

## Optimasi Pengendalian PID pada *Load Frequency Control* Menggunakan *Artificial Hummingbird Algorithm*

**Gilang Esa Muhammad**

Program Studi S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya, Ketintang 60231, Indonesia

e-mail: [gilangesa.22005@mhs.unesa.ac.id](mailto:gilangesa.22005@mhs.unesa.ac.id)

**Rifqi Firmansyah**

Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya, Ketintang 60231, Indonesia

e-mail: [rifqifirmansyah@unesa.ac.id](mailto:rifqifirmansyah@unesa.ac.id)

### Abstrak

Load Frequency Control (LFC) merupakan bagian penting dalam sistem tenaga listrik yang berfungsi menjaga kestabilan frekuensi saat terjadi perubahan beban. Ketidakstabilan frekuensi dapat menurunkan keandalan sistem dan mengganggu kinerja peralatan yang terhubung. Salah satu metode umum untuk meningkatkan kestabilan sistem adalah dengan menggunakan pengendali Proportional-Integral-Derivative (PID). Namun, penyetelan parameter PID secara konvensional sering kali belum menghasilkan performa yang optimal. Penelitian ini bertujuan untuk mengoptimalkan parameter kontrol PID menggunakan Artificial Hummingbird Algorithm (AHA), yaitu algoritma metaheuristik yang terinspirasi dari perilaku terbang dan pencarian makan burung kolibri. Implementasi dilakukan menggunakan MATLAB dengan fungsi objektif *Integral of Time-weighted Absolute Error* (ITAE) untuk meminimalkan kesalahan sistem. Algoritma AHA dimodifikasi agar dapat menampilkan evolusi nilai fitness serta konvergensi terhadap solusi optimal selama proses iterasi. Hasil simulasi menunjukkan bahwa pengendali PID yang dioptimalkan dengan AHA mampu menghasilkan respon sistem yang lebih cepat, memiliki Overshoot lebih rendah, dan mencapai kondisi yang lebih stabil dibandingkan metode konvensional. Temuan ini membuktikan bahwa Artificial Hummingbird Algorithm merupakan metode optimasi yang efektif dan efisien untuk penyetelan parameter kontrol PID pada sistem Load Frequency Control.

**Kata kunci:** Optimalisasi, Load Frequency Control, Kontrol PID, Artificial Hummingbird Algorithm, ITAE.

### Abstract

*Load Frequency Control (LFC) is an important part of the electrical power system that functions to maintain frequency stability when load changes occur. Frequency instability can reduce system reliability and disrupt the performance of connected equipment. One common method to improve system stability is to use a Proportional-Integral-Derivative (PID) controller. However, conventional PID parameter tuning often fails to produce optimal performance. This study aims to optimize PID control parameters using the Artificial Hummingbird Algorithm (AHA), a metaheuristic algorithm inspired by the flight and foraging behavior of hummingbirds. The implementation was carried out using MATLAB with the Integral of Time-weighted Absolute Error (ITAE) objective function to minimize system errors. The AHA algorithm was modified to display the evolution of fitness values and convergence to the optimal solution during the iteration process. The simulation results show that the PID controller optimized with AHA is capable of producing a faster system response, has lower Overshoot, and achieves a more stable condition compared to conventional methods. These findings prove that the Artificial Hummingbird Algorithm is an effective and efficient optimization method for tuning PID control parameters in Load Frequency Control systems.*

**Keywords:** Optimization, Load Frequency Control, PID Control, Artificial Hummingbird Algorithm, ITAE.

### PENDAHULUAN

Dalam sistem tenaga listrik modern, kestabilan frekuensi menjadi faktor penting untuk menjamin kualitas dan keandalan suplai listrik kepada konsumen. Aplikasi Load Frequency Control (LFC) banyak digunakan pada sistem pembangkit listrik, baik di PLTU, PLTA, maupun sistem interkoneksi antar wilayah, untuk menjaga agar frekuensi sistem tetap berada pada nilai nominal meskipun terjadi perubahan beban secara tiba-tiba. Ketika permintaan daya meningkat, LFC secara otomatis menyesuaikan keluaran daya generator agar sistem tetap stabil tanpa menimbulkan gangguan besar pada frekuensi. Dengan demikian, LFC berperan penting dalam menjaga keseimbangan antara daya pembangkitan dan permintaan beban, sehingga frekuensi sistem tetap konstan pada nilai nominalnya (Kamarudin et

al., 2021).

Pengendali Proportional-Integral-Derivative (PID) masih menjadi pilihan utama dalam sistem LFC karena kesederhanaannya, keandalannya, dan kemampuan untuk menghasilkan respon dinamis yang memadai. Namun demikian, kinerja PID sangat dipengaruhi oleh pemilihan parameter proporsional, integral, dan derivative yang tepat (Vijay Kumar & Ranga Reddy, 2024). Proses tuning secara konvensional sering kali membutuhkan waktu lama dan tidak memberikan hasil optimal, terutama pada sistem yang bersifat nonlinier dan memiliki ketidakpastian parameter. Oleh sebab itu, banyak penelitian terkini yang menggunakan metode optimasi berbasis metaheuristik untuk menentukan parameter PID secara otomatis dan efisien (Ravikumar et al.,

2015).

Berbagai algoritma metaheuristic seperti Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Grey Wolf Optimizer (GWO), hingga algoritma hibrid telah diterapkan dalam tuning PID pada sistem LFC. Pendekatan ini terbukti mampu meningkatkan performa sistem dengan meminimalkan Integral Time Absolute Error (ITAE) serta mempercepat waktu tunak (Settling time). Metode optimasi ini memberikan kemampuan eksplorasi dan eksploitasi yang lebih baik terhadap ruang solusi dibandingkan pendekatan klasik seperti Ziegler–Nichols (Ogar, 2023).

Salah satu algoritma optimasi baru yang dikembangkan untuk berbagai masalah nonlinier adalah Artificial Hummingbird Algorithm (AHA). AHA terinspirasi dari perilaku burung kolibri dalam mencari makanan, mempertahankan lokasi, dan kemampuan terbangnya yang unik dalam tiga mode: axial, diagonal, dan omnidirectional. Algoritma ini memiliki mekanisme eksplorasi dan eksploitasi yang seimbang melalui pembaruan posisi populasi secara dinamis (Emam et al., 2023). AHA telah menunjukkan hasil menjanjikan pada berbagai aplikasi, termasuk optimasi sistem energi, perancangan kontroler PID untuk motor DC, dan manajemen energi mikrogrid. Studi terbaru menunjukkan bahwa AHA mampu mencapai konvergensi yang cepat serta menghindari jebakan solusi lokal berkat keseimbangan eksplorasi dan eksploitasi yang efisien (Du et al., 2024).

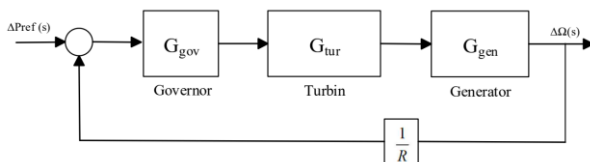
## TINJAUAN PUSTAKA

### Load Frequency Control (LFC)

Load Frequency Control (LFC) bertujuan menjaga keseimbangan daya aktif sehingga frekuensi sistem kembali ke nilai nominal setelah terjadi gangguan beban (Saadat, 1999). Untuk area tunggal, LFC mengatur keluaran pembangkit melalui governor/turbin sehingga deviasi frekuensi diminimalkan dan dipulihkan (Rasolomampionona et al., 2024).

Dalam praktiknya, LFC sangat penting karena frekuensi sistem tenaga listrik harus dijaga tetap dalam batas yang diizinkan, yaitu umumnya 49–51 Hz sesuai standar di Indonesia. Ketika terjadi perubahan beban secara mendadak, frekuensi dapat mengalami kenaikan atau penurunan, sehingga diperlukan mekanisme kontrol yang responsif untuk mempercepat waktu pemulihan, mengurangi *Overshoot*, serta menjaga kestabilan operasi sistem tenaga (Laksono & Tiftazani, 2025).

Diagram blok model LFC area tunggal dapat dilihat pada gambar 1.



Gambar 1. Diagram blok LFC area tunggal

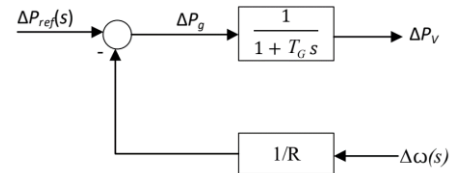
Komponen utama pada sistem LFC area tunggal adalah governor, turbin, generator, dan beban.

#### 1. Governor

Governor adalah alat kontrol otomatis yang berfungsi

untuk mengatur dan menstabilkan kecepatan putaran mesin atau penggerak mula (seperti turbin dan mesin diesel). governor bekerja dengan mengatur jumlah bahan bakar, uap, air, atau gas yang masuk ke mesin, sehingga mesin dapat mempertahankan putaran atau RPM yang konstan walaupun terjadi perubahan beban (Mesin et al., 2023).

Governor bertindak sebagai pengendali utama perubahan daya keluaran pembangkit agar tetap stabil. Diagram blok model governor dapat dilihat pada gambar 2.



Gambar 2. Diagram blok governor (Saadat, 1999)

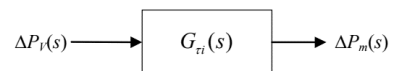
Model matematis dari diagram blok model governor didapat dari persamaan 1.

$$\Delta P_V(s) = \frac{1}{1 + T_G s} \Delta P_g(s) \quad (1)$$

Dimana  $\Delta P_V(s)$  merupakan sinyal perubahan posisi katup keluaran dari suatu sistem dinamik orde satu dengan konstanta waktu governor  $T_G$ , yang menerima masukan berupa perubahan daya governor  $\Delta P_g(s)$ .

#### 2. Turbin

Turbin adalah mesin penggerak yang mengubah energi potensial fluida kerja (seperti air, uap, atau angin) menjadi energi mekanik dalam bentuk putaran poros turbin. Energi mekanik dari turbin ini umumnya dimanfaatkan untuk menggerakkan generator sehingga menghasilkan energi listrik, atau untuk aplikasi lain yang memerlukan energi mekanik (Khattak et al., 2016). Diagram blok model turbin dapat dilihat pada gambar 3.



Gambar 3. Diagram blok turbin (Saadat, 1999)

Model matematis dari diagram blok model turbin didapat dari persamaan 2.

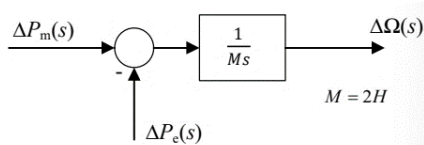
$$\frac{\Delta P_m(s)}{\Delta P_V(s)} = \frac{1 + sF_{HP}T_{RH}}{(1 + T_{CH}s)(1 + T_{RH}s)} \quad (2)$$

Dimana  $\Delta P_m(s)$  merupakan perubahan daya mekanik turbin. Parameter  $F_{HP}$  menyatakan fraksi daya pada bagian turbin tekanan tinggi (high pressure), sedangkan  $T_{RH}$  adalah konstanta waktu reheater yang merepresentasikan dinamika pemanas ulang uap dan  $T_{CH}$  menyatakan konstanta waktu chest (ruang uap masuk turbin) yang menggambarkan keterlambatan akibat volume dan kompresibilitas uap sebelum masuk ke turbin.

### 3. Generator

Sebuah generator adalah perangkat pengubah energi mekanik yang menerapkan prinsip induksi magnet. Terdapat generator arus bolak-balik (alternator) dan arus searah (DC). Kedua jenis ini menggunakan konduktor berputar dalam medan magnet untuk menghasilkan tegangan listrik (Faturachman et al., 2021).

Diagram blok model generator dapat dilihat pada gambar 4.



Gambar 4. Diagram blok generator (Saadat, 1999)

Model matematis dari diagram blok model generator didapat dari persamaan 3.

Dimana  $\Delta\Omega(s)$  merupakan perubahan kecepatan sudut atau deviasi frekuensi sistem,  $\Delta P_e(s)$  adalah perubahan daya listrik beban. Parameter  $H$  menyatakan konstanta inersia sistem (inertia constant) yang menggambarkan kemampuan generator dalam menahan perubahan kecepatan.

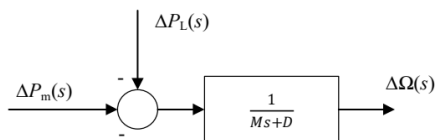
### 4. Beban

Beban merupakan perilaku beban listrik dalam sistem tenaga listrik yang digunakan untuk mengatur tanggapan beban terhadap perubahan frekuensi (Laksono & Tiftazani, 2025). Model beban dapat dimodelkan yang tertulis pada persamaan 3.

$$\Delta P_e = \Delta P_L + D\Delta\omega \quad (3)$$

Dimana  $\Delta P_L$  adalah perubahan daya beban, dan  $D\Delta\omega$  menyatakan komponen perubahan daya yang dipengaruhi oleh karakteristik redaman beban akibat perubahan frekuensi. Parameter  $D$  merupakan koefisien redaman beban (load damping coefficient), sedangkan  $\Delta\omega$  adalah deviasi frekuensi sistem.

Dari persamaan 4 didapatkan diagram blok pada gambar 5.

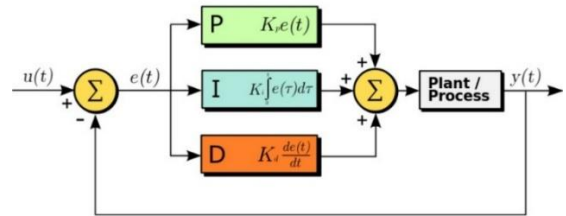


Gambar 5. Diagram blok Model Generator dengan Beban (Saadat, 1999)

### Sistem Kontrol PID

Kontrol *Proporsional-Integral-Derivatif* (PID) merupakan salah satu pengendali paling luas digunakan dalam berbagai sistem industri karena strukturnya yang sederhana, mudah dituning, dan efektif dalam meningkatkan stabilitas serta kinerja sistem. Kontrol PID bekerja berdasarkan sinyal error, yaitu selisih antara nilai referensi (setpoint) dan keluaran sistem. Menurut (Ogata, 2009), kontroler PID mampu memperbaiki respon transien dan

kondisi tunak secara simultan, sehingga banyak diterapkan pada sistem mekanik, elektronik, hingga sistem tenaga seperti *Load Frequency Control* (LFC). Diagram blok model sistem kontrol PID dapat dilihat pada gambar 6.



Gambar 6. Diagram blok model sistem kontrol PID

Rumus matematis dari diagram blok sistem kontrol PID didapat dari persamaan 4.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (4)$$

$$\Delta\Omega(s) = \frac{1}{2Hs} [\Delta P_m(s) - \Delta P_e(s)] \quad (5)$$

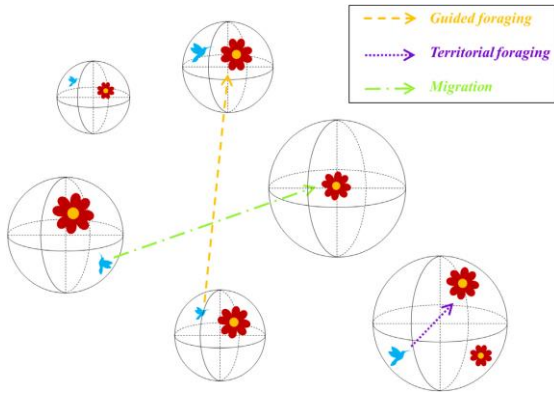
dimana,  $u(t)$  merepresentasikan sinyal kontrol yang dihasilkan oleh pengendali untuk mengatur dinamika sistem. Sinyal ini dibentuk berdasarkan besarnya error  $e(\tau)$ , yaitu selisih antara nilai setpoint dan keluaran sistem aktual. Parameter  $K_p$  merupakan konstanta proporsional yang menentukan besar kecilnya respon pengendali terhadap error pada saat itu. Selanjutnya,  $K_i$  adalah konstanta integral yang berfungsi memperbaiki kesalahan secara kumulatif sehingga mampu menghilangkan error keadaan tunak (*steady-state error*). Sementara itu,  $K_d$  bertindak sebagai konstanta derivatif yang memberikan aksi kontrol berdasarkan laju perubahan error, sehingga dapat meningkatkan stabilitas sistem dan mengurangi *overshoot*.

### *Artificial Hummingbird Algorithm* (AHA)

Metode metaheuristik populer untuk menyelesaikan masalah optimisasi nyata karena kemampuannya menyeimbangkan *eksplorasi* dan *eksploitasi* tanpa memerlukan turunan fungsi. Dalam konteks ini, (Zhao et al., 2022) memperkenalkan AHA sebuah algoritma terinspirasi perilaku foraging (mencari makanan) dan pola terbang kolibri dengan tujuan menyediakan optimasi yang sederhana, efisien, dan hanya membutuhkan sedikit parameter kontrol.

AHA terinspirasi tiga sifat menonjol kolibri: (1) keterampilan terbang (axial, diagonal, omnidirectional) yang memungkinkan perpindahan di berbagai arah; (2) kapasitas memori yang kuat (mengingat lokasi, kualitas nektar, dan waktu kunjungan terakhir); dan (3) strategi foraging yang adaptif (guided, territorial, migrating) (Hosseinzadeh et al., 2025).

Tiga perilaku mencari makan (foraging) burung kolibri dalam optimasi AHA dapat dilihat pada gambar 7.

Gambar 7. Tiga perilaku *foraging* AHA (Zhao et al., 2022)

Secara ringkas AHA terdiri dari langkah-langkah berikut: inisialisasi populasi & visit table, lalu iterasi dengan probabilitas 50% memilih antara Guided foraging atau Territorial foraging; setiap  $2n$  iterasi ( $n$  = ukuran populasi) dilakukan migration foraging untuk mencegah stagnasi. Ketiga strategi foraging ini memakai tiga Flight skills untuk mengontrol arah dan cakupan pencarian.

#### 1. Inisialisasi

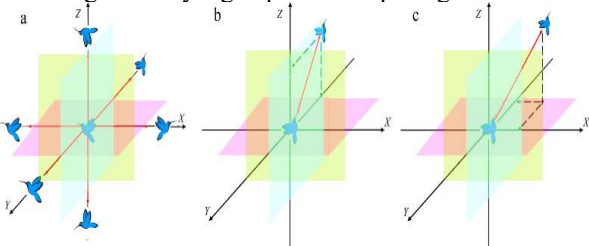
Setiap Suatu populasi  $n$  burung kolibri ditempatkan pada  $n$  sumber makanan, diinisialisasi secara acak pada persamaan 6.

$$x_i = Low + r \cdot (Up - Low) \quad i = 1, \dots, n \quad (6)$$

di mana  $Low$  dan  $Up$  masing-masing adalah batas atas dan batas bawah untuk masalah berdimensi,  $r$  adalah vektor acak dalam  $[0, 1]$ , dan  $x_i$  merepresentasikan posisi sumber makanan ke- $i$  yang merupakan solusi dari masalah yang diberikan.

#### 2. Flight skills

*Flight skills* merupakan arah gerak atau terbang dari burung kolibri yang dapat dilihat pada gambar 8.

Gambar 8. Tiga perilaku terbang burung kolibri, (a) *axial flight*, (b) *diagonal flight*, (c) *omnidirectional flight* (Zhao et al., 2022)

*Axial flight* adalah pola gerak burung pada satu sumbu saja, sehingga pencarian dilakukan dengan memvariasikan posisi hanya pada satu dimensi tertentu. *Diagonal flight* merupakan pola gerak pada kombinasi beberapa sumbu sekaligus, di mana kolibri mengubah posisinya dengan bergerak di beberapa dimensi dalam ruang pencarian secara bersamaan. Adapun *omnidirectional flight* memungkinkan burung kolibri untuk bergerak pada semua sumbu secara bersamaan sehingga proses eksplorasi menjadi lebih menyeluruh dan peluang menemukan solusi optimal di ruang

multidimensi semakin besar.

#### 3. Guided foraging

Persamaan matematika yang mensimulasikan perilaku *Guided foraging* dan sumber makanan kandidat dapat dilihat pada persamaan 7.

$$v_i(t+1) = x_{i,tar}(t) + a \cdot D \cdot (x_i(t) - x_{i,tar}(t)), \quad (7)$$

$$a \sim N(0, 1)$$

di mana  $x_i(t)$  adalah posisi sumber makanan ke- $i$  pada waktu  $t$ ,  $x_{i,tar}(t)$  adalah posisi sumber makanan target yang akan dikunjungi oleh burung kolibri ke- $i$ , dan  $a$  adalah faktor pemandu, yang tunduk pada distribusi normal  $N(0, 1)$  dengan nilai rata-rata = 0 dan standar deviasi = 1. *Guided foraging* menekankan eksplorasi pada tahap awal (jarak antar sumber besar) dan perlahan beralih ke eksploitasi.

#### 4. Territorial foraging

Setelah mengunjungi sumber makanan yang targetnya sudah habis nektarnya, burung kolibri cenderung mencari sumber makanan baru daripada kembali ke sumber yang telah ada. Oleh karena itu, kolibri biasanya berpindah ke area di sekitar wilayahnya untuk mencari sumber makanan alternatif yang berpotensi memberikan kualitas lebih baik. Proses inilah yang dimodelkan sebagai *Territorial foraging*, di mana burung kolibri mengeksplorasi lingkungan terdekat guna membentuk solusi kandidat baru. Persamaan matematis yang menggambarkan mekanisme *Territorial foraging* ditunjukkan pada persamaan 8.

$$v_i(t+1) = x_i(t) + b \cdot D \cdot x_i(t), \quad b \sim N(0, 1) \quad (8)$$

Pada persamaan 8,  $b$  merupakan faktor teritorial yang mengikuti distribusi normal  $N(0, 1)$  dengan nilai rata-rata 0 dan standar deviasi 1. Melalui Persamaan 8, setiap burung kolibri dapat membangkitkan sumber makanan baru di area sekitar posisinya, sehingga memungkinkan pencarian solusi lokal yang efektif dengan memanfaatkan kemampuan terbang spesifik yang dimilikinya.

#### 5. Migration foraging

Ketika ketersediaan makanan menurun, burung kolibri akan bermigrasi ke sumber makanan lain yang lebih jauh. Dalam AHA, perilaku ini dimodelkan melalui koefisien migrasi. Jika jumlah iterasi telah melewati nilai koefisien tersebut, burung kolibri pada sumber dengan tingkat nektar terburuk dipindahkan ke sumber baru yang dibangkitkan secara acak dalam ruang pencarian. Selanjutnya, posisi dan tabel kunjungan diperbarui. Proses migrasi ini dijelaskan pada Persamaan 9.

$$x_{wor}(t+1) = Low + r \cdot (Up - Low) \quad (9)$$

di mana  $x_{wor}$  adalah sumber makanan dengan tingkat

pengisian nektar terburuk dalam populasi.

Pada strategi *Guided foraging*, burung kolibri bergerak menuju sumber makanan lain untuk meningkatkan eksplorasi dan menghindari optimum lokal. Jika terdapat sumber makanan yang diperbarui, sumber tersebut lebih cenderung dipilih sebagai target sehingga proses eksploitasi meningkat. Berdasarkan Persamaan 7, eksplorasi lebih dominan pada awal iterasi, sedangkan eksploitasi meningkat seiring berkurangnya jarak antar sumber.

Strategi *Territorial foraging* berfokus pada eksploitasi di sekitar lingkungan lokal, sedangkan *Migration foraging* berperan dalam meningkatkan eksplorasi pada seluruh ruang pencarian. Pada AHA, selain ukuran populasi dan jumlah iterasi, terdapat parameter kontrol migrasi. Strategi *Migration foraging* diterapkan untuk mengurangi stagnasi dan mencegah konvergensi pada optimum lokal ketika proses pencarian mengalami keterbatasan eksplorasi.

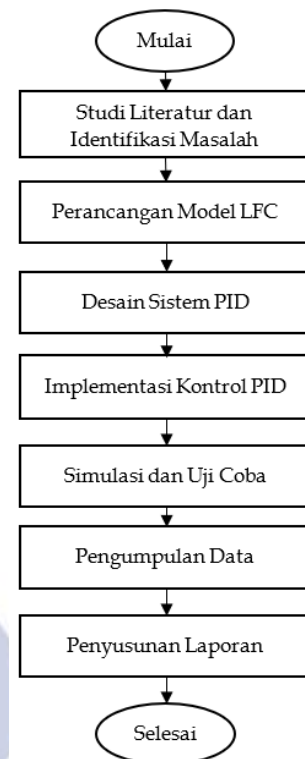
Secara keseluruhan, *Artificial Hummingbird Algorithm* (AHA) menawarkan pendekatan optimisasi yang menarik karena mampu menyeimbangkan proses eksplorasi dan eksploitasi melalui tiga mekanisme utama, yaitu *Guided foraging*, *Territorial foraging*, dan *migration foraging*. Penggunaan *visit table* juga menjadi keunggulan khas AHA karena dapat mengurangi kunjungan berulang dan meningkatkan penyebaran pencarian di ruang solusi. Selain itu, struktur algoritma yang sederhana dan jumlah parameter yang minim menjadikan AHA mudah diterapkan pada berbagai masalah optimisasi rekayasa. Meskipun demikian, AHA masih memiliki beberapa keterbatasan, seperti potensi stagnasi pada ruang pencarian berdimensi tinggi serta sensitivitas terhadap parameter migrasi apabila tidak disetel secara tepat. Selain itu, performanya tidak selalu unggul untuk semua jenis fungsi atau masalah. Oleh karena itu, berbagai pengembangan lanjutan seperti hibridisasi, penambahan operator pencarian lokal, dan adaptasi untuk masalah multi-objektif masih menjadi bidang penelitian yang relevan untuk meningkatkan kinerja AHA di masa mendatang.

## METODE PENELITIAN

### Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif, yaitu pendekatan yang berfokus pada pengumpulan serta analisis data numerik yang dapat diukur secara objektif. Pendekatan kuantitatif menekankan penggunaan data hasil pengukuran yang sistematis dan akurat sebagai dasar analisis. Dalam penelitian ini, metode optimasi AHA diterapkan untuk menentukan parameter kontroler PID, sehingga memungkinkan peneliti memperoleh hasil analisis yang lebih mendalam mengenai performa dan karakteristik pengendali PID.

Penelitian ini melibatkan beberapa tahapan yang terdapat pada gambar 9:

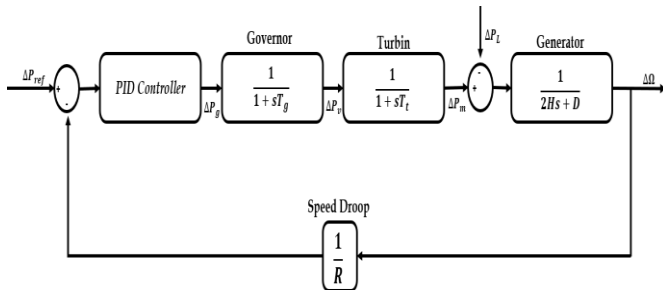


Gambar 9. Flowchart rancangan penelitian

Rancangan penelitian ini dimulai dengan tahap mulai yang menandai awal kegiatan penelitian, dilanjutkan dengan studi literatur dan identifikasi masalah melalui pengkajian referensi terkait *Load Frequency Control* (LFC) *single-area* dan kontrol PID untuk menentukan permasalahan utama, yaitu deviasi frekuensi akibat perubahan beban. Selanjutnya dilakukan perancangan model LFC yang mencakup komponen governor, turbin, dan generator-beban sebagai representasi dinamika sistem tenaga listrik. Setelah model terbentuk, dilakukan desain sistem PID dengan menentukan struktur dan parameter pengendali yang sesuai untuk meningkatkan respon frekuensi sistem. Pengendali PID yang telah dirancang kemudian diterapkan pada tahap implementasi kontrol PID ke dalam model LFC. Tahap berikutnya adalah simulasi dan uji coba untuk mengevaluasi kinerja sistem terhadap gangguan perubahan beban dengan menganalisis parameter respon seperti *rise time*, *undershoot*, *overshoot*, *settling time*, *steady-state error*, dan *ITAE*. Hasil simulasi selanjutnya dilakukan pengumpulan data sebagai dasar analisis kinerja sistem, dan seluruh proses penelitian diakhiri dengan penyusunan laporan yang memuat hasil, analisis, serta kesimpulan penelitian hingga tahap selesai.

### Blok Diagram

Diagram blok pada Gambar 10 menunjukkan alur kerja sistem pada penelitian ini.



Gambar 10. Blok diagram sistem LFC

Blok diagram sistem *Load Frequency Control* (LFC) dengan pengendali PID menggambarkan alur pengaturan frekuensi pada sistem tenaga berdasarkan perubahan beban. Pada bagian awal, sinyal referensi perubahan daya ( $\Delta P_{ref}$ ) dibandingkan dengan perubahan frekuensi keluaran sistem, kemudian menghasilkan sinyal error. Sinyal error ini menjadi masukan bagi PID Controller yang parameternya ( $K_p$ ,  $K_i$ ,  $K_d$ ). Kemudian, keluaran kontroler berupa perubahan perintah governor ( $\Delta P_g$ ) diteruskan menuju blok Governor. Governor menghasilkan sinyal perubahan posisi katup ( $\Delta P_v$ ), yang kemudian masuk ke blok Turbin. Turbin mengubah sinyal kontrol tersebut menjadi perubahan daya mekanik ( $\Delta P_m$ ). Selanjutnya, sinyal ( $\Delta P_m$ ) dikombinasikan dengan gangguan beban ( $\Delta P_L$ ) pada titik penjumlahan, lalu diteruskan ke blok Generator. Keluaran blok ini berupa perubahan frekuensi sistem ( $\Delta \Omega$ ). Sinyal ( $\Delta \Omega$ ) kemudian diberikan kembali ke titik awal melalui jalur umpan balik yang melewati blok *Speed Droop*, sehingga sistem dapat menyesuaikan perubahan frekuensi terhadap perubahan beban yang terjadi.

#### Parameter Komponen

Parameter komponen LFC yang digunakan pada penelitian ini terdapat pada tabel 1 sebagai berikut:

Tabel 1. Parameter komponen LFC

| Parameter         | Model                | Keterangan                                                               |
|-------------------|----------------------|--------------------------------------------------------------------------|
| Frekuensi nominal | $f$                  | 50 Hz                                                                    |
| Base power        | $S_{base}$           | 100 MVA                                                                  |
| Pengendali        | PID(s)               | Parameter PID digunakan untuk koreksi error sistem                       |
| Governor          | $\frac{1}{1 + sT_g}$ | Konstanta waktu governor $T_g = 0,2s$                                    |
| Turbin            | $\frac{1}{1 + sT_t}$ | Konstanta waktu turbin $T_t = 0,5s$                                      |
| Generator & beban | $\frac{1}{2Hs + D}$  | Konstanta inersia generator $H = 5s$ , koefisien redaman beban $D = 0,8$ |
| Speed droop       | $\frac{1}{R}$        | $R = 0,05 pu$                                                            |

#### Evaluasi Sistem

Evaluasi kinerja sistem merupakan tahapan penting dalam penelitian ini untuk menilai keberhasilan pengoptimalan pengendali PID pada sistem *Load Frequency Control* (LFC) menggunakan *Artificial Hummingbird Algorithm* (AHA). Evaluasi dilakukan dengan menganalisis respons dinamik frekuensi sistem tenaga listrik terhadap

gangguan perubahan beban.

Pengujian sistem dilakukan melalui simulasi menggunakan perangkat lunak MATLAB. Gangguan diberikan dalam bentuk perubahan beban secara tiba-tiba (*step load disturbance*) untuk merepresentasikan kondisi nyata pada sistem tenaga listrik, seperti perubahan konsumsi daya oleh beban. Respons sistem kemudian dianalisis berdasarkan kurva deviasi frekuensi terhadap waktu.

Parameter kinerja yang digunakan dalam penelitian ini meliputi *rise time*, *undershoot*, *overshoot*, *settling time*, *steady-state error*, dan *ITAE*. parameter tersebut dipilih karena mampu merepresentasikan karakteristik respons transien dan kestabilan sistem LFC secara menyeluruh.

Ringkasan standar evaluasi yang digunakan ditunjukkan pada tabel 2.

Tabel 2. Standar evaluasi sistem LFC

| Parameter Evaluasi                  | Standar PLN Indonesia (Grid Code)                               | Standar yang Digunakan dalam Penelitian     | Keterangan                                                          |
|-------------------------------------|-----------------------------------------------------------------|---------------------------------------------|---------------------------------------------------------------------|
| Steady-State Error (ESS)            | $\leq \pm 0,2$ Hz (frekuensi operasi normal 50 Hz $\pm 0,2$ Hz) | $\leq 0,2$ Hz                               | Standar resmi PLN sesuai Grid Code / Permen ESDM                    |
| Deviasi Frekuensi Transien Maksimum | Diperbolehkan hingga $\pm 0,5$ Hz untuk waktu singkat           | Tidak melebihi $\pm 0,5$ Hz selama transien | Digunakan untuk validasi keamanan sistem                            |
| Overshoot (Mp)                      | Tidak ditentukan secara eksplisit oleh PLN                      | $\leq 5\%$                                  | Digunakan sebagai standar kinerja kontrol berdasarkan literatur LFC |
| Undershoot (Mu)                     | Tidak ditentukan secara eksplisit oleh PLN                      | $\leq 5\%$                                  | Digunakan sebagai standar kinerja kontrol berdasarkan literatur LFC |
| Rise time (tr)                      | Tidak ditentukan oleh PLN                                       | $\leq 5$ detik                              | Kriteria performa dinamis sistem kontrol                            |
| Settling time (ts)                  | Tidak ditentukan oleh PLN                                       | $\leq 10-20$ detik                          | Umum digunakan pada penelitian LFC                                  |

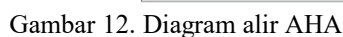
#### HASIL DAN PEMBAHASAN

##### Implementasi AHA pada Sistem LFC

Sistem *Load Frequency Control* (LFC) area tunggal pada penelitian ini dimodelkan menggunakan perangkat lunak MATLAB sebagaimana ditunjukkan pada gambar 11.



File Simulink diatas akan diintegrasikan dengan MATLAB Code yang didalamnya terdapat algoritma AHA. Tahapan bagaimana algoritma AHA bekerja dijelaskan pada diagram alir gambar 12.



- ### 1. Inisialisasi Parameter AHA

Pada tahap ini ditentukan parameter utama algoritma AHA yang digunakan dalam proses optimasi, yaitu:

- PopSize = 25 (jumlah populasi kolibri)
- MaxIter = 60 (jumlah iterasi maksimum)
- Dim = 3 (dimensi solusi dari 3 parameter  $K_p$ ,  $K_i$ ,  $K_d$ )
- Lower Bound : Low = [0 0 0] (batas bawah  $K_p$ ,  $K_i$ ,  $K_d$ )
- Upper Bound: Up = [9 9.5 1.6] (batas atas  $K_p$ ,  $K_i$ ,  $K_d$ )

Parameter ini bertujuan untuk membatasi ruang pencarian agar solusi tetap stabil dan realistis secara sistem kendali.

- ## 2. Inisialisasi Populasi Kolibri

Setiap kolibri merepresentasikan satu kandidat solusi berupa vektor dengan nilai awal  $K_p$ ,  $K_i$ ,  $K_d$  dihasilkan secara acak dalam rentang batas bawah dan batas atas yang telah ditentukan.

- ### 3. Evaluasi Fitness Awal

Setiap solusi awal diuji dengan cara:

- a. Mengimplementasikan parameter PID ke dalam model LFC area tunggal di MATLAB.

- b. Melakukan simulasi respon frekuensi sistem akibat gangguan beban.

- c. Menghitung nilai fungsi objektif ITAE (Integral of Time Absolute Error):

Nilai ITAE digunakan sebagai fitness, di mana nilai yang lebih kecil menunjukkan performa kendali yang lebih baik.

- #### 4. Pengecekan Iterasi Maksimum

Dilakukan pengecekan jika iterasi  $<$  MaxIter, maka proses optimasi dilanjutkan dan jika iterasi  $\geq$  MaxIter, maka algoritma dihentikan dan solusi terbaik diambil sebagai hasil akhir.

- ## 5. Proses Foraging Kolibri

Pada tahap ini, posisi kolibri diperbarui menggunakan tiga mekanisme pencarian AHA:

- a. *Guided foraging*

Kolibri bergerak menuju solusi terbaik global yang telah ditemukan sebelumnya untuk mempercepat proses konvergensi.

- b. *Territorial foraging*

Kolibri melakukan eksplorasi lokal di sekitar posisinya saat ini untuk menemukan solusi yang lebih baik di area terdekat.

- c. *Migration foraging*

Kolibri dengan performa terburuk dipindahkan ke posisi baru secara acak untuk menghindari jebakan local optimum dan meningkatkan eksplorasi global.

- ## 6. Pembaruan Posisi Kolibri

Hasil dari mekanisme foraging digunakan untuk memperbarui nilai  $K_n$ ,  $K_i$ ,  $K_d$  setiap kolibri.

- ### 7. Pembatasan Posisi (*Boundary Check*)

Nilai parameter PID yang baru diperoleh dicek agar tetap berada dalam batas,  $\text{Lower} \leq K_p, K_i, K_d \leq \text{Upper}$  jika ada nilai yang melampaui batas, maka dikembalikan ke batas terdekat.

- #### 8. Simulasi Ulang dan Perhitungan ITAE Baru

Parameter PID yang telah diperbarui diuji kembali pada sistem LFC melalui simulasi, kemudian dihitung nilai ITAE baru.

## 9. Evaluasi Perbaikan Solusi

Dilakukan perbandingan jika ITAE baru  $<$  ITAE lama, maka solusi baru diterima dan disimpan sebagai solusi terbaik dan jika ITAE baru  $\geq$  ITAE lama, maka solusi lama dipertahankan.

## 10. Update Iterasi

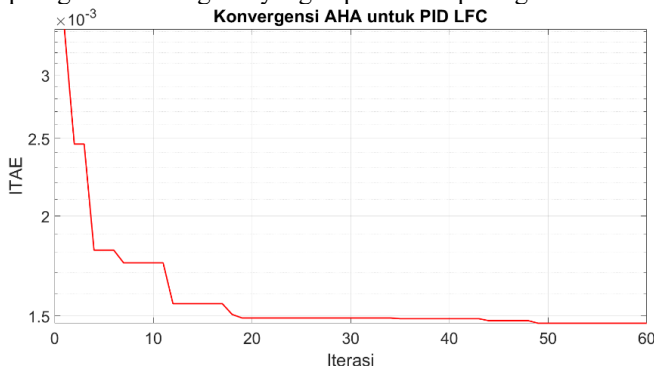
Nilai iterasi dinaikkan iterasi = iterasi + 1 kemudian proses kembali ke tahap pengecekan iterasi maksimum.

## 11. Solusi Terbaik (Optimal)

Setelah iterasi maksimum tercapai, algoritma menghasilkan solusi optimal berupa nilai optimal dari  $K_p$ ,  $K_i$ ,  $K_d$ .

Nilai ini merupakan parameter PID terbaik yang menghasilkan respon frekuensi sistem LFC dengan nilai ITAE minimum, *Overshoot* kecil, dan waktu pemulihan yang cepat.

Setelah melakukan semua tahapan di atas, didapatkan plot grafik konvergensi yang dapat dilihat pada gambar 13.



Gambar 13. Grafik konvergensi AHA

Sehingga didapatkan parameter nilai  $K_p$ ,  $K_i$ ,  $K_d$  yang paling optimal yang nantinya akan dibandingkan dengan LFC tanpa pengendali, PID-konvensional, dan PID-PSO. Tabel parameter nilai  $K_p$ ,  $K_i$ ,  $K_d$  dapat dilihat pada tabel 3.

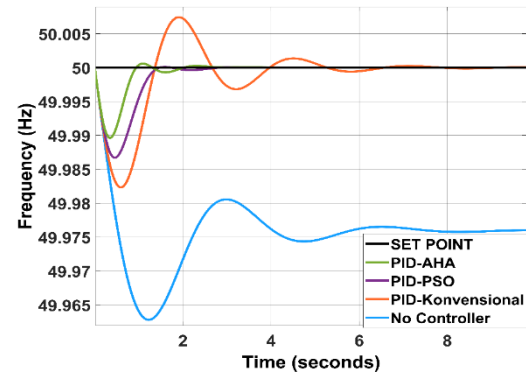
Tabel 3. Parameter sistem

| Nilai | PID-konvensional | PID-PSO | PID-AHA |
|-------|------------------|---------|---------|
| $K_p$ | 2.22             | 3.1060  | 4.8537  |
| $K_i$ | 2.3344           | 2.4746  | 4.567   |
| $K_d$ | 0.5278           | 1.0267  | 1.6     |

Setelah mendapatkan nilai tiap parameter ini akan dijalankan pada sistem LFC area tunggal dan diuji terkait performa sistem terhadap deviasi frekuensi.

## Respon Frekuensi Terhadap Gangguan Beban Kecil

Pengujian gangguan beban kecil dilakukan untuk mengevaluasi respons awal sistem LFC terhadap perubahan daya beban yang relatif ringan. Pada skenario ini, gangguan beban diberikan sebesar  $\Delta PL = 0,01$  pu setara 1 MW pada base 100 MVA. Pengujian ini bertujuan untuk mengamati sensitivitas sistem serta kemampuan masing-masing metode pengendalian dalam meredam deviasi frekuensi sejak awal terjadinya gangguan. Grafik gangguan beban kecil dapat dilihat pada gambar 14.



Gambar 14. Grafik gangguan beban kecil

Tabel 4. Gangguan beban kecil

| Parameter                 | Tanpa pengendali | PID-Konvensional | PID-PSO | PID-AHA |
|---------------------------|------------------|------------------|---------|---------|
| <i>Rise time</i> (s)      | —                | 1,356            | 1,466   | 0,937   |
| <i>Overshoot</i> (%)      | —                | 0,014            | 0,000   | 0,00    |
| <i>Undershoot</i> (%)     | -0,074           | -0,036           | 0,0264  | 0,0206  |
| <i>Settling time</i> (s)  | 10               | 7,995            | 5,761   | 4       |
| <i>Steady State Error</i> | 0,02             | 0                | 0       | 0       |
| $\Delta f$ (Hz)           | 0,04             | 0,018            | 0,0132  | 0,0103  |

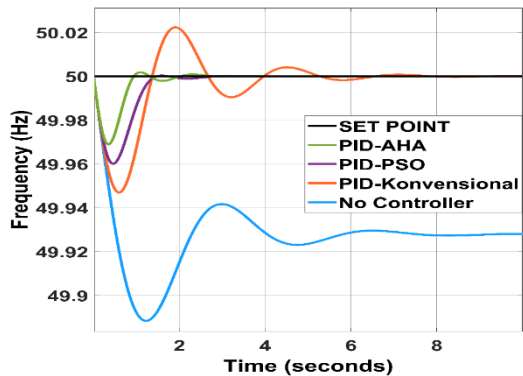
Berdasarkan gambar 14 dan tabel 4, respon frekuensi sistem pada gangguan beban kecil menunjukkan bahwa sistem tanpa pengendali memiliki performa terburuk. Hal ini ditandai dengan deviasi frekuensi paling besar sebesar 0,04 Hz, steady-state error 0,02 Hz, serta *Settling time* mencapai 10 detik, yang mengindikasikan sistem tidak mampu mengembalikan frekuensi ke nilai nominal secara cepat dan stabil. Kondisi ini menunjukkan pentingnya penggunaan pengendali dalam sistem *Load Frequency Control*, meskipun gangguan yang diberikan relatif kecil.

Penerapan PID konvensional mampu meningkatkan kinerja sistem dengan menghilangkan steady-state error dan menurunkan deviasi frekuensi menjadi 0,018 Hz. Namun, respon sistem masih menunjukkan osilasi awal dengan *Overshoot* 0,014%, *Undershoot* -0,036%, serta *Settling time* 7,995 detik. Selanjutnya, penggunaan PID-PSO memberikan perbaikan lebih lanjut dengan *Settling time* 5,761 detik dan deviasi frekuensi yang lebih kecil (0,0132 Hz), serta *Overshoot* dan *Undershoot* yang semakin teredam, meskipun *Rise time* masih relatif lebih lambat dibandingkan PID-AHA.

Pengendali PID-AHA menunjukkan performa terbaik pada gangguan beban kecil. Hal ini ditunjukkan oleh *Rise time* tercepat sebesar 0,937 detik, *Settling time* paling singkat yaitu 4 detik, serta deviasi frekuensi terkecil sebesar 0,0103 Hz. Selain itu, nilai *Overshoot* dan *Undershoot* yang paling rendah menandakan karakteristik redaman yang baik dan respon yang lebih stabil.

### Respon Frekuensi Terhadap Gangguan Beban Sedang

Pengujian gangguan beban sedang dirancang untuk mengevaluasi efektivitas sistem LFC saat terjadi perubahan daya beban yang lebih besar dibandingkan dengan gangguan beban kecil. Dalam situasi ini, gangguan beban diberi nilai  $\Delta PL = 0,03$  pu setara 3 MW pada base 100 MVA, yang mencerminkan fluktuasi beban biasa yang sering kali terjadi dalam pengoperasian sistem kelistrikan. Dari pengujian ini, variasi kinerja antara berbagai metode pengendalian dapat terlihat dengan lebih jelas, terutama berkaitan dengan kecepatan reaksi dan stabilitas frekuensi. Grafik gangguan beban sedang dapat dilihat pada gambar 15.



Gambar 11. Grafik gangguan beban sedang

Tabel 5. Gangguan beban sedang

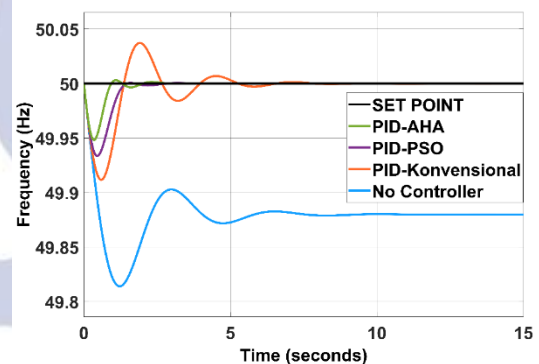
| Parameter          | Tanpa pengendali | PID-Konvensional | PID-PSO | PID-AHA |
|--------------------|------------------|------------------|---------|---------|
| Rise time (s)      | —                | 1,371            | 1,484   | 0,947   |
| Overshoot (%)      | —                | 0,0441           | 0,00091 | 0,0035  |
| Undershoot (%)     | -0,2228          | -0,106           | -0,0792 | -0,0618 |
| Settling time (s)  | 10,25            | 9,202            | 4,536   | 4,05    |
| Steady State Error | 0,0719           | 0                | 0       | 0       |
| $\Delta f$ (Hz)    | 0,1114           | 0,053            | 0,0396  | 0,0309  |

Berdasarkan gambar 15 dan tabel 5, respon frekuensi sistem saat terjadi gangguan beban sedang menunjukkan bahwa peningkatan beban sebesar 0,03 pu menyebabkan perubahan frekuensi yang lebih besar dibandingkan saat terjadi gangguan beban kecil. Sistem tanpa pengendali mengalami penurunan frekuensi yang terbesar, dengan deviasi maksimum mencapai 0,1114 Hz, *Undershoot* terbesar sebesar -0,2228%, dan waktu stabilisasi terlalu lama sekitar 10,25 detik. Selain itu, sistem ini masih memiliki kesalahan steady-state sebesar 0,0719 Hz, yang menunjukkan bahwa sistem tidak mampu mengembalikan frekuensi ke nilai semula secara tepat tanpa adanya pengendali. Penerapan pengendali PID konvensional mampu meningkatkan respon sistem dengan menghilangkan kesalahan steady-state dan menurunkan deviasi frekuensi menjadi 0,053 Hz. Namun, respon sementara masih menunjukkan osilasi awal yang cukup besar, terlihat dari *Overshoot* sebesar 0,0441% dan *Undershoot* sebesar -0,106%, serta waktu stabilisasi sebesar 9,202 detik. Selanjutnya, PID-PSO memberikan peningkatan kinerja

yang lebih baik dengan waktu penstabilan berkurang menjadi 4,536 detik, deviasi frekuensi sebesar 0,0396 Hz, serta *Overshoot* dan *Undershoot* yang lebih rendah dibandingkan dengan PID biasa, menunjukkan penurunan osilasi yang lebih efektif. Pengendali PID-AHA kembali menunjukkan performa terbaik saat menghadapi gangguan beban sedang. Ini terbukti dari waktu naik yang paling cepat sebesar 0,947 detik, waktu stabil yang terpendek yaitu 4,05 detik, serta deviasi frekuensi maksimum yang terkecil sebesar 0,0309 Hz. Selain itu, nilai *Overshoot* (0,0035%) dan *Undershoot* (-0,0618%) lebih rendah dibandingkan metode lainnya, sehingga respon frekuensi lebih cepat mencapai kondisi stabil dengan osilasi yang terbatas.

### Respon Frekuensi Terhadap Gangguan Beban Besar

Pengujian gangguan beban besar dilakukan agar bisa mengetahui seberapa tangguh dan stabil sistem kontrol frekuensi beban saat menghadapi perubahan beban yang sangat besar. Dalam kondisi ini, gangguan beban diberikan dengan nilai  $\Delta PL = 0,05$  pu setara 5 MW pada base 100 MVA, yang berpotensi menyebabkan peningkatan deviasi frekuensi dan mengganggu kestabilan sistem. Skenario ini digunakan untuk menguji kemampuan pengendali dalam mempertahankan stabilitas sistem dan mengurangi osilasi saat terjadi gangguan yang lebih parah. Grafik gangguan beban besar dapat dilihat pada gambar 16.



Gambar 16. Grafik gangguan beban besar

Tabel 6. Gangguan beban besar

| Parameter          | Tanpa pengendali | PID-Konvensional | PID-PSO | PID-AHA |
|--------------------|------------------|------------------|---------|---------|
| Rise time (s)      | —                | 1,385            | 1,474   | 0,95    |
| Overshoot (%)      | —                | 0,0735           | 0,00109 | 0,0058  |
| Undershoot (%)     | -0,372           | -0,175           | -0,133  | -0,104  |
| Settling time (s)  | 10,517           | 9,513            | 4,736   | 4,278   |
| Steady State Error | 0,1202           | 0                | 0       | 0       |
| $\Delta f$ (Hz)    | 0,186            | 0,0875           | 0,0665  | 0,0518  |

Berdasarkan gambar 16 dan tabel 6, respon frekuensi sistem saat mengalami gangguan beban besar menunjukkan deviasi frekuensi yang lebih besar dibandingkan gangguan beban kecil dan sedang. Pada sistem tanpa pengendali, terjadi penurunan frekuensi paling besar dengan nilai *undershoot* sebesar 0,372%, deviasi frekuensi maksimum ( $\Delta f$ ) sebesar 0,186 Hz, dan error steady-state sebesar 0,1202 Hz. Selain itu, waktu settling mencapai 10,517 detik,

sehingga respon sistem masih lambat dan mengalami osilasi yang cukup besar.

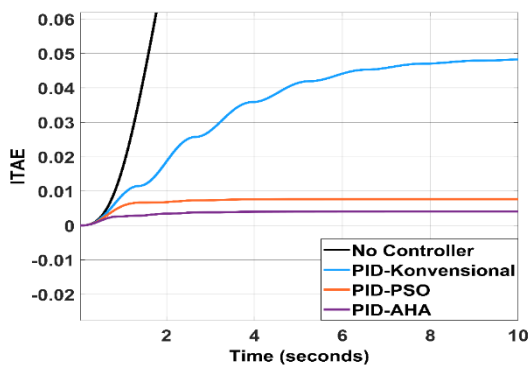
Penggunaan PID konvensional mampu menghilangkan error steady-state, namun respon transien masih menunjukkan osilasi dengan overshoot sebesar 0,0735%, undershoot 0,175%, dan waktu settling 9,513 detik. Deviasi frekuensi maksimum yang dihasilkan sebesar 0,0875 Hz, lebih kecil dibandingkan sistem tanpa pengendali.

Metode PID-PSO memberikan respon yang lebih baik dengan overshoot sebesar 0,00109%, undershoot -0,133%, dan waktu settling 4,736 detik. Deviasi frekuensi maksimum yang dihasilkan sebesar 0,0665 Hz.

Sementara itu, PID-AHA menghasilkan kinerja terbaik dengan rise time 0,95 detik, overshoot 0,0058%, undershoot -0,104%, waktu settling 4,278 detik, serta deviasi frekuensi maksimum sebesar 0,0518 Hz dengan error steady-state bernilai nol.

### Analisis Indeks Kinerja

Pada tahap ini dilakukan analisis indeks kinerja menggunakan *Integral of Time Absolute Error* (ITAE) untuk menilai performa pengendali *Load Frequency Control* secara kuantitatif. ITAE digunakan sebagai parameter evaluasi karena mampu merepresentasikan akumulasi error frekuensi terhadap waktu, dengan memberikan bobot yang lebih besar pada error yang berlangsung lebih lama. Melalui analisis ITAE, kinerja masing-masing metode tanpa pengendali, PID konvensional, PID-PSO, dan PID-AHA dapat dibandingkan secara objektif dalam meredam deviasi frekuensi pada berbagai kondisi gangguan beban. Grafik ITAE dapat dilihat pada gambar 17.



Gambar 17. Grafik ITAE

Tabel 7. Indeks kinerja ITAE

| Parameter | Tanpa pengendali | PID-Konvensional | PID-PSO | PID-AHA |
|-----------|------------------|------------------|---------|---------|
| ITAE      | 1,452            | 0,0483           | 0,0077  | 0,004   |

Mengacu pada gambar 17 dan tabel 7, ITAE memperlihatkan perbedaan performa yang cukup signifikan antar metode pengendalian yang digunakan. Sistem tanpa pengendali memiliki nilai ITAE tertinggi sebesar 1,452, yang pada grafik ditunjukkan oleh kenaikan kurva yang sangat tajam sejak awal waktu simulasi. Kondisi ini mengindikasikan bahwa deviasi frekuensi yang terjadi bersifat besar dan bertahan cukup lama, sehingga menghasilkan akumulasi error terhadap waktu yang tinggi.

Penggunaan PID konvensional mampu menurunkan nilai ITAE secara signifikan menjadi 0,0483. Namun demikian,

grafik ITAE masih menunjukkan laju peningkatan yang relatif lebih cepat dibandingkan metode optimasi, yang menandakan bahwa error frekuensi masih berlangsung dalam interval waktu tertentu sebelum sistem benar-benar mencapai kondisi stabil. Selanjutnya, PID-PSO menghasilkan nilai ITAE yang lebih kecil, yaitu 0,0077, dengan kecenderungan kurva yang lebih landai, menunjukkan bahwa deviasi frekuensi dapat dikompensasi dalam waktu yang lebih singkat dibandingkan PID konvensional.

Di sisi lain, PID-AHA menghasilkan nilai ITAE terendah sebesar 0,004, yang ditunjukkan oleh kurva ITAE paling rendah dan relatif stabil sepanjang durasi simulasi. Hal ini mencerminkan kemampuan pengendali dalam menekan akumulasi error frekuensi secara lebih efektif, khususnya pada interval waktu akhir yang memiliki pengaruh lebih besar dalam perhitungan ITAE. Variasi nilai ITAE yang diperoleh tersebut menunjukkan perbedaan tingkat efektivitas masing-masing metode pengendalian dalam meredam deviasi frekuensi terhadap waktu.

### Ucapan Terima Kasih

Penulis menyampaikan terima kasih yang sebesar-besarnya kepada Bapak Rifqi Firmansyah, S.T., M.T., Ph.D. selaku dosen pembimbing yang dengan penuh kesabaran, ketelitian, dan ketulusan memberikan bimbingan, arahan, serta masukan berharga sepanjang proses penelitian. Berkat dukungan beliau, penelitian ini dapat penulis selesaikan dengan lebih terarah dan mendalam serta motivasi yang sangat berarti bagi penulis dalam menyelesaikan artikel ini.

### PENUTUP

#### Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa perancangan sistem kontrol PID pada *Load Frequency Control* (LFC) area tunggal menggunakan metode *Artificial Hummingbird Algorithm* (AHA) berhasil dilakukan secara efektif. Proses perancangan dilakukan dengan memanfaatkan AHA sebagai metode optimasi untuk menentukan parameter pengendali PID, yaitu  $K_p$ ,  $K_i$ ,  $K_d$ , berdasarkan fungsi objektif yang meminimalkan error frekuensi sistem. Mekanisme eksplorasi dan eksploitasi pada AHA memungkinkan pencarian solusi optimal secara sistematis serta mampu menghindari jebakan solusi lokal yang dapat menyebabkan respon sistem berosilasi dan tidak stabil. Hasil optimasi menunjukkan bahwa parameter PID yang dihasilkan mampu meningkatkan kecepatan respon dan kestabilan sistem LFC dalam menghadapi gangguan perubahan beban.

Hasil simulasi menunjukkan bahwa PID-AHA memberikan performa respon frekuensi yang lebih baik dibandingkan sistem tanpa pengendali, PID konvensional, dan PID-PSO. Pada kondisi gangguan beban besar, PID-AHA menghasilkan rise time sebesar 0,95 s, lebih cepat dibandingkan PID konvensional sebesar 1,385 s dan PID-PSO sebesar 1,474 s. Selain itu, settling time PID-AHA sebesar 4,278 s, lebih singkat dibandingkan PID konvensional (9,513 s) dan PID-PSO (4,736 s). Nilai

overshoot dan undershoot yang dihasilkan PID-AHA juga lebih kecil, masing-masing sebesar 0,0058% dan -0,104%, serta mampu menghilangkan steady-state error hingga bernilai nol.

Analisis indeks kinerja menggunakan Integral of Time Absolute Error (ITAE) menunjukkan bahwa PID-AHA menghasilkan nilai ITAE paling kecil, yaitu 0,004, dibandingkan PID-PSO sebesar 0,0077, PID konvensional sebesar 0,0483, dan sistem tanpa pengendali sebesar 1,452. Nilai ITAE yang lebih kecil ini menunjukkan bahwa PID-AHA mampu meminimalkan akumulasi error frekuensi terhadap waktu secara efektif, sehingga pengendali PID yang dioptimasi menggunakan Artificial Hummingbird Algorithm memiliki kinerja terbaik dalam menjaga kestabilan frekuensi pada sistem Load Frequency Control area tunggal.

### Saran

Penelitian selanjutnya disarankan untuk mengembangkan model *Load Frequency Control* multi-area dengan mempertimbangkan keterkaitan antararea melalui *tie-line power* agar kinerja PID-AHA dapat diuji pada sistem yang lebih kompleks dan mendekati kondisi nyata. Selain itu, perlu dipertimbangkan ketidakpastian parameter sistem, seperti variasi konstanta inersia, perubahan karakteristik beban, dan *delay* komunikasi untuk menguji robustnes pengendali. Penelitian berikutnya juga dapat membandingkan *Artificial Hummingbird Algorithm* dengan algoritma metaheuristik lainnya, seperti *Grey Wolf Optimizer* (GWO), *Sine Cosine Algorithm* (SCA), maupun algoritma hibrida untuk memperoleh evaluasi yang lebih komprehensif.

### DAFTAR PUSTAKA

- Du, J., Zhang, J., Li, S., & Yang, Z. (2024). Enhanced Artificial Hummingbird Algorithm with chaotic traversal flight. 1–37.
- Emam, M. M., Houssein, E. H., Tolba, M. A., Zaky, M. M., & Ali, M. H. (2023). Application of modified Artificial Hummingbird Algorithm in optimal power flow and generation capacity in power networks considering renewable energy sources. *Scientific Reports*. <https://doi.org/10.1038/s41598-023-48479-6>
- ESDM, P. (2020). *Aturan Jaringan Sistem Tenaga Listrik (Grid Code)*. 3.
- Faturachman, D., Manan, M. P. De, & Sulaeman, K. A. (2021). Literature Study on Review Emergency Generator Usage on Landing Craft Tank. *6495*(6), 288–294.
- Hosseinazadeh, M., Masoud, A., Fatimatelbatoul, R., Husari, M., Mutab, O., Marzougui, M., Nhu, G., & Lee, S. W. (2025). A Survey of Artificial Hummingbird Algorithm and Its Variants: Statistical Analysis, Performance Evaluation, and Structural Reviewing. *Archives of Computational Methods in Engineering*. <https://doi.org/10.1007/s11831-024-10135-1>
- Hote, Y. V., & Jain, S. (2018). PID controller design for Load Frequency Control: Past, Present and future challenges.
- Jangid, P. (2025). Design a PID controller based on grey wolf optimization algorithm for single-area Load Frequency Control. *3*(1), 1–8.
- Kamarudin, M. N., Shahrudin, N., Haqkimi, N., & Rahman, A. (2021). Review on Load Frequency Control for power system stability. *TELKOMNIKA*, *19*(2), 638–644. <https://doi.org/10.12928/TELKOMNIKA.v19i2.16118>
- Khattak, M. A., Ali, N. S. M., Abidin, N. H. Z., Azhar, N. S., & Omar, M. H. (2016). Common Type of Turbines in Power Plant: A Review. *3*(1), 77–100.
- Laksono, H. D., & Tiftazani, A. (2025). Analisis Kinerja Pengendali dengan Filter pada Sistem Load Frequency Control Tipe Reheat untuk Stabilisasi Frekuensi. *15*(1), 42–51.
- Mesin, J. T., Negeri, P., & Pandang, U. (2023). Analisis pengaruh perubahan frekuensi sistem terhadap kinerja governor pada plta bakaru unit 1 skripsi.
- Ogar, V. N. (2023). Load Frequency Control Using the Particle Swarm Optimisation Algorithm and PID Controller for Effective Monitoring of Transmission Line.
- Ogata, K. (2009). *Modern Control Engineering*. Prentice Hall.
- Rasolomampionona, D. D., Połeczki, M., Zagrajek, K., Wróblewski, W., & Januszewski, M. (2024). A Comprehensive Review of Load Frequency Control Technologies. *Energies*, *17*(12). <https://doi.org/10.3390/en17122915>
- Ravikumar, B., Ashok, G., & Sivaprasad, B. (2015). Tuning PID Controller Parameters for Load Frequency Control Considering System Uncertainties. *5*(5), 42–47.
- Saadat, H. (1999). *power system analysis*. McGraw-Hill.
- Samuel, N. (2011). A single area Load Frequency Control: a comparative study based on pi, optimal and fuzzy logic controllers. 748–754. <https://doi.org/10.5251/ajsir.2011.2.5.748.754>
- Sunca, K. Y., & Boz, A. F. (2025). Artificial Hummingbird Algorithm- Based PID Controller for DC Motor Speed Control. *Politeknik Dergisi*, *0900*, 1–1. <https://doi.org/10.2339/politeknik.1687239>
- Vijay Kumar, A., & Ranga Reddy, G. P. (2024). Prototype Of Fuzzy PID Controller For Load Frequency Control Based On Hybrid Optimization Algorithm. *Ispes 2023*, 38–44. <https://doi.org/10.5220/0012506000003808>
- Zhang, Z. (2023). High - precision angle adaptive control simulation of synchronous motor for automatic lifting and boarding equipment of aircraft platform. *Scientific Reports*, 1–12. <https://doi.org/10.1038/s41598-023-30742-5>
- Zhao, W., Methods, C., Mech, A., Wang, L., & Mirjalili, S. (2022). Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Computer Methods in Applied Mechanics and Engineering*, *388*, 114194. <https://doi.org/10.1016/j.cma.2021.114194>