL		
4	drift	float			Yes	NULL		
5	pga	float			Yes	NULL		
6	anglepeak	float			Yes	NULL		
7	height	float			Yes	NULL		
8	displacepeak	float			Yes	NULL		
9	driftpeak	float			Yes	NULL		
10	pgadr1	float			Yes	NULL		
11	score	float			Yes	NULL		
12	time	timestamp			No	current_timestamp()		


Gambar 4 lampiran 3 Struktur tabel datalogger

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	bigint(20)			No	None		AUTO_INCREMENT
2	pga	float			Yes	NULL		
3	botanglepeak	float			Yes	NULL		
4	topanglepeak	float			Yes	NULL		
5	height	float			Yes	NULL		
6	displacepeak	float			Yes	NULL		
7	driftpeak	float			Yes	NULL		
8	pgadr1	float			Yes	NULL		
9	score	float			Yes	NULL		
10	time	timestamp			No	current_timestamp()		


Gambar 5 lampiran 3 Struktur tabel topcalib

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	bigint(20)			No	None		AUTO_INCREMENT
2	pidCal	int(11)			Yes	NULL		
3	axOff	int(11)			Yes	NULL		
4	ayOff	int(11)			Yes	NULL		
5	azOff	int(11)			Yes	NULL		
6	gxOff	int(11)			Yes	NULL		
7	gyOff	int(11)			Yes	NULL		
8	gzOff	int(11)			Yes	NULL		
9	time	timestamp			No	current_timestamp()		

Gambar 6 lampiran 3 Struktur tabel topcommand

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	bigint(20)			No	None		AUTO_INCREMENT
2	command	char(1)	utf8mb4_general_ci		Yes	NULL		
3	time	timestamp			No	current_timestamp()		

Gambar 7 lampiran 3 Struktur tabel topdata

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	bigint(20)			No	None		AUTO_INCREMENT
2	angle	float			Yes	NULL		
3	anglepeak	float			Yes	NULL		
4	time	timestamp			No	current_timestamp()		

Lampiran 4 Listing program komunikasi server

Gambar 1 lampiran 4 Listing program komunikasi server nodebawah.php

```

1  <?php
2  include_once('fungsi.php');
3
4  if($_GET['gacc'] != '' and $_GET['angle'] != '' and
5     $_GET['drift'] != '' and $_GET['pga'] != '' and
6     $_GET['anglepeak'] != '' and $_GET['height'] != '' and
7     $_GET['displacepeak'] != '' and
8     $_GET['driftpeak'] != '' and $_GET['pgadr1'] != '' and
9     $_GET['score'] != '' and $_GET['command'] != '') {
10     $gndacc = $_GET['gacc'];
11     $angle = $_GET['angle'];
12     $drift = $_GET['drift'];
13     $pga = $_GET['pga'];
14     $anglepeak = $_GET['anglepeak'];
15     $height = $_GET['height'];
16     $displacepeak = $_GET['displacepeak'];
17     $driftpeak = $_GET['driftpeak'];
18     $pgadr1 = $_GET['pgadr1'];
19     $score = $_GET['score'];
20     $command = $_GET['command'];
21
22     // Create connection
23     $conn = new mysqli($servername, $username,
24                       $password, $dbname);
25     // Check connection
26     if ($conn->connect_error) {
27         die("Connection failed: " . $conn->connect_error);
28     }
29
30     $sql = "INSERT INTO botdata (gndacc, angle, drift, pga,
31                                anglepeak, height,
32                                displacepeak, driftpeak,
33                                pgadr1, score)
34     VALUES ('".$gndacc."', '".$angle."', '".$drift."',
35             '".$pga."', '".$anglepeak."', '".$height."',
36             '".$displacepeak."', '".$driftpeak."',
37             '".$pgadr1."', '".$score."')";
38
39     $conn->query($sql);
40
41     $last_reading = getLastCommand();
42     $last_reading_command = $last_reading["command"];
43
44     if ($last_reading_command != 'x') {
45         if ($command == 'x') {
46             $sql = "INSERT INTO botcommand (command)
47             VALUES ('" . $command . "')";
48
49             $conn->query($sql);
50             $last_reading_command = 'x';
51         }
52     }
53

```

```

54     $last_reading = getLastReadings();
55     $last_reading_time = $last_reading["time"];
56
57     $last_reading = getLastReadings2();
58     $last_reading_anglepeak2 = $last_reading["anglepeak"];
59
60     echo "<br>Command&Time&TopAngle: *";
61     echo $last_reading_command;
62     echo "*";
63     echo $last_reading_time;
64     echo "*";
65     echo $last_reading_anglepeak2;
66     echo "#";
67
68     $conn->close();
69 }
70
71
72
73

```

Gambar 2 lampiran 4 Listing program komunikasi server nodeatas.php

```

1  <?php
2  include_once('fungsi.php');
3
4  if($_GET['imuCal'] != '' and
5      $_GET['axOff'] != '' and $_GET['ayOff'] != '' and
6      $_GET['azOff'] != '' and $_GET['gxOff'] != '' and
7      $_GET['gyOff'] != '' and $_GET['gzOff'] != '') {
8      $pidCal = $_GET['imuCal'];
9      $axOff = $_GET['axOff'];
10     $ayOff = $_GET['ayOff'];
11     $azOff = $_GET['azOff'];
12     $gxOff = $_GET['gxOff'];
13     $gyOff = $_GET['gyOff'];
14     $gzOff = $_GET['gzOff'];
15
16     if ($pidCal=='1') {
17         $conn = new mysqli($servername, $username,
18                             $password, $dbname);
19         if ($conn->connect_error) {
20             die("Connection failed: " . $conn->connect_error);
21         }
22
23         $sql= "INSERT INTO topcalib (pidCal, axOff, ayOff,
24                                     azOff, gxOff, gyOff, gzOff)
25             VALUES ('".$_pidCal."', '".$_axOff."', '".$_ayOff."',
26                     '".$_azOff."', '".$_gxOff."',
27                     '".$_gyOff."', '".$_gzOff.'")";
28
29         $conn->query($sql);
30         $conn->close();
31     }
32 }
33
34 if($_GET['angle'] != '' and $_GET['anglepeak'] != '' and

```

```

35     $_GET['command'] != '') {
36         $angle = $_GET['angle'];
37         $anglepeak = $_GET['anglepeak'];
38         $command = $_GET['command'];
39
40         $conn = new mysqli($servername, $username,
41                             $password, $dbname);
42         if ($conn->connect_error) {
43             die("Connection failed: " . $conn->connect_error);
44         }
45
46         $sql = "INSERT INTO topdata (angle, anglepeak)
47             VALUES ('" . $angle . "', '" . $anglepeak . "')";
48
49         $conn->query($sql);
50
51         $last_reading = getLastCommand2();
52         $last_reading_command = $last_reading["command"];
53
54         if ($last_reading_command != 'x') {
55             if ($command == 'x') {
56                 $sql = "INSERT INTO topcommand (command)
57                     VALUES ('" . $command . "')";
58
59                 $conn->query($sql);
60                 $last_reading_command = 'x';
61             }
62         }
63
64         $last_reading = getLastReadings2();
65         $last_reading_time = $last_reading["time"];
66
67         echo "<br>Command&Time: *";
68         echo $last_reading_command;
69         echo "*";
70         echo $last_reading_time;
71         echo "#";
72
73         $conn->close();
74     }

```

Lampiran 5 Listing program graphical user interface

Gambar 1 lampiran 5 Listing program gui.php

```

1  <?php
2      include_once('fungsi.php');
3
4      $command="x";
5
6      if (isset($_GET["command"])){$command = $_GET["command"];}
7      if ($command!="x"){writeCommand($command);}
8      if ($command=="a"){writeLog();}
9  ?>
10
11  <!DOCTYPE html>
12  <html>
13  <head><meta http-equiv="Content-Type" content="text/html;
14  charset=utf-8">
15      <link rel="stylesheet" type="text/css" href="style.css">
16      <meta name="viewport" content="width=device-width,
17  initial-scale=1">
18      <script src="jquery.min.js"></script>
19      <script>
20          setInterval(function (){$('#latest-time')
21              .load('read-time.php');}, 100);
22          setInterval(function (){$('#latest-data')
23              .load('read-latest.php');}, 100);
24          setInterval(function (){$('#peak-data')
25              .load('read-peak.php');}, 100);
26          setInterval(function (){$('#score-gauge')
27              .load('read-score.php');}, 100);
28          setInterval(function (){$('#logs-table')
29              .load('read-logs.php');}, 100);
30      </script>
31  </head>
32  <header class="header">
33      <h1> WALL STIFFNESS</h1>
34      <form method="get" style="margin-bottom: 13px">
35          <input type="text" name="command" placeholder="Node
36  command">
37          <input type="submit" value="Send">
38      </form>
39      <form class="button" method="get">
40          <input type="hidden" name="command" value="q">
41          <input type="submit" class="buttonGreen" value="Start">
42      </form>
43      <form class="button" method="get">
44          <input type="hidden" name="command" value="a">
45          <input type="submit" class="buttonGreen" value="Stop">
46      </form>
47      <form class="button" method="get">
48          <input type="hidden" name="command" value="w">
49          <input type="submit" class="buttonGreen" value="Reset">
50      </form>
51  </header>
52  <body>
53      <p id="latest-time">

```



```

51 <div class="box gauge--1">
52   <h3>SCORE</h3>
53   <center id="score-gauge">
54 </div>
55 <section class="content">
56   <table cellspacing="5" cellpadding="5">
57     <tr>
58       <th colspan="4">Latest reading</th>
59     </tr>
60     <tr>
61       <td>Ground<br>Acc</td>
62       <td>Bot. Inc.<br>Angle</td>
63       <td>Top Inc.<br>Angle</td>
64       <td>Drift<br>Ratio</td>
65     </tr>
66     <tr id="latest-data">
67   </table>
68   <table cellspacing="5" cellpadding="5">
69     <tr>
70       <th colspan="4">Peak Data</th>
71     </tr>
72     <tr>
73       <td>Ground<br>Acc</td>
74       <td>Bot. Inc.<br>Angle</td>
75       <td>Top Inc.<br>Angle</td>
76       <td>Drift<br>Ratio</td>
77     </tr>
78     <tr id="peak-data">
79   </table>
80 </section>
81 <?php
82   echo '<h2>View Recent Data Logs</h2>
83     <table cellspacing="5" cellpadding="5" id="logs-
84 table">';
85 ?>
86 </body>
87 </html>

```

Gambar 2 lampiran 5 Listing program style.css

```

1  body {
2      width: 60%;
3      margin: auto;
4      text-align: center;
5      font-family: Arial;
6      top: 50%;
7      left: 50%;
8  }
9
10 @media screen and (max-width: 800px) {
11     body {
12         width: 100%;
13     }
14 }
15
16 table {
17     margin-left: auto;
18     margin-right: auto;
19 }
20
21 div {
22     margin-left: auto;
23     margin-right: auto;
24 }
25
26 h2 { font-size: 2.5rem; }
27
28 .header {
29     padding: 1rem;
30     margin: 0 0 2rem 0;
31     background: #f2f2f2;
32 }
33
34 .button {
35     display: inline-block;
36 }
37
38 .buttonGreen {
39     background-color: #4CAF50; /* Green */
40     border: none;
41     color: white;
42     padding: 10px 25px;
43     text-align: center;
44     text-decoration: none;
45     display: inline-block;
46     font-size: 15px;
47     cursor: pointer;
48 }
49
50 .buttonGreen:hover {
51     background-color: #3e8e41;
52 }
53
54 h1 {
55     font-size: 2rem;

```

```

56     font-family: Arial, sans-serif;
57     text-align: center;
58     text-transform: uppercase;
59 }
60
61 .content {
62     display: flex;
63 }
64
65 @media screen and (max-width: 500px) /* Mobile */ {
66     .content {
67         flex-direction: column;
68     }
69 }
70
71 .mask {
72     position: relative;
73     overflow: hidden;
74     display: block;
75     width: 12.5rem;
76     height: 6.25rem;
77     margin: 1.25rem;
78 }
79
80 .semi-circle {
81     position: relative;
82     display: block;
83     width: 12.5rem;
84     height: 6.25rem;
85     background: linear-gradient(to right, #ff0000 0%,
#05b027 100%);
86     border-radius: 50% 50% 50% 50% / 100% 100% 0% 0%;
87 }
88
89 .semi-circle::before {
90     content: "";
91     position: absolute;
92     bottom: 0;
93     left: 50%;
94     z-index: 2;
95     display: block;
96     width: 8.75rem;
97     height: 4.375rem;
98     margin-left: -4.375rem;
99     background: #fff;
100     border-radius: 50% 50% 50% 50% / 100% 100% 0% 0%;
101 }
102
103 .semi-circle--mask {
104     position: absolute;
105     top: 0;
106     left: 0;
107     width: 12.5rem;
108     height: 12.5rem;
109     background: transparent;
110     transform: rotate(120deg) translate3d(0, 0, 0);

```

```

111     transform-origin: center center;
112     backface-visibility: hidden;
113     transition: all 0.3s ease-in-out;
114 }
115
116 .semi-circle--mask::before {
117     content: "";
118     position: absolute;
119     top: 0;
120     left: 0%;
121     z-index: 2;
122     display: block;
123     width: 12.625rem;
124     height: 6.375rem;
125     margin: -1px 0 0 -1px;
126     background: #f2f2f2;
127     border-radius: 50% 50% 50% 50% / 100% 100% 0% 0%;
128 }
129
130 #logs-table { border-collapse: collapse; }
131
132 #logs-table td, #logs-table th {
133     border: 1px solid #ddd;
134     padding: 10px;
135 }
136
137 #logs-table tr:nth-child(even) {background-color: #f2f2f2;}
138
139 #logs-table tr:hover {background-color: #ddd;}
140
141 #logs-table th {
142     padding: 10px;
143     background-color: #2f4468;
144     color: white;
145 }

```

Gambar 3 lampiran 5 Listing program read-time.php

```

1  <?php
2  include_once('fungsi.php');
3
4  $last_reading = getLastReadings();
5  $last_reading_time = $last_reading["time"];
6
7  echo "Last reading: ";
8  echo $last_reading_time;

```

Gambar 4 lampiran 5 Listing program read-latest.php

```

1  <?php
2      include_once('fungsi.php');
3
4      $last_reading = getLastReadings();
5      $last_reading_acc = $last_reading["gndacc"];
6      $last_reading_angle = $last_reading["angle"];
7      $last_reading_drift = $last_reading["drift"];
8
9      $last_reading = getLastReadings2();
10     $last_reading_angle2 = $last_reading["angle"];
11
12     echo '<td>' . $last_reading_acc . 'g</td>'
13         '<td>' . $last_reading_angle . '°</td>'
14         '<td>' . $last_reading_angle2 . '°</td>'
15         '<td>' . $last_reading_drift . '%</td>';

```

Gambar 5 lampiran 5 Listing program read-peak.php

```

1  <?php
2      include_once('fungsi.php');
3
4      $last_reading = getLastReadings();
5      $last_reading_accpeak = $last_reading["pga"];
6      $last_reading_anglepeak = $last_reading["anglepeak"];
7      $last_reading_driftpeak = $last_reading["driftpeak"];
8
9      $last_reading = getLastReadings2();
10     $last_reading_anglepeak2 = $last_reading["anglepeak"];
11
12     echo '<td>' . $last_reading_accpeak . 'g</td>'
13         '<td>' . $last_reading_anglepeak . '°</td>'
14         '<td>' . $last_reading_anglepeak2 . '°</td>'
15         '<td>' . $last_reading_driftpeak . '%</td>';

```

Gambar 6 lampiran 5 Listing program read-score.php

```

1  <?php
2      include_once('fungsi.php');
3
4      $last_reading = getLastReadings();
5      $last_reading_score = $last_reading["score"];
6  ?>
7
8  <script>
9      var value1 = <?php echo $last_reading_score; ?>;
10     setScore(value1);
11
12     function setScore(curVal){
13         var minScore = 0;
14         var maxScore = 4;
15
16         var newVal = scaleValue(curVal, [minScore, maxScore],
17                                 [0, 180]);
18         $(' .gauge--1 .semi-circle--mask').attr({
19             style: '-webkit-transform: rotate('+ newVal+ 'deg);'+
20             '-moz-transform: rotate('+ newVal+ 'deg);'+
21             'transform: rotate(' + newVal + 'deg);'
22         });
23         $('#scoreVal').text(curVal + '/4');
24     }
25
26     function scaleValue(value, from, to) {
27         var scale = (to[1] - to[0]) / (from[1] - from[0]);
28         var capped = Math.min(from[1],
29                               Math.max(from[0],value)) - from[0];
30         return ~~(capped * scale + to[0]);
31     }
32 </script>
33 <div class="mask">
34     <div class="semi-circle"></div>
35     <div class="semi-circle--mask"></div>
36 </div>
37 <p style="font-size: 30px;
38     margin-bottom: 15px" id="scoreVal">--</p>
39
40 <?php
41 if ($last_reading_score>=4) {
42     echo '<p style="font-size: 20px; color: #05b027;
43         margin: 0">Good</p>';
44 }
45 else if ($last_reading_score==0) {
46     echo '<p style="font-size: 20px; margin: 0">--</p>';
47 }
48 else {
49     echo '<p style="font-size: 20px; color: #ff0000;
50         margin: 0">Bad</p>';
51 }
52 ?>

```

Gambar 7 lampiran 5 Listing program read-logs.php

```

1  <?php
2      echo '<tr>
3          <th>ID</th>
4          <th>Peak Ground Acc</th>
5          <th>Peak Bottom Inc. Angle</th>
6          <th>Peak Top Inc. Angle</th>
7          <th>Height</th>
8          <th>Peak Displacement</th>
9          <th>Peak Drift Ratio</th>
10         <th>PGA at DR = 1%</th>
11         <th>Score</th>
12         <th>Timestamp</th>
13     </tr>';
14
15     include_once('fungsi.php');
16
17     $result = getAllLogs(20);
18     while ($row = $result->fetch_assoc()) {
19         $row_id = $row["id"];
20         $row_pga = $row["pga"];
21         $row_botanglepeak = $row["botanglepeak"];
22         $row_topanglepeak = $row["topanglepeak"];
23         $row_height = $row["height"];
24         $row_displacepeak = $row["displacepeak"];
25         $row_driftpeak = $row["driftpeak"];
26         $row_pgadr1 = $row["pgadr1"];
27         $row_score = $row["score"];
28         $row_time = $row["time"];
29
30         echo '<tr>
31             <td>' . $row_id . '</td>
32             <td>' . $row_pga . 'g</td>
33             <td>' . $row_botanglepeak . '°</td>
34             <td>' . $row_topanglepeak . '°</td>
35             <td>' . $row_height . ' cm</td>
36             <td>' . $row_displacepeak . ' cm</td>
37             <td>' . $row_driftpeak . '%</td>
38             <td>' . $row_pgadr1 . 'g</td>
39             <td>' . $row_score . '</td>
40             <td>' . $row_time . '</td>
41         </tr>';
42     }

```

Lampiran 6 Listing program fungsi.php

Gambar lampiran 6 Listing program web server fungsi.php

```

1  <?php
2  $servername = "localhost";
3  $username = "hanipm97";
4  $password = "hanifmaliki";
5  $dbname = "skripsi";
6
7  function getAllLogs($limit) {
8      global $servername, $username, $password, $dbname;
9
10     $conn = new mysqli($servername, $username,
11                        $password, $dbname);
12     if ($conn->connect_error) {
13         die("Connection failed: " . $conn->connect_error);
14     }
15
16     $sql = "SELECT id, pga, botanglepeak, topanglepeak,
17            height, displacepeak, driftpeak,
18            pgadrl, score, time FROM datalogger
19            ORDER BY ID DESC limit " . $limit;
20
21     if ($result = $conn->query($sql)) {
22         return $result;
23     }
24     else {return false;}
25     $conn->close();
26 }
27
28 function getLastReadings() {
29     global $servername, $username, $password, $dbname;
30
31     $conn = new mysqli($servername, $username,
32                        $password, $dbname);
33     if ($conn->connect_error) {
34         die("Connection failed: " . $conn->connect_error);
35     }
36
37     $sql = "SELECT id, gndacc, angle, drift, pga, anglepeak,
38            height, displacepeak, driftpeak, pgadrl, score,
39            time FROM botdata ORDER BY ID DESC limit 1";
40
41     if ($result = $conn->query($sql)) {
42         return $result->fetch_assoc();
43     }
44     else {return false;}
45     $conn->close();
46 }
47
48 function getLastReadings2() {
49     global $servername, $username, $password, $dbname;
50
51     $conn = new mysqli($servername, $username,
52                        $password, $dbname);
53     if ($conn->connect_error) {

```



```

54     die("Connection failed: " . $conn->connect_error);
55 }
56
57 $sql = "SELECT id, angle, anglepeak, time FROM topdata
58         ORDER BY ID DESC limit 1";
59
60 if ($result = $conn->query($sql)) {
61     return $result->fetch_assoc();
62 }
63 else {return false;}
64 $conn->close();
65 }
66
67 function getLastCommand() {
68     global $servername, $username, $password, $dbname;
69
70     $conn = new mysqli($servername, $username,
71                       $password, $dbname);
72     if ($conn->connect_error) {
73         die("Connection failed: " . $conn->connect_error);
74     }
75
76     $sql = "SELECT command FROM botcommand
77             ORDER BY ID DESC limit 1";
78
79     if ($result = $conn->query($sql)) {
80         return $result->fetch_assoc();
81     }
82     else {return false;}
83     $conn->close();
84 }
85
86 function getLastCommand2() {
87     global $servername, $username, $password, $dbname;
88
89     $conn = new mysqli($servername, $username,
90                       $password, $dbname);
91     if ($conn->connect_error) {
92         die("Connection failed: " . $conn->connect_error);
93     }
94
95     $sql = "SELECT command FROM topcommand
96             ORDER BY ID DESC limit 1";
97
98     if ($result = $conn->query($sql)) {
99         return $result->fetch_assoc();
100     }
101     else {return false;}
102     $conn->close();
103 }
104
105 function writeCommand($command) {
106     global $servername, $username, $password, $dbname;
107
108     $conn = new mysqli($servername, $username,
109                       $password, $dbname);

```

```

110     if ($conn->connect_error) {
111         die("Connection failed: " . $conn->connect_error);
112     }
113
114     $sql = "INSERT INTO botcommand (command)
115     VALUES ('" . $command . "')";
116
117     $conn->query($sql);
118
119     $sql = "INSERT INTO topcommand (command)
120     VALUES ('" . $command . "')";
121
122     $conn->query($sql);
123     $conn->close();
124 }
125
126 function writeLog() {
127     $last_reading = getLastReadings();
128     $pga = $last_reading["pga"];
129     $anglepeak = $last_reading["anglepeak"];
130     $height = $last_reading["height"];
131     $displacepeak = $last_reading["displacepeak"];
132     $driftpeak = $last_reading["driftpeak"];
133     $pgadr1 = $last_reading["pgadr1"];
134     $score = $last_reading["score"];
135
136     $last_reading = getLastReadings2();
137     $anglepeak2 = $last_reading["anglepeak"];
138
139     global $servername, $username, $password, $dbname;
140
141     $conn = new mysqli($servername, $username,
142                       $password, $dbname);
143     if ($conn->connect_error) {
144         die("Connection failed: " . $conn->connect_error);
145     }
146
147     $sql = "INSERT INTO datalogger (pga, botanglepeak,
148     topanglepeak, height, displacepeak,
149     driftpeak, pgadr1, score)
150     VALUES ('" . $pga . "', '" . $anglepeak . "', '" . $anglepeak2 . "',
151     '" . $height . "', '" . $displacepeak . "',
152     '" . $driftpeak . "', '" . $pgadr1 . "', '" . $score . "')";
153
154     $conn->query($sql);
155     $conn->close();
156 }

```

Lampiran 7 Data pembacaan paket FIFO sensor IMU

Data *raw* yang dibaca pada paket FIFO adalah nilai *accelerometer*, *gyroscope*, dan *quaternion*. Node bawah akan mengkonversikan data tersebut menjadi *ground acceleration*, *bottom inclination angle*, *displacement*, dan *drift ratio*. Node atas hanya akan mengkonversikan data tersebut menjadi *top inclination angle*. Berikut adalah sampel pembacaan paket FIFO dari sensor IMU selama 1 detik menggunakan *serial monitor*:

Tabel 1 lampiran 7 Pembacaan Paket FIFO Node Bawah

No.	Timestamp	Ground Acceleration	Bottom Inc. Angle	Displacement	Drift Ratio
1	11:10:30.023	-0.0001	0.036	0.18	0.063
2	11:10:30.023	-0.0006	0.036	0.18	0.063
3	11:10:30.023	-0.0007	0.036	0.18	0.063
4	11:10:30.023	-0.0009	0.036	0.18	0.063
5	11:10:30.071	-0.0006	0.036	0.18	0.063
6	11:10:30.071	-0.0001	0.036	0.18	0.063
7	11:10:30.071	0.0002	0.036	0.18	0.063
8	11:10:30.071	0.0000	0.036	0.18	0.063
9	11:10:30.118	0.0000	0.036	0.18	0.063
10	11:10:30.118	-0.0001	0.036	0.18	0.063
11	11:10:30.118	0.0001	0.036	0.18	0.063
12	11:10:30.118	0.0002	0.036	0.18	0.063
13	11:10:30.118	0.0005	0.036	0.18	0.063
14	11:10:30.154	0.0001	0.029	0.14	0.051
15	11:10:30.154	-0.0002	0.029	0.14	0.051
16	11:10:30.154	0.0004	0.029	0.14	0.051
17	11:10:30.154	0.0006	0.029	0.14	0.051
18	11:10:30.192	0.0001	0.029	0.14	0.051
19	11:10:30.192	0.0000	0.029	0.14	0.051
20	11:10:30.192	-0.0009	0.029	0.14	0.051
21	11:10:30.243	-0.0005	0.029	0.14	0.051
22	11:10:30.243	-0.0001	0.029	0.14	0.051
23	11:10:30.243	0.0000	0.029	0.14	0.051
24	11:10:30.243	0.0000	0.036	0.18	0.063
25	11:10:30.243	0.0001	0.036	0.18	0.063
26	11:10:30.292	0.0000	0.036	0.18	0.063
27	11:10:30.292	-0.0012	0.036	0.18	0.063
28	11:10:30.292	-0.0012	0.036	0.18	0.063

29	11:10:30.292	0.0000	0.036	0.18	0.063
30	11:10:30.292	0.0002	0.036	0.18	0.063
31	11:10:30.333	0.0000	0.029	0.14	0.051
32	11:10:30.333	-0.0002	0.029	0.14	0.051
33	11:10:30.333	-0.0001	0.029	0.14	0.051
34	11:10:30.333	-0.0009	0.029	0.14	0.051
35	11:10:30.333	-0.0011	0.029	0.14	0.051
36	11:10:30.383	-0.0007	0.036	0.18	0.063
37	11:10:30.383	0.0000	0.036	0.18	0.063
38	11:10:30.383	0.0000	0.029	0.14	0.051
39	11:10:30.383	0.0000	0.029	0.14	0.051
40	11:10:30.383	0.0000	0.036	0.18	0.063
41	11:10:30.423	-0.0006	0.036	0.18	0.063
42	11:10:30.423	-0.0005	0.036	0.18	0.063
43	11:10:30.423	-0.0001	0.043	0.21	0.075
44	11:10:30.423	0.0006	0.043	0.21	0.075
45	11:10:30.471	0.0013	0.043	0.21	0.075
46	11:10:30.471	0.0016	0.043	0.21	0.075
47	11:10:30.471	0.0015	0.050	0.25	0.087
48	11:10:30.471	0.0012	0.050	0.25	0.087
49	11:10:30.524	0.0013	0.050	0.25	0.087
50	11:10:30.524	0.0018	0.050	0.25	0.087
51	11:10:30.524	0.0016	0.057	0.28	0.100
52	11:10:30.524	0.0009	0.057	0.28	0.100
53	11:10:30.524	0.0001	0.057	0.28	0.100
54	11:10:30.524	0.0007	0.057	0.28	0.100
55	11:10:30.572	0.0002	0.064	0.32	0.112
56	11:10:30.572	-0.0010	0.064	0.32	0.112
57	11:10:30.572	-0.0011	0.071	0.35	0.124
58	11:10:30.572	-0.0006	0.071	0.35	0.124
59	11:10:30.616	-0.0005	0.071	0.35	0.124
60	11:10:30.616	-0.0016	0.071	0.35	0.124
61	11:10:30.616	-0.0024	0.071	0.35	0.124
62	11:10:30.616	-0.0020	0.071	0.35	0.124
63	11:10:30.616	-0.0007	0.064	0.32	0.112
64	11:10:30.654	0.0000	0.064	0.32	0.112
65	11:10:30.654	0.0000	0.057	0.28	0.100
66	11:10:30.654	-0.0011	0.057	0.28	0.100
67	11:10:30.654	-0.0017	0.057	0.28	0.100
68	11:10:30.724	-0.0013	0.057	0.28	0.100
69	11:10:30.724	-0.0007	0.057	0.28	0.100

70	11:10:30.724	-0.0007	0.057	0.28	0.099
71	11:10:30.724	-0.0010	0.057	0.28	0.099
72	11:10:30.724	-0.0012	0.050	0.25	0.087
73	11:10:30.724	-0.0006	0.050	0.25	0.087
74	11:10:30.752	-0.0009	0.050	0.25	0.087
75	11:10:30.752	-0.0004	0.050	0.25	0.087
76	11:10:30.752	0.0000	0.050	0.25	0.087
77	11:10:30.794	0.0001	0.043	0.21	0.075
78	11:10:30.794	0.0009	0.043	0.21	0.075
79	11:10:30.794	0.0011	0.043	0.21	0.075
80	11:10:30.794	0.0011	0.043	0.21	0.075
81	11:10:30.794	0.0010	0.043	0.21	0.075
82	11:10:30.835	0.0000	0.043	0.21	0.075
83	11:10:30.835	0.0002	0.043	0.21	0.075
84	11:10:30.835	0.0004	0.043	0.21	0.075
85	11:10:30.835	0.0011	0.043	0.21	0.075
86	11:10:30.869	0.0010	0.050	0.25	0.087
87	11:10:30.869	0.0012	0.050	0.25	0.087
88	11:10:30.869	0.0015	0.050	0.25	0.087
89	11:10:30.906	0.0004	0.050	0.25	0.087
90	11:10:30.906	0.0002	0.050	0.25	0.087
91	11:10:30.906	0.0001	0.050	0.25	0.087
92	11:10:30.906	0.0001	0.057	0.28	0.099
93	11:10:30.954	0.0006	0.057	0.28	0.099
94	11:10:30.954	0.0006	0.057	0.28	0.099
95	11:10:30.954	0.0002	0.064	0.32	0.112
96	11:10:30.954	0.0005	0.064	0.32	0.112
97	11:10:30.954	0.0006	0.064	0.32	0.112
98	11:10:30.991	0.0001	0.064	0.32	0.112
99	11:10:30.991	0.0000	0.064	0.32	0.112
100	11:10:30.991	-0.0001	0.064	0.32	0.112

Tabel 2 lampiran 7 Pembacaan Paket FIFO Node Atas

No.	Timestamp	Top Inclination Angle	No.	Timestamp	Top Inclination Angle
1	11:31:30.011	-1.08	51	11:31:30.513	-1.98
2	11:31:30.011	-1.53	52	11:31:30.513	-1.97
3	11:31:30.011	-1.14	53	11:31:30.513	-1.97
4	11:31:30.052	-0.39	54	11:31:30.513	-1.97
5	11:31:30.052	-0.19	55	11:31:30.562	-1.98
6	11:31:30.052	-1.16	56	11:31:30.562	-1.99

7	11:31:30.052	-2.37	57	11:31:30.562	-1.99
8	11:31:30.052	-1.63	58	11:31:30.562	-1.99
9	11:31:30.092	-1.01	59	11:31:30.562	-1.99
10	11:31:30.092	-1.79	60	11:31:30.602	-1.99
11	11:31:30.092	-2.57	61	11:31:30.602	-2.00
12	11:31:30.092	-2.42	62	11:31:30.602	-2.00
13	11:31:30.127	-1.88	63	11:31:30.602	-2.00
14	11:31:30.127	-1.80	64	11:31:30.642	-2.00
15	11:31:30.127	-1.94	65	11:31:30.642	-2.00
16	11:31:30.161	-1.67	66	11:31:30.642	-2.00
17	11:31:30.161	-1.34	67	11:31:30.642	-2.00
18	11:31:30.161	-1.46	68	11:31:30.677	-2.00
19	11:31:30.197	-1.97	69	11:31:30.677	-2.00
20	11:31:30.197	-2.36	70	11:31:30.677	-2.00
21	11:31:30.197	-2.47	71	11:31:30.711	-2.00
22	11:31:30.197	-2.35	72	11:31:30.711	-2.00
23	11:31:30.244	-2.08	73	11:31:30.711	-1.99
24	11:31:30.244	-1.81	74	11:31:30.762	-1.99
25	11:31:30.244	-1.71	75	11:31:30.762	-1.98
26	11:31:30.244	-1.77	76	11:31:30.762	-1.97
27	11:31:30.244	-1.90	77	11:31:30.762	-1.97
28	11:31:30.282	-2.03	78	11:31:30.762	-1.97
29	11:31:30.282	-2.15	79	11:31:30.762	-1.97
30	11:31:30.282	-2.16	80	11:31:30.802	-1.97
31	11:31:30.282	-2.07	81	11:31:30.802	-1.97
32	11:31:30.322	-1.97	82	11:31:30.802	-1.97
33	11:31:30.322	-1.94	83	11:31:30.802	-1.97
34	11:31:30.322	-1.98	84	11:31:30.842	-1.97
35	11:31:30.322	-2.03	85	11:31:30.842	-1.97
36	11:31:30.359	-2.07	86	11:31:30.842	-1.97
37	11:31:30.359	-2.04	87	11:31:30.842	-1.97
38	11:31:30.359	-2.00	88	11:31:30.876	-1.97
39	11:31:30.393	-1.98	89	11:31:30.876	-1.97
40	11:31:30.393	-1.98	90	11:31:30.876	-1.97
41	11:31:30.393	-1.99	91	11:31:30.910	-1.98
42	11:31:30.393	-2.02	92	11:31:30.910	-1.98
43	11:31:30.427	-2.03	93	11:31:30.910	-1.98
44	11:31:30.427	-2.01	94	11:31:30.963	-1.98
45	11:31:30.427	-2.01	95	11:31:30.963	-1.98
46	11:31:30.461	-1.99	96	11:31:30.963	-1.98
47	11:31:30.461	-1.97	97	11:31:30.963	-1.98

48	11:31:30.461	-1.98	98	11:31:30.963	-1.98
49	11:31:30.513	-1.99	99	11:31:30.963	-1.98
50	11:31:30.513	-1.99			

Lampiran 8 Data pengujian fuzzy logic

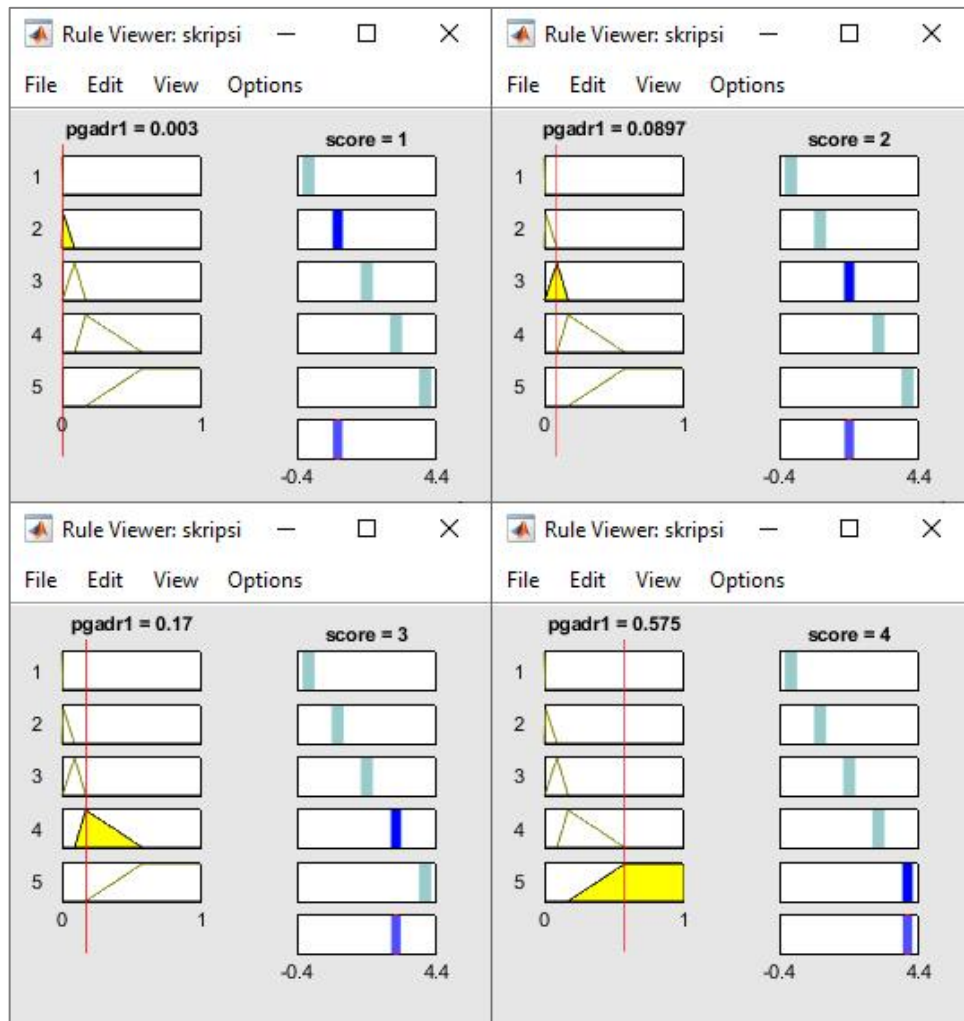
Untuk mengetes logika *fuzzy* yang digunakan pada sistem, dilakukan uji coba dengan cara memanipulasi masukan *fuzzy* dengan data *dummy*. Hasil dari *fuzzy logic* pada sistem ditampilkan pada *serial monitor*, bagian kiri adalah *input* (PGA_{DR1}), sedangkan bagian kanan adalah *output* (*score*). Berikut adalah hasil pengujian *fuzzy logic* pada sistem:

Gambar 1 lampiran 8 Hasil Pengujian Fuzzy Logic pada Sistem

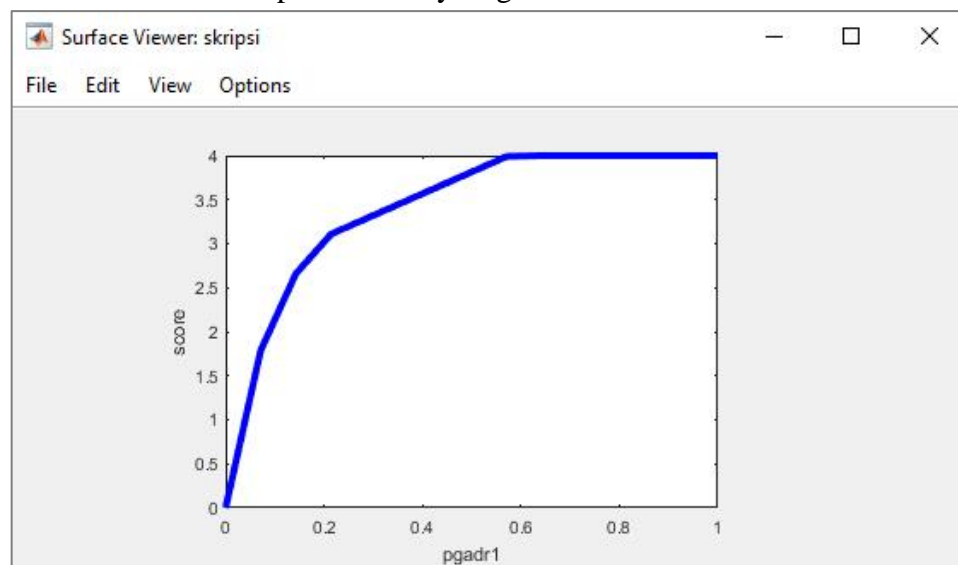
COM3	COM3	COM3	COM3
0.0000 0.00	0.0880 1.98	0.1697 2.99	0.5745 4.00
0.0010 0.33	0.0890 1.99	0.1698 2.99	0.5746 4.00
0.0020 0.67	0.0891 1.99	0.1699 3.00	0.5747 4.00
0.0030 1.00	0.0892 1.99	0.1700 3.00	0.5748 4.00
0.0130 1.12	0.0893 2.00	0.1701 3.00	0.5749 4.00
0.0230 1.23	0.0894 2.00	0.1702 3.00	0.5750 4.00
0.0330 1.35	0.0895 2.00	0.1703 3.00	0.5751 4.00
0.0430 1.46	0.0896 2.00	0.2703 3.25	0.6751 4.00
0.0530 1.58	0.0897 2.00	0.3703 3.49	0.7751 4.00
0.0630 1.69	0.0997 2.12	0.4703 3.74	0.8751 4.00
0.0730 1.81	0.1097 2.25	0.5703 3.99	0.9751 4.00
0.0830 1.92	0.1197 2.37	0.5713 3.99	1.0751 4.00
0.0840 1.93	0.1297 2.50	0.5723 3.99	1.1751 4.00
0.0850 1.95	0.1397 2.62	0.5733 4.00	1.2751 4.00
0.0860 1.96	0.1497 2.74	0.5743 4.00	1.3751 4.00
0.0870 1.97	0.1597 2.87	0.5744 4.00	1.4751 4.00
<input checked="" type="checkbox"/> Autoscroll <input type="checkbox"/> Show	<input checked="" type="checkbox"/> Autoscroll <input type="checkbox"/> Show	<input checked="" type="checkbox"/> Autoscroll <input type="checkbox"/> Show	<input checked="" type="checkbox"/> Autoscroll <input type="checkbox"/> Show

Hasil dari *fuzzy logic* pada sistem akan dibandingkan dengan hasil *fuzzy logic* yang dibuat pada MATLAB untuk menghitung akurasi *fuzzy logic*. Berikut adalah hasil *fuzzy logic* pada MATLAB:

Gambar 2 lampiran 8 Fuzzy Logic MATLAB Rule Viewer



Gambar 3 lampiran 8 Fuzzy Logic MATLAB Surface Viewer



Lampiran 9 Data Pengujian pada Dinding

Data hasil pengujian pada dinding diambil dari *data logger* yang telah dibuat, berikut adalah data hasil pengujian pada setiap dinding beserta nilai *offset* pada sensor saat pengujian dilaksanakan:

a) Pengujian pada Dinding 1

Tabel 1 Lampiran 9 Kalibrasi Pengujian Dinding 1 Node Bawah

No.	Time	Offset ax	Offset ay	Offset az	Offset gx	Offset gy	Offset gz
1	13:59:09	-2880	376	3044	122	-5	-18
2	14:39:07	-2878	376	3042	122	-4	-18
3	15:27:40	-2876	376	3042	122	-4	-18

Tabel 2 Lampiran 9 Kalibrasi Pengujian Dinding 1 Node Atas

No.	Time	Offset ax	Offset ay	Offset az	Offset gx	Offset gy	Offset gz
1	13:59:06	420	1950	3532	85	311	58
2	14:39:04	418	1950	3534	86	312	59
3	15:27:43	420	1950	3536	86	312	59

Tabel 3 Lampiran 9 Hasil Pengujian Dinding 1 (1)

No.	Time	Wall Height	Peak Bottom Inclination Angle	Peak Top Inclination Angle	Peak Displacement
1	14:02:05	284 cm	0.053°	0.060°	0.26 cm
2	14:05:24	284 cm	0.031°	0.034°	0.15 cm
3	14:14:12	284 cm	0.051°	0.047°	0.25 cm
4	14:19:03	284 cm	0.043°	0.048°	0.21 cm
5	14:27:55	284 cm	0.035°	0.042°	0.17 cm
6	14:41:44	284 cm	0.047°	0.043°	0.23 cm
7	14:46:06	284 cm	0.035°	0.031°	0.17 cm
8	14:53:38	284 cm	0.049°	0.042°	0.24 cm
9	15:31:14	284 cm	0.027°	0.031°	0.13 cm
10	15:35:35	284 cm	0.043°	0.050°	0.21 cm

Tabel 4 Lampiran 9 Hasil Pengujian Dinding 1 (2)

No.	Time	PGA	Peak Drift Ratio	PGA _{DR1}	Score
1	14:02:05	0.0146g	0.092%	0.1580g	2.85
2	14:05:24	0.0104g	0.054%	0.1912g	3.05
3	14:14:12	0.0128g	0.089%	0.1437g	2.67
4	14:19:03	0.0138g	0.074%	0.1854g	3.04
5	14:27:55	0.0102g	0.060%	0.1690g	2.98

6	14:41:44	0.0127g	0.082%	0.1545g	2.80
7	14:46:06	0.0090g	0.061%	0.1482g	2.73
8	14:53:38	0.0133g	0.086%	0.1548g	2.81
9	15:31:14	0.0088g	0.047%	0.1853g	3.04
10	15:35:35	0.0146g	0.075%	0.1934g	3.06

b) Pengujian pada Dinding 2

Tabel 5 Lampiran 9 Kalibrasi Pengujian Dinding 2 Node Bawah

No.	Time	Offset ax	Offset ay	Offset az	Offset gx	Offset gy	Offset gz
1	15:30:22	-2824	3834	2954	120	-5	-16

Tabel 6 Lampiran 9 Kalibrasi Pengujian Dinding 2 Node Atas

No.	Time	Offset ax	Offset ay	Offset az	Offset gx	Offset gy	Offset gz
1	15:30:53	390	-1560	3512	86	312	57

Tabel 7 Lampiran 9 Hasil Pengujian Dinding 2 (1)

No.	Time	Wall Height	Peak Bottom Inclination Angle	Peak Top Inclination Angle	Peak Displacement
1	15:35:03	268 cm	0.042°	0.041°	0.20 cm
2	15:36:54	268 cm	0.030°	0.026°	0.14 cm
3	15:38:49	268 cm	0.032°	0.040°	0.15 cm
4	15:40:28	268 cm	0.033°	0.027°	0.16 cm
5	15:42:22	268 cm	0.034°	0.034°	0.16 cm
6	15:44:22	268 cm	0.028°	0.024°	0.13 cm
7	15:46:42	268 cm	0.036°	0.044°	0.17 cm
8	15:49:27	268 cm	0.026°	0.020°	0.12 cm
9	15:51:03	268 cm	0.027°	0.025°	0.13 cm
10	15:53:13	268 cm	0.036°	0.044°	0.17 cm
11	15:56:14	268 cm	0.037°	0.034°	0.17 cm

Tabel 8 Lampiran 9 Hasil Pengujian Dinding 2 (2)

No.	Time	PGA	Peak Drift Ratio	PGA _{DR1}	Score
1	15:35:03	0.0142g	0.074%	0.1923g	3.05
2	15:36:54	0.0088g	0.052%	0.1688g	2.98
3	15:38:49	0.0092g	0.055%	0.1670g	2.96
4	15:40:28	0.0118g	0.058%	0.2030g	3.08
5	15:42:22	0.0118g	0.060%	0.1959g	3.06
6	15:44:22	0.0080g	0.049%	0.1623g	2.90
7	15:46:42	0.0091g	0.063%	0.1452g	2.69

8	15:49:27	0.0086g	0.045%	0.1918g	3.05
9	15:51:03	0.0089g	0.047%	0.1897g	3.05
10	15:53:13	0.0123g	0.063%	0.1955g	3.06
11	15:56:14	0.0128g	0.065%	0.1967g	3.07

c) Pengujian pada Dinding 3

Tabel 9 Lampiran 9 Kalibrasi Pengujian Dinding 3 Node Bawah

No.	Time	Offset ax	Offset ay	Offset az	Offset gx	Offset gy	Offset gz
1	15:39:06	-2848	3832	2930	119	-5	-16
2	16:03:53	-2848	3834	2930	119	-6	-16

Tabel 10 lampiran 9 Kalibrasi Pengujian Dinding 3 Node Atas

No.	Time	Offset ax	Offset ay	Offset az	Offset gx	Offset gy	Offset gz
1	15:39:10	436	-1562	3478	86	311	58
2	16:03:58	436	-1562	3478	86	312	58

Tabel 11 Lampiran 9 Hasil Pengujian Dinding 3 (1)

No.	Time	Wall Height	Peak Bottom Inclination Angle	Peak Top Inclination Angle	Peak Displacement
1	15:41:42	248 cm	0.028°	0.034°	0.12 cm
2	15:43:03	248 cm	0.029°	0.033°	0.13 cm
3	15:46:53	248 cm	0.038°	0.044°	0.17 cm
4	15:49:14	248 cm	0.037°	0.037°	0.16 cm
5	15:54:00	248 cm	0.036°	0.030°	0.15 cm
6	16:06:12	248 cm	0.033°	0.039°	0.14 cm
7	16:08:13	248 cm	0.040°	0.033°	0.17 cm
8	16:11:26	248 cm	0.041°	0.035°	0.18 cm
9	16:14:27	248 cm	0.030°	0.030°	0.13 cm
10	16:18:21	248 cm	0.034°	0.042°	0.15 cm
11	16:23:20	248 cm	0.035°	0.041°	0.15 cm
12	16:26:40	248 cm	0.031°	0.038°	0.13 cm

Tabel 12 Lampiran 9 Hasil Pengujian Dinding 3 (2)

No.	Time	PGA	Peak Drift Ratio	PGA _{DR1}	Score
1	15:41:42	0.0067g	0.048%	0.1406g	2.63
2	15:43:03	0.0076g	0.051%	0.1498g	2.75
3	15:46:53	0.0089g	0.067%	0.1324g	2.53
4	15:49:14	0.0094g	0.064%	0.1468g	2.71
5	15:54:00	0.0095g	0.062%	0.1536g	2.79

6	16:06:12	0.0091g	0.058%	0.1569g	2.83
7	16:08:13	0.0112g	0.069%	0.1624g	2.90
8	16:11:26	0.0110g	0.071%	0.1544g	2.80
9	16:14:27	0.0074g	0.052%	0.1421g	2.65
10	16:18:21	0.0090g	0.060%	0.1505g	2.75
11	16:23:20	0.0072g	0.061%	0.1182g	2.35
12	16:26:40	0.0065g	0.054%	0.1208g	2.39