

Sistem Proteksi Aktif Berbasis Logika OR Perangkat Lunak Untuk Mencegah Overheat Dan Overcurrent Pada Power Amplifier Kelas D

Gaguk Firasanto^{1*}, Andriani¹

^{1,2}Teknik Elektro, Universitas Pamulang, Tangerang Selatan
*dosen02634@unpam.ac.id

Abstrak— Perangkat penguat daya audio (*Power Amplifier*) kelas D memiliki efisiensi tinggi namun rentan mengalami kerusakan fatal akibat panas berlebih (*overheat*) dan arus pendek (*short circuit*) pada beban speaker. Sistem proteksi konvensional yang mengandalkan sekering (*fuse*) dinilai memiliki respon lambat, bersifat pasif, dan tidak menyediakan data historis untuk analisis pasca-kegagalan. Penelitian ini bertujuan merancang sistem proteksi aktif yang cerdas dan informatif berbasis mikrokontroler Arduino Nano. Sistem dirancang mengintegrasikan sensor LM35 untuk mendeteksi suhu *heatsink* dan sensor INA219 untuk memantau arus beban secara digital via protokol I2C. Untuk mengatasi fluktuasi sinyal sensor (*noise*), diterapkan metode *hybrid filtering* yang menggabungkan resistor *pull-down* 10k Ω pada perangkat keras dan filter rata-rata bergerak (*Moving Average Filter*) pada perangkat lunak. Pengambilan keputusan pemutusan arus dilakukan menggunakan Gerbang Logika OR pada antarmuka Python. Hasil pengujian menunjukkan sistem mampu memutus arus (*cut-off*) secara otomatis saat suhu *heatsink* melampaui 60°C atau arus beban melebihi 2.0 A. Berdasarkan hasil kalibrasi, sensor suhu LM35 memiliki rata-rata kesalahan (*error*) sebesar 0.35% dan sensor arus INA219 memiliki rata-rata kesalahan sebesar 1.29% dibandingkan alat ukur standar. Sistem proteksi ini terbukti efektif mencegah kerusakan komponen amplifier akibat panas berlebih dan lonjakan arus.

Kata Kunci — *Arduino Nano, Data Logger, Hybrid Filtering, Power Amplifier, Sistem Proteksi.*

DOI: 10.22441/jte.2026.v17i2.006

I. PENDAHULUAN

Penguat daya audio (*Power Amplifier*), khususnya topologi Kelas D, kini mendominasi sistem tata suara modern karena efisiensinya yang mampu mencapai lebih dari 90% [1]. Namun, efisiensi tinggi ini dibarengi dengan kompleksitas pada tahap *switching* kecepatan tinggi yang melibatkan komponen semikonduktor MOSFET. Kerentanan utama perangkat ini terletak pada risiko operasi di luar batas aman (*Safe Operating Area*), di mana panas berlebih (*overheat*) akibat ventilasi yang buruk [2], [3] atau lonjakan arus akibat hubungan singkat pada beban *speaker* [4], [5] dapat menyebabkan kerusakan permanen pada modul daya dalam hitungan milidetik. Kegagalan pada satu komponen sering kali berdampak domino pada keseluruhan sistem, yang pada akhirnya meningkatkan biaya perawatan dan risiko kerusakan beban *speaker* yang mahal.

Secara teknis, proteksi pada *amplifier* Kelas D sering kali gagal bukan karena ketiadaan fitur keamanan, melainkan karena keterbatasan sirkuit proteksi analog tradisional dalam menangani impedansi beban *speaker* yang bersifat reaktif dan kompleks. Proteksi konvensional yang mengandalkan sekering (*fuse*), saklar termal bimetal, atau sirkuit transistor diskrit [6] memiliki kelemahan mendasar pada waktu respon yang lambat (*thermal delay*) dan sifatnya yang pasif. Penggunaan sekering elektronik telah dikaji oleh Ramadhan dan Sukir [7], yang menunjukkan bahwa sekering konvensional sering kali terlambat merespon lonjakan arus *transient* yang sangat singkat namun mematikan bagi transistor *output*. Selain itu, ketiadaan data historis pada sistem konvensional membuat teknisi sulit melakukan analisis pasca-kegagalan untuk menentukan apakah kerusakan disebabkan oleh faktor beban, lonjakan catu daya [8], atau kelemahan internal perangkat.

Berbagai penelitian telah mencoba mengembangkan sistem proteksi aktif, namun umumnya masih berfokus pada parameter tunggal. Iskandar dan Wijaya [2] melakukan optimasi sistem pendinginan menggunakan PWM untuk mencegah *thermal runaway*, namun sistem ini bersifat preventif tanpa pemutusan arus jika kegagalan termal sudah terjadi. Santoso [9] mengembangkan sistem pemantauan suhu nirkabel menggunakan sensor MLX90614, namun memiliki keterbatasan dalam respon pemutusan beban secara *real-time* saat kondisi kritis. Dalam aspek proteksi arus, Kim [10] merancang sirkuit *overcurrent protection* (OCP) terintegrasi yang cepat, tetapi memiliki ambang batas (*threshold*) tetap yang tidak dapat dikustomisasi oleh pengguna. Di sisi lain, Al-Farizi [11] serta Raj dan Sujith [12] menggunakan mikrokontroler sebagai relai proteksi tegangan dan arus, namun mengabaikan parameter suhu *heatsink* yang merupakan faktor risiko utama pada *amplifier* Kelas D yang bekerja secara kontinu.

Penerapan sensor digital melalui protokol komunikasi I2C menjadi salah satu solusi untuk meningkatkan presisi pemantauan arus dan tegangan tanpa membebani pin analog mikrokontroler secara berlebihan. Sensor INA219 [13], misalnya, menawarkan resolusi tinggi dan kemampuan pemrosesan sinyal digital internal yang dapat meminimalisir kesalahan pembacaan akibat gangguan elektromagnetik (EMI) yang sering dihasilkan oleh sirkuit *switching amplifier* Kelas D. Sinergi antara perangkat keras dan algoritma pemrosesan sinyal digital diperlukan untuk memastikan sistem proteksi tidak

bersifat "sensitif berlebihan" (*false triggering*) terhadap dinamika sinyal audio, namun tetap responsif terhadap kondisi anomali yang nyata. Penyediaan data historis (*data logging*) melalui mekanisme ini menjadi sangat krusial untuk evaluasi sistem secara mendalam, sebagaimana dibuktikan oleh Prasetyo [14] dan Jadin dkk. [15] dalam efektivitas pencatatan data untuk pemantauan sistem jarak jauh.

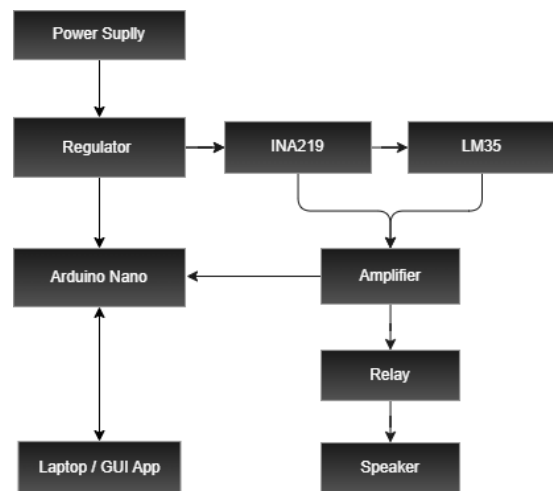
Berdasarkan berbagai literatur tersebut, terdapat celah penelitian (*research gap*) di mana sistem proteksi yang ada umumnya memiliki ambang batas yang kaku atau tidak menyediakan fitur penyimpanan data kegagalan yang komprehensif untuk analisis teknis audio profesional. Untuk mengatasi hal tersebut, penelitian ini merancang sistem proteksi aktif yang cerdas berbasis mikrokontroler Arduino Nano [16]. Kebaruan (*novelty*) dari penelitian ini terletak pada integrasi pemantauan suhu menggunakan sensor LM35 [17] dan arus beban secara digital via INA219 [13], [18] dengan penerapan logika OR berbasis perangkat lunak (*software-based logic*). Kontribusi utama penelitian ini adalah fleksibilitas penyesuaian parameter proteksi melalui PC *host* dan penggunaan metode *Hybrid Filtering* untuk memastikan stabilitas sistem. Integrasi *Data Logger* juga memungkinkan perekaman kondisi teknis sesaat sebelum kegagalan terjadi, memberikan dimensi informatif yang tidak ditemukan pada sistem proteksi konvensional maupun penelitian terdahulu.

Selain aspek fungsionalitas perangkat keras, penelitian ini juga menekankan pada pentingnya sinergi antara kontrol otomatis dengan antarmuka pengguna berbasis grafis (GUI) yang dibangun menggunakan bahasa pemrograman Python [18]. Pendekatan ini memungkinkan visualisasi data suhu dan arus secara dinamis, sehingga memudahkan pengguna dalam melakukan diagnosis awal sebelum terjadi kegagalan sistem yang lebih serius. Dengan optimasi waktu respons yang dijaga pada rentang 1,5 hingga 2,5 detik melalui algoritma filter rata-rata bergerak, sistem mampu membedakan antara dinamika sinyal audio sesaat dengan kondisi anomali yang berbahaya secara presisi. Integrasi antarmuka yang informatif ini tidak hanya meningkatkan faktor keamanan perangkat, tetapi juga memberikan fleksibilitas dalam proses kalibrasi ambang batas (*threshold*) yang dapat disesuaikan secara *real-time*, menjadikannya solusi yang adaptif untuk berbagai varian penguat daya audio profesional di masa depan.

II. METODOLOGI PENELITIAN

A. Diagram Blok Sistem

Sistem proteksi ini dirancang dengan arsitektur kendali *closed-loop* yang memisahkan jalur daya instrumen dengan jalur daya kontrol utama. Secara visual, aliran data dan distribusi tegangan pada sistem digambarkan pada Diagram Blok berikut.



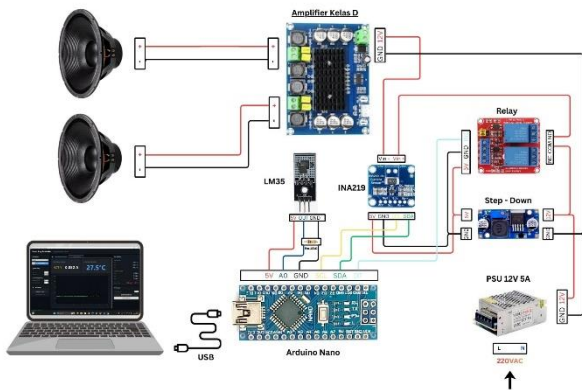
Gambar 6. Diagram Blok Sistem

Berdasarkan diagram di atas, sistem bekerja dalam dua domain terpisah yang saling terintegrasi. Pada domain perangkat keras, sumber daya utama didistribusikan menjadi dua jalur: jalur tegangan tinggi menuju Amplifier yang melewati kontak Modul Relay, dan jalur regulasi yang masuk ke *Step-Down Converter* untuk menyuplai tegangan 5V stabil bagi sensor dan koil relay. Konfigurasi ini memastikan bahwa fluktuasi beban pada amplifier tidak mempengaruhi kestabilan pembacaan sensor.

Pada domain data dan kendali, Arduino Nano yang ditenagai langsung melalui *port* USB komputer bertindak sebagai jembatan komunikasi (*interface*). Sensor Suhu LM35 dan Sensor Arus INA219 secara berkelanjutan memantau parameter fisik amplifier dan mengirimkan data tersebut ke mikrokontroler. Melalui protokol serial, data diteruskan ke komputer untuk diolah oleh algoritma Python. Apabila perangkat lunak mendeteksi parameter yang melampaui ambang batas aman (Suhu > 60°C atau Arus > 2.0A), sistem secara otomatis mengirimkan sinyal umpan balik (*feedback*) ke Arduino untuk memicu Relay agar memutus aliran daya utama, sehingga amplifier terlindungi dari kerusakan fatal.

B. Perancangan Perangkat Keras (Hardware Design)

Perancangan perangkat keras mencakup integrasi antara sumber daya, unit pemroses, dan instrumen pengukuran. Skema rangkaian dirancang secara modular dengan memisahkan sumber daya mikrokontroler dan beban sensor untuk menjamin keamanan perangkat komputer dari potensi konflik tegangan. Skema keseluruhan rangkaian elektronika sistem ditunjukkan pada Gambar Perancangan Perangkat Keras (Hardware Design) berikut.



Gambar 7. Perancangan Perangkat Keras

Unit Catu Daya (*Power Supply Unit*) Sumber daya utama sistem berasal dari transformator 5A yang disearahkan menjadi tegangan DC. Tegangan utama ini dibagi menjadi dua jalur distribusi. Jalur pertama menyuplai daya beban ke Amplifier melalui kontak Relay. Jalur kedua masuk ke modul *Step-Down Converter* LM2596 untuk menurunkan tegangan menjadi 5V stabil. Tegangan 5V eksternal ini digunakan khusus untuk menyuplai daya ke sensor dan koil Relay agar beban arus komponen tersebut tidak membebani *port* USB komputer.

Mikrokontroler (Arduino Nano) Arduino Nano beroperasi menggunakan daya 5V yang bersumber langsung dari sambungan kabel USB komputer. Mengingat sistem menggunakan protokol *StandardFirmata* yang mengharuskan koneksi data terus-menerus, penggunaan daya USB merupakan opsi paling aman untuk menghindari arus balik (*back-feeding*) yang dapat merusak *port* USB. Meskipun sumber daya terpisah, terminal *Ground* (GND) Arduino tetap dihubungkan dengan *Ground* Step-Down (*Common Ground*) sebagai referensi sinyal data.

Rangkaian Sensor Suhu LM35 Sensor suhu LM35 diposisikan menempel pada pelat pendingin (*heatsink*) IC TPA3116D2. Pin VCC sensor dihubungkan ke positif 5V, sedangkan pin GND ke jalur *Common Ground*. Pin sinyal keluaran dihubungkan ke pin Analog A0 Arduino. Resistor *pull-down* 10k Ω dipasang secara paralel antara kaki sinyal dan *ground* untuk menghilangkan tegangan ambang (*floating*) dan menstabilkan pembacaan suhu.

Rangkaian Sensor Arus INA219 Sensor INA219 dipasang secara seri (*High-Side*) pada jalur positif catu daya amplifier. Pin VCC modul ini juga mengambil daya dari 5V *Step-Down*. Jalur komunikasi data I2C, yaitu SDA dan SCL, masing-masing dihubungkan ke pin A4 dan A5 pada Arduino. Penggunaan antarmuka digital I2C dipilih karena memiliki akurasi yang lebih tinggi dan tahan terhadap penurunan tegangan (*voltage drop*) dibandingkan pengukuran analog.

Aktuator Proteksi (Relay) Modul Relay 1-Channel disuplai oleh tegangan 5V dari *Step-Down*. Sinyal pemicu relay

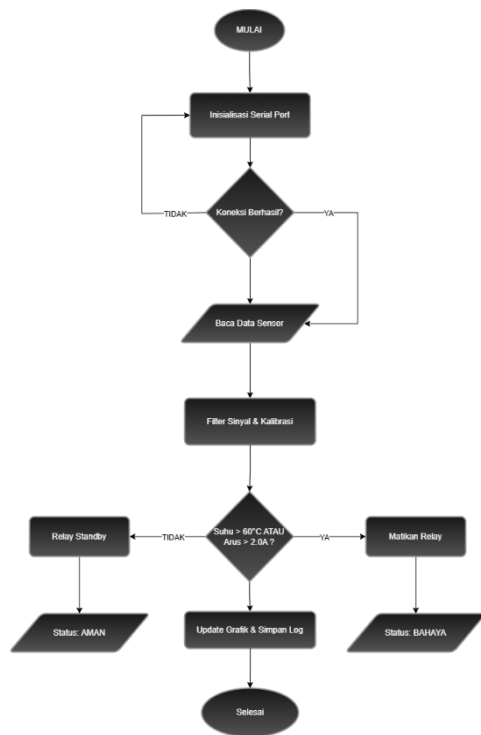
dihubungkan ke pin Digital D7 Arduino Nano. Sistem menggunakan logika *Active Low*, di mana relay akan aktif menyambungkan arus saat diberi logika '0' dan memutus arus saat logika '1'. Konfigurasi ini juga berfungsi sebagai pengaman pasif (*fail-safe*), di mana relay akan otomatis terbuka (memutus daya amplifier) jika kabel USB atau Arduino kehilangan daya.

C. Perancangan Perangkat Lunak

Perangkat lunak dikembangkan menggunakan bahasa pemrograman Python 3.11 yang dieksekusi pada komputer *host*. Pemilihan Python didasarkan pada ketersediaan pustaka (*library*) yang luas untuk keperluan teknis. Dalam pengembangan ini, pustaka Tkinter digunakan untuk membangun antarmuka grafis (*GUI*), sementara Matplotlib diimplementasikan untuk memvisualisasikan data sensor ke dalam bentuk grafik dinamis. Aplikasi ini berfungsi sebagai pusat kendali (*Master Controller*) yang tidak hanya menerima data telemetri dari mikrokontroler melalui protokol komunikasi serial, tetapi juga bertindak sebagai pengambil keputusan proteksi secara otomatis berdasarkan algoritma yang telah ditanamkan.

Komunikasi Data (Protokol Firmata) Interaksi dwiarah antara perangkat keras (Arduino) dan perangkat lunak (Python) dijemput oleh pustaka PyFirmata. Pada sisi mikrokontroler, ditanamkan *firmware StandardFirmata* yang mengubah fungsi Arduino menjadi *client* pasif, memungkinkan seluruh pin Input/Output (I/O) dikendalikan sepenuhnya oleh skrip Python dari komputer. Pendekatan ini secara efektif memindahkan beban komputasi berat dari mikrokontroler ke prosesor komputer, sehingga memungkinkan pemrosesan data *floating-point* yang lebih presisi dan visualisasi grafik yang responsif tanpa membebani memori Arduino.

Alur Algoritma Sistem Logika pemrograman dirancang dengan struktur infinite loop (perulangan tak terbatas) untuk memastikan pemantauan yang berkelanjutan. Alur dimulai dengan inisialisasi port serial dan sinkronisasi baudrate. Setelah koneksi stabil, sistem melakukan pencuplikan (*sampling*) nilai sensor Suhu (Analog A0) dan sensor Arus (I2C) dengan interval waktu 500 milidetik. Data mentah yang masuk kemudian diproses menggunakan filter rata-rata bergerak (*Moving Average Filter*) dengan jendela sampel $N=10$. Filter ini bertujuan meredam fluktuasi sinyal (*noise*) parasitik agar data yang ditampilkan pada grafik GUI lebih halus dan akurat. Selanjutnya, data yang telah bersih dibandingkan dengan nilai ambang batas (*threshold*); jika parameter melampaui batas aman, sistem seketika mengirimkan instruksi digital untuk memutus relay. Tindakan pemutusan ini juga diikuti dengan pembaruan status visual pada antarmuka pengguna secara seketika. Bersamaan dengan hal tersebut, sistem akan menyimpan riwayat parameter anomali beserta informasi waktu kejadian ke dalam bentuk berkas log agar dapat digunakan sebagai bahan evaluasi perbaikan perangkat ke depannya. Pendekatan terintegrasi ini secara signifikan meningkatkan keandalan operasional penguat daya, sekaligus mempermudah pengguna dalam melakukan pemeliharaan sistem audio secara berkala.



Gambar 8. Diagram Alir (Flowchart) Logika Proteksi

Teknik Kalibrasi Perangkat Lunak Mengingat sensor analog memiliki toleransi akurasi, sistem menerapkan metode kalibrasi satu titik (*One-Point Calibration*) pada perangkat lunak. Nilai pembacaan sensor dikoreksi dengan menambahkan nilai *offset* yang didapat dari perbandingan dengan alat ukur standar. Proses ini dilakukan bersamaan dengan *filtering* sinyal sebelum data masuk ke logika pengambilan keputusan.

Implementasi Gerbang Logika (Logic Gate) Sistem menerapkan prinsip Gerbang Logika OR untuk memicu mekanisme proteksi. Variabel keputusan (Y) akan bernilai '1' (Bahaya/Cut-Off) jika salah satu dari dua kondisi terpenuhi: Suhu (T) melebihi batas 60°C ATAU Arus (I) melebihi batas 2.0A. Persamaan logika tersebut diformulasikan sebagai berikut:

$$Y = (T > 60) \vee (I > 2.0)$$

Pemilihan nilai ambang batas (*threshold*) pada sistem ini didasarkan pada spesifikasi komponen dan pertimbangan keamanan preventif. Merujuk pada *datasheet* IC TPA3116D2, perangkat sebenarnya memiliki fitur *thermal shutdown* internal pada suhu 150°C. Namun, untuk mencegah kelelahan termal (*thermal fatigue*) pada komponen di sekitar PCB (seperti kapasitor elco) akibat paparan panas berulang, batas suhu eksternal pada *heatsink* ditetapkan pada 60°C sebagai tindakan

preventif dini. Sementara itu, batas arus 2.0A dipilih untuk menyesuaikan spesifikasi catu daya dan karakteristik beban uji, guna memastikan sistem memutuskan daya sebelum lonjakan arus berpotensi merusak kumparan *speaker* atau mengakibatkan penurunan tegangan (*drop voltage*) ekstrem pada catu daya utama.

Jika kondisi Y bernilai True, aplikasi Python mengirimkan sinyal digital 'HIGH' ke pin Relay untuk memutuskan sirkuit beban, dan secara otomatis mencatat waktu kejadian (timestamp) ke dalam berkas log (CSV) untuk keperluan evaluasi.

Terkait implementasi logika, sistem menggunakan pernyataan kondisional (if/or) dalam skrip Python, yang merupakan logika berbasis perangkat lunak (software-based logic), bukan komponen perangkat keras fisik. Untuk mencegah fenomena chattering (relai berkedip cepat) saat parameter berfluktuasi di titik ambang batas, sistem menerapkan hysteresis band. Sebagai contoh, jika relai memutuskan arus pada suhu 60°C, pemulihan (recovery) relai ditunda hingga suhu *heatsink* turun mencapai 50°C.

Selain itu, terdapat batasan operasional berupa ketergantungan penuh pada PC host. Karena seluruh logika pengambilan keputusan dan perekaman data dieksekusi oleh skrip Python di komputer, sistem mewajibkan PC dalam keadaan terus menyala. Jika koneksi serial terputus atau aplikasi pada PC terhenti, mekanisme proteksi tidak dapat dieksekusi meskipun mikrokontroler dalam keadaan aktif.

III. HASIL DAN PEMBAHASAN

Bab ini memaparkan hasil implementasi perangkat keras dan perangkat lunak, serta analisis data pengujian yang meliputi akurasi pembacaan sensor dan respon sistem proteksi terhadap gangguan.

A. Implementasi Alat

Tahap implementasi merupakan proses integrasi seluruh komponen perangkat keras dan perangkat lunak yang telah dirancang menjadi satu kesatuan sistem yang fungsional. Pada sisi perangkat keras, modul mikrokontroler Arduino Nano, sensor arus INA219, sensor suhu LM35, dan modul Relay ditempatkan di dalam boks panel kontrol yang terbuat dari bahan isolator. Penempatan ini dipisahkan secara fisik dari ruang resonansi speaker untuk meminimalisir pengaruh getaran mekanis yang dapat mengganggu pembacaan sensor analog LM35. Tata letak komponen disusun dengan memperhatikan manajemen pengkabelan (*cable management*) untuk mengurangi risiko interferensi elektromagnetik (EMI) antar jalur sinyal dan jalur daya utama.

Pada sisi perangkat lunak, aplikasi antarmuka (*Graphical User Interface/GUI*) berbasis Python 3.11 telah berhasil dijalankan pada komputer *host*. Komunikasi data antara komputer dan mikrokontroler terjalin melalui protokol *StandardFirmata* dengan kecepatan *baudrate* 57600 bps. Pengujian awal menunjukkan koneksi serial berjalan stabil tanpa adanya *packet loss* yang signifikan, memungkinkan

pemantauan parameter suhu dan arus secara *real-time* dengan latensi yang sangat rendah.

Untuk memvalidasi kinerja sistem, dilakukan serangkaian skenario pengujian dengan melibatkan alat ukur standar sebagai pembanding. Realisasi fisik perangkat keras beserta konfigurasi pengujian (*testbed*) ditunjukkan pada Gambar 9. Pada gambar tersebut, terlihat sistem terhubung dengan beban *dummy* resistor geser dan alat ukur multimeter digital yang digunakan untuk proses kalibrasi dan validasi pembacaan sensor.



Gambar 9. Realisasi Perangkat Keras dan Konfigurasi Pengujian

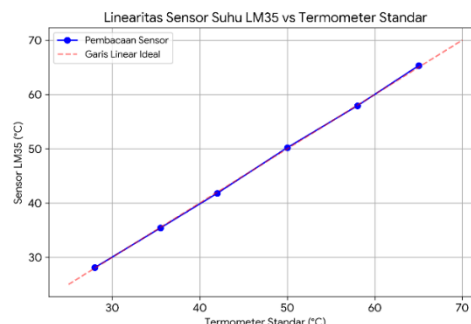
B. Pengujian Akurasi Sensor Suhu

Pengujian ini bertujuan untuk memvalidasi akurasi sensor LM35 yang telah melalui proses kalibrasi perangkat lunak (*software calibration*). Pengujian dilakukan dengan membandingkan pembacaan sensor LM35 pada aplikasi Python terhadap termometer digital standar industri. Sumber panas dihasilkan dari *heatsink* amplifier yang dibebani sinyal sinus 1kHz secara bertahap. Data hasil perbandingan ditunjukkan pada Tabel 1.

Tabel 1. Data Pengujian Sensor Suhu LM35 vs Termometer Standar

No	Sensor Suhu LM35 vs Termometer Standar			
	Termometer Standar (°C)	Sensor LM35 GUI (°C)	Selisih (Error)	Error (%)
1	28.0	28.1	0.1	0.35%
2	35.5	35.4	0.1	0.28%
3	42.0	41.8	0.2	0.47%
4	50.0	50.2	0.2	0.40%
5	58.0	57.9	0.1	0.17%
6	65.0	65.3	0.3	0.46%
Rata-rata			0.16	0.35%

Berdasarkan Tabel 1, terlihat bahwa sensor LM35 memiliki tingkat akurasi yang sangat baik setelah diterapkan filter *Moving Average*. Rata-rata kesalahan relatif (*relative error*) yang dihasilkan hanya sebesar **0.35%** dengan selisih suhu maksimal 0.3°C. Hal ini menunjukkan bahwa penambahan resistor *pull-down* 10kΩ dan kalibrasi *offset* pada Python efektif menghilangkan *noise* sinyal analog.



Gambar 10. Grafik Linearitas Sensor Suhu LM35

Berdasarkan hasil analisis linearitas, diperoleh nilai koefisien determinasi (R^2) sebesar 0.999 yang menunjukkan bahwa pembacaan sensor LM35 memiliki hubungan yang sangat linear terhadap termometer standar.

C. Pengujian Akurasi Sensor Arus

Pengujian sensor arus INA219 dilakukan dengan memberikan beban resistif variabel pada output amplifier. Arus aktual diukur menggunakan *Digital Multimeter* (DMM) yang dipasang seri (amperemeter), kemudian dibandingkan dengan data digital yang diterima aplikasi Python via I2C.

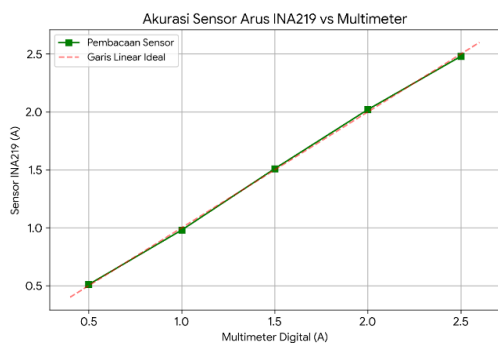
Tabel 2. Data Pengujian Sensor Arus INA219 vs Multimeter

No	Sensor Arus INA219 vs Multimeter			
	Multimeter (Ampere)	Sensor INA219 (Ampere)	Selisih (Ampere)	Error (%)
1	0.50	0.51	0.01	2.00%
2	1.00	0.98	0.02	2.00%
3	1.50	1.51	0.01	0.66%
4	2.00	2.02	0.02	1.00%
5	2.50	2.48	0.02	0.80%
Rata-rata			0.016	1.29%

Hasil pada Tabel 2 kinerja sensor INA219 menunjukkan tingkat presisi yang tinggi dalam memantau beban arus dinamis, dengan rata-rata kesalahan relatif (*relative error*)

hanya sebesar 1.29%. Jika dikorelasikan dengan ambang batas proteksi (*cut-off threshold*) yang ditetapkan pada angka 2.0 Ampere, deviasi pengukuran absolut sebesar 0.01-0.02 Ampere dapat dikategorikan sebagai toleransi yang sangat kecil. Kondisi ini memberikan jaminan stabilitas operasional yang signifikan, di mana sistem dipastikan tidak akan mengalami kesalahan pemicuan proteksi (*false triggering*) akibat ketidakakuratan pembacaan sensor, sehingga proteksi hanya akan aktif ketika kondisi bahaya yang sebenarnya benar-benar terjadi.

Gambar 11. Grafik Linearitas Sensor INA219



Berdasarkan grafik linearitas pada Gambar 11, sensor INA219 juga menunjukkan tingkat korelasi yang sangat tinggi terhadap alat ukur standar

D. Visualisasi Antarmuka Pengguna (Graphical User Interface)

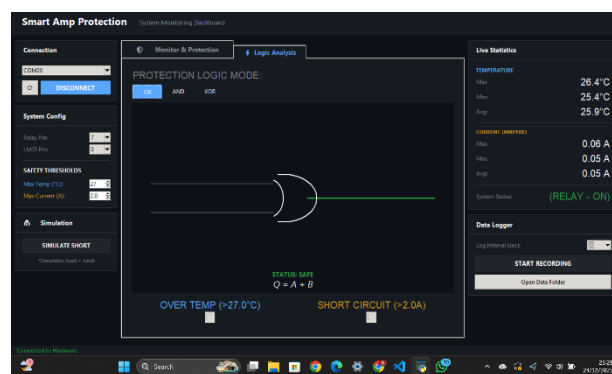
Untuk memfasilitasi pemantauan parameter secara *real-time*, sistem dilengkapi dengan aplikasi antarmuka (GUI) yang dikembangkan menggunakan bahasa pemrograman Python. Antarmuka ini berfungsi memvisualisasikan data akuisisi dari sensor LM35 dan INA219 yang diterima melalui komunikasi serial dengan kecepatan (*baudrate*) 57600 bps.



Gambar 12. Visualisasi Antarmuka Pengguna (Graphical User Interface)

Seperti terlihat pada Gambar 12, *dashboard* aplikasi menampilkan tiga komponen informasi utama:

1. Indikator Numerik: Menampilkan nilai suhu aktual (°C) dan arus beban (Ampere) yang diperbarui secara otomatis dengan interval 1 detik.
2. Status Sistem: Kotak indikator visual keamanan yang akan berubah warna (Hijau untuk "AMAN" dan Merah untuk "BAHAYA") sesuai dengan keputusan logika mikrokontroler.
3. Grafik Riwayat: Kurva *plot* yang merekam fluktuasi data suhu dan arus terhadap waktu. Grafik ini memiliki skala sumbu vertikal yang adaptif, di mana rentang nilai pada tepi grafik akan berubah secara otomatis menyesuaikan tinggi-rendahnya data sensor yang terukur, sehingga lonjakan suhu atau arus dapat terpantau dengan jelas. E. Pengujian Respon Sistem Proteksi (Logika OR)



Gambar 13. Tampilan Tab Analisis Logika pada Antarmuka Pengguna.

Selain itu, komponen *Logic Analysis* diintegrasikan pada antarmuka untuk memberikan transparansi terhadap proses pengambilan keputusan internal sistem. Visualisasi ini menampilkan status biner dari input gerbang logika OR yang diturunkan dari parameter ambang batas suhu dan arus, terhadap sinyal *output* yang dihasilkan. Melalui visualisasi analisis logika ini (Gambar 13), pengguna dapat memverifikasi secara langsung bahwa sistem secara tepat mengidentifikasi kondisi kegagalan spesifik seperti 'Input A' untuk kondisi *overheat* atau 'Input B' untuk *short circuit* dan berhasil mengeksekusi perintah *Output* untuk memicu aktuator relai

E. Analisis Waktu Respons

Berdasarkan rancangan algoritma dengan interval sampling 500 ms dan filter $N=10$, sistem memiliki waktu respons desain berkisar antara 1.5 hingga 2.5 detik. Durasi ini memastikan stabilitas sistem sehingga relai hanya memutus arus pada kondisi bahaya yang berkelanjutan, bukan karena lonjakan sesaat (*noise*). Hasil pengujian untuk berbagai skenario bahaya dirangkum dalam Tabel 3

Tabel 3. Hasil Pengujian Logika Proteksi

No	Hasil Pengujian Logika Proteksi					
	Kondisi Pengujian	Suhu (°C)	Arus (A)	Status Logika (OR)	Respon Relay	Status Sistem
1	Normal (Aman)	45.0	1.20	0 OR 0 = 0	ON (Connect)	Normal
2	<i>Overheat</i>	65.0	1.50	1 OR 0 = 1	OFF (Cut)	BAHAYA
3	<i>Recovery</i> (Turun)	55.0	1.20	0 OR 0 = 0	ON (Connect)	Normal
4	<i>Overcurrent</i>	48.0	2.50	0 OR 1 = 1	OFF (Cut)	BAHAYA
5	Bahaya Ganda	70.0	3.00	1 OR 1 = 1	OFF (Cut)	BAHAYA

Analisis data pada Tabel 3 menunjukkan bahwa algoritma proteksi beroperasi sesuai tabel kebenaran logika OR.

1. Pada skenario ke-2, sistem seketika memutus Relay (OFF) saat suhu mencapai 65.0°C, melebihi ambang batas aman.
2. Pada skenario ke-4, proteksi arus bekerja efektif dengan memutus Relay saat terdeteksi lonjakan arus beban sebesar 2.50 A.
3. Mekanisme pemulihan otomatis (*auto-recovery*) terverifikasi pada skenario ke-3, di mana sistem mengaktifkan kembali Relay setelah parameter suhu dan arus turun ke level normal.

F. Hasil Perekaman Data Logger

Fitur *Data Logger* berfungsi merekam riwayat aktivitas sensor untuk keperluan audit dan analisis pasca-gangguan. Data dikirimkan setiap detik oleh mikrokontroler dan disimpan secara otomatis ke dalam format CSV oleh aplikasi antarmuka. Tabel 4 memperlihatkan cuplikan data log saat terjadi transisi dari kondisi aman ke bahaya.

Tabel 4. Cuplikan Data Log (CSV Output)

No	Hasil Cuplikan Data Log (CSV Output)				
	Timestamp	Suhu (°C)	Arus (A)	Status	Keterangan Kejadian
1	26/12/2025 01:22:45	58.50	1.80	AMAN	Kondisi beban tinggi
2	26/12/2025 01:22:46	59.10	1.85	AMAN	Suhu mulai naik
3	26/12/2025 01:22:47	60.20	1.90	BAHAYA	Pemicuan (Trigger)
4	26/12/2025 01:22:48	60.50	0.00	BAHAYA	Relay Putus (Cut-Off)
5	26/12/2025 01:22:49	60.30	0.00	BAHAYA	Pendinginan (Cooling)

Berdasarkan data rekaman pada Tabel 4, dinamika perubahan sistem dapat dianalisis sebagai berikut:

1. Pada pukul 01:22:46, sistem masih berstatus AMAN meskipun suhu mulai naik mencapai 59.10°C.
2. Transisi kritis terjadi pada detik berikutnya (01:22:47), di mana suhu terdeteksi menyentuh angka 60.20°C (melewati ambang batas). Sistem seketika merespon dengan mengubah status menjadi BAHAYA.
3. Efektivitas proteksi terkonfirmasi pada pukul 01:22:48. Terlihat pembacaan arus beban turun drastis menjadi 0.00 A (dari sebelumnya 1.90 A), yang membuktikan bahwa Relay telah memutus aliran listrik secara fisik (*Cut-Off*) untuk mengamankan amplifier dari kerusakan lebih lanjut.

IV. KESIMPULAN

Berdasarkan hasil pengujian, sistem proteksi aktif berbasis mikrokontroler Arduino Nano dan antarmuka Python berhasil beroperasi secara *real-time* untuk memutus aliran daya ke perangkat *amplifier* saat kondisi melampaui batas aman. Aspek keakuratan data menjadi poin krusial dalam dalam efektivitas

sistem ini, di mana penerapan metode kalibrasi perangkat lunak dan algoritma filter rata-rata bergerak (*Moving Average Filter*) terbukti sangat efektif dalam mereduksi *noise* serta meningkatkan presisi pembacaan sensor. Hal ini dibuktikan melalui data pengujian yang menunjukkan persentase rata-rata kesalahan yang sangat rendah, yaitu sebesar 0,35% pada sensor suhu LM35 dan 1,29% pada sensor arus INA219 jika dibandingkan dengan alat ukur standar industri. Pemanfaatan gerbang logika OR pada tingkat perangkat lunak memberikan fleksibilitas yang presisi dalam pengambilan keputusan untuk memicu modul *relay* ke status *cut-off* apabila salah satu parameter kritis terpenuhi, yakni suhu *heatsink* melampaui 60°C atau arus beban melebihi 2,0 A. Selain itu, kehadiran fitur *Data Logger* yang merekam riwayat anomali dalam format CSV menjadi nilai tambah yang signifikan bagi teknisi dalam melakukan evaluasi teknis serta analisis penyebab gangguan pasca-kejadian secara mendalam. Untuk pengembangan di masa mendatang, sangat disarankan untuk melakukan migrasi platform ke modul mikrokontroler ESP32 guna memfasilitasi akses pemantauan data secara nirkabel dari jarak jauh, serta mengintegrasikan sistem pendinginan aktif berupa kipas DC berbasis kendali *Pulse Width Modulation* (PWM) untuk manajemen termal yang lebih proaktif sebelum sistem mencapai ambang batas kritis pemutusan daya.

DAFTAR PUSTAKA

- [1] Texas Instruments, "TPA3116D2 2x50-W Class-D Stereo Amplifier Family Datasheet," Dallas, TX, 2016.
- [2] J. Iskandar and H. Wijaya, "Optimasi Heat Sink dan Kipas Otomatis Berbasis PWM untuk Mencegah Thermal Runaway pada Amplifier Transistor Bipolar," *Jurnal Energi dan Kelistrikan*, vol. 15, no. 1, pp. 55–63, 2024.
- [3] R. Hidayat, M. Ibrahim, and T. Ardiansyah, "Analisis Sistem Proteksi Thermal dan DC Offset pada Power Amplifier OCL 150 Watt untuk Keamanan Loudspeaker," *Jurnal Ilmiah Teknik Elektronika*, vol. 5, no. 1, pp. 12–19, 2021.
- [4] I. G. N. Putu and M. Rusli, "Rancang Bangun Proteksi Hubung Singkat pada Power Amplifier Mobil Menggunakan Komparator LM393," *Jurnal Riset Teknik Elektro*, vol. 2, no. 1, pp. 23–30, 2022.
- [5] A. D. Prasetyo and S. Muryanto, "Perancangan Sistem Proteksi Arus Lebih pada Audio Power Amplifier Berbasis Mikrokontroler Arduino Uno," *Jurnal Teknik Elektro dan Komputer (JTEK)*, vol. 9, no. 2, pp. 45–52, 2020.
- [6] A. Z. Muttaqin, "Studi Komparasi Proteksi Speaker Menggunakan IC UPC1237 dan Rangkaian Transistor Diskrit," *Jurnal Elektronika Terapan*, vol. 11, no. 1, pp. 10–17, 2024.
- [7] M. F. Ramadhan and S. Sukir, "Efektivitas Penggunaan Sekring Otomatis (Circuit Breaker) Elektronik pada Penguat Daya Audio High-Fidelity," *Jurnal Pendidikan Teknik Elektro*, vol. 7, no. 2, pp. 115–122, 2022.
- [8] D. Kurniawan, "Sistem Proteksi Tegangan Berlebih (Overvoltage) pada Catu Daya Amplifier Menggunakan Zener Crowbar," *Gema Teknologi*, vol. 22, no. 3, pp. 140–145, 2023.
- [9] B. Santoso, "Pengembangan Sistem Monitoring Suhu Power Amplifier Kelas D Menggunakan Sensor MLX90614 Berbasis IoT," *Jurnal Teknologi Informasi dan Terapan (JTT)*, vol. 8, no. 2, pp. 88–94, 2021.
- [10] J. Kim, "Self-recovering over current protection circuit for class-D audio power amplifier," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 3, pp. 1303–1309, 2008.
- [11] F. Al-Farizi and W. Sunanda, "Implementasi Relay Proteksi untuk Menghindari Lonjakan Arus (Soft Start) pada Amplifier Daya Besar," in *Prosiding Seminar Nasional Penelitian dan Pengabdian pada Masyarakat (SNPP)*, Pangkalpinang, 2021, pp. 102–108.
- [12] A. Raj and M. S. Sujith, "Arduino based Over-current and Over-voltage Protection Relay," *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, no. 5, pp. 433–437, 2020.
- [13] Texas Instruments, "INA219 Zero-Drift, Bidirectional Current/Power Monitor Datasheet," Dallas, TX, 2015.
- [14] T. Prasetyo, "Desain Penguat Daya Audio dengan Sistem Proteksi Terintegrasi Berbasis NodeMCU untuk Notifikasi Kerusakan Jarak Jauh," *Jurnal Teknik dan Ilmu Komputer*, vol. 13, no. 4, pp. 210–218, 2025.
- [15] M. S. Jadin, "Design and Development of a Low-Cost Data Logging System for Solar Photovoltaic Monitoring," *International Journal of Power Electronics and Drive Systems (IJPEDS)*, vol. 10, no. 4, pp. 1974–1981, 2019.
- [16] M. Banzi and M. Shiloh, *Getting Started with Arduino*, 3rd ed. Sebastopol, CA: Maker Media, Inc., 2014.
- [17] Texas Instruments, "LM35 Precision Centigrade Temperature Sensors Datasheet," Dallas, TX, 2017.
- [18] Python Software Foundation, "Python 3.11.0 Documentation." [Online]. Available: <https://docs.python.org/3/>