BAB I

PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi sensor dan sistem pengukuran telah berkembang pesat dalam beberapa tahun terakhir. Sensor-sensor ini digunakan dalam berbagai aplikasi, seperti pengukuran suhu, kelembaban, tekanan, dan lain-lain. Namun, sensor-sensor ini seringkali mengalami gangguan atau *noise* yang dapat mempengaruhi akurasi pengukuran.

Salah satu masalah yang sering dihadapi dalam penggunaan sensor adalah gangguan atau *noise* yang dapat mempengaruhi akurasi pengukuran. Gangguan ini dapat berasal dari berbagai sumber, seperti interferensi elektromagnetik, perubahan suhu, atau getaran mekanik.

Menurut sebuah studi yang dilakukan oleh International Society of Automation (ISA), lebih dari 50% kesalahan dalam pengendalian proses industri disebabkan oleh pembacaan sensor yang tidak akurat (ISA, 2019). Selain itu, sebuah studi lain yang dilakukan oleh National Institute of Standards and Technology (NIST) menemukan bahwa pembacaan sensor yang tidak stabil dapat menyebabkan kerugian ekonomi yang signifikan (NIST, 2017).

Oleh karena itu, diperlukan suatu metode untuk meningkatkan kualitas dan stabilitas pembacaan sensor. Salah satu metode yang dapat digunakan adalah dengan merancang rangkaian filter adaptif yang dapat menghilangkan gangguan dan meningkatkan akurasi pembacaan sensor.

B. Rumusan Masalah

Berdasarkan uraian yang dijelaskan pada latar belakang, mendapatkan rumusan masalah sebagai berikut:

1. Bagaimana merancang rangkaian filter adaptif yang mampu meningkatkan kualitas dan stabilitas sensor ?
2. Bagaimana pengaruh parameter adaptasi filter terhadap performa sistem pengukuran sensor ?
3. Bagaimana implementasi filter adaptif dalam sistem berbasis sensor LM35 dan Load Cell untuk meningkatkan stabilitas dan keandalan pengukuran ?

C. Tujuan Penelitaan

1. Merancang rangkaian filter adaptif yang dapat menghilangkan gangguan dan meningkatkan akurasi pembacaan sensor.
2. Meningkatkan kualitas dan stabilitas pembacaan sensor dengan menggunakan rangkaian filter adaptif yang dirancang.
3. Mengimplementasikan filter adaptif pada sistem berbasis sensor LM35 dan Load Cell untuk menguji peningkatan stabilitas.

D. Batasan masalah

Adapun batasan masalah yang akan dibahas, maka ada beberapa hal sebagai berikut:

1. Penelitian ini hanya fokus pada perancangan filter adaptif untuk sensor suhu LM35 dan Load Cell, tidak membahas sensor lainnya.
2. Rangkaian filter adaptif yang dirancang hanya dapat digunakan dalam rentang suhu tertentu

3. Penelitian ini hanya fokus pada perancangan rangkaian filter adaptif dan tidak membahas tentang implementasi rangkaian filter adaptif pada sistem yang lebih besar.

E. Manfaat Penelitian

Manfaat yang ingin dicapai pada peneletian ini ialah:

1. Meningkatkan kualitas dan stabilitas pembacaan sensor dengan menggunakan rangkaian filter adaptif yang dirancang.
2. Mengurangi gangguan dan noise pada sinyal pembacaan sensor, sehingga meningkatkan akurasi dan reliabilitas pembacaan sensor
3. Mengurangi gangguan dan noise pada sinyal pembacaan sensor, sehingga meningkatkan akurasi dan reliabilitas pembacaan sensor.

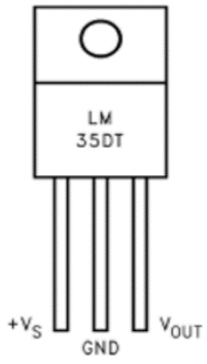
BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Sensor LM35 dan Kualitas sinyal

Sensor LM35 adalah sensor suhu yang banyak digunakan dalam sistem elektronik karena keakuratannya yang tinggi dan kemampuannya menghasilkan output linier terhadap perubahan suhu. Sensor ini beroperasi dengan tegangan DC dan memiliki sensitivitas sebesar 10mV/°C. Namun, dalam aplikasi praktis, sinyal dari sensor LM35 dapat terpengaruh oleh *noise* yang berasal dari interferensi elektromagnetik, variasi tegangan sumber, dan gangguan lingkungan. Oleh karena itu, diperlukan teknik pemrosesan sinyal seperti penggunaan filter untuk meningkatkan kualitas data yang diperoleh. (Sumber: Texas Instruments, 2020).



Gambar 2. 1 Sensor LM35

Kelebihan Sensor LM35

a. Rentang suhu cukup jauh, yaitu antara -55 sampai + 150 ºC

b. Low self-heating yang sangat rendah, sebesar 0,08 ºC

c. Beroprasi pada pada rentang tegangan 4 sampai 30 V

d. Rangkaian sangat sederhana.

e. Harga relatif murah.

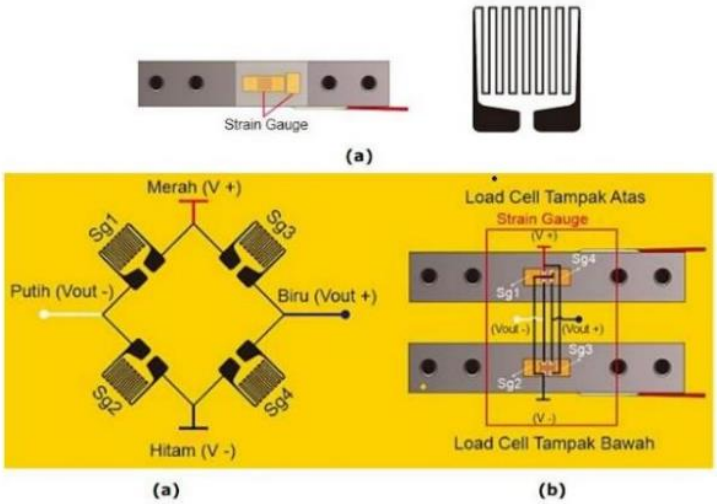
Tabel 2. 1 Spesifikasi Sensor LM35

Parameter	Spesifikasi
Tipe sensor	Sensor suhu analog
Tegangan oprasi	4 V – 30 V DC
Arus oprasi	< 60 µA
Rentang suhu oprasi	0℃- 100℃ (LM35DZ) hingga 150℃ LM35)
Akurasi	± 0,5℃ pada 25℃
Resolusi output	10 mV/℃
Tipe output	Tegangan analog linier terhadap suhu
Waktu respon	~ 100 ms (tergantung kondisi termal)
Bentuk fisik	TO-92,SO-8,TO-220
Kalibrasi	Internal,tidak perlu eksternal
Konektivitas	VCC= tegangan masuk, Output=keluaran, GND=*ground*

2. Sensor Load Cell

Load cell adalah sensor yang digunakan untuk mengukur gaya atau berat berdasarkan perubahan resistansi yang dihasilkan oleh tekanan yang diberikan. Load cell umumnya bekerja dengan prinsip *strain gauge* yang menghasilkan sinyal listrik kecil yang perlu diperkuat sebelum digunakan. Sama seperti sensor LM35, load cell juga rentan terhadap *noise* akibat faktor lingkungan, variasi tegangan suplai, dan interferensi elektromagnetik. Oleh karena itu, penggunaan filter adaptif

dapat membantu meningkatkan stabilitas dan akurasi pengukuran load cell. (Sumber: Natarajan, 2018)



Gambar 2. 2 Sensor loadcell

Tabel 2. 2 Spesifikasi load cell

Parameter	Spesifikasi umum
Tipe	Strain gauge (tipe paling umum)
Tegangan kerja	5V – 15V DC (exitacion voltage)
Output tegangan	~1mV/V hingga 3mV/V(tergantung beban)
Kapasitas beban	1kg – beberapa ton (tergantung model)
Akurasi	0,02%-0,1%
Non – linearity	<0,03% FS
Histeresis	<0,02% FS
Material	Aluminium, baja tahan karat
Bentuk hokum	Bar, S-Type,*Pancake*,*canister*
Sinyal outpt	Diferensial analog (perlu amplifier seperti HX711)
Sensivitas	Sekitar 2mV/V umum
Suhu oprasi	-10℃ – 50℃(tergantung spesifikasi

3. Modul HX711

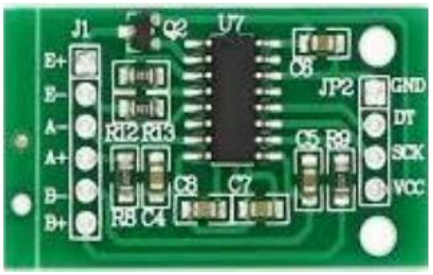
HX711 adalah modul penguat dan konverter ADC (*Analog to Digital Converter*) yang digunakan untuk mengolah sinyal dari sensor load cell. Modul ini memiliki kemampuan untuk memperkuat sinyal yang sangat kecil sehingga dapat dibaca oleh mikrokontroler seperti Arduino. Penggunaan HX711 memungkinkan akurasi yang lebih baik dalam pembacaan data berat dari load cell. Namun, sinyal dari modul ini masih dapat dipengaruhi oleh *noise* eksternal, sehingga penggunaan filter adaptif menjadi penting untuk meningkatkan keakuratan pengukuran. (Sumber: Avtar & Singh, 2020).

Komponen terintegrasi dari HX711 presisi 24-bit *analog to digital conventer* (ADC) yang didesain untuk sensor timbangan digital dal industrial control aplikasi yang terkoneksi sensor jembatan. Struktur yang sederhana, mudah dalam penggunaan, hasil yang stabil dan reliable, memiliki sensitivitas tinggi, dan mampu mengukur perubahan dengan cepat. Spesifikasinya dapat dilihat pada Tabel 2.3 Spesifikasi HX711.(Sumber:UNIKOM Rizzywan Maolana Sunarya,2011)

Tabel 2. 3 Spesifikasi modul HX711

Spesifikasi	Keterangan
Tegangan input diferensial	±40mV
Akurasi data	24 bit (chip konverter A/d 24)
Frekuensi	80Hz
Tegangan Pengoprasian	5V DC
Oprasi saat ini	<10 mA
Ukuran	38mm * 21mm * 10mm

Bentuk fisik dari modul HX711 dapat ditunjukkan pada Gambar 2.3 Modul HX711



Gambar 2. 3 Modul HX711

4. Mikrokontroler

Mikrokontroller adalah sistem *mikroprosesor* lengkap yang terkandung di dalam sebuah chip. *Mikrokontroler* berbeda dari *mikroprosesor* serba guna yang digunakan dalam sebuah *PC*, karena didalam sebuah *mikrokontroler* umumnya juga telah berisi komponen pendukung sistem minimal *mikroprosesor*, yakni memori dan antarmuka, sedangkan didalam *mikroprosesor* umumnya hanya berisi *CPU* saja. (Wikipedia, 2015).

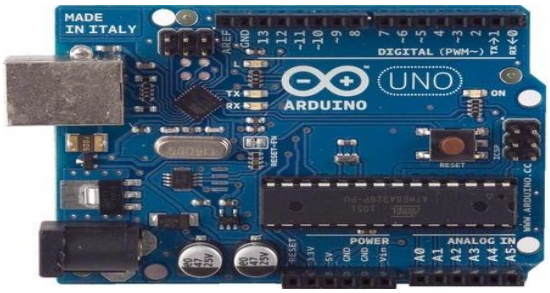
Adapun *mikrokontroler* yang dipakai peneliti adalah *mikrokontroler* jenis Arduino Uno sebagai pengontrol eletronik untuk membaca dan menulis data untuk tersambung ke komputer.

Arduino uno merupakan sebuah board *mikrokontroller* yang mana didalamnya terdapat *mikrokontroller ATmega328* yang memiliki 14 pin digital (6 pin dapat digunakan sebagai output PWM), 6 input analog sebuah header ICSP dan sebuah tombol reset yang berfungsi untuk mengulang program. Untuk mendukung *mikrokontroller* agar dapat digunakan, hanya dengan menghubungkan board arduino ke komputer dengan menggunakan kabel USB atau listrik dengan AC ke adaptor DC. Arduino adalah sebuah platform komputasi fisik *open source*

berbasiskan Rangkaian *input/output* sederhana dan lingkungan pengembangan yang mengimplementasikan bahasa *Processing*. (Aulia et al., 2021).

Setiap 14 pin digital pada arduino uno dapat di gunakan sebagai input dan output, menggunakan pin mode.digital writer, dan digital read. Pada fungsi tersebut beroperasi di tegangan 5V, setiap pin dapat memberikan atau menerima suatu arus maksimum 40 mA dan mempunyai sebuah resistor *pull-up* (terputus secara *default*) 20-50 Kohm.

Secara umum, arduino dengan sebuah *mikrokontroler* ini mampu menciptakan suatu program yang dapat mengendalikan berbagai komponen elektronika. Sehingga cukup jelas, bila fungsi yang dimiliki arduino uno adalah untuk memudahkan pengguna dalam melakukan *prototyping* dan memprogram *mikrokontroller* serta menciptakan berbagai macam alat canggih berbasis *mikrokontroller*. (Arga, 2020)



Gambar 2. 4 Mikrokontroler Arduino Uno

Arduino uno dan ekosistemnya punya kelebihan sebagai berikut:

a. Di dukung oleh arduiouno IDE dan bahasa pemrograman yang sudah lengkap librarinya.

b. Terdapat modul yang siap pakai dan dapat langsung dipasang pada board Arduiuno.

c. Dukungan dokumentasi yang bagus dan memiliki komunikasi yang solid.

Spesifikasi arduino uno:

Tabel 2. 4 spesifikasi Arduino Uno

Data Teknis Arduino Uno	
Microkontroler	ATMega 328P
Tegangan operasi	5V
Tegangan *input*	7-12 V
Pin digital I/O/1	14 (6 diantaranya PWM output)
Pin Analog input	6 pin
Arus DC per pin I/O	40 Ma
Arus/1DC untuk pin 3,3 V	50 Ma
Flash memory	32 KB (ATmega328p)
SRAM	2 KB
EEPROM	1 KB
Clock speed	16 MHz
Panjang	68.6 mm
Lebar	53.44 mm
Berat	25 g

Beberapa power pin pada arduino adalah sebagai berikut:

a. VIN:Tegangan input pada *board* arduino uno ketika menggunakan sumber daya (5V dari sambungan USB atau dari sumber regulator lain) anda dapat men suplay tegangan pada pin ini, dan jika suplai tegangan lewat *power jack,* dapat diakses melalui pin ini atau langsung ke jack power 5V.

b. 5 V;Catu daya yang digunakan untuk power *mikrokontroller* dan beberapa komponen lainnya telah diatur tegangan sebesar 5 V dari regulator pada board.dapat disuplai melalui DC jack power (7-12 Volt).
c. 3V3:Tegangan sebesar 3,3 Volt diperoleh dari FTDI chip yang ada pada board. Arus maksimum sebesar 50 mA.
d. Pin ground:GND /pin ground berfungsi sebagai lintasan ground pada Arduino
e. Memory:ATmega328 memiliki 32 KB memori *flash* Dengan 0,5 digunakan sebagai *bootloader*, dan juga mempunyai 2 KB SRAM dan memori EEPROM sebesar 1 KB.
f. *Input* dan *output*:setiap pin/1pada arduino uno dapat digunakan sebagai input maupun/*1output* yang menggunakan funsgsi *pin mode, digital write, dan digital read,* pada pin tersebut dapat beroperasi dengan tegangan 5V. Setiap pin juga mampu memberikan atau menerima arus maksimal hingga 40 mA dan resistor internal *pull-up* antara 20-50 Kohm. Selain itu, terdapat beberapa fungsi khusus dari pin tersebut:
    1. Serial : 0 (RX) dan 1 (TX) berfungsi untuk menerima dan mengirimkan data serial. Pin ini terhubung pada pin yang sesuai dari USB ke TTL chip
    2. Interupsi eksternal : 2 dan 3 pada pin ini difungsikan untuk memicu interupsi pada nilai yang rendah, naik turun, ataupun perubahan nilai.
    3. PWM :2,5,6,9,10 dan 11 menyediakan *output* 8 bit dengan fungsi *analog write*.
    4. SPI : pin 10 (SS), 11 (MISO), dan 13 (SCK).pin ini mendukung komunikasi SPI dengan menggunakan *library* SPI

5. LED : pin 13 terdapat LED internal yang dihubungkan ke pin 13 pada board. Ketika pin berbilai tinggi maka LED akan menyala(*ON*), dan begitu pula sebaliknya.
6. Pada arduino uno memiliki pin analog dengan 6 *input* yang berlabel A0 sampai A5, dimana maing-masing menyediakan 10 bit dengan resolusi 1024. Secara default, 5 volt dari *ground*.

6. Filter Digital

Filter digital adalah suatu sistem yang digunakan untuk memproses sinyal digital dan menghilangkan komponen yang tidak diinginkan, seperti *noise* atau gangguan. Filter digital dapat digunakan untuk berbagai aplikasi, seperti pengolahan sinyal audio, pengolahan sinyal gambar, dan pengolahan sinyal lainnya

a. Filter Rata-Rata (*Moving Average Filter*)

Filter ini menghitung rata-rata nilai sinyal pada beberapa sampel terakhir dan menghasilkan *output* yang merupakan rata-rata nilai sinyal tersebut.

Rumus :

$$y[n] = (1/N) * (x[n] + x[n-1] + ... + x[n-N+1]) \text{ …….}$$

di mana,

y[n] : *output* filter,

x[n] : *input* filter,

N : jumlah sampel yang digunakan untuk menghitung rata-rata.

b. Filter Exponential Smoothing (ES)

Filter ini menggunakan konstanta yang menentukan seberapa besar pengaruh input terhadap *output*, dan menghasilkan output yang merupakan kombinasi antara *input* dan *output* sebelumnya.

Rumus :

y[n] = alpha * x[n] + (1-alpha) * y[n-1] ……

di mana,

y[n] : *output* filter,

x[n] : *input* filter,

alpha : konstanta yang menentukan seberapa besar pengaruh input terhadap output.

c. Filter Finite Impulse Response (FIR)

Filter ini memiliki respon *impuls* yang terbatas, dan menghasilkan *output* yang merupakan kombinasi antara input dan output sebelumnya

Rumus :

y[n] = b0 * x[n] + b1 * x[n-1] + ... + bN * x[n-N] ……..

di mana,

y[n] : *output* filter,

x[n] : *input* filter,

b0, b1, ..., bN : koefisien filter.

d. Filter Infinite Impulse Response (IIR)

Filter ini memiliki respon *impuls* yang tidak terbatas, dan menghasilkan *output* yang merupakan kombinasi antara *input* dan *output* sebelumnya.

Rumus :

y[n] = b0 * x[n] + b1 * x[n-1] + ... + bN * x[n-N] + a1 * y[n-1] + ... + aN * y[n-N]

dimana :

y[n] : *output* filter,

x[n] : *input* filter,

b0, b1, ..., bN : koefisien filter,

a1, a2, ...,aN :koefisien filter yang menentukan seberapa besar pengaruh output sebelumnya terhadap *output* saat ini.

e. Filter Butterworth

Filter ini memiliki respon frekuensi yang datar dan menghasilkan *output* yang merupakan kombinasi antara *input* dan *output* sebelumnya.

Rumus :

y[n] = (b0 + b1 * z^-1 + b2 * z^-2 + ... + bN * z^-N) / (1 + a1 * z^-1 + a2 * z^-2 + ... + aN * z^-N) ……

di mana,

y[n] : *output* filter,

x[n] : *input* filter,

b0, b1, ..., bN : koefisien filter,

a1, a2, ...,aN : koefisien filter yang menentukan seberapa besar pengaruh *output* sebelumnya terhadap *output* saat ini, z adalah variabel kompleks yang menentukan frekuensi filter.

f. Filter Chebyshev

Filter ini memiliki respon frekuensi yang tidak datar dan menghasilkan output yang merupakan kombinasi antara *input* dan *output* sebelumnya.

Rumus :

y[n] = (b0 + b1 * z^-1 + b2 * z^-2 + ... + bN * z^-N) / (1 + a1 * z^-1 + a2 * z^-2 + ... + aN * z^-N) ……

di mana,

y[n] : *output* filter,

x[n] : *input* filter,

b0, b1, ..., b : koefisien filter,

a1, a2, ..., aN : koefisien filter yang menentukan seberapa besar pengaruh *output* sebelumnya terhadap *output* saat ini, z adalah variabel kompleks yang menentukan frekuensi filter.

7. Komponen Filter Digital

1. Input : Sinyal digital yang akan diproses oleh filter
2. Koefisien Filter : Nilai yang menentukan seberapa pengaruh *input* terhadap *output*
3. *Output* : Sinyal digital yang dihasilkan oleh filter.
4. *Delay* : waktu yang dibutuhkan oleh filter untuk menghasilkan *output*.

8. Kelebihan Filter Digital

1. Menghilangkan *Noise* : Filter digital dapat menghilangkan noise atau gangguan yang tidak diinginkan dari sinyal digital.
2. Menghasilkan sinyal yang stabil : Filter digital dapat menghasilkan sinyal yang stabil dan tidak berubah-ubah.

3. Mengurangi Distorsi : Filter digital dapat mengurangi distorsi yang terjadi pada sinyal digital.

9. Kekurangan Filter Digital

1. Mengurangi Kualitas Sinyal : Filter digital dapat mengurangi kualiatas sinyal digital jika tidak dirancang dengan benar.
2. Menghasilkan *Delay* : Filter digital dapat menghasilkan *delay* yang tidak diinginkan pada sinyal digital.
3. Mengurangi Frekuensi : filter digital dapat mengurangi fekuensi sinyal digital jika tidak dirancang dengan benar

Dalam keseluruhan, filter digital adalah suatu sistem yang sangat berguna untuk memproses sinyal digital dan menghilangkan sinyal yang tidak diinginkan.Namu, perlu dirancang dengan benar untuk menghasilkan *output* yang diinginkan.

B. Kajian Penelitian Terdahulu

Kajian terhadap penelitian terdahulu bertujuan untuk memahami perkembangan teknologi filter adaptif dalam peningkatan kualitas sinyal sensor. Pada penelitian kali ini, peneliti mendapat *referensi* dari beberapa peneliti yang terdahulu yang mengunakan sistem yang hampir sama.Berikut beberapa jurnal penelitian yang relevan.

Penelitian oleh Smith et al. (2018): Penelitian ini membahas implementasi filter adaptif berbasis algoritma LMS untuk mengurangi *noise* pada sensor suhu dalam sistem kendali industri. Hasil penelitian menunjukkan

peningkatan akurasi pembacaan sensor hingga 30% dibandingkan metode filter konvensional.

Penelitian oleh Zhang, L., & Lee (2020): Studi ini mengembangkan filter adaptif berbasis RLS untuk meningkatkan stabilitas sinyal pada sensor tekanan dalam aplikasi medis. Dengan metode ini, sinyal yang dihasilkan lebih stabil meskipun dalam kondisi lingkungan yang berubah-ubah.

Penelitian ini menerapkan filter berbasis jaringan saraf tiruan (ANN) pada sensor load cell untuk meningkatkan akurasi pengukuran berat. Hasilnya menunjukkan bahwa kombinasi filter adaptif dengan ANN memberikan kinerja yang lebih baik dibandingkan dengan filter digital konvensional.

Dalam penelitian ini, filter Kalman digunakan untuk menyaring noise pada sensor LM35 dalam sistem monitoring suhu berbasis IoT. Studi ini membuktikan bahwa metode tersebut mampu mengurangi fluktuasi sinyal hingga 40%.

Dari kajian ini, dapat disimpulkan bahwa penggunaan filter adaptif, telah menunjukkan hasil yang positif dalam peningkatan kualitas sinyal sensor. Penelitian ini akan mengadopsi metode serupa dengan fokus pada sensor LM35 dan load cell dalam berbagai kondisi operasional.

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Penelitian ini merupakan penelitian eksperimen yang bertujuan untuk mengembangkan dan menguji efektivitas filter adaptif dalam meningkatkan kualitas sinyal sensor. Metode eksperimen dilakukan melalui serangkaian pengujian pada sistem sensor LM35 dan load cell dengan berbagai tingkat *noise* untuk mengevaluasi performa filter adaptif yang diterapkan.

B. Lokasi Dan Waktu

Penelitian ini akan dilaksanakan di laboratorium Teknik Elektro Universitas Muhammadiayah Parepare**,** yang memiliki fasilitas untuk pengujian sensor dan pemrosesan sinyal.

Waktu pelaksanaan penelitian direncanakan selama 1 bulan

Tabel 3. 1 Jadwal Pelaksanaan Penelitian

No	Kegiatan	Waktu kegiatan			
		Bulan ke-1			
		1	2	3	4
1	Studi literature				
2	Perancangan perangkat keras				
3	Perancangan perangkat lunak				
4	Realisasi perangkat keras				
5	Realisasi perangkat lunak				

No	Kegiatan	Waktu kegiatan			
		Bulan ke-1			
		1	2	3	4
6	Integrasi perangkat keras dan perangkat lunak				
7	Pengujian				
8	Analisis hasil pengujian				
9	Pembuatan laporan hasil penelitian				

C. Alat dan Bahan

Adapun alat dan bahan yang digunakan dalam melaksanakan penelitian ini adalah sebagai berikut:

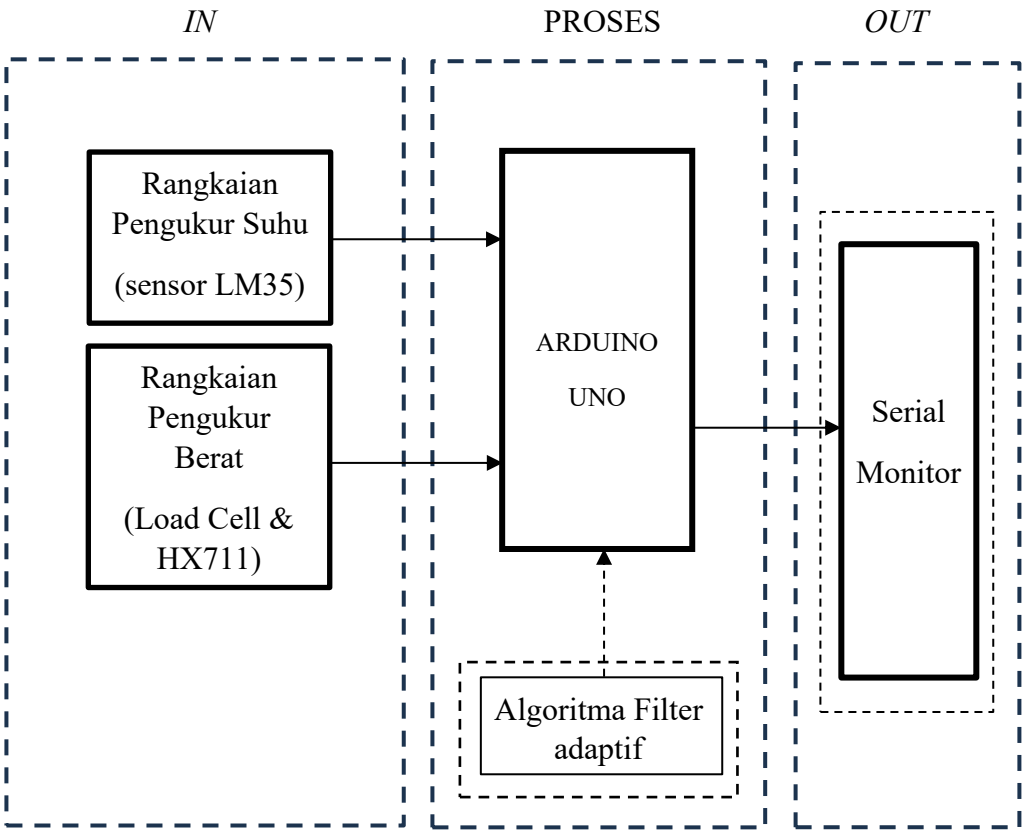
Tabel 3. 2 Alat dan Bahan

No.	Alat dan Bahan	Jumlah
1.	Sensor LM35	1
2.	Load Cell	1
3.	Modul HX711	1
4.	Mikrokontroller Arduiuno UNO	1
5.	Kabel jumper	-
6.	Arduiuno IDE	1

D. Rancangan Penelitian

Dalam pembuatan sistem ini perlu adanya diagram rancangan penelitian yang dimaksud agar lebih mudah dipahami dan mudah dilihat berdasarkan urutan

.



Gambar 3.1 Diagram Perencanaan

Berdasarkan diagram perencanaan diatas, cara kerja Sistem ini dirancang untuk meningkatkan kualitas pembacaan data dari sensor suhu dan beban dengan menggunakan filter adaptif. Pertama, sensor LM35 digunakan untuk mengukur suhu, menghasilkan sinyal analog yang proporsional terhadap suhu dalam satuan derajat Celsius. Di sisi lain, Load Cell digunakan untuk mengukur beban. Sinyal dari Load Cell sangat kecil (*mikrovolt*) dan masih dalam bentuk analog, sehingga perlu diperkuat dan dikonversi menjadi sinyal digital.

Proses penguatan dan konversi sinyal dilakukan oleh modul HX711. Modul ini berfungsi sebagai penguat sinyal sekaligus konverter analog ke digital dengan resolusi tinggi (24-bit). Hasil dari HX711 kemudian dikirim ke mikrokontroler

Arduino UNO, sementara sinyal dari LM35 dibaca langsung melalui ADC internal Arduino.

Selanjutnya, data dari kedua sensor diproses menggunakan algoritma filter adaptif, algoritma ini bertujuan untuk mengurangi *noise* yang mungkin timbul akibat gangguan lingkungan atau fluktuasi sinyal dari sensor. Dengan filter adaptif, sistem mampu menyaring data sehingga lebih stabil dan akurat.

Yang membedakan sistem ini adalah kemampuannya menyesuaikan parameter filter secara otomatis. Sistem terus memantau perubahan karakteristik *noise* dan melakukan penyesuaian agar filter tetap optimal. Ini menciptakan suatu mekanisme umpan balik (*closed-loop*) yang adaptif terhadap perubahan lingkungan.

Akhirnya, data yang telah difilter ditampilkan dalam bentuk output melalui tampilan serial monitor ardiuuno IDE. Karena telah melalui proses penyaringan adaptif, data ini menjadi lebih bersih dan dapat diandalkan untuk keperluan analisis atau kendali lebih lanjut. Dengan demikian, sistem bekerja secara otomatis dan cerdas dalam menyaring informasi dari lingkungan sekitar.

E. Teknik Pengumpulan Data

Data dikumpulkan dari hasil pembacaan sensor secara langsung dalam kondisi dengan dan tanpa filter.Setiap eksperimen akan dijalankan beberapa kali untuk memperoleh data yang konsisten.Data akan direkam dalam bentuk waktunyata (*real-time*) menggunakan *software* Arduiuno IDE.

F. Teknik Analisis Data

Data yang diperoleh akan dianalisis secara kuantitatif menggunakan metode statistik untuk mengevaluasi efektivitas filter adaptif dalam meningkatkan kualitas dan stabilitas pembacaan sensor.

BAB IV

HASIL DAN PEMBAHASAN

A. Percancangan Perangkat keras (*hardware)*

Perancangan alat keras merupakan tahap penting dalam penelitian ini karena sistem yang dirancang harus mampu membaca data dari sensor secara akurat, memprosesnya dengan algoritma filter adaptif, serta menampilkan hasil pembacaan secara *real-time*. Sistem dirancang dalam bentuk *prototipe* berbasis mikrokontroler Arduino Uno yang mengintegrasikan dua jenis sensor, yaitu sensor suhu LM35 dan sensor berat Load Cell yang dipasangkan dengan modul HX711 sebagai penguat dan konverter ADC.

Salah satu pertimbangan utama dalam perancangan perangkat keras ini adalah kestabilan sumber tegangan, karena fluktuasi daya dapat memengaruhi akurasi pembacaan sensor, terutama untuk sinyal analog dari LM35 dan sinyal *mikrovolt* dari Load Cell. Oleh karena itu, digunakan catu daya DC stabil 5V dari adaptor atau melalui USB komputer.

Sinyal dari sensor LM35 dihubungkan ke salah satu pin analog Arduino (A0), sedangkan Load Cell dikoneksikan ke modul HX711 yang kemudian dihubungkan ke dua pin digital Arduino (D2 untuk data dan D3 untuk clock).Selain pembacaan dan pemrosesan data, sistem ini juga dirancang untuk memiliki kemampuan menampilkan hasil melalui *Serial Monitor* dan *Serial plotter* Pada Arduiuno IDE. Hal ini bertujuan agar sistem dapat digunakan secara mandiri tanpa tergantung pada komputer. Untuk debugging dan pengujian lanjutan, Arduino juga

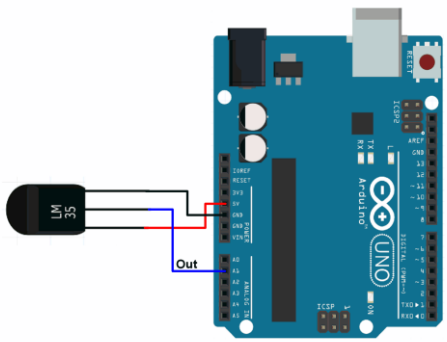
dikoneksikan ke *software serial monitor* melalui antarmuka USB. Hasil pengukuran sebelum dan sesudah proses filter juga dapat dicatat untuk analisis lebih lanjut dalam perangkat lunak seperti Excel.

Dengan rancangan ini, alat keras tidak hanya berfungsi sebagai pembaca sinyal dari sensor, tetapi juga sebagai platform eksperimen untuk pengujian algoritma filter adaptif dengan metode *Eksponential Smoothing* secara langsung dalam lingkungan fisik yang menyerupai kondisi operasional nyata.

Adapun uraian masing-masing perangkat keras yang digunakan sebagai berikut :

1. Rangkaian Pengukur Suhu

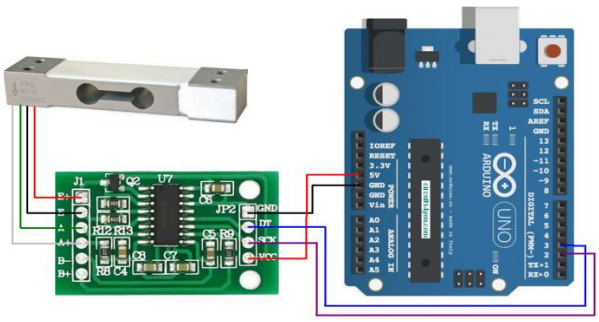
Untuk melakukan koneksi antara LM35 dan Arduino, digunakan skema sambungan sebagai berikut: pin 1 (VCC) dari LM35 dihubungkan ke pin 5V Arduino, pin 2 (VOUT) dihubungkan ke pin A0 Arduino sebagai input analog, dan pin 3 (GND) dihubungkan ke GND Arduino.



Gambar 4. 1 *Skema rangkaian LM35 ke Arduiuno*

2. Rangkaian Pengukur Beban

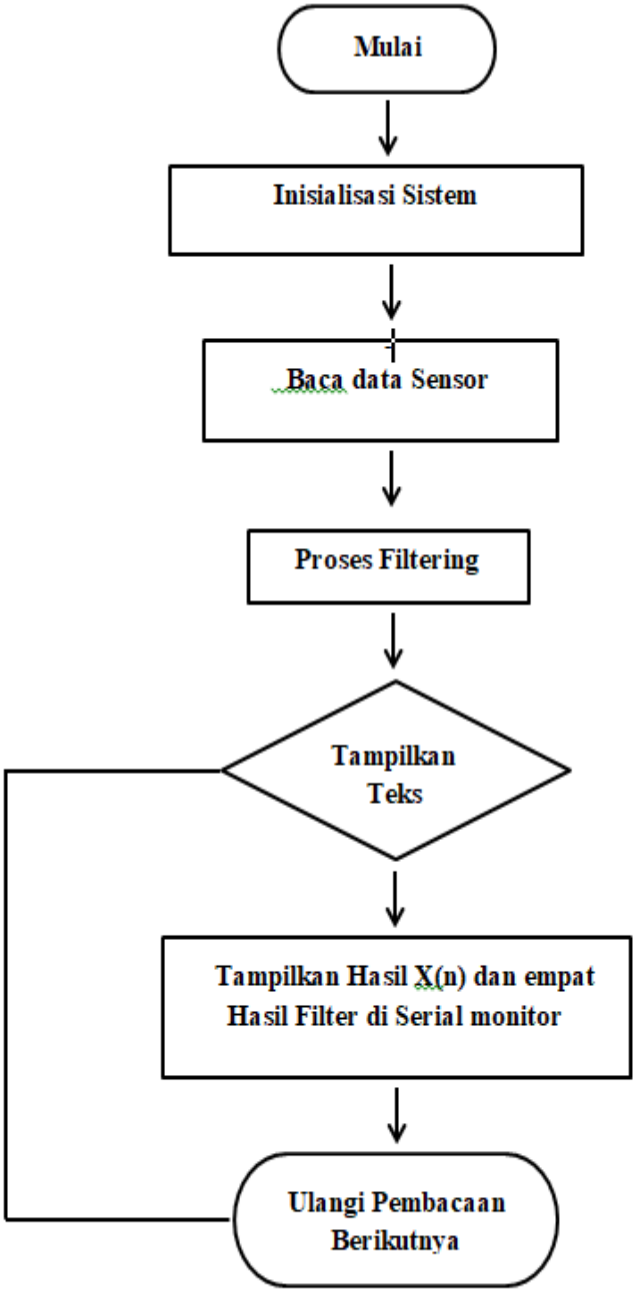
Pada konfigurasi ini, sensor Load Cell dihubungkan terlebih dahulu ke modul HX711, yang berfungsi sebagai penguat sinyal dan konverter analog-ke-digital. Kabel merah dari Load Cell, yang merupakan pin E+, disambungkan ke pin VCC pada modul HX711 sebagai sumber tegangan positif untuk strain gauge. Kabel hitam, yaitu pin E−, disambungkan ke GND HX711 sebagai ground. Kemudian, kabel hijau sebagai A+ dihubungkan ke pin A+ pada HX711, dan kabel putih sebagai A− dihubungkan ke pin A− pada modul HX711.Selanjutnya, dari sisi modul HX711 ke Arduino Uno, pin VCC pada HX711 disambungkan ke pin 5V pada Arduino untuk catu daya, dan pin GND dihubungkan ke GND Arduino. Untuk komunikasi data, pin DT (data) pada HX711 dihubungkan ke pin digital D3 Arduino, sedangkan pin SCK (clock) pada HX711 dihubungkan ke pin digital D2 Arduino. Dengan alur pengkabelan ini, Arduino dapat membaca sinyal dari Load Cell melalui HX711 dengan akurasi yang tinggi.



Gambar 4. 2 *Skema Rangkaian Load Cell ke Arduiuno*

B. Perancangan Perangkat Lunak (*software)*

1. Alur Pemrograman (*Flowchart)*



Alur kerja sistem filter adaptif untuk stabilisasi pembacaan sensor dimulai dengan **inisialisasi sistem**, yaitu mengaktifkan sensor dan melakukan pengaturan awal pada perangkat keras serta perangkat lunak. Pada tahap ini, nilai-nilai variabel seperti pembacaan awal sensor, konstanta filter (*alpha1, alpha2, alpha3, alpha4*), serta variabel pendukung untuk perhitungan *Exponential Smoothing* diatur.

Setelah inisialisasi selesai, sistem masuk ke tahap **pembacaan data sensor**, di mana sensor membaca nilai aktual dari lingkungan. Nilai ini disebut X(n) dan merupakan data mentah yang belum difilter. Data ini kemudian dikirim ke tahap berikutnya, yaitu **proses filtering**.

Pada proses filtering, data X(n) diproses menggunakan metode *Exponential Smoothing* dengan empat nilai alpha yang berbeda. Setiap alpha menghasilkan tingkat peredaman noise yang berbeda — alpha kecil menghasilkan hasil yang lebih halus namun lambat merespon perubahan, sedangkan alpha besar lebih cepat merespon tetapi kurang stabil. Proses ini menghasilkan empat data hasil filter masing-masing untuk *alpha* 1 hingga *alpha* 4.

Setelah hasil filtering diperoleh, sistem melanjutkan ke tahap penampilan hasil. Pada tahap ini, nilai asli sensor (X(n)) dan keempat nilai hasil filter ditampilkan ke *serial monitor*.Tampilan ini memudahkan analisis kinerja filter dengan membandingkan pembacaan asli dan hasil filter.Sistem kemudian masuk kembali ke *loop* pembacaan berikutnya, mengulang proses mulai dari pembacaan sensor hingga penampilan hasil secara terus menerus.

Dengan alur ini, sistem dapat memantau kondisi sensor secara *real-time* sekaligus menstabilkan pembacaan menggunakan filter adaptif.

2. Kode Program Untuk Membaca Suhu (LM35)

a) kode pemanrogram sebelum menggunakan filter adaptif berbasis *exponential smoothing* :

```
const int lm35Pin = A0;

void setup() {
Serial.begin(9600);
}

void loop() {
int sensorValue = analogRead(lm35Pin);
float voltage = sensorValue * (5.0 / 1023.0);
float temperatureC = voltage * 100.0;

Serial.print("Suhu: ");
Serial.print(temperatureC);
Serial.println(" °C");

delay(1000); //
}
```

b) Implementasi filter adaptif berbasis *exponential smoothing* pada kode program

```
float alpha1 = 0.05;
float alpha2 = 0.1;
float alpha3 = 0.3;
float alpha4 = 0.5;

float filteredValue1 = 0;
float filteredValue2 = 0;
float filteredValue3 = 0;
float filteredValue4 = 0;
float lm35Suhu = 0;

void setup() {
  Serial.begin(9600);
}
```

```
void loop() {
  int lm35suhu1 = analogRead(A0);
  lm35Suhu = lm35suhu1 * 0.48828125;
  Serial.print(lm35Suhu);

filteredValue1=alpha1 * lm35Suhu + (1-alpha1) *
filteredValue1;
Serial.print("  ");
Serial.print(filteredValue1);

filteredValue2=alpha2 * lm35Suhu + (1-alpha2) *
filteredValue2;
Serial.print("  ");
Serial.print(filteredValue2);

filteredValue3=alpha3 * lm35Suhu + (1-alpha3) *
filteredValue3;
Serial.print("  ");
Serial.print(filteredValue3);

filteredValue4=alpha4 * lm35Suhu + (1-alpha4) *
filteredValue4;
Serial.print("  ");
Serial.println(filteredValue4);
delay(250);
```

c) Penjelasan kode Program

```
float alpha1 = 0.05;
float alpha2 = 0.1;
float alpha3 = 0.3;
float alpha4 = 0.5;
```

Mendefinisikan 4 nilai **koefisien *smoothing* (α)** untuk filter eksponensial.

```
float filteredValue1 = 0;
float filteredValue2 = 0;
float filteredValue3 = 0;
float filteredValue4 = 0;
```

Menyimpan nilai keluaran dari iterasi sebeluemnya untuk masing masing *alpha*.

```
float lm35Suhu = 0;
```

Variabel untuk menyimpan konversi hasil konversi suhu.

```
void setup() {
  Serial.begin(9600);
}
```

Menyiapkan komunikasi *serial* dengan *baud rate* 9600 dan menampilan data ke *Serial monitor*.

```
void loop() {
  int lm35suhu1 = analogRead(A0);
```

Membaca nilai ADC 10-bit (0–1023) dari pin analog A0. Ini adalah representasi tegangan keluaran dari sensor LM35.

```
lm35Suhu = lm35suhu1 * 0.48828125;
Serial.print(lm35Suhu);
```

Mengubah nilai ADC menjadi tegangan aktual dalam volt.

1'C = 10mV (sesuai *datasheet*)

5v /1023 = 4,883 mV (5v = tegangan refrensi, 1023 = resolusi 10 bit)

setiap kenaikan 1'C --> 10 / 4.883 = 2.0479

sehingga didapat rumus lm35Suhu = lm35suhu1 / 2.0479.

```
filteredValue1=alpha1 * lm35Suhu + (1-alpha1) *
filteredValue1;
```

Implementasi dari metode *Exponential Smoothing* untuk menyaring data sensor suhu dari sensor LM35. Dalam persamaan ini, lm35Suhu mewakili data suhu terbaru yang dibaca dari sensor, sedangkan *filteredValue1* menyimpan nilai suhu yang telah difilter pada iterasi sebelumnya. Nilai *alpha*1 adalah konstanta peluruhan (*smoothing factor*) yang menentukan seberapa besar pengaruh data terbaru terhadap hasil akhir semakin kecil nilainya (misalnya 0.05), maka filter akan lebih lambat merespons perubahan mendadak, namun memberikan hasil yang lebih halus dan stabil.

Formula tersebut bekerja dengan mengambil sebagian dari data baru (*alpha*1 * lm35Suhu) dan sebagian dari hasil sebelumnya (1 - *alpha1*) * *filteredValue*1) untuk menghasilkan nilai baru yang lebih stabil. Dengan demikian, hasil filter ini mampu mengikuti tren data suhu yang sebenarnya, namun tanpa terlalu terpengaruh oleh gangguan sesaat atau lonjakan data yang tidak wajar.

```
Serial.print("  ");
```

Menambahkan dua spasi sebagai pemisah kolom di serial monitor

```
Serial.print(filteredValue1);
```

Mencetak nilai hasil filter di *serial monitor* .

```
filteredValue2=alpha2 * lm35Suhu + (1-alpha2) *
filteredValue2;
Serial.print("  ");
Serial.print(filteredValue2);

filteredValue3=alpha3 * lm35Suhu + (1-alpha3) *
filteredValue3;
Serial.print("  ");
Serial.print(filteredValue3);

filteredValue4=alpha4 * lm35Suhu + (1-alpha4) *
filteredValue4;
Serial.print("  ");
Serial.println(filteredValue4);
```

Diulangi oleh *alpha* lainnya.

```
Delay (250);
```

Menunda eksekusi selama 250 milidetik atau 0,25 detik, agar pembacaan tidak terlalu lambat dan mudah dipantau.

3. Kode Program Untuk Membaca berat (Load Cell + HX711)

1) Kode program sebelum menggunakan filter adaptif berbasis *exponential smoothing* **:**

```
 #include <HX711.h>
#include <EEPROM.h>

HX711 scale;

float scale_factor = 1.0;
const int eepromAddr = 0;

Void setup() {
  Serial.begin(9600);
  scale.begin(DT, SCK);

  // Ambil scale factor dari EEPROM
  EEPROM.get(eepromAddr, scale_factor);
  if (scale_factor <= 0.01 || scale_factor > 100000)
scale_factor = 1.0;

  scale.set_scale(scale_factor);
  scale.tare(); // Nolkan timbangan

  Serial.println("Ketik 'k' lalu Enter untuk
kalibrasi.");
}

void loop() {
  if (Serial.available()) {
    if (Serial.read() == 'k') kalibrasi();
  }

  float berat = scale.get_units(10);
  Serial.print("Berat: ");
  Serial.print(berat, 2);
  Serial.println(" gram");
  delay(1000);
}

void kalibrasi() {
  Serial.println("\n--- Kalibrasi ---");
  Serial.println("Kosongkan timbangan, lalu tekan
Enter...");
  while (Serial.available()) Serial.read();
  while (!Serial.available());
  scale.tare();
```

```
  Serial.println("Letakkan beban diketahui, tekan
Enter...");
  while (Serial.available()) Serial.read();
  while (!Serial.available());

  float nilaiMentah = scale.get_units(10);
  Serial.print("Nilai mentah: ");
  Serial.println(nilaiMentah);
Serial.println("Masukkan berat sebenarnya (gram):");
  while (!Serial.available());
  float beratAktual = Serial.parseFloat();

  if (beratAktual > 0) {
    scale_factor = nilaiMentah / beratAktual;
    scale.set_scale(scale_factor);
    EEPROM.put(eepromAddr, scale_factor);
    Serial.print("Kalibrasi selesai. Faktor skala
disimpan: ");
    Serial.println(scale_factor, 4);
  } else {
    Serial.println("Berat tidak valid.");
  }
}
```

2) Implementasi filter adaptif berbasis *exponential smoothing* pada kode program

```
#include <HX711.h>
#include <EEPROM.h>

#define DOUT 3
#define CLK 2

HX711 scale;
float scale_factor = 1.0; // Default scale factor
bool modeKalibrasi = false;
bool sudahKalibrasi = false;

float alpha1 = 0.05;
float alpha2 = 0.10;
float alpha3 = 0.30;
float alpha4 = 0.50;
float Xn = 0.0, Yn1_1 = 0.0, Yn2_1 = 0.0, Yn3_1 = 0.0,
Yn4_1 = 0.0;

void setup() {
  Serial.begin(9600);
  scale.begin(DOUT, CLK);

  EEPROM.get(0, scale_factor);
  Serial.print("Scale factor dari EEPROM: ");
  Serial.println(scale_factor, 6);

  scale.set_scale(scale_factor);
  scale.tare();
  Serial.println("Load cell siap.");
  Serial.println("Ketik 'k' untuk mulai kalibrasi
  baru.");
}

void loop() {
  // Mode Kalibrasi
  if (Serial.available()) {
    String input = Serial.readStringUntil('\n');
    input.trim();

    if (input == "k") {
      modeKalibrasi = true;
      Serial.println("\n== MODE KALIBRASI ==");

      scale.set_scale();  // Reset sementara
      scale.tare();

      Serial.println("Letakkan beban yang diketahui,
      lalu tekan ENTER.");
      while (!Serial.available()) delay(10);
      Serial.readStringUntil('\n');
```

```
      float raw = scale.get_units(10);
      Serial.print("Nilai mentah HX711: ");
      Serial.println(raw, 6);

      Serial.println("Masukkan BERAT aktual
(gram):");
      while (!Serial.available()) delay(10);
      float berat =
      Serial.readStringUntil('\n').toFloat();

    if (berat > 0) {
        scale_factor = raw / berat;
        scale.set_scale(scale_factor);
        EEPROM.put(0, scale_factor);
        Serial.print("Skala baru disimpan: ");
        Serial.println(scale_factor, 6);
        sudahKalibrasi = true;

      } else {
        Serial.println("Berat tidak valid.");
      }

      modeKalibrasi = false;
    }
  }

      // Pembacaan dan Filtering
  if (!modeKalibrasi) {
    Xn = scale.get_units(10);

    float Yn1 = alpha1 * Xn + (1 - alpha1) * Yn1_1;
    float Yn2 = alpha2 * Xn + (1 - alpha2) * Yn2_1;
    float Yn3 = alpha3 * Xn + (1 - alpha3) * Yn3_1;
    float Yn4 = alpha4 * Xn + (1 - alpha4) * Yn4_1;

    Yn1_1 = Yn1;
    Yn2_1 = Yn2;
    Yn3_1 = Yn3;
    Yn4_1 = Yn4;

   Serial.print(Xn);    // Asli
    Serial.print("  ");
    Serial.print(Yn1);   // alpha 0.05
    Serial.print("  ");
    Serial.print(Yn2);   // alpha 0.1
    Serial.print("  ");
    Serial.print(Yn3);   // alpha 0.3
    Serial.print("  ");
    Serial.println(Yn4); // alpha 0.5

    delay(250);
  }
}
```

3) **Penjelasan kode program**

```
#include <HX711.h>
#include <EEPROM.h>
```

Memuat *library* HX711 untuk membaca sinyal dari sensor load cell melalui modul HX711 dan Memuat library EEPROM agar nilai kalibrasi bisa disimpan permanen, walaupun Arduino dimatikan.

```
#define DOUT 3
#define CLK 2
```

Mendefinisikan pin DOUT dan CLK HX711 terhubung ke pin digital 3 dan 2 di Arduiuno.

```
HX711 Scale ;
```

Membuat objek *scale* untuk mengakses fungsi dari HX711.

```
float scale_factor = 1.0;
bool modeKalibrasi = false;
bool sudahKalibrasi = false;
```

Variabel scale factor menyimpan faktor kalibrasi, mode kalibrasi mengatur apakah sistem sedang dalam mode kalibaraasi atau tidak, sudah kalibrasi sebagai indikator bahwa kalibarsai berhasil dilakukan.

```
float alpha1 = 0.05;
float alpha2 = 0.10;
float alpha3 = 0.30;
float alpha4 = 0.50;= false;
```

4 metode alpha berbeda untuk metode *exponential smoothing.*

```
 float Xn = 0.0, Yn1_1 = 0.0, Yn2_1 = 0.0, Yn3_1 = 0.0,
Yn4_1 = 0.0;
```

Xn adalah pembacaan berat saat ini YnX_1 adalah hasil *smoothing* sebleumnnya (Y(n-1)) untuk masing-masing alpha.

```
void setup() {
  Serial.begin(9600);
  scale.begin(DOUT, CLK);
```

Fungsi set up () untuk menginisialisasi komunikasi serial dan load cell.

```
EEPROM.get(0, scale_factor);
  Serial.print("Scale factor dari EEPROM: ");
  Serial.println(scale_factor, 6);
```

Membaca nilai kalibrasi dan menampilkannya

```
scale.set_scale(scale_factor);
  scale.tare();
```

Menerapkan faktor kalibrasi dan menyetel nol (*tare*)

```
serial.println("Load cell siap.");
  Serial.println("Ketik 'k' untuk mulai kalibrasi
baru.");
}
```

Menampilakan instruksi ke pengguna.

```
void loop() {
  // Mode Kalibrasi
  if (Serial.available()) {
    String input = Serial.readStringUntil('\n');
    input.trim()
```

Fungsi loop, jika ada input dari pengguna via *serial monitor*, baca dan bersihkan spasi/enter-nya .

```
  if (input == "k") {
      modeKalibrasi = true;
      Serial.println("\n== MODE KALIBRASI ==");
```

Masuk ke mode kalibrasi jika penggunaka mengetik ”k”.

```
scale.set_scale
  scale.tare();
```

Reset sementara skala dan lakukan *tare* ulang.

```
  Serial.println("Letakkan beban yang diketahui, lalu
tekan ENTER.");
      while (!Serial.available()) delay(10);
      Serial.readStringUntil('\n');
```

Minta pengguna letakkan beban lalu tekan *enter*, menunggu hingga ada input.

```
float raw = scale.get_units(10);
      Serial.print("Nilai mentah HX711: ");
      Serial.println(raw, 6);
```

Ambil 10 kali pembacaan untuk dirata-rata kan, lalu tampilkan nilainya.

```
Serial.println("Masukkan BERAT aktual (gram):");
      while (!Serial.available()) delay(10);
      float berat =
Serial.readStringUntil('\n').toFloat();
```

Minta pengguna memasukkan nilai berat aktual dalam bentuk gram.

```
  if (berat > 0) {
        scale_factor = raw / berat;
        scale.set_scale(scale_factor);
        EEPROM.put(0, scale_factor);
        Serial.print("Skala baru disimpan: ");
        Serial.println(scale_factor, 6);
        sudahKalibrasi = true;
      } else {
        Serial.println("Berat tidak valid.");
      }
```

Jika valild hitung *scale factor,* terapkan ke HX711, simpan ke EEPROM, dan beri konfirmasi.

```
modeKalibrasi = false;
```

Keluar dari mode kalibrasi.

```
if (!modeKalibrasi) {
 Xn = scale.get_units(10);
```

Jika bukan dalam mode kalibrasi, ambil data dari sensor rata-rata 10 kali pembacaan.

```
float Yn1 = alpha1 * Xn + (1 - alpha1) * Yn1_1;
float Yn2 = alpha2 * Xn + (1 - alpha2) * Yn2_1;
float Yn3 = alpha3 * Xn + (1 - alpha3) * Yn3_1;
float Yn4 = alpha4 * Xn + (1 - alpha4) * Yn4_1;
```

Menghitung hasil *smoothimg* (Yn) untuk masing-masing *alpha*.

```
Yn1_1 = Yn1;
Yn2_1 = Yn2;
Yn3_1 = Yn3;
Yn4_1 = Yn4;
```

Menyimpan hasil *smoothing* sekarang sebargai refrensi untul iterasi berikutnya (Y(n-1.)).

```
Serial.print(Xn);    // Asli
Serial.print("  ");
Serial.print(Yn1);   // alpha 0.05
Serial.print("  ");
Serial.print(Yn2);   // alpha 0.1
Serial.print("  ");
Serial.print(Yn3);   // alpha 0.3
Serial.print("  ");
Serial.println(Yn4); // alpha 0.5
```

Menampilkan hasil pembacaan dan semua hasil *smoothing* di satu baris, dipisahkan spasi.

```
delay(250);
```

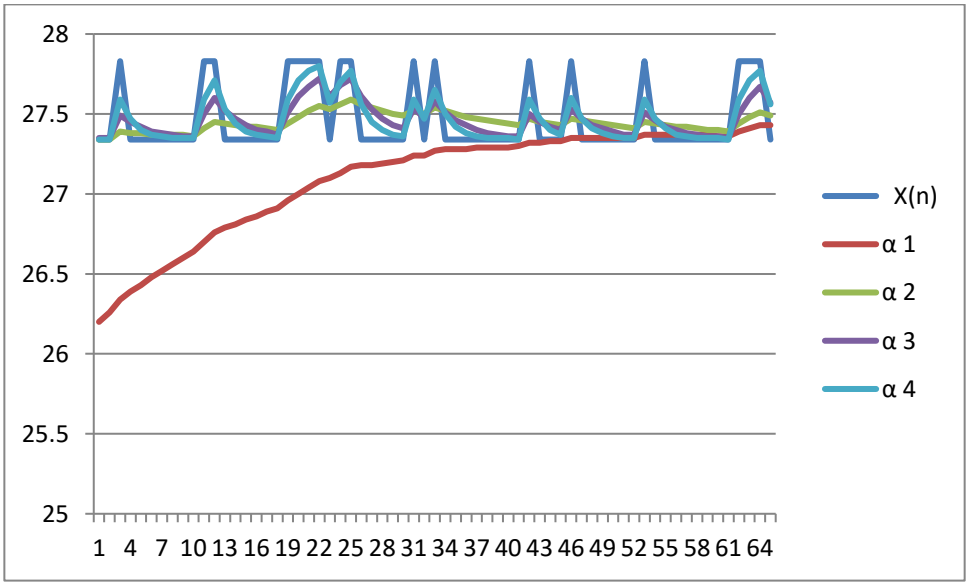
Delay untuk memberi waktu antar pembacaan selama 0,25 .

C. Pengujian Sistem

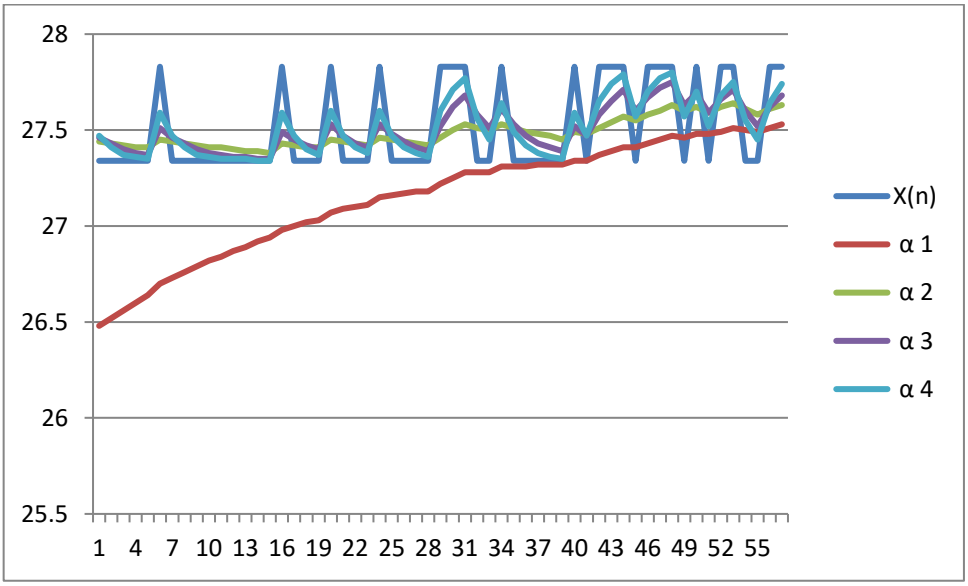
1. Pengujiam filter pada sesor LM35

Pengujian dilakukan sebanyak tiga kali percobaan untuk membandingkan pembacaan sensor LM35 dengan dan tanpa filter adaptif berbasis metode *Exponential Smoothing*, menggunakan empat variasi nilai α (*alpha*): 0.05, 0.1, 0.3, dan 0.5. Nilai X(n) merupakan data suhu aktual yang dibaca sensor, sedangkan masing-masing α menghasilkan data hasil penyaringan (*filtering*) terhadap sinyal tersebut.

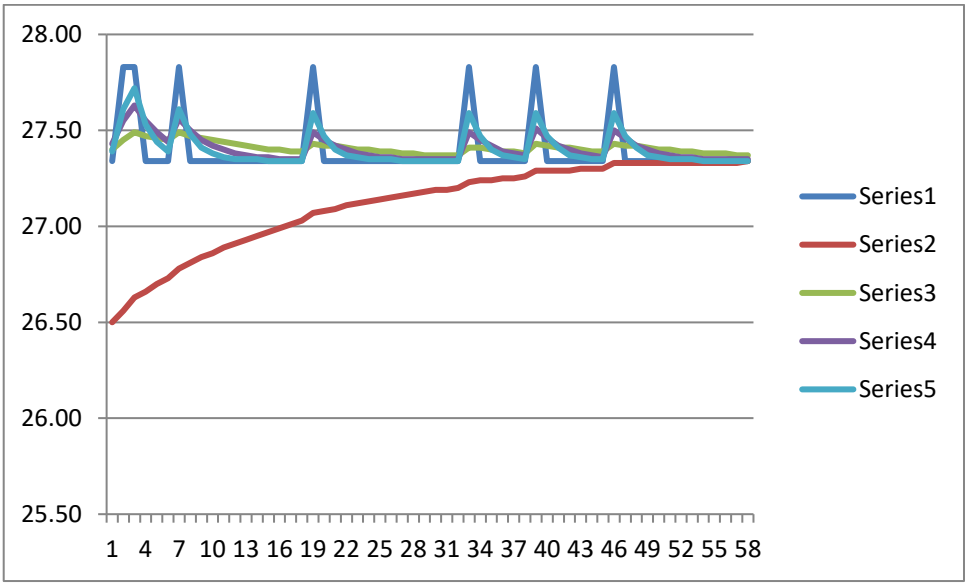
a. Visualisasi dan Data Pengujian Sensor Suhu LM35



Gambar 4. 2 Grafik *Serial Plotter* Percobaan 1 LM35



Gambar 4. 3 Grafik *Serial Plotter* Percobaan 2 LM35



Gambar 4. 4 Grafik *Serial Plotter* Percobaan 3 LM35

Tabel 4. 1 Percobaan 1

X(n)	α 1	α 2	α 3	α 4
27.34	26.20	27.34	27.35	27.34
27.34	26.26	27.34	27.35	27.34
27.83	26.34	27.39	27.49	27.59
27.34	26.39	27.38	27.45	27.47
27.34	26.43	27.38	27.42	27.40
27.34	26.48	27.37	27.39	27.37
27.34	26.52	27.37	27.38	27.36
27.34	26.56	27.37	27.37	27.35
27.34	26.60	27.37	27.36	27.35
27.34	26.64	27.36	27.36	27.35
27.83	26.70	27.41	27.50	27.59
27.83	26.76	27.45	27.60	27.71
27.34	26.79	27.44	27.52	27.53
27.34	26.81	27.43	27.47	27.44
27.34	26.84	27.42	27.43	27.39
27.34	26.86	27.42	27.40	27.37
27.34	26.89	27.41	27.39	27.36
27.34	26.91	27.40	27.37	27.35
27.83	26.96	27.44	27.51	27.59
27.83	27.00	27.48	27.61	27.71
27.83	27.04	27.52	27.67	27.77
27.83	27.08	27.55	27.72	27.80
27.34	27.10	27.53	27.61	27.57
27.83	27.13	27.56	27.68	27.70
27.83	27.17	27.59	27.72	27.77
27.34	27.18	27.56	27.61	27.56

X(n)	α 1	α 2	α 3	α 4
27.34	27.18	27.54	27.53	27.45
27.34	27.19	27.52	27.47	27.40
27.34	27.20	27.50	27.43	27.37
27.34	27.21	27.49	27.41	27.36
27.83	27.24	27.52	27.53	27.59
27.34	27.24	27.50	27.48	27.47
27.83	27.27	27.54	27.58	27.65
27.34	27.28	27.52	27.51	27.50
27.34	27.28	27.50	27.46	27.42
27.34	27.28	27.48	27.43	27.38
27.34	27.29	27.47	27.40	27.36
27.34	27.29	27.46	27.38	27.35
27.34	27.29	27.45	27.37	27.35
27.34	27.29	27.44	27.36	27.35
27.34	27.30	27.43	27.36	27.34
27.83	27.32	27.47	27.50	27.59
27.34	27.32	27.45	27.45	27.47
27.34	27.33	27.44	27.42	27.40
27.34	27.33	27.43	27.40	27.37
27.83	27.35	27.47	27.53	27.60
27.34	27.35	27.46	27.47	27.47
27.34	27.35	27.45	27.43	27.41
27.34	27.35	27.44	27.41	27.38
27.34	27.35	27.43	27.39	27.36
27.34	27.35	27.42	27.37	27.35
27.34	27.35	27.41	27.37	27.35
27.83	27.37	27.45	27.51	27.59

X(n)	α 1	α 2	α 3	α 4
27.34	27.37	27.44	27.46	27.47
27.34	27.37	27.43	27.42	27.41
27.34	27.37	27.42	27.40	27.37
27.34	27.37	27.42	27.38	27.36
27.34	27.37	27.41	27.37	27.35
27.34	27.37	27.40	27.36	27.35
27.34	27.36	27.40	27.36	27.35
27.34	27.36	27.39	27.35	27.34
27.83	27.39	27.44	27.50	27.59
27.83	27.41	27.48	27.60	27.71
27.83	27.43	27.51	27.67	27.77
27.34	27.43	27.49	27.57	27.56

Tabel 4. 2 Percobaan 2

X(n)	α 1	α 2	α 3	α 4
27.34	26.48	27.44	27.46	27.47
27.34	26.52	27.43	27.43	27.41
27.34	26.56	27.42	27.40	27.37
27.34	26.60	27.41	27.38	27.36
27.34	26.64	27.41	27.37	27.35
27.83	26.70	27.45	27.51	27.59
27.34	26.73	27.44	27.46	27.47
27.34	26.76	27.43	27.43	27.41
27.34	26.79	27.42	27.40	27.37
27.34	26.82	27.41	27.38	27.36
27.34	26.84	27.41	27.37	27.35
27.34	26.87	27.40	27.36	27.35
27.34	26.89	27.39	27.36	27.35
27.34	26.92	27.39	27.35	27.34
27.34	26.94	27.38	27.35	27.34
27.83	26.98	27.43	27.49	27.59
27.34	27.00	27.42	27.45	27.47
27.34	27.02	27.41	27.42	27.40
27.34	27.03	27.41	27.40	27.37
27.83	27.07	27.45	27.53	27.60
27.34	27.09	27.44	27.47	27.47
27.34	27.10	27.43	27.43	27.41
27.34	27.11	27.42	27.41	27.38
27.83	27.15	27.46	27.53	27.60

X(n)	α 1	α 2	α 3	α 4
27.34	27.16	27.45	27.48	27.47
27.34	27.17	27.44	27.44	27.41
27.34	27.18	27.43	27.41	27.38
27.34	27.18	27.42	27.39	27.36
27.83	27.22	27.46	27.52	27.60
27.83	27.25	27.50	27.62	27.71
27.83	27.28	27.53	27.68	27.77
27.34	27.28	27.51	27.58	27.56
27.34	27.28	27.50	27.51	27.45
27.83	27.31	27.53	27.61	27.64
27.34	27.31	27.51	27.53	27.49
27.34	27.31	27.49	27.47	27.42
27.34	27.32	27.48	27.43	27.38
27.34	27.32	27.47	27.41	27.36
27.34	27.32	27.45	27.39	27.35
27.83	27.34	27.49	27.52	27.59
27.34	27.34	27.48	27.47	27.47
27.83	27.37	27.51	27.58	27.65
27.83	27.39	27.54	27.65	27.74
27.83	27.41	27.57	27.71	27.79
27.34	27.41	27.55	27.60	27.57
27.83	27.43	27.58	27.67	27.70
27.83	27.45	27.60	27.72	27.77
27.83	27.47	27.63	27.75	27.80
27.34	27.46	27.60	27.63	27.57
27.83	27.48	27.62	27.69	27.70

X(n)	α 1	α 2	α 3	α 4
27.34	27.48	27.59	27.59	27.52
27.83	27.49	27.62	27.66	27.68
27.83	27.51	27.64	27.71	27.75
27.34	27.50	27.61	27.60	27.55
27.34	27.49	27.58	27.52	27.45
27.83	27.51	27.61	27.62	27.64
27.83	27.53	27.63	27.68	27.74

Tabel 4. 3 Percobaan 3

X(n)	α 1	α 2	α 3	α 4
27.34	26.50	27.40	27.43	27.39
27.83	26.56	27.45	27.55	27.61
27.83	26.63	27.49	27.63	27.72
27.34	26.66	27.47	27.55	27.53
27.34	26.70	27.46	27.49	27.44
27.34	26.73	27.45	27.44	27.39
27.83	26.78	27.49	27.56	27.61
27.34	26.81	27.47	27.50	27.48
27.34	26.84	27.46	27.45	27.41
27.34	26.86	27.45	27.42	27.38
27.34	26.89	27.44	27.40	27.36
27.34	26.91	27.43	27.38	27.35
27.34	26.93	27.42	27.37	27.35
27.34	26.95	27.41	27.36	27.35
27.34	26.97	27.40	27.36	27.34
27.34	26.99	27.40	27.35	27.34
27.34	27.01	27.39	27.35	27.34

X(n)	α 1	α 2	α 3	α 4
27.34	27.03	27.39	27.35	27.34
27.83	27.07	27.43	27.49	27.59
27.34	27.08	27.42	27.45	27.47
27.34	27.09	27.42	27.42	27.40
27.34	27.11	27.41	27.40	27.37
27.34	27.12	27.40	27.38	27.36
27.34	27.13	27.40	27.37	27.35
27.34	27.14	27.39	27.36	27.35
27.34	27.15	27.39	27.36	27.35
27.34	27.16	27.38	27.35	27.34
27.34	27.17	27.38	27.35	27.34
27.34	27.18	27.37	27.35	27.34
27.34	27.19	27.37	27.35	27.34
27.34	27.19	27.37	27.35	27.34
27.34	27.20	27.37	27.35	27.34
27.83	27.23	27.41	27.49	27.59
27.34	27.24	27.41	27.45	27.47
27.34	27.24	27.40	27.42	27.40
27.34	27.25	27.39	27.39	27.37
27.34	27.25	27.39	27.38	27.36
27.34	27.26	27.38	27.37	27.35
27.83	27.29	27.43	27.51	27.59
27.34	27.29	27.42	27.46	27.47
27.34	27.29	27.41	27.42	27.41
27.34	27.29	27.41	27.40	27.37
27.34	27.30	27.40	27.38	27.36

X(n)	α 1	α 2	α 3	α 4
27.34	27.30	27.39	27.37	27.35
27.34	27.30	27.39	27.36	27.35
27.83	27.33	27.43	27.50	27.59
27.34	27.33	27.42	27.46	27.47
27.34	27.33	27.42	27.42	27.41
27.34	27.33	27.41	27.40	27.37
27.34	27.33	27.40	27.38	27.36
27.34	27.33	27.40	27.37	27.35
27.34	27.33	27.39	27.36	27.35
27.34	27.33	27.39	27.36	27.35
27.34	27.33	27.38	27.35	27.34
27.34	27.33	27.38	27.35	27.34
27.34	27.33	27.38	27.35	27.34
27.34	27.33	27.37	27.35	27.34
27.34	27.34	27.37	27.35	27.34

b. Pola Data Awal (Tanpa Filter) LM35

Dari hasil pembacaan X(n)**,** terlihat bahwa nilai suhu mengalami fluktuasi yang cukup sering meskipun dalam kondisi lingkungan yang cenderung stabil. Pada percobaan pertama, data X(n) berosilasi di sekitar angka 27.34 hingga 27.83 °C, menunjukkan adanya *noise* yang mengganggu.

c. Efek Penggunaan Filter pada LM35

α = 0.05 dan 0.1 meredam fluktuasi dengan sangat baik, menghasilkan data yang lebih halus dan stabil. Cocok digunakan ketika tujuan utama adalah mendapatkan data jangka panjang yang akurat dengan toleransi terhadap respons

lambat terhadap perubahan suhu, Sedangkan α = 0.3 dan 0.5 Memiliki respon lebih cepat terhadap perubahan suhu, namun kurang optimal dalam menekan *noise* kecil. Data cenderung mengikuti fluktuasi X(n) dengan lebih agresif.

d. Evaluasi Kinerja Filter

Evaluasi respon filter terhaadap *noise* , dapat ditinjau dari seberapa besar filter mengikuti lonjakan atau gangguan mendadak biasa disebut *impuls noise* atau *transien* pada data x(n), karena filter yang bagus tidak langsung menuruti perubahan *(noise),* tapi juga tidak terlalu lambat saat data benar benar berubah.

Adapun pada hasil percobaan 1,2,3 memiliki karakter data yang sama yaitu ketika data X(n) mengalami *impuls noise* atau perubahan mendadak maka data Y(n) atau filter mengikuti perubahan berdasarkan masing-masing alpha nya, dapat dilihat dari potongan data yang mengalami *impuls noise* ;

Tabel 4. 4 Potongan Data Percobaan 1

Percobaan 1				
X(n)	α 1	α 2	α 3	α 4
27.34	27.36	27.40	27.36	27.35
27.34	27.36	27.39	27.35	27.34
27.83	27.39	27.44	27.50	27.59
27.83	27.41	27.48	27.60	27.71
27.83	27.43	27.51	27.67	27.77

→ α 4 cepat mengikuti *noise*

Tabel 4. 5 Potongan Data Percobaan 2

Percobaan 2				
X(n)	α 1	α 2	α 3	α 4
27.34	27.48	27.59	27.59	27.52
27.83	27.49	27.62	27.66	27.68
27.83	27.51	27.64	27.71	27.75
27.34	27.50	27.61	27.60	27.55
27.34	27.49	27.58	27.52	27.45

→ α 4 cepat mengikuti *noise*

Tabel 4. 6 Potongan Data Percobaan 3

Percobaan 3				
X(n)	α 1	α 2	α 3	α 4
27.34	27.30	27.39	27.36	27.35
27.83	27.33	27.43	27.50	27.59
27.34	27.33	27.42	27.46	27.47
27.34	27.33	27.42	27.42	27.41
27.34	27.33	27.41	27.40	27.37

→ α 4 cepat mengikuti *noise*

Nilai Y(n) pada *alpha* 1 (0.05) hanya naik atau turun sedikit saat nilai X(n) mengalami perubahan mendadak, sebaliknya alpha 4 (0.5) mengalami perubahan drastis hampir mendekati nilai X(n).

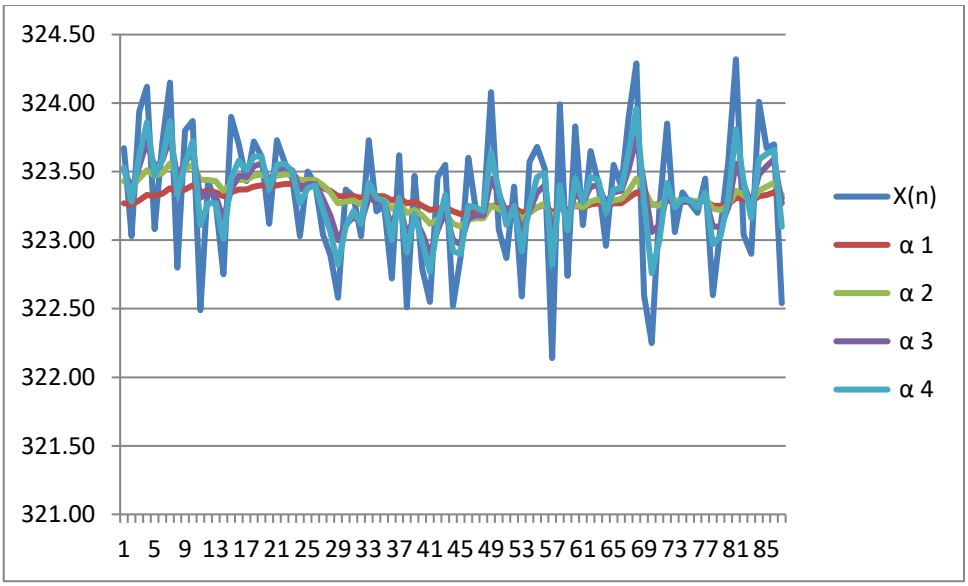
2. Pengujiam filter pada sensor berat Load Cell + HX711

Pada pengujian ini, data pembacaan sensor ditampilakan dalam bentuk nilai mentah (Xn) dan hasil penyaringan menggunakan metode *exponential smoothing* dengan empat variasi nilai *alpha*, yaitu α1, α2, α3, α4.

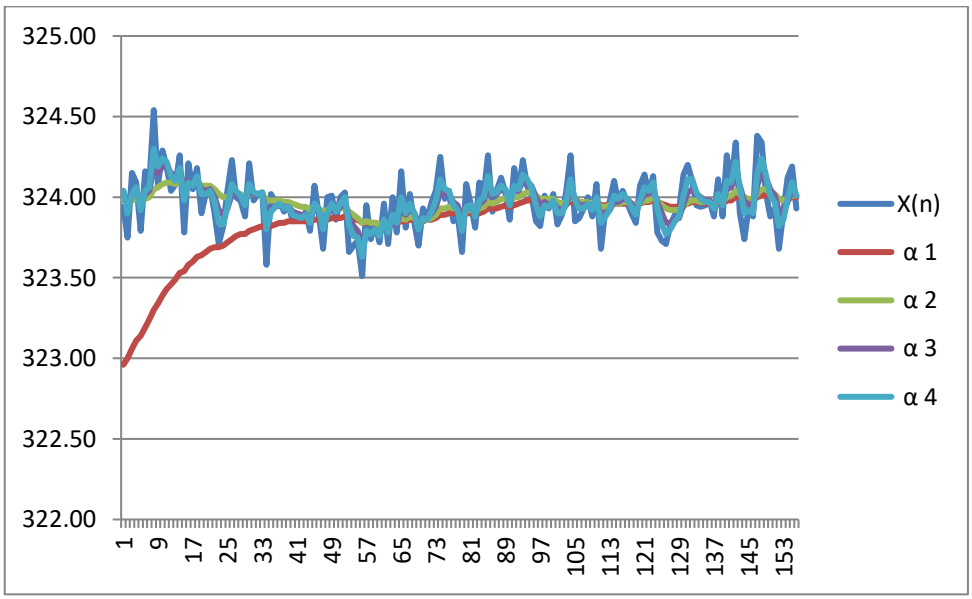
Dalam pengujian performa filter terhadap pembacaan sensor load cell, penelitian ini hanya menggunakan satu beban dengan nilai tetap yang diletakkan ssecara statis

tanpa adanya perpindahan atau penggantian beban. Pemilihan metode ini bertujuan untuk memfokuskan analisis terhadap kualitas sinyal yang dihasilkan sensor, khususnya dalam mendeteksi dan mereduksi noise yang muncul seacara alami pada kondisi beban tetap.

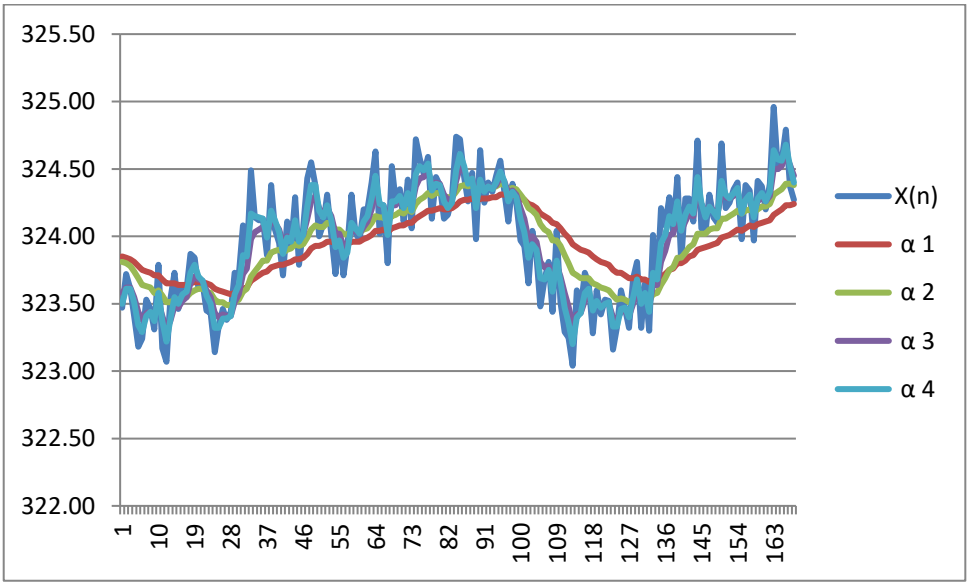
a. Visualisasi dan Data Pengujian Sensor berat Load Cell + HX711



Gambar 4. 5 Grafik *Plotter* Pengujian 1 Load Cell



Gambar 4. 6 Grafik *Plotter* pengujian 2 Load cell



Gambar 4. 7 Grafik *Plotter* Pngujian 3 Load Cell

Tabel 4. 7 Percobaan 1

X(n)	α 1	α 2	α 3	α 4
323.67	323.27	323.43	323.52	323.53
323.03	323.26	323.39	323.37	323.28
323.94	323.29	323.45	323.54	323.61
324.12	323.33	323.51	323.72	323.86
323.08	323.32	323.47	323.53	323.47
323.71	323.34	323.49	323.58	323.59
324.15	323.38	323.56	323.75	323.87
322.80	323.35	323.48	323.47	323.33
323.80	323.37	323.52	323.56	323.56
323.87	323.40	323.55	323.66	323.72
322.49	323.35	323.44	323.31	323.11
323.43	323.36	323.44	323.35	323.27
323.27	323.35	323.43	323.32	323.27
322.75	323.32	323.36	323.15	323.01
323.90	323.35	323.41	323.37	323.45
323.70	323.37	323.44	323.47	323.58
323.43	323.37	323.44	323.46	323.51
323.72	323.39	323.47	323.54	323.61
323.61	323.40	323.48	323.56	323.61
323.12	323.39	323.45	323.43	323.37
323.73	323.40	323.47	323.52	323.55

323.57	323.41	323.48	323.53	323.56
323.45	323.41	323.48	323.51	323.50
323.03	323.40	323.44	323.37	323.27
323.50	323.40	323.44	323.41	323.38
323.41	323.40	323.44	323.41	323.40
323.04	323.38	323.40	323.30	323.22
322.89	323.36	323.35	323.18	323.06
322.58	323.32	323.27	323.00	322.82
323.37	323.32	323.28	323.11	323.10
323.32	323.32	323.29	323.17	323.21
323.03	323.31	323.26	323.13	323.12
323.73	323.33	323.31	323.31	323.42
323.21	323.32	323.30	323.28	323.31
323.25	323.32	323.29	323.27	323.28
322.72	323.29	323.23	323.10	323.00
323.62	323.31	323.27	323.26	323.31
322.51	323.27	323.20	323.04	322.91
323.47	323.28	323.22	323.16	323.19
322.79	323.25	323.18	323.05	322.99
322.55	323.22	323.12	322.90	322.77
323.46	323.23	323.15	323.07	323.11
323.55	323.24	323.19	323.21	323.33
322.52	323.21	323.12	323.00	322.92
322.88	323.19	323.10	322.97	322.90
323.60	323.21	323.15	323.16	323.25
323.23	323.21	323.16	323.18	323.24
323.18	323.21	323.16	323.18	323.21
324.08	323.25	323.25	323.45	323.64
323.08	323.25	323.23	323.34	323.36
322.87	323.23	323.20	323.20	323.11
323.39	323.24	323.22	323.26	323.25
322.59	323.20	323.15	323.05	322.92
323.57	323.22	323.20	323.21	323.25
323.68	323.24	323.24	323.35	323.46
323.52	323.26	323.27	323.40	323.49
322.14	323.20	323.16	323.03	322.82
323.99	323.24	323.24	323.31	323.40
322.74	323.22	323.19	323.14	323.07
323.83	323.25	323.26	323.35	323.45
323.11	323.24	323.24	323.28	323.28

323.65	323.26	323.28	323.39	323.46
323.42	323.27	323.30	323.40	323.44
322.96	323.25	323.26	323.27	323.20
323.55	323.27	323.29	323.35	323.37
323.39	323.27	323.30	323.36	323.38
323.90	323.31	323.36	323.52	323.64
324.29	323.35	323.45	323.75	323.96
322.60	323.32	323.37	323.41	323.28
322.25	323.26	323.26	323.06	322.76
323.23	323.26	323.25	323.11	323.00
323.85	323.29	323.31	323.33	323.42
323.06	323.28	323.29	323.25	323.24
323.35	323.28	323.29	323.28	323.30
323.27	323.28	323.29	323.28	323.29
323.20	323.28	323.28	323.26	323.24
323.45	323.29	323.30	323.31	323.35
322.60	323.25	323.23	323.10	322.97
323.09	323.25	323.22	323.10	323.03
323.57	323.26	323.25	323.24	323.30
324.32	323.31	323.36	323.56	323.81
323.04	323.30	323.33	323.41	323.43
322.90	323.28	323.28	323.26	323.16
324.01	323.32	323.36	323.48	323.59
323.67	323.33	323.39	323.54	323.63
323.70	323.35	323.42	323.59	323.66
322.54	323.31	323.33	323.27	323.10

Tabel 4. 8 Percobaan 2

X(n)	α 1	α 2	α 3	α 4
323.94	322.96	323.99	324.04	324.04
323.75	323.00	323.96	323.95	323.90
324.15	323.06	323.98	324.01	324.02
324.09	323.11	323.99	324.04	324.06
323.79	323.14	323.97	323.96	323.92
324.16	323.19	323.99	324.02	324.04
324.08	323.24	324.00	324.04	324.06

324.54	323.30	324.05	324.19	324.30
324.07	323.34	324.06	324.15	324.19
324.29	323.39	324.08	324.19	324.24
324.19	323.43	324.09	324.19	324.22
324.04	323.46	324.09	324.15	324.13
324.08	323.49	324.08	324.13	324.10
324.26	323.53	324.10	324.17	324.18
323.78	323.54	324.07	324.05	323.98
324.21	323.58	324.08	324.10	324.09
324.05	323.60	324.08	324.08	324.07
324.18	323.63	324.09	324.11	324.13
323.90	323.64	324.07	324.05	324.02
324.03	323.66	324.07	324.04	324.02
324.06	323.68	324.07	324.05	324.04
323.90	323.69	324.05	324.00	323.97
323.71	323.69	324.02	323.92	323.84
323.82	323.70	324.00	323.89	323.83
324.04	323.72	324.00	323.93	323.94
324.23	323.74	324.02	324.02	324.08
324.00	323.76	324.02	324.02	324.04
323.98	323.77	324.02	324.01	324.01
323.88	323.77	324.00	323.97	323.95
324.21	323.79	324.02	324.04	324.08
323.98	323.80	324.02	324.02	324.03
324.02	323.81	324.02	324.02	324.02
324.03	323.82	324.02	324.02	324.03
323.58	323.81	323.98	323.89	323.80

324.02	323.82	323.98	323.93	323.91
323.97	323.83	323.98	323.94	323.94
323.97	323.84	323.98	323.95	323.96
323.91	323.84	323.97	323.94	323.93
323.96	323.85	323.97	323.94	323.94
323.85	323.85	323.96	323.91	323.89
323.87	323.85	323.95	323.90	323.88
323.86	323.85	323.94	323.89	323.87
323.91	323.85	323.94	323.89	323.89
323.79	323.85	323.92	323.86	323.84
324.07	323.86	323.94	323.93	323.96
323.90	323.86	323.93	323.92	323.93
323.68	323.85	323.91	323.85	323.80
324.00	323.86	323.92	323.89	323.90
324.01	323.87	323.93	323.93	323.96
323.86	323.87	323.92	323.91	323.91
324.00	323.87	323.93	323.94	323.96
324.03	323.88	323.94	323.97	324.00
323.66	323.87	323.91	323.87	323.83
323.70	323.86	323.89	323.82	323.76
323.73	323.86	323.87	323.79	323.75
323.51	323.84	323.84	323.71	323.63
323.95	323.84	323.85	323.78	323.79
323.74	323.84	323.84	323.77	323.77
323.84	323.84	323.84	323.79	323.80
323.72	323.83	323.83	323.77	323.76
323.96	323.84	323.84	323.83	323.86

323.71	323.83	323.83	323.79	323.78
324.00	323.84	323.84	323.85	323.89
323.78	323.84	323.84	323.83	323.84
324.16	323.85	323.87	323.93	324.00
323.81	323.85	323.86	323.89	323.90
324.02	323.86	323.88	323.93	323.96
323.84	323.86	323.87	323.90	323.90
323.70	323.85	323.86	323.84	323.80
323.93	323.85	323.86	323.87	323.87
323.86	323.86	323.86	323.87	323.86
323.97	323.86	323.87	323.90	323.91
324.04	323.87	323.89	323.94	323.98
324.25	323.89	323.93	324.03	324.11
323.99	323.89	323.93	324.02	324.05
324.04	323.90	323.94	324.02	324.04
323.85	323.90	323.93	323.97	323.95
323.90	323.90	323.93	323.95	323.92
323.66	323.89	323.90	323.86	323.79
324.08	323.90	323.92	323.93	323.94
323.96	323.90	323.92	323.94	323.95
323.81	323.89	323.91	323.90	323.88
324.09	323.90	323.93	323.96	323.98
324.02	323.91	323.94	323.98	324.00
324.26	323.93	323.97	324.06	324.13
323.91	323.93	323.96	324.01	324.02
324.04	323.93	323.97	324.02	324.03
324.12	323.94	323.99	324.05	324.07

324.02	323.95	323.99	324.04	324.05
323.86	323.94	323.98	323.99	323.95
324.18	323.95	324.00	324.05	324.07
324.03	323.96	324.00	324.04	324.05
324.23	323.97	324.02	324.10	324.14
324.07	323.98	324.03	324.09	324.11
324.01	323.98	324.03	324.07	324.06
323.85	323.97	324.01	324.00	323.95
323.82	323.96	323.99	323.95	323.88
324.01	323.97	323.99	323.97	323.95
323.93	323.96	323.99	323.95	323.94
324.02	323.97	323.99	323.97	323.98
323.83	323.96	323.97	323.93	323.90
323.89	323.96	323.96	323.92	323.90
324.03	323.96	323.97	323.95	323.96
324.26	323.97	324.00	324.04	324.11
323.85	323.97	323.99	323.99	323.98
323.87	323.96	323.97	323.95	323.93
323.93	323.96	323.97	323.95	323.93
324.00	323.96	323.97	323.96	323.96
323.88	323.96	323.96	323.94	323.92
324.08	323.97	323.97	323.98	324.00
323.68	323.95	323.94	323.89	323.84
323.93	323.95	323.94	323.90	323.88
323.97	323.95	323.95	323.92	323.92
324.10	323.96	323.96	323.97	324.01
323.97	323.96	323.96	323.97	323.99

324.04	323.96	323.97	323.99	324.02
323.97	323.96	323.97	323.99	323.99
323.90	323.96	323.96	323.96	323.94
323.84	323.95	323.95	323.92	323.89
324.07	323.96	323.96	323.97	323.98
324.14	323.97	323.98	324.02	324.06
324.02	323.97	323.98	324.02	324.04
324.13	323.98	324.00	324.05	324.09
323.78	323.97	323.98	323.97	323.93
323.73	323.96	323.95	323.90	323.83
323.71	323.95	323.93	323.84	323.77
323.83	323.94	323.92	323.84	323.80
323.89	323.94	323.92	323.86	323.85
323.92	323.94	323.92	323.87	323.88
324.14	323.95	323.94	323.96	324.01
324.20	323.96	323.97	324.03	324.11
324.10	323.97	323.98	324.05	324.11
323.95	323.97	323.98	324.02	324.03
323.94	323.97	323.97	324.00	323.99
323.95	323.96	323.97	323.98	323.97
323.98	323.96	323.97	323.98	323.97
323.88	323.96	323.96	323.95	323.93
324.11	323.97	323.98	324.00	324.02
323.88	323.96	323.97	323.96	323.95
324.26	323.98	324.00	324.05	324.10
324.08	323.98	324.01	324.06	324.09
324.34	324.00	324.04	324.14	324.22

323.90	324.00	324.02	324.07	324.06
323.74	323.98	324.00	323.97	323.90
323.93	323.98	323.99	323.96	323.92
323.88	323.98	323.98	323.94	323.90
324.38	324.00	324.02	324.07	324.14
324.34	324.01	324.05	324.15	324.24
324.03	324.01	324.05	324.12	324.14
323.88	324.01	324.03	324.05	324.01
323.93	324.00	324.02	324.01	323.97
323.68	323.99	323.99	323.91	323.82
323.91	323.98	323.98	323.91	323.86
324.12	323.99	323.99	323.97	323.99
324.19	324.00	324.01	324.04	324.09
323.93	324.00	324.01	324.01	324.01

Tabel 4. 9 Percobaan 3

X(n)	α 1	α 2	α 3	α 4
323.47	323.85	323.81	323.58	323.50
323.72	323.84	323.80	323.62	323.61
323.61	323.83	323.78	323.62	323.61
323.41	323.81	323.74	323.55	323.51
323.18	323.78	323.69	323.44	323.34
323.24	323.75	323.64	323.38	323.29
323.53	323.74	323.63	323.43	323.41
323.46	323.73	323.62	323.44	323.44
323.31	323.71	323.58	323.40	323.37
323.79	323.71	323.60	323.51	323.58
323.17	323.68	323.56	323.41	323.37

323.07	323.65	323.51	323.31	323.22
323.53	323.65	323.51	323.37	323.38
323.73	323.65	323.53	323.48	323.55
323.46	323.64	323.53	323.47	323.50
323.64	323.64	323.54	323.52	323.57
323.61	323.64	323.55	323.55	323.59
323.87	323.65	323.58	323.65	323.73
323.84	323.66	323.60	323.70	323.79
323.61	323.66	323.61	323.68	323.70
323.65	323.66	323.61	323.67	323.67
323.45	323.65	323.59	323.60	323.56
323.43	323.64	323.58	323.55	323.50
323.14	323.61	323.53	323.43	323.32
323.33	323.60	323.51	323.40	323.32
323.46	323.59	323.51	323.42	323.39
323.38	323.58	323.49	323.41	323.39
323.44	323.57	323.49	323.41	323.41
323.73	323.58	323.51	323.51	323.57
323.71	323.59	323.53	323.57	323.64
324.08	323.61	323.59	323.72	323.86
323.84	323.62	323.61	323.76	323.85
324.49	323.67	323.70	323.98	324.17
324.14	323.69	323.74	324.03	324.15
324.12	323.71	323.78	324.05	324.14
324.13	323.73	323.82	324.07	324.13
323.86	323.74	323.82	324.01	323.99
324.38	323.77	323.88	324.12	324.19

324.05	323.78	323.89	324.10	324.12
323.95	323.79	323.90	324.05	324.03
323.71	323.79	323.88	323.95	323.87
324.11	323.80	323.90	324.00	323.99
323.92	323.81	323.91	323.98	323.96
324.29	323.83	323.94	324.07	324.12
323.79	323.83	323.93	323.99	323.96
324.06	323.84	323.94	324.01	324.01
324.43	323.87	323.99	324.13	324.22
324.55	323.91	324.05	324.26	324.38
324.38	323.93	324.08	324.30	324.38
324.00	323.93	324.07	324.21	324.19
324.09	323.94	324.07	324.17	324.14
324.31	323.96	324.10	324.21	324.23
324.01	323.96	324.09	324.15	324.12
323.72	323.95	324.05	324.02	323.92
324.03	323.95	324.05	324.02	323.97
323.71	323.94	324.02	323.93	323.84
323.92	323.94	324.01	323.93	323.88
324.31	323.96	324.04	324.04	324.10
324.03	323.96	324.04	324.04	324.06
323.97	323.96	324.03	324.02	324.02
324.20	323.98	324.05	324.07	324.11
324.20	323.99	324.06	324.11	324.15
324.40	324.01	324.10	324.20	324.28
324.63	324.04	324.15	324.33	324.45
324.02	324.04	324.14	324.23	324.24

324.22	324.05	324.14	324.23	324.23
323.80	324.03	324.11	324.10	324.01
324.52	324.06	324.15	324.23	324.26
324.26	324.07	324.16	324.23	324.26
324.35	324.08	324.18	324.27	324.30
324.11	324.08	324.17	324.22	324.21
324.42	324.10	324.20	324.28	324.32
324.06	324.10	324.18	324.22	324.19
324.72	324.13	324.24	324.37	324.46
324.58	324.15	324.27	324.43	324.52
324.46	324.17	324.29	324.44	324.49
324.59	324.19	324.32	324.49	324.54
324.13	324.19	324.30	324.38	324.34
324.44	324.20	324.32	324.40	324.39
324.38	324.21	324.32	324.39	324.38
324.13	324.20	324.30	324.31	324.26
324.16	324.20	324.29	324.27	324.21
324.29	324.21	324.29	324.27	324.25
324.74	324.23	324.33	324.41	324.49
324.72	324.26	324.37	324.51	324.61
324.44	324.27	324.38	324.49	324.52
324.26	324.27	324.37	324.42	324.39
324.47	324.28	324.38	324.43	324.43
323.98	324.26	324.34	324.30	324.21
324.64	324.28	324.37	324.40	324.42
324.25	324.28	324.36	324.35	324.33
324.40	324.28	324.36	324.37	324.37

324.33	324.29	324.36	324.36	324.35
324.45	324.29	324.37	324.38	324.40
324.56	324.31	324.39	324.44	324.48
324.34	324.31	324.38	324.41	324.41
324.11	324.30	324.35	324.32	324.26
324.39	324.30	324.36	324.34	324.33
324.25	324.30	324.35	324.31	324.29
323.97	324.28	324.31	324.21	324.13
323.93	324.27	324.27	324.13	324.03
323.65	324.24	324.21	323.98	323.84
324.04	324.23	324.19	324.00	323.94
323.85	324.21	324.16	323.96	323.90
323.48	324.17	324.09	323.81	323.69
323.68	324.15	324.05	323.77	323.68
323.81	324.13	324.03	323.78	323.75
323.44	324.10	323.97	323.68	323.59
324.04	324.09	323.97	323.79	323.82
323.51	324.06	323.93	323.70	323.66
323.29	324.02	323.86	323.58	323.47
323.25	323.99	323.80	323.48	323.36
323.04	323.94	323.73	323.35	323.20
323.60	323.92	323.71	323.42	323.40
323.46	323.90	323.69	323.43	323.43
323.73	323.89	323.69	323.52	323.58
323.65	323.88	323.69	323.56	323.61
323.28	323.85	323.65	323.48	323.45
323.60	323.83	323.64	323.51	323.52

323.42	323.81	323.62	323.48	323.47
323.53	323.80	323.61	323.50	323.50
323.52	323.79	323.60	323.50	323.51
323.16	323.75	323.56	323.40	323.33
323.32	323.73	323.53	323.38	323.33
323.60	323.73	323.54	323.44	323.46
323.47	323.71	323.53	323.45	323.47
323.32	323.69	323.51	323.41	323.40
323.69	323.69	323.53	323.50	323.55
323.81	323.70	323.56	323.59	323.68
323.32	323.68	323.53	323.51	323.50
323.65	323.68	323.55	323.55	323.58
323.30	323.66	323.52	323.48	323.44
324.01	323.68	323.57	323.64	323.73
323.64	323.68	323.58	323.64	323.68
324.21	323.70	323.64	323.81	323.95
324.08	323.72	323.69	323.89	324.01
324.29	323.75	323.75	324.01	324.15
324.03	323.76	323.77	324.02	324.09
324.44	323.80	323.84	324.14	324.26
323.81	323.80	323.84	324.04	324.04
324.28	323.82	323.88	324.11	324.16
324.28	323.85	323.92	324.17	324.22
324.11	323.86	323.94	324.15	324.17
324.71	323.90	324.02	324.32	324.44
324.06	323.91	324.02	324.24	324.25
324.03	323.92	324.02	324.18	324.14

324.31	323.93	324.05	324.22	324.22
324.14	323.94	324.06	324.19	324.18
324.10	323.95	324.06	324.16	324.14
324.69	323.99	324.13	324.32	324.41
324.21	324.00	324.13	324.29	324.31
324.25	324.01	324.15	324.28	324.28
324.35	324.03	324.17	324.30	324.32
324.40	324.05	324.19	324.33	324.36
323.98	324.04	324.17	324.22	324.17
324.38	324.06	324.19	324.27	324.27
324.34	324.08	324.20	324.29	324.31
323.97	324.07	324.18	324.19	324.14
324.41	324.09	324.20	324.26	324.28
324.37	324.10	324.22	324.29	324.32
324.20	324.11	324.22	324.26	324.26
324.39	324.12	324.24	324.30	324.33
324.96	324.16	324.31	324.50	324.64
324.50	324.18	324.33	324.50	324.57
324.56	324.20	324.35	324.52	324.56
324.79	324.23	324.39	324.60	324.68
324.37	324.23	324.39	324.53	324.52
324.27	324.24	324.38	324.45	324.40

b. Pola Data Awal (nilai berat aktual)

Data X(n) menunjukkan variasi nilai yang cukup sering dan turun secara acak meskipun beban tidak berubah. Hal ini terlihat dari banyaknya perubahan nilai

antar waktu yang tidak memiliki pola pasti. Fluktuasi ini menjadi indikasi bahwa hasil pembacaan sensor masih terpengaruh oleh noise.

c. Evaluasi Kinerja Filter

1) Perbandingan numerik

Data pembacaan sensor dievaluasi dengan membandingkan nilai mentah (Xn) terhadap hasil *smoothing* mengguakan metode *exponential smoothing* debgan empat variasi nilai alpha (α1, α2, α3, α4).Setiap data X(n) diproroses menggunakan rumus smoothing untuk memperoleh respons filter terhadap sinyal input.

ketika data X(n) mengalami *impuls noise* atau perubahan mendadak maka data Y(n) atau filter mengikuti perubahan berdasarkan masing-masing alpha nya, dapat dilihat dari potongan data yang mengalami *impuls noise* ;

Tabel 4. 10 Potongan tabel percobaan 1 Load Cell

X(n)	α 1	α 2	α 3	α 4
323.40	323.02	323.24	323.22	323.22
323.47	323.04	323.26	323.29	323.34
323.20	323.05	323.25	323.27	323.27
323.24	323.06	323.25	323.26	323.26
323.25	323.07	323.25	323.26	323.25

→ α 4 naik mendekati X(n)

Tabel 4. 11 Potongan tabel percobaan 2 Load Cell

X(n)	α 1	α 2	α 3	α 4
324.10	323.97	323.98	324.05	324.11
323.95	323.97	323.98	324.02	324.03
323.94	323.97	323.97	324.00	323.99
323.95	323.96	323.97	323.98	323.97
323.98	323.96	323.97	323.98	323.97

→ α 4 naik mendekati X(n)

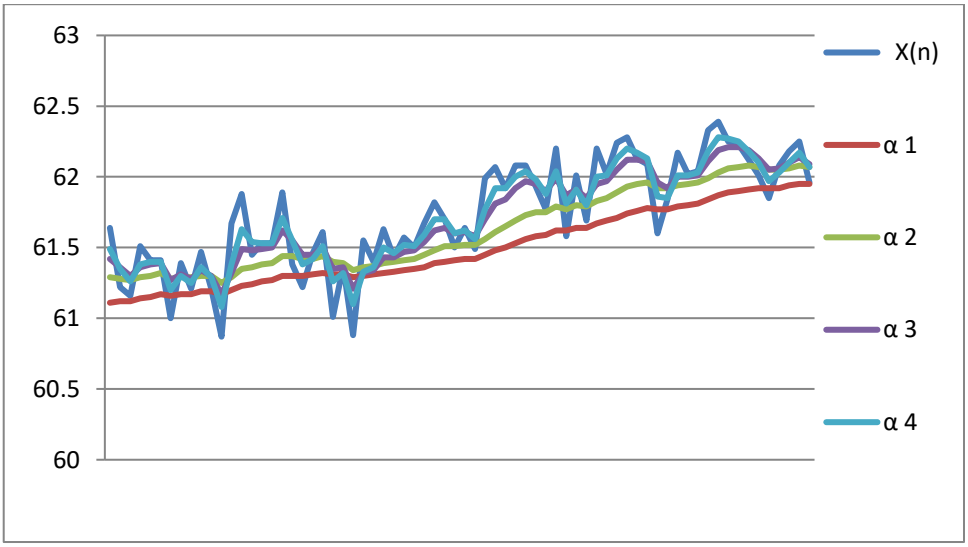
Tabel 4. 12 Percobaan tabel percobaan 3 Load Cell

X(n)	α 1	α 2	α 3	α 4
323.61	323.83	323.78	323.62	323.61
323.41	323.81	323.74	323.55	323.51
323.18	323.78	323.69	323.44	323.34
323.24	323.75	323.64	323.38	323.29
323.53	323.74	323.63	323.43	323.41

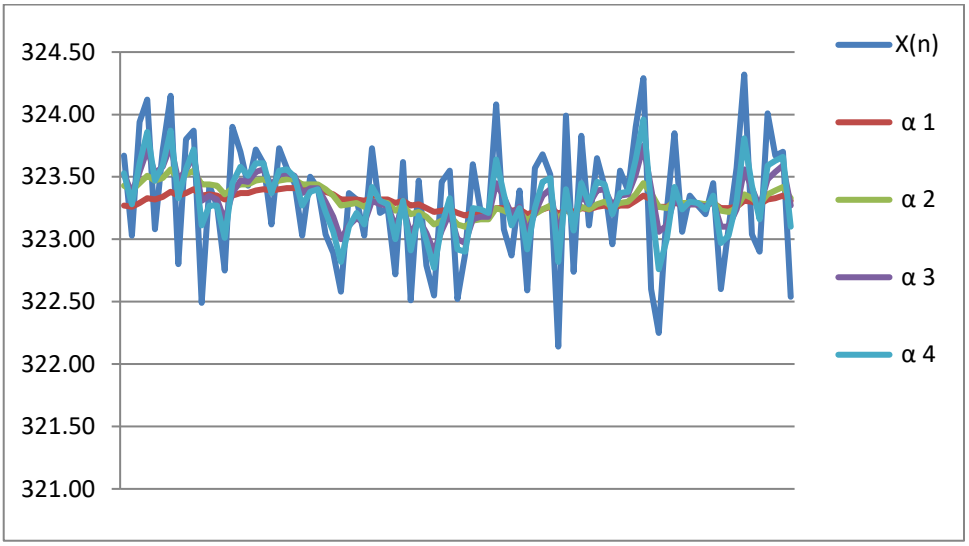
→ α 4 turun mendekati X(n)

Nilai Y(n) pada alpha 1 (0.05) naik atau turun sedikit saat nilai X(n) mengalami perubahan mendadak, sebaliknya alpha 4 (0.50) mengalami perubahan drastis hampir mendekati nilai X(n).

2) Perbandingan grafik



Gambar 4. 8 Grafik pada sampel beban lain



Gambar 4. 9 grafik pada sampel beban penelitian

Pada kedua grafik diatas menunjukkan konsistensi filter dalam memproses *noise* tidak berubah.dapat di simpulkan bahwa adanya perubahan berat beban tidak mempengaruhi performa filter dalam meredam fluktuasi.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil perancangan, inplementasi, dan pengujian, maka diperoleh beberapa kesimpulan sebagai berikut :

1. Filter adaptif berbasis metode *Exponential Smoothing* berhasil meningkatkan kualitas dan stabilitas pembacaan sensor LM35 dan Load Cell.
2. Penggunaan variasi nilai *alpha* memberikan fleksibilitas dalam merespons perubahan data. *alpha* kecil (0,05 dan 0,10) menghasilkan pembacaan yang lebih halus dan stabil, sementara alpha besar (0,30 dan 0,50) lebih responsif terhadap perubahan tetapi dengan kestabilan yang lebih rendah (fluktuasi 0,31–0,42 °C).
3. Pada pengujian sensor LM35, data mentah berfluktuasi antara 27,34–27,83 °C (selisih 0,49 °C). Setelah diterapkan filter dengan alpha 0,05, fluktuasi berkurang menjadi hanya 0,07 °C, sedangkan alpha 0,10 menghasilkan fluktuasi 0,11 °C. Hal ini menunjukkan peningkatan kestabilan pembacaan lebih dari 80% dibandingkan data mentah.

B. Saran

Untuk pengembangan dan penelitian lebih lanjut, berikut beberapa saran yang dapat dipertimbangkan:

1. Penelitian dapat diperluas dengan menerapkan algoritma filter adaptif lainnya seperti Kalman Filter atau LMS (Least Mean Square) untuk dibandingkan efektivitasnya.
2. Perlu dilakukan pengujian dalam berbagai kondisi lingkungan yang lebih ekstrem untuk menguji ketahanan sistem filter.
3. Integrasi sistem ke dalam aplikasi berbasis Internet of Things (IoT) dapat meningkatkan pemanfaatannya dalam pengawasan jarak jauh secara real-time.

DAFTAR PUSTAKA

Ahmad, M., Rahman, A., & Latif, S. (2021). ANN-based filtering for load cell sensors. International Journal of Electrical and Computer Engineering, 11(5), 4454–4461.

Arga, B. (2020). Penggunaan Arduino dalam prototyping sistem otomasi. Jurnal Elektro dan Telekomunikasi, 4(1), 45–50.

Aulia, D., Santoso, R., & Pratama, M. (2021). Rancang bangun sistem monitoring suhu dan kelembaban berbasis Arduino Uno dan sensor DHT11. Jurnal Teknologi dan Sistem Komputer, 9(2), 87–94.

Avtar, R., & Singh, K. (2020). Load cell interfacing using HX711 with Arduino. International Journal of Innovative Research in Technology, 7(6), 152–158.

International Society of Automation. (2019). Reducing process errors through accurate sensor readings. Research Triangle Park, NC: ISA.

Kusuma, D., & Wijaya, A. (2022). Implementasi filter Kalman untuk sensor LM35 dalam monitoring suhu berbasis IoT. Jurnal Teknologi dan Sistem Komputer, 10(1), 23–30.

Maolana Sunarya, R. (2011). Perancangan modul penguat HX711 . Bandung: Universitas Komputer Indonesia.

Natarajan, S. (2018). Introduction to load cell technology. Singapore: Springer.

National Institute of Standards and Technology. (2017). The economic impact of inaccurate sensor measurements. Gaithersburg, MD: U.S. Department of Commerce.

Smith, J., Chen, Y., & Davis, K. (2018). Adaptive LMS filtering for industrial temperature sensors. IEEE Transactions on Instrumentation and Measurement, 67(4), 904–911.

Texas Instruments. (2020). LM35 precision centigrade temperature sensors (Rev. S). Diperoleh dari:https://www.ti.com/lit/ds/symlink/lm35.pdf.

Wikipedia. (2015) . Microcontroller. Diperoleh dari:https://en.wikiped ia.org/wiki/ Micr ocont roller.

Zhang, L., & Lee, T. H. (2020). Recursive least squares filtering in medical pressure sensors. Sensors, 20(3), 920–929.