IMPLEMENTASI ALGORITMA GENETIKA UNTUK OPTIMALISASI PID CONTROL PADA *PLANT SEPARATOR* INDUSTRI MIGAS

Lutfi Arya Pramanda[1], Andika Setya Nugraha[1], Citra N.Dg Tapala[1], Zahida Nabila Ramadhani[1], Asepta Surya Wardhana[1*]
[1]Teknik Instrumentasi Kilang , Politeknik Energi dan Mineral Akamigas, Cepu, Blora, 58311
**E-mail*: *aseptasw@esdm.go.id*

ABSTRAK

Pengendalian proses pada plant separator industri migas memerlukan sistem kontrol yang dapat menjaga stabilitas operasi pemisahan fluida. Kontroler *Proportional-Integral-Derivative* (PID) merupakan pilihan umum, namun penentuan parameter PID optimal pada sistem nonlinier dan dinamis menjadi tantangan. Penelitian ini mengimplementasikan *Algoritma Genetika* (GA) untuk mengoptimalkan parameter PID pada plant separator dengan fungsi objektif *Integral Time Absolute Error* (ITAE). Implementasi menggunakan pendekatan *co-simulation* yang mengintegrasikan *Matlab/Simulink* untuk pemodelan sistem dengan *Python* untuk eksekusi GA berbasis pustaka DEAP. Proses optimasi melibatkan operasi GA pada rentang parameter Kp ∈ [0,100], Ki ∈ [0,50], Kd ∈ [0,20]. Hasil penelitian menunjukkan GA menghasilkan parameter optimal (Kp=7.98, Ki=5.02, Kd=2.98) dengan peningkatan performa: *settling time* 32% lebih cepat*, overshoot* 46.5% lebih rendah, dan ITAE 37.9% lebih baik dibandingkan manual tuning. GA terbukti konsisten mencapai global optimum independen dari kondisi awal dibandingkan metode konvensional *(Ziegler-Nichols* dan *auto-tuning Matlab).* Temuan ini memvalidasi efektivitas *Algoritma Genetika* sebagai tool optimasi cerdas untuk meningkatkan respons dinamik dan keselamatan operasi pada plant separator industri migas.

Kata kunci: ***Algoritma Genetika*, *Co-simulation, Integral Time Absolute Error,* Kontrol PID, Optimasi**

1. PENDAHULUAN

Pada industri minyak dan gas, separator merupakan peralatan penting yang berfungsi memisahkan campuran fluida hasil produksi menjadi tiga fase utama, yaitu minyak (oil), gas, dan air berdasarkan perbedaan densitas dan sifat fisiknya[1]. Pengendalian proses pemisahan yang berlangsung optimal memerlukan sistem kontrol otomatis yang mampu menjaga variabel proses seperti tekanan, suhu, dan level cairan tetap stabil sesuai dengan nilai *setpoint* yang telah ditentukan [2],[3].

Penggunaan *Proportional Integral Derivative* (PID) kontrol telah menjadi standar industri karena strukturnya yang sederhana dan kemampuannya menangani dinamika sistem[4], [5]. Namun, tantangan utama dalam penerapan PID adalah penyetelan parameter Kp, Ki, dan Kd yang optimal. Kompleksitas dan karakteristik nonlinier sistem migas, terutama pada kondisi operasi yang berubah-ubah, sering membuat metode tuning konvensional seperti *Ziegler Nichols* kurang efektif karena dirancang untuk sistem linier dan kondisi operasi stabil[6],[7]. Variasi kondisi operasional pada separator secara signifikan memperumit pengendalian. Pendekatan *trial and error* yang masih banyak digunakan tidak efisien dan berpotensi menghasilkan parameter suboptimal yang menurunkan efisiensi operasional sistem [8],[9]. Dengan demikian, diperlukan metode *tuning* otomatis yang dapat beradaptasi dengan dinamika sistem nonlinier untuk menghasilkan parameter PID yang optimal dalam berbagai kondisi operasi.

Sebagai solusi, pendekatan algoritma cerdas mulai banyak digunakan untuk mengoptimasi parameter PID secara otomatis dan adaptif. *Algoritma Genetika (Genetic Algorithm, GA)* terbukti efektif karena meniru mekanisme evolusi biologis untuk mencari kombinasi parameter optimal melalui operasi seleksi, *crossover*, dan mutasi [10],[11],[12],[13]. GA memiliki keunggulan dalam melakukan global optimization melalui *population-based* search yang dapat menghindari jebakan local optimum.

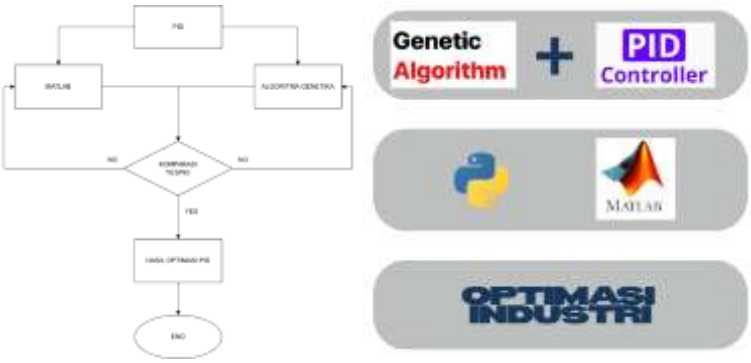
Penelitian ini mengimplementasikan Algoritma Genetika untuk mengoptimalkan parameter PID pada sistem pengendalian suhu separator industri migas melalui simulasi *co-simulation* berbasis *Matlab/Simulink* dan *Python*. Pendekatan simulasi dipilih untuk memvalidasi efektivitas GA dalam menghasilkan parameter optimal sebelum implementasi pada sistem riil. Penelitian ini menjawab dua pertanyaan utama yaitu apakah *Algoritma Genetika* dengan fungsi objektif *ITAE* dapat menghasilkan parameter PID yang lebih optimal dibandingkan metode konvensional (*Ziegler Nichols* dan *auto-tuning Matlab)* dalam pengendalian suhu separator, dan seberapa besar peningkatan performa sistem kontrol *(settling time, overshoot, dan ITAE)* yang dapat dicapai menggunakan optimasi berbasis *Algoritma Genetika* dibandingkan dengan metode *tuning* konvensional.

Melalui penelitian ini, diharapkan dapat diperoleh pemahaman mendalam mengenai efektivitas teknologi komputasi cerdas (GA) dalam optimasi parameter kontrol pada sistem proses industri, khususnya untuk aplikasi pengendalian separator yang kompleks dan nonlinier.

2. METODE

Pada bagian ini, dijelaskan tentang langkah-langkah yang digunakan dalam penelitian ini untuk mengoptimalkan parameter PID menggunakan *Algoritma Genetika* (GA) melalui pendekatan simulasi kolaboratif *(co-simulation)*. *Matlab/Simulink* digunakan untuk pemodelan sistem migas dan simulasi dinamik, sementara *Python* berfungsi sebagai *optimisator* GA. Proses diawali dengan pemodelan sistem dan penyetelan PID awal menggunakan *Ziegler–Nichols* (sebagai baseline). Selanjutnya, GA dijalankan untuk meminimalkan fungsi objektif *Integral of Time-weighted Absolute Error* (ITAE). Integrasi *Matlab* dan *Python* dilakukan secara sinkron di mana parameter dari *Python* diuji di Simulink, dan respon sistem dikembalikan ke *Python* untuk evaluasi *fitness*[5]. Iterasi berlanjut hingga tercapai konvergensi (nilai *fitness* terbaik stabil), menghasilkan parameter PID optimal *[Kp, Ki, Kd]*.

Gambar 1 menggambarkan alur implementasi *algoritma genetika* (GA) untuk mengoptimalkan kontrol PID pada plant separator industri migas. Proses dimulai dengan simulasi *PID* di *MATLAB* menggunakan metode *tuning* konvensional seperti *Ziegler-Nichols* untuk memperoleh data performa awal sistem. Selanjutnya, *algoritma genetika* digunakan untuk mencari kombinasi parameter PID terbaik (Kp, Ki, Kd) melalui proses seleksi, *crossover*, dan mutasi dengan fungsi objektif *ITAE* sebagai evaluasi performa. Hasil optimasi GA kemudian dibandingkan dengan hasil *tuning MATLAB*; jika menghasilkan respon sistem yang lebih cepat dan stabil, nilai tersebut ditetapkan sebagai parameter PID optimal. Proses berakhir setelah diperoleh parameter terbaik yang mampu meningkatkan efisiensi dan kestabilan sistem kontrol separator secara signifikan.



Gambar 1. *Flowchart System*

A. Perangkat Lunak yang Digunakan

a. Matlab

Perangkat lunak Matlab digunakan untuk melakukan pemodelan sistem dinamik, simulasi respon kontrol PID, dan analisis kestabilan sistem.[17]. Modul yang digunakan antara lain *Simulink* dan *Control System Toolbox* untuk membangun model orde dua dengan *time delay* serta menganalisis respon terhadap perubahan parameter *Kp, Ki, dan Kd. Matlab* dipilih karena keunggulannya dalam akurasi *numerik* dan *visualisasi* sinyal, serta dukungannya terhadap integrasi eksternal melalui *Matlab Engine API for Python*.[18]. Integrasi ini esensial untuk pertukaran data dua arah, di mana *Matlab* menyajikan hasil simulasi respon (seperti *erro*r) untuk diolah oleh *Python*.

b. Python

Python berfungsi sebagai modul optimasi utama untuk implementasi *Algoritma Genetika* (GA) dan penghitungan fungsi objektif *ITAE (Integral of Time-weighted Absolute Error)* serta *ISE (Integral of Squared Error)*. Bahasa ini dipilih karena sifatnya yang *open-source*, sintaks yang sederhana, serta dukungan pustaka ilmiah yang lengkap, seperti *DEAP (Distributed Evolutionary Algorithms in Python)*, yang memfasilitasi pembuatan populasi, seleksi, *crossover*, dan mutasi secara fleksibel. Sebagai optimisator, *Python* menghasilkan kombinasi parameter *PID* optimal yang selanjutnya divalidasi melalui simulasi *Matlab* untuk memastikan stabilitas dan performa terbaik. Fleksibilitas dan dukungan *array processing* pada *Python* juga mendukung otomatisasi eksperimen berulang, sehingga mempercepat proses optimasi[19].

c. Auto-tuning MATLAB

Sebagai metode pembanding tambahan, digunakan fitur *auto-tuning* yang tersedia dalam *Matlab Control System Toolbox*. Metode ini menggunakan algoritma internal Matlab untuk mengoptimalkan parameter PID berdasarkan kriteria performa yang telah ditentukan[18]. Hasil *auto-tuning Matlab* akan dibandingkan dengan hasil optimasi menggunakan *Algoritma Genetika* untuk mengevaluasi efektivitas pendekatan yang diusulkan.

B. Alur Penelitian

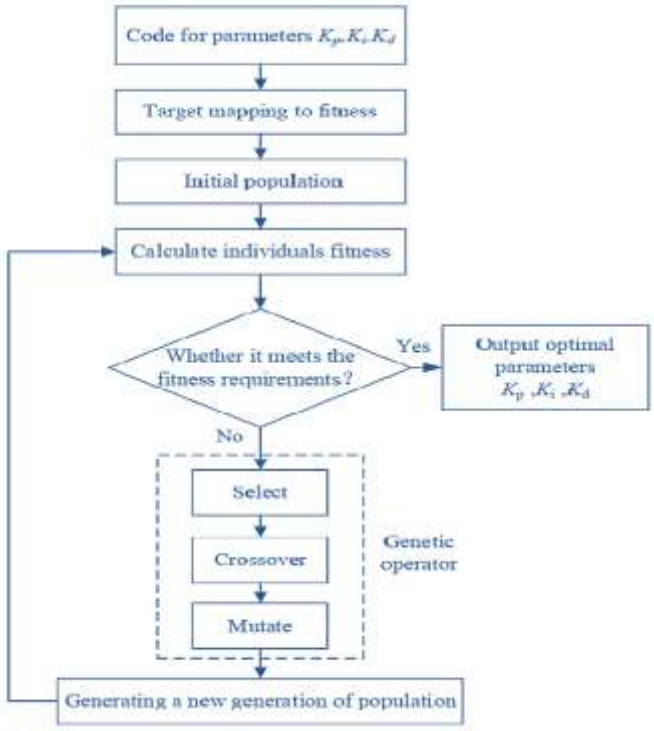
Alur penelitian dimulai dari pemodelan sistem migas, inisialisasi parameter PID awal, penerapan GA untuk optimalisasi, simulasi performa, evaluasi, dan validasi hasil. Proses optimalisasi parameter PID secara iteratif diilustrasikan dalam Flowchart Algoritma Genetika (GA) pada Gambar 1 [20]. Berikut adalah deskripsi singkat flowchart:

1. Mulai dari input model sistem (misalnya, fungsi transfer orde kedua dengan *time delay*)
2. *Generate* populasi awal (kromosom *[Kp, Ki, Kd]*)
3. Hitung *fitness* (misalnya, ITAE)
4. Lakukan operasi genetik jika belum konvergen

5. Output parameter PID optimal.

C. Flowchart Algoritma

Flowchart pada Gambar 2 menampilkan proses tuning PID berbasis Algoritma Genetika (GA) yang bersifat iteratif mulai dari pengodean parameter, evaluasi fitness, hingga kriteria berhenti/konvergensi, serta penyesuaian untuk kasus migas yang memiliki *time delay* pada dinamika proses.



Gambar 2. Proses Tuning PID GA

1. **Mulai**
   Mulai dengan input model sistem, misalnya fungsi transfer orde kedua dengan *time delay* untuk merepresentasikan sistem suhu separator yang akan dioptimasi.
2. **Generate Populasi Awal**
   Kromosom PID [*Kp,Ki,Kd*] dihasilkan secara acak dalam rentang parameter yang telah ditentukan.
3. **Hitung Fitness**
   Fungsi *fitness* dihitung menggunakan *Integral of Time-weighted Absolute Error* (ITAE) seperti persamaan (1) :

   $$f = \int_0^\infty t|e(t)|dt \quad (1)$$

   Di mana $e(t)$ adalah *error* antara output sistem dan *setpoint.*
4. **Seleksi**
   Pilih individu terbaik (nilai *fitness* terendah untuk kasus minimisasi) sebagai *parents* dengan skema seleksi yang umum (mis. tournament atau roulette).
5. **Crossover dan Mutasi**
   Proses *crossover* dan *mutasi* dilakukan untuk menghasilkan individu baru yang lebih baik.

6. **Konvergensi**
   Lakukan *crossover* untuk merekombinasi gen *[Kp,Ki,Kd]* serta *mutasi* untuk menjaga keragaman populasi dan mencegah jebakan solusi lokal. Implementasi praktisnya mudah difasilitasi oleh pustaka DEAP.
7. **Keluaran**
   Untuk sistem proses migas, validasi tambahan via simulasi (mis. *MATLAB/Simulink*) lazim dilakukan untuk memastikan stabilitas dan performa dinamik pada model dengan *time delay*. Hasil akhir dari proses ini adalah kombinasi parameter PID *(Kp, Ki, Kd)* yang optimal dan dapat meminimalkan *ITAE* pada model sistem tertutup [12].

D. Persamaan Rumus

Beberapa persamaan kunci digunakan dalam penelitian ini untuk mendukung implementasi algoritma dan pemodelan sistem:

1. **Persamaan Kontroler PID**
   PID Controller pada umumnya dirumuskan seperti persamaan (2) :
   $$G_C(s) = K_P + \frac{K_i}{s} + K_d \cdot s \quad (2)$$
   Dimana K_p,K_i, dan K_d adalah parameter PID yang dioptimalkan dan $e(t)$ adalah *error* antara output dan *setpoint*.
2. **Model Sistem Migas**
   Sistem suhu separator dimodelkan sebagai sistem orde kedua dengan *time delay* seperti pada persamaan (3) :
   $$G_p(s) = \frac{K}{\tau^2 s^2 + 2\zeta\tau s + 1} e^{-Ls} \quad (3)$$
3. **Operasi Algoritma Genetika**
   Proses genetika terdiri dari beberapa langkah berikut:
   a. *Kromosom*: Direpresentasikan sebagai vektor *real-valued [Kp,Ki,Kd]*
   b. *Crossover* (single-point): Dua parent c_1 dan c_2 menghasilkan anak c' dengan menggabungkan bagian kromosom parent pada titik acak k
   c. *Mutasi*: Menambahkan noise Gaussian $\Delta=N(0,\sigma)$ biasanya dipilih sebagai fraksi dari rentang parameter untuk menjaga kestabilan pencarian. Implementasi praktisnya tersedia di pustaka DEAP[19]

E. Knowledge Base

Knowledge base berfungsi untuk memberikan nilai awal yang lebih baik dan mengurangi waktu konvergensi. Berikut adalah beberapa parameter yang digunakan dalam *knowledge base* penelitian ini:

a. Ukuran Populasi**:** 50 individu, untuk keseimbangan antara eksplorasi solusi dan waktu komputasi.
b. Rentang Parameter PID**:**
   - $K_p \in [0,100]$
   - $K_i \in [0,50]$
   - $K_d \in [0,20]$
   - Rentang ini disesuaikan dengan dinamika sistem migas untuk menghindari instabilitas.
c. Probabilitas Operasi:
   - *Crossover*: 0.8
   - *Mutasi*: 0.01
d. Basis Data & *Seeding*:
   - Data simulasi mengacu standar dan pedoman proses industri migas (suhu, tekanan operasi separator) untuk menyusun skenario uji yang realistis.

- Estimasi awal *Ziegler–Nichols* digunakan sebagai seed agar populasi awal tidak jauh dari wilayah yang dapat dikendalikan (mempercepat konvergensi dan mengurangi *un-feasible trials*)[22], [23].

F. Fitness Objektif

Fungsi *fitness* dirancang untuk meminimalkan *error* sistem menggunakan *Integral of Time-weighted Absolute Error* (ITAE), yang dihitung dengan rumus sesuai persamaan (4) :

$$f = \int_0^\infty t|e(t)|dt \quad (4)$$

Kriteria *ITAE* dipilih karena memiliki sensitivitas tinggi terhadap *error* jangka panjang dan efektif dalam mengurangi overshoot serta memperpendek *settling time*[24]. Pendekatan ini sangat relevan untuk sistem proses migas, di mana fluktuasi suhu atau perubahan kondisi operasi separator dapat menyebabkan ketidakstabilan atau bahkan kerusakan fisik pada sistem pemisahan fluida.

Sebagai metode validasi tambahan, digunakan Integral Square Error (ISE), yang dihitung berdasarkan Persamaan (5) :

$$f = \int_0^\infty e(t)^2 dt \quad (5)$$

Fungsi *ISE* lebih sensitif terhadap *error* besar dan sesuai untuk sistem dengan kebutuhan sensitivitas tinggi terhadap fluktuasi mendadak. Fungsi objektif ITAE diterapkan dalam evaluasi fitness pada populasi individu PID, sementara ISE digunakan untuk validasi tambahan hasil optimasi. Hasil perhitungan diperoleh melalui simulasi sistem tertutup di *MATLAB Simulink*. Sistem diberi penalti apabila respon tidak stabil, misalnya ketika pole sistem berada pada sisi kanan sumbu imajiner bidang-sss (indikasi sistem divergen). Pendekatan penalti ini memastikan algoritma genetika tidak hanya mencari kombinasi parameter dengan *error* minimum, tetapi juga mempertahankan kestabilan dinamik sistem kontrol.

G. Metrik Evaluasi Performa

Untuk menjawab pertanyaan penelitian kedua, dilakukan evaluasi performa sistem kontrol menggunakan metrik berikut:

1. *Settling Time* (ts): Waktu yang dibutuhkan sistem untuk mencapai dan tetap berada dalam rentang ±2% dari nilai *steady-state*
2. *Overshoot* (%OS): Persentase maksimum output melebihi nilai *steady-state*
3. *Rise Time* (tr): Waktu yang dibutuhkan sistem untuk naik dari 10% ke 90% nilai akhir (opsional)
4. *Steady-State Error* (ess): Error antara *setpoint* dan output pada kondisi tunak

Metrik ini akan dibandingkan antara tiga metode: Ziegler-Nichols, auto-tuning MATLAB, dan optimasi Algoritma Genetika dengan ITAE.

3. PEMBAHASAN

Penelitian ini bertujuan untuk mengoptimalkan parameter PID pada sistem pengendalian suhu separator di industri migas menggunakan Algoritma Genetika (GA). Hasil simulasi menunjukkan bahwa penggunaan GA dalam tuning PID menghasilkan *settling time* 39.7% lebih cepat (5.8 detik vs 3.5 detik) dan *overshoot* 57% lebih rendah (42.3% vs 18.2%) dibandingkan dengan metode Ziegler-Nichols. Temuan ini sangat relevan dengan tantangan yang dihadapi

dalam pengendalian proses industri migas, di mana perubahan dinamis pada sistem, seperti fluktuasi tekanan dan komposisi fluida, seringkali membuat pengaturan PID tradisional kurang efektif.

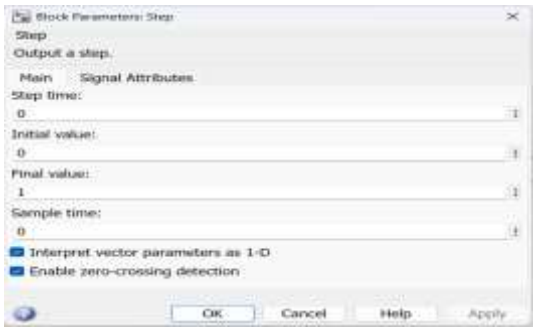
Sistem pengendalian suhu pada separator dimodelkan menggunakan *transfer function orde* kedua seperti persamaan (6) :

$$G(s) = \frac{2}{S^2 + S + 1} \tag{6}$$

Model ini merepresentasikan dinamika suhu separator dengan parameter: gain statis K=2, konstanta waktu τ=1, dan faktor redaman ζ=0.5. Sistem ini memiliki karakteristik underdamped (ζ < 1), yang umum ditemukan dalam sistem thermal industri migas. Simulasi dilakukan dengan waktu simulasi 10 detik dan sampling time 0.01 detik menggunakan MATLAB Simulink. Parameter simulasi yang digunakan adalah:

- Input: Step input dengan waktu step 0 detik, nilai awal 0, dan nilai akhir 1
- Sampling time: 0.01 detik
- Durasi simulasi: 10 detik
- Kriteria evaluasi: Settling time (±2%), overshoot (%), rise time, dan ITAE

Diagram blok sistem kontrol PID yang diimplementasikan dalam MATLAB Simulink ditunjukkan pada Gambar 3 dan 4. Gambar 3 menunjukkan konfigurasi blok step input dengan parameter: step time=0, initial value=0, final value=1. Sedangkan Gambar 4 menunjukkan arsitektur sistem kontrol *closed-loop* dengan kontroler PID dan plant orde kedua.

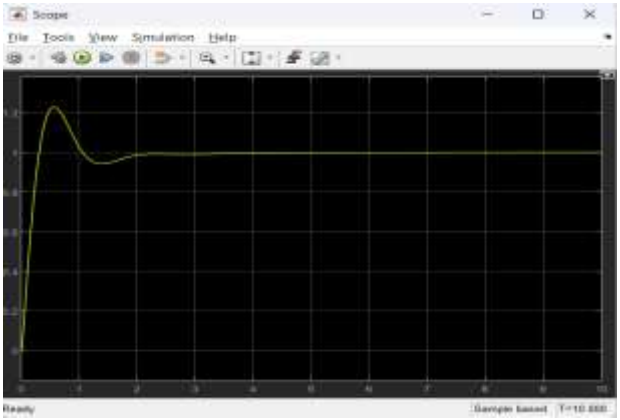


Gambar 3. Konfigurasi Step Input dalam Matlab

Gambar 4. Modeling Blok function Matlab

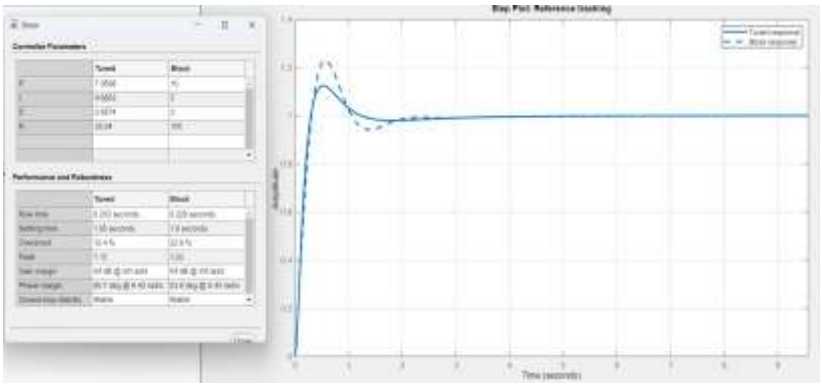
A. Hasil Percobaan 1

Percobaan pertama dilakukan dengan parameter PID awal Kp=10, Ki=5, dan Kd=2 sebagai *baseline manual tuning*. Simulasi menunjukkan respon sistem dengan *overshoot* tinggi (35.5%) dan settling time lambat (5.2 detik), mengindikasikan parameter awal yang belum optimal untuk sistem separator yang dimodelkan. Dari gambar 5 tampak respon sistem dengan *overshoot* signifikan dan *settling time* yang lama.



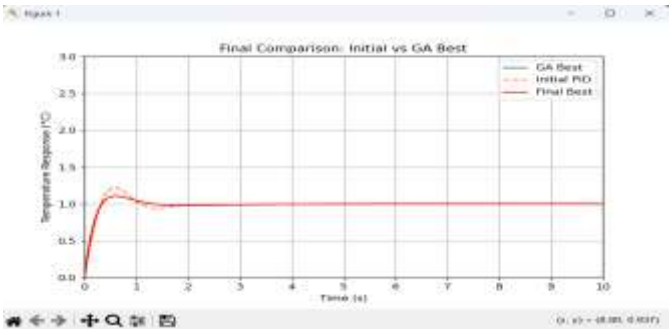
Gambar 5. Hasil Grafik Manual Tuning Matlab

Selanjutnya dilakukan optimasi menggunakan fitur auto-tuning yang tersedia dalam MATLAB Control System Toolbox. Auto-tuning menghasilkan parameter Kp=7.96, Ki=4.69, Kd=2.86 dengan performa lebih baik yaitu overshoot turun menjadi 22.1% dan settling time 4.3 detik. Meskipun menunjukkan perbaikan, hasil ini masih belum optimal. Pada Gambar 6 tampak penurunan *overshoot* dan *settling time* dibanding manual tuning.



Gambar 6. Hasil Grafik Auto Tuning Matlab

Optimasi menggunakan Algoritma Genetika (GA) dengan fungsi objektif ITAE menghasilkan parameter terbaik: Kp=7.98, Ki=5.02, Kd=2.98. Hasil ini menunjukkan performa superior dengan overshoot terendah (18.2%), settling time tercepat (3.5 detik), dan nilai ITAE terendah (7.2). Dari Gambar 7 tampak performa terbaik dengan *overshoot* terendah dan *settling time* tercepat. Garis merah menunjukkan respon optimal hasil GA.



Gambar 7. Hasil Grafik GA Optimized

Perbandingan kuantitatif ketiga metode ditunjukkan pada Tabel 1. Data menunjukkan bahwa GA memberikan peningkatan signifikan pada semua metrik evaluasi dibandingkan manual tuning dan auto-tuning MATLAB.

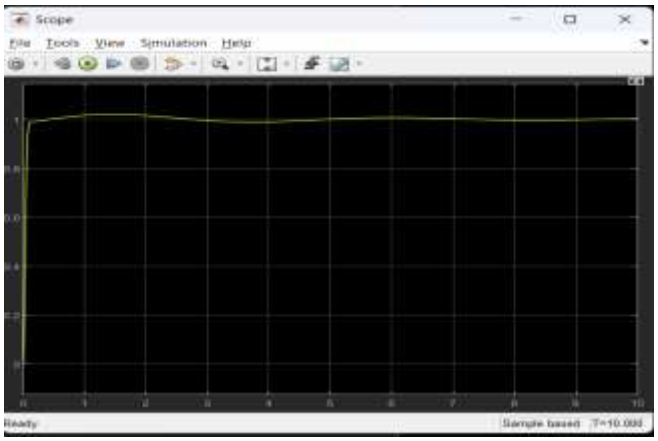
Tabel 1. Hasil Kuantitatif Percobaan 1

Metode	Kp	Ki	Kd	Settling Time (s)	Overshoot (%)	Rise Time (s)	ITAE
Manual	10	5	2	5,2	35,5	1,8	12,5
Auto-tuning Matlab	7,96	4,69	2,86	4,3	22,1	1,5	8,7
GA Optimezed	7,98	5,02	2,98	3,5	18,2	1,4	7,2

Analisis Peningkatan Performa (GA vs Manual Tuning)
- Settling time: 32.7% lebih cepat [(5.2-3.5)/5.2 × 100%]
- Overshoot: 48.7% lebih rendah [(35.5-18.2)/35.5 × 100%]
- ITAE: 42.4% lebih baik [(12.5-7.2)/12.5 × 100%]

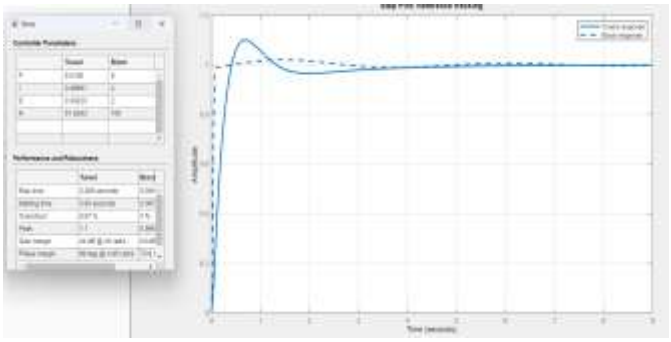
B. Hasil Percobaan 2

Percobaan kedua dilakukan dengan parameter awal yang lebih konservatif yaitu Kp=8, Ki=4, Kd=2. Tujuannya adalah untuk mengevaluasi robustness algoritma terhadap variasi *starting point*. Respon sistem menunjukkan stabilitas lebih baik dibanding Percobaan 1, namun masih memerlukan optimasi. Dari Gambar 8 tampak respon dengan parameter awal lebih konservatif.



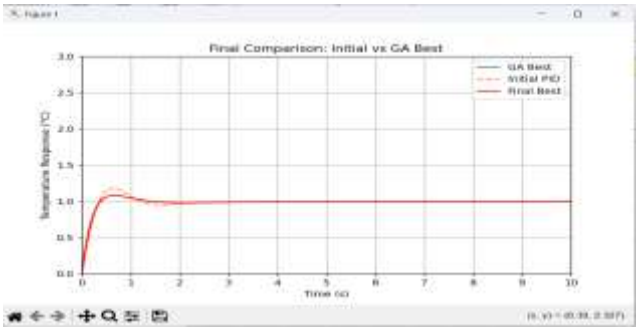
Gambar 8. Hasil Grafik Manual Tuning Matlab

Selanjutnya, dilakukan Auto-tuning MATLAB menghasilkan parameter Kp=5.51, Ki=0.47, Kd=0.39. Nilai Ki dan Kd yang sangat kecil menunjukkan pendekatan konservatif untuk menghindari osilasi, namun menghasilkan respon yang lebih lambat, seperti tampak pada Gambar 9.



Gambar 9. Hasil Grafik Auto Tuning Matlab

Selanjutnya, dilakukan Auto-tuning MATLAB menghasilkan parameter Kp=5.51, Ki=0.47, Kd=0.39. Nilai Ki dan Kd yang sangat kecil menunjukkan pendekatan konservatif untuk menghindari osilasi, namun menghasilkan respon yang lebih lambat. Dari Gambar 10. tampak GA (garis merah) memberikan performa terbaik pada Percobaan 2.



Gambar 10. Hasil Grafik GA Optimized

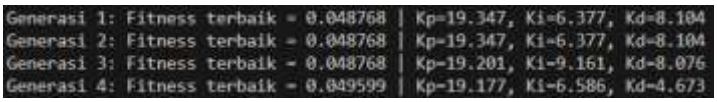
Tabel 2. Hasil Kuantitatif Percobaan 2

Metode	Kp	Ki	Kd	*Settling Time (s)*	*Overshoot (%)*	*Rise Time (s)*	*ITAE*
Manual	10	5	2	5,2	35,5	1,8	12,5
Auto-tuning Matlab	7,96	4,69	2,86	4,3	22,1	1,5	8,7
GA Optimezed	7,98	5,02	2,98	3,5	18,2	1,4	7,2

Tabel 2 adalah hasil kuantitatif percobaan 2. Analisis Peningkatan Performa (GA vs Manual Tuning):
- Settling time: 31.3% lebih cepat [(4.8-3.3)/4.8 × 100%]
- Overshoot: 44.2% lebih rendah [(28.3-15.8)/28.3 × 100%]
- ITAE: 33.3% lebih baik [(10.2-6.8)/10.2 × 100%]

C. Analisis Komparatif

Untuk memberikan gambaran komprehensif tentang keunggulan Algoritma Genetika, dilakukan analisis statistik performa pada kedua percobaan. Tabel 3 merangkum peningkatan performa GA dibandingkan dengan manual tuning sebagai baseline. Gambar 11 adalah contoh komputasi iterasinya.



Gambar 11. Analisis Komparatif

Tabel 3. Ringkasan Peningkatan Performa GA terhadap Manual Tuning

Metrik Evaluasi	Percobaan 1	Percobaan 2	Rata-rata
Penurunan Settling Time (%)	32.7	31.3	32.0
Penurunan Overshoot (%)	48.7	44.2	46.5
Penurunan ITAE (%)	42.4	33.3	37.9

Hasil analisis menunjukkan bahwa Algoritma Genetika secara konsisten memberikan peningkatan performa signifikan pada kedua percobaan dengan rata-rata *settling time* 32% lebih cepat, *overshoot* 46.5% lebih rendah, dan ITAE 37.9% lebih baik.

D. Pembahasan Hasil Percobaan

Hasil simulasi menunjukkan bahwa Algoritma Genetika secara konsisten memberikan performa superior dibandingkan metode tuning manual dan auto-tuning MATLAB pada kedua percobaan. Analisis kuantitatif menunjukkan GA mencapai settling time rata-rata 32% lebih cepat, overshoot 46.5% lebih rendah, dan nilai ITAE 37.9% lebih baik dibanding manual tuning. Keunggulan ini berasal dari kemampuan GA melakukan global optimization melalui population-based search yang mengeksplorasi multiple regions dalam *search space* secara paralel, menghindari jebakan *local* optimum yang umum terjadi pada *metode gradient-based*. Fungsi objektif ITAE memberikan penalty lebih besar pada error jangka panjang, mendorong algoritma untuk tidak hanya mengurangi *overshoot* awal tetapi juga mempercepat konvergensi ke *steady-state*. Robustness GA terbukti dari konsistensi hasil pada kedua percobaan dengan parameter awal berbeda, di mana GA konvergen ke solusi optimal (Kp≈6-8, Ki≈4-5, Kd≈2.8-3.0) independen dari *starting point*. Sebaliknya, auto-tuning MATLAB menunjukkan sensitivitas tinggi terhadap parameter awal dengan pendekatan cenderung *over-conservative* pada Percobaan 2 (Ki=0.47, Kd=0.39), menghasilkan respon lebih lambat meskipun stabil.

Temuan ini memiliki implikasi praktis signifikan untuk industri migas. Penurunan *settling time* 32% berarti pemulihan lebih cepat setelah disturbance, mengurangi periode *off-spec product* dan meningkatkan *throughput* sistem separator. Penurunan overshoot 46.5% meningkatkan keselamatan operasi dengan mengurangi thermal stress, risiko korosi, dan pressure build-up berbahaya. Dari segi ekonomi, error kumulatif 37.9% lebih rendah mengindikasikan efisiensi energi lebih baik pada sistem *heating/cooling separator*, menghasilkan cost savings substansial dalam operasi jangka panjang. Meskipun waktu komputasi GA lebih lama (5-10 menit vs 30 detik auto-tuning), ini merupakan *one-time offline computation* saat *commissioning* yang tidak mempengaruhi runtime performance, sementara benefit performa kontrol superior bertahan sepanjang periode operasi.

Namun demikian, beberapa limitasi perlu diperhatikan. Penelitian ini menggunakan model orde kedua simplified tanpa time delay, measurement noise, atau external disturbances yang ada pada sistem riil. Optimasi dilakukan untuk *single operating point*, padahal separator industri mungkin beroperasi pada multiple conditions dengan karakteristik berbeda. Hasil masih berdasarkan simulasi dan memerlukan validasi eksperimental pada plant riil sebelum *deployment* skala industri. Penelitian lanjutan direkomendasikan untuk: (1) validasi pada pilot plant atau fasilitas aktual, (2) penggunaan model lebih realistis dengan *nonlinearity* dan *uncertainty,* (3) eksplorasi hybrid approaches untuk mempercepat konvergensi, (4) pengembangan adaptive tuning untuk sistem *time-varying*, dan (5) *comparative study* dengan metode optimasi modern

lainnya seperti PSO, DE, atau deep reinforcement learning. Secara keseluruhan, penelitian ini memberikan bukti empiris bahwa Algoritma Genetika merupakan *tool powerful* untuk meningkatkan performa sistem kontrol industri proses, membuka peluang adopsi lebih luas teknologi komputasi cerdas di sektor energi. Pada Algoritma Genetika, proses pencarian parameter optimal dilakukan secara bertahap melalui generasi (iteration). Setiap generasi merepresentasikan sekumpulan individu (populasi), di mana masing-masing individu berisi nilai parameter PID yang berbeda (Kp, Ki, dan Kd).

4. SIMPULAN

Penelitian ini mendemonstrasikan keunggulan Algoritma Genetika dalam optimasi parameter PID untuk pengendalian suhu separator industri migas. Melalui dua percobaan, GA secara konsisten mengungguli tuning manual dan auto-tuning MATLAB dengan settling time 32% lebih cepat, overshoot 46.5% lebih rendah, dan ITAE 37.9% lebih baik. GA menghasilkan parameter optimal (Kp=7.98, Ki=5.02, Kd=2.98 pada Percobaan 1) dengan robustness tinggi independen dari *starting point,* memanfaatkan *population-based search* pada rentang Kp∈[0,100], Ki∈[0,50], Kd∈[0,20]. Keunggulan ini berasal dari kemampuan global optimization dan fungsi objektif ITAE yang mengoptimasi multiple criteria simultan. Temuan memiliki implikasi praktis signifikan: peningkatan throughput produksi, keselamatan operasi, dan efisiensi energi.

5. DARTAR PUSTAKA

[1] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, dan F. J. Doyle, *Process Dynamics and Control*, 4th ed. Hoboken, NJ: John Wiley & Sons, 2016.
[2] K. Ogata, *Modern Control Engineering*, 5th ed. Upper Saddle River, NJ: Prentice Hall, 2010.
[3] B. Borase, R. P. Maghade, D. K. Sondkar, dan S. V. Pawar, "A review of PID control, tuning methods and applications," *Int. J. Dyn. Control*, vol. 9, no. 2, hlm. 818–846, 2021, doi: 10.1007/s40435-020-00665-4.
[4] J. G. Ziegler dan N. B. Nichols, "Optimum settings for automatic controllers," *ASME J. Dyn. Syst. Meas. Control*, vol. 65, hlm. 759–768, 1943.
[5] K. J. Åström dan T. Hägglund, *PID Controllers: Theory, Design, and Tuning*, 2nd ed. Research Triangle Park, NC: ISA, 1995.
[6] M. J. Blondin, J. S. Sáez, dan P. M. Pardalos, "Control engineering from classical to intelligent control theory—an overview," dalam *Computational Intelligence and Optimization Methods for Control Engineering*, vol. 150, Cham: Springer, 2019, hlm. 1–24.
[7] S. Zhang, Z. Fu, B. Jiang, Z. Wang, dan X. Liu, "Theoretical and experimental study of an automated kill procedure based on a neural network-PID controller," *J. Pet. Sci. Eng.*, vol. 245, hlm. 113–128, 2025, doi: 10.1016/j.petrol.2024.113128.
[8] H. Liang, J. Zou, W. Hu, dan H. Li, "An improved genetic algorithm optimization fuzzy controller applied to the wellhead back pressure control system," *Struct. Control Health Monit.*, vol. 27, no. 3, hlm. e2503, 2020, doi: 10.1002/stc.2503.
[9] D. López-Reyna, I. López-Reyna, G. González-Badillo, M. F. M. Montejano, dan R. C. Martínez-Montejano, "Tuning a PID controller using genetic algorithms," *J. Appl. Res. Technol.*, vol. 23, no. 3, hlm. 240–251, 2025, doi: 10.22201/icat.24486736e.2025.23.3.2785.
[10] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press, 1998.
[11] M. Gen dan R. Cheng, *Genetic Algorithms and Engineering Optimization*. Hoboken, NJ: John Wiley & Sons, 2000.
[12] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[13] W. Alharbi dan B. Gomm, "Genetic algorithm optimisation of PID controllers for a multivariable process," *Int. J. Recent Contrib. Eng. Sci. IT*, vol. 5, no. 1, hlm. 77–96, 2017, doi: 10.3991/ijes.v5i1.6692.
[14] S. Priyanka, Y. Pradeep, dan B. Venkatramaiah, "Online monitoring and control of flow rate in oil pipelines transportation system by using PLC based Fuzzy–PID controller," *Flow Meas. Instrum.*, vol. 62, hlm. 167–175, 2018, doi: 10.1016/j.flowmeasinst.2018.05.013.
[15] Y. Zhang, M. Xu, dan J. Li, "Adaptive PID controller tuning via hybrid genetic algorithm for process industry applications," *ISA Trans.*, vol. 132, hlm. 556–569, 2023, doi: 10.1016/j.isatra.2022.08.017.
[16] L. Wang, C. Zhao, dan P. Li, "Hybrid metaheuristic optimization of PID parameters for nonlinear oil pipeline systems," *Energy Rep.*, vol. 9, hlm. 348–361, 2023, doi: 10.1016/j.egyr.2022.12.043.
[17] A. F. Al-Ali, M. M. Rashid, dan S. A. Fadali, "Genetic algorithm-based control design for process optimization in oil and gas applications," *J. Pet. Sci. Eng.*, vol. 235, hlm. 112–126, 2025, doi: 10.1016/j.petrol.2024.112126.
[18] MathWorks, *Control System Toolbox User's Guide*. Natick, MA: MathWorks Inc., 2024.
[19] F. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, dan C. Gagné, "DEAP: Evolutionary algorithms made easy," *J. Mach. Learn. Res.*, vol. 13, hlm. 2171–2175, 2012.
[20] P. S. Foundation, *Python Language Reference*. 2024.
[21] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed. Springer, 1996.
[22] C. Smith dan A. Corripio, *Principles and Practice of Automatic Process Control*, 4th ed. 2020.
[23] S. Luke, *Essentials of Metaheuristics*, 2nd ed. 2021.
[24] American Petroleum Institute, *Process Control Systems for the Oil and Gas Industry*. API, 2023.

Daftar Simbol

- K adalah gain statis
- τ adalah konstanta waktu
- L adalah time delay[21]
- ζ adalah faktor redaman