

Desain Optimal Mini Turbin Angin untuk Penerangan Perahu Nelayan Menggunakan Algoritma Genetika dan Optimasi Partikel Swarm

Parman^{1*}, Fais Hamzah², Rahmat Basya Shahryts Tsany³, Dicki Nizar Zulfika⁴, Thomas Brian⁵

¹Jurusan Teknik Bangunan Kapal, Politeknik Perkapalan Negeri Surabaya

^{2,3,4}Jurusan Teknik Permesinan Kapal, Politeknik Perkapalan Negeri Surabaya

⁵Jurusan Teknik Kelistrikan Kapal, Politeknik Perkapalan Negeri Surabaya

e-mail: parman@ppns.ac.id¹, fais.hamzah@ppns.ac.id², rahmatbasya@ppns.ac.id³, nizar.zulfika@ppns.ac.id⁴, thomasbrian@ppns.ac.id⁵

Abstrak

Penelitian ini mengkaji konfigurasi optimal mini turbin angin untuk penerangan perahu nelayan, sebuah solusi krusial untuk mengatasi tantangan keterbatasan energi dan ruang di laut. Mengingat kompleksitas desain turbin yang multifaktor, pendekatan konvensional sering kali tidak efisien. Oleh karena itu, penelitian ini memperkenalkan metodologi optimasi berbasis komputasi menggunakan Algoritma Genetika (GA) dan Particle Swarm Optimization (PSO) untuk secara sistematis mengidentifikasi parameter desain terbaik. Sistem dimodelkan dalam Matlab Simulink untuk mendapatkan parameter awal seperti kecepatan angin, ukuran blade, kecepatan sudut, dan torsi. Sebuah fungsi objektif kemudian ditetapkan untuk meminimalkan deviasi daya dari target 100 Watt sekaligus mengoptimalkan dimensi blade, dimana Algoritma PSO dan GA secara iteratif menyesuaikan parameter desain turbin. Hasil optimasi menunjukkan bahwa kedua algoritma secara konsisten menghasilkan daya output sebesar 102.55 W dengan ukuran blade optimal 0.45 m, yang melampaui target 100 Watt. Desain optimal ini memberikan solusi sistem energi untuk perahu nelayan dan berkinerja tinggi untuk peningkatan keselamatan dan operasional nelayan, sekaligus berkontribusi signifikan pada kemajuan teknologi energi angin skala kecil yang berkelanjutan.

Kata kunci: GA, PSO, mini turbin angin, optimasi, perahu nelayan

Abstract

This study aims to identify the optimal configuration of a mini wind turbine for fishing boat illumination, addressing challenges of energy efficiency and space constraints, by implementing Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) to optimize its design. The system was modeled in Matlab Simulink to obtain initial parameters such as wind speed, blade size, angular velocity, and torque. An objective function was then established to minimize power deviation from a 100 Watt target while simultaneously optimizing blade dimensions, where both PSO and GA algorithms iteratively adjusted the turbine design parameters. Optimization results consistently showed that both algorithms yielded a power output of 102.55 W with an optimal blade size of 0.45 m, exceeding the 100 Watt target. Specifically, PSO optimization achieved this result at a wind speed of 2.06 m/s, an angular velocity of 12.63 rad/s, and a torque of 0.50 Nm. The validated optimal design provides a pragmatic and high-performance solution for enhancing the safety and operational efficiency of fishermen, while also contributing significantly to the advancement of sustainable small-scale wind energy technology.

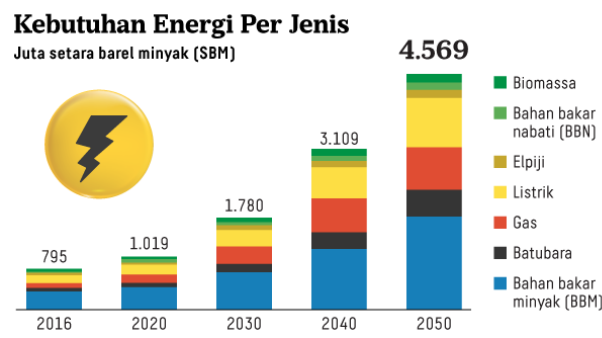
Keywords: GA, PSO, mini wind turbine, optimization, fishing boat

^{1*} parman@ppns.ac.id

1. Pendahuluan

Energi angin, khususnya melalui mini turbin angin, menawarkan harapan besar sebagai solusi energi terbarukan. Pemanfaatannya menjadi krusial untuk mengurangi ketergantungan pada bahan bakar fosil dan mengatasi perubahan iklim. Di antara semua energi terbarukan, tenaga angin menonjol sebagai teknologi paling menjanjikan dan berkembang pesat dalam transisi energi global. Pertumbuhan kapasitas pembangkit listrik tenaga angin didorong oleh dukungan kebijakan dan penurunan biaya yang signifikan, khususnya pada teknologi surya dan angin. Pada tahun 2023, kapasitas angin terpasang secara global mencapai 1015 GW, dengan mayoritas (93%) di darat dan sisanya (7%) di lepas pantai. Produksi listrik tenaga angin global juga sangat impresif, mencapai lebih dari 2330 TWh pada tahun 2023, naik 10% dari tahun sebelumnya, menjadikannya teknologi terbarukan non-hidro terkemuka (IEA, 2025).

Meskipun tren global ini positif terhadap energi angin, kebutuhan energi di Indonesia diproyeksikan masih sangat bergantung pada bahan bakar fosil, seperti yang terlihat dari grafik "Kebutuhan Energi Per Jenis". Grafik ini menunjukkan bahwa total kebutuhan energi nasional akan melonjak dari 795 juta SBM pada tahun 2016 menjadi 4.569 juta SBM pada tahun 2050. Lonjakan kebutuhan ini, terutama pada sektor listrik, menyoroti pentingnya diversifikasi energi. Oleh karena itu, pengembangan mini turbin angin, seperti yang dijelaskan dalam penelitian ini, tidak hanya krusial untuk mengurangi ketergantungan pada bahan bakar fosil dan mengatasi perubahan iklim, tetapi juga menjadi solusi inovatif untuk memenuhi kebutuhan energi spesifik di lokasi terpencil, seperti perahu nelayan, yang tidak terdakup oleh jaringan listrik konvensional.



Gambar 1. *Outlook* energi Indonesia (Perpustakaan Emil Salim, 2018)

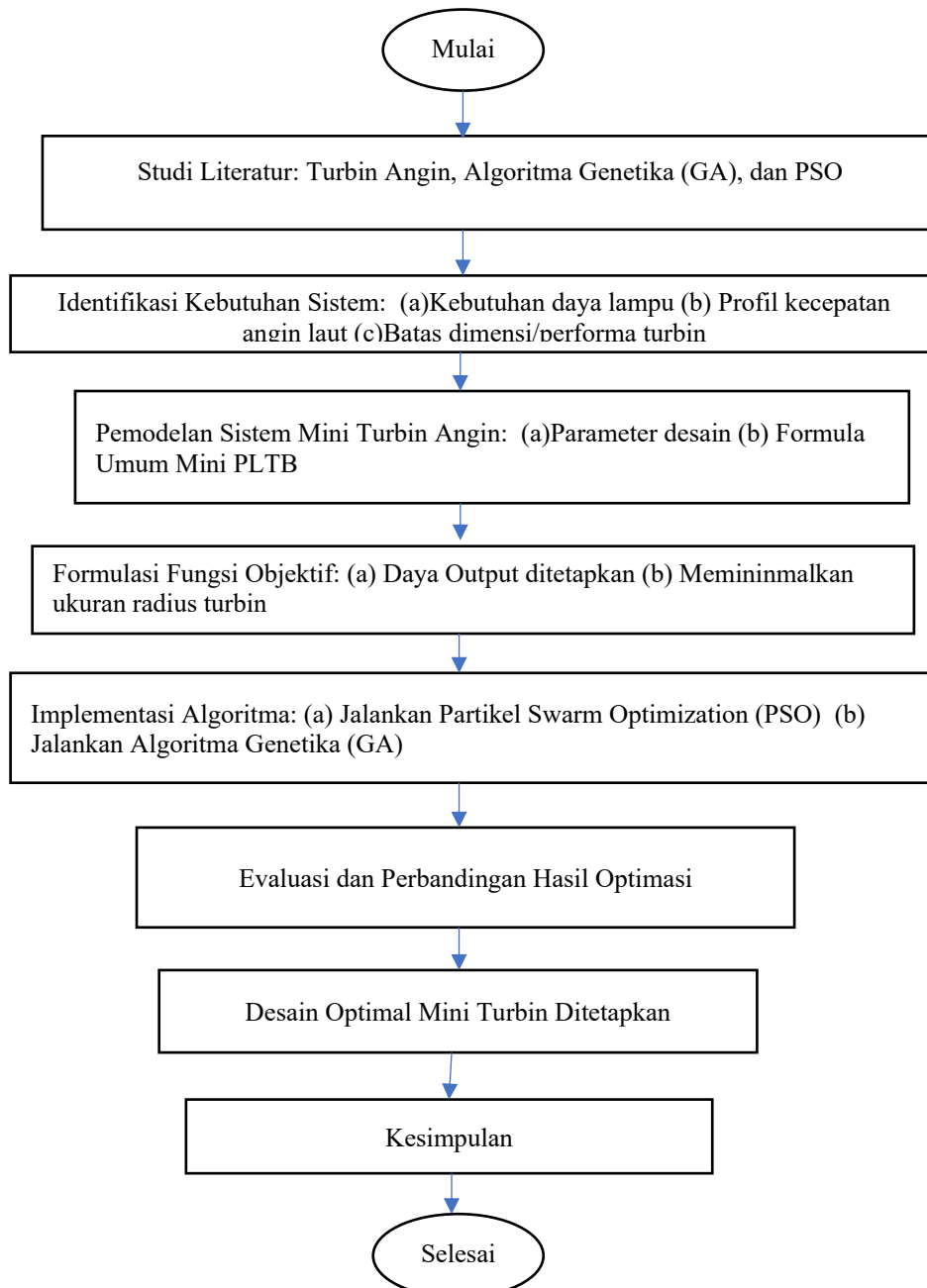
Di tengah perkembangan skala utilitas energi angin, aplikasi skala kecil, seperti mini turbin angin, juga menunjukkan relevansi yang tinggi, terutama untuk memenuhi kebutuhan energi di lokasi terpencil atau spesifik. Salah satu sektor yang sangat diuntungkan adalah perikanan, di mana nelayan seringkali beroperasi di perairan lepas pantai pada malam hari dan memerlukan sumber penerangan yang andal. Saat ini, banyak perahu nelayan masih bergantung pada sumber listrik konvensional seperti generator berbahan bakar fosil. Ketergantungan ini tidak hanya membebani nelayan dengan biaya operasional yang tinggi akibat fluktuasi harga bahan bakar, tetapi juga berkontribusi pada polusi udara dan emisi karbon yang mempercepat dampak perubahan iklim. Selain itu, keterbatasan ruang di kapal dan kebutuhan akan sistem yang ringan serta tahan terhadap kondisi laut yang keras menjadi kendala signifikan dalam implementasi solusi energi konvensional. Oleh karena itu, pengembangan mini turbin angin yang optimal untuk penerangan perahu nelayan menjadi krusial untuk meningkatkan efisiensi, mengurangi biaya, dan mendukung praktik perikanan yang lebih berkelanjutan. Salah satu aplikasi inovatif dari teknologi tenaga angin adalah mini turbin angin, yang memiliki potensi besar untuk memenuhi kebutuhan energi listrik skala kecil dan spesifik, seperti penerangan pada perahu nelayan. Sektor perikanan, yang sangat bergantung pada operasi malam hari, seringkali menghadapi tantangan dalam penyediaan sumber energi listrik yang efisien dan berkelanjutan. Penggunaan generator berbahan bakar fosil tidak hanya meningkatkan biaya operasional, tetapi juga berkontribusi pada emisi gas rumah kaca. Oleh karena itu, pengembangan dan desain optimal mini turbin angin untuk aplikasi ini menjadi sangat relevan.

Mengingat kompleksitas desain dan kendala operasional mini turbin angin pada perahu nelayan, dalam beberapa tahun terakhir, teknik machine learning (pembelajaran mesin) telah menjadi pendekatan yang sangat menjanjikan untuk mengatasi berbagai tantangan dan meningkatkan kinerja sistem rekayasa. Kemampuan pembelajaran mesin untuk belajar dari data, mengidentifikasi pola, dan membuat prediksi atau keputusan tanpa pemrograman eksplisit memberikan keunggulan signifikan dalam mengoptimalkan sistem yang kompleks. Sejalan dengan itu, Algoritma Genetika (GA) dan Particle Swarm Optimization (PSO) telah terbukti sebagai metode optimasi berbasis populasi yang sangat efektif. GA,

terinspirasi dari prinsip seleksi alam, menggunakan operator genetik untuk mengevolusikan solusi kandidat menuju kondisi optimal (Bhattacharjee & Bhattacharya, 2023), sementara PSO meniru perilaku sosial kawanan hewan, di mana partikel-partikel berbagi informasi untuk menemukan solusi terbaik dalam ruang pencarian (Belgaid dkk., 2020; Zorić, 2023). Mengingat potensi besar mini turbin angin dan kemampuan optimasi dari algoritma cerdas, paper ini bertujuan untuk menyajikan desain optimal mini turbin angin untuk sistem penerangan perahu nelayan. Kami akan mengimplementasikan dan mengevaluasi kinerja GA dan PSO untuk menemukan parameter-parameter terbaik yang dapat mengoptimalkan ukuran dan keandalan sistem (Prasetyo dkk., 2022).

Berdasarkan urgensi permasalahan penerangan perahu nelayan dan kebutuhan akan solusi energi yang efisien dan berkelanjutan, penelitian ini mengusulkan pengembangan desain optimal mini turbin angin. Dengan memanfaatkan keunggulan GA dan PSO sebagai dua metode optimasi yang telah terbukti, studi ini secara spesifik bertujuan untuk mengidentifikasi konfigurasi turbin yang paling efektif dan efisien dalam menghasilkan energi listrik yang stabil untuk kebutuhan penerangan. Diharapkan, solusi optimal yang ditawarkan tidak hanya memberikan dampak positif signifikan terhadap operasional dan keselamatan nelayan, tetapi juga berkontribusi pada pengembangan teknologi energi terbarukan skala kecil yang lebih ramah lingkungan.

2. Metode Penelitian



2.1 Model Mini Turbin Angin

Desain awal turbin mengutamakan performa aerodinamis dan ketahanan struktural di lingkungan laut. Model turbin tersebut kemudian disimulasikan dalam MATLAB Simulink untuk mengevaluasi pembangkitan dayanya pada berbagai kecepatan angin, dengan tujuan mengukur efisiensi konversi energi serta stabilitas sistem dalam kondisi angin yang bervariasi. Dengan kecepatan angin rata-rata di area perahu nelayan Indonesia sekitar 15 knot atau 7,7 m/s (BMKG, 2025), terdapat potensi besar untuk pembangkit listrik tenaga angin. Mini turbin angin ini dirancang untuk menghasilkan listrik secara efisien, bahkan pada kecepatan angin rendah. Desainnya yang ringkas dan portabel memungkinkan pemasangan yang mudah tanpa memerlukan infrastruktur tambahan. Terbuat dari bahan tahan korosi yang ringan, turbin ini dibangun untuk menghadapi elemen pesisir yang keras seperti paparan air laut dan angin kencang.

Dalam perancangan model Pembangkit Listrik Tenaga Bayu (PLTB) di MATLAB Simulink, langkah awal adalah menentukan nilai C_p (koefisien daya), λ (rasio kecepatan ujung bilah), dan β (sudut pitch bilah). Selanjutnya, nilai C_p dapat diperkirakan sekitar 0,39 berdasarkan persamaan (4) dengan $\beta=0$, untuk mendapatkan koefisien daya maksimum (Slah dkk., 2016a).

$P_m = \frac{1}{2} \rho \pi R^2 V^3 C_p(\lambda, \beta)$	(1)
--	-----

Persamaan (1) menjelaskan daya mekanik yang dihasilkan turbin angin, dipengaruhi oleh parameter aerodinamika dan kondisi lingkungan. Daya ini berbanding lurus dengan kerapatan udara, luas sapuan rotor, dan kecepatan angin. Koefisien daya (C_p), yang bergantung pada rasio kecepatan ujung bilah (λ) dan sudut pitch bilah (β), menentukan efisiensi konversi energi angin menjadi energi mekanik. Konstanta 0.5 (setengah) dalam persamaan berasal dari prinsip mekanika fluida yang mengatur seberapa banyak daya bisa diambil dari aliran udara melewati rotor. Persamaan ini menjadi dasar analisis kinerja turbin untuk mengoptimalkan desain dan operasinya (Prmod Jain, 2011).

2.2 Metode PSO

Kode PSO di Python bertujuan untuk mensimulasikan dan menganalisis parameter yang mempengaruhi performa turbin angin. Dataset yang diperoleh dari model yang dibuat pada Matlab Simulink yang mencakup variabel seperti kecepatan angin, ukuran blade, kecepatan sudut, dan torsi, yang dibuat menggunakan distribusi normal untuk mencerminkan variasi alami dalam lingkungan operasi turbin. Setelah data dibuat, kode ini menggunakan teknik normalisasi agar setiap fitur berada dalam rentang skala yang seimbang, memungkinkan model pembelajaran mesin untuk melakukan prediksi dengan lebih akurat.

Selanjutnya, dataset yang telah diproses digunakan dalam model regresi berbasis XGBoost, yang dikenal dengan kemampuannya menangani fitur non-linear dan memberikan performa tinggi dalam tugas prediksi. Model ini diuji menggunakan metrik seperti r-squared (R^2) untuk mengevaluasi keakuratan hasil prediksi. Dengan pendekatan ini, kode dapat membantu memahami bagaimana setiap parameter mempengaruhi keluaran turbin, serta memberikan wawasan tentang optimasi desain dan operasional sistem. Selain pemodelan dengan regresi, kode juga mengimplementasikan algoritma PSO untuk mencari kombinasi optimal dari parameter turbin yang dapat meningkatkan efisiensi daya keluaran. Dengan PSO, kode mencoba menemukan konfigurasi terbaik yang memaksimalkan performa sistem berdasarkan pola pergerakan sekumpulan partikel dalam ruang solusi. Metode ini sangat berguna dalam konteks teknik dan keuangan, di mana optimasi berbasis heuristik dapat membantu meningkatkan hasil tanpa memerlukan perhitungan yang terlalu kompleks (Li dkk., 2020).

Error Daya untuk memastikan prediksi model (*daya output*) sedekat mungkin dengan 100 W menggunakan error kuadrat, yaitu $(\text{predicted_power} - 100)^2$. Penalti parameter dengan menambahkan penalti untuk setiap parameter yang melebihi nilai minimumnya. Karena batas minimum untuk kecepatan angin, ukuran blade, dan torsi masing-masing adalah 0.14-2 Nm, perbedaan relatif (dengan membagi *range parameters*) sehingga tiap penalti bernilai nol bila parameter berada pada nilai minimum.

$f(x) = \frac{(\bar{P}(x) - 100)^2}{\text{Target daya}} + \frac{\lambda \left[\frac{v-2}{12-2} + \frac{r-0.3}{0.65-0.3} + \frac{T-0.14}{2.0-0.14} \right]}{\text{Penalti: semakin kecil semakin baik}}$	(2)
--	-----

Dimana:

x : $[v, r, \omega, T]$

v : kecepatan angin (rentang 2 m/s hingga 12 m/s)

r : ukuran blade (rentang 0.3 hingga 0.65)

ω : kecepatan sudut (rentang 10 hingga 150)

T : torsi (rentang 0.14 hingga 2.0)

$\tilde{P}(x)$ adalah prediksi daya output dari model (di sini menggunakan model XGBoost yang telah dilatih)

λ adalah faktor bobot (misalnya 100) yang mengatur seberapa besar penalti parameter dibandingkan error daya.

Untuk evaluasi model dan optimasi parameter turbin angin menggunakan Particle Swarm Optimization (PSO). Pada bagian pertama, model yang sudah dilatih diuji menggunakan dataset uji ($X_{\text{test_scaled}}$) R-squared (R^2) sebagai indikator performa. Nilai R^2 yang tinggi menunjukkan bahwa variabel input memiliki pengaruh kuat terhadap daya keluaran. Bagian utama dari kode adalah fungsi objektif komposit untuk PSO, yang bertujuan untuk meminimalkan error dalam prediksi daya sambil mengontrol penggunaan sumber daya. Fungsi ini bekerja dengan standar input (x) yang mewakili kecepatan angin, ukuran blade, kecepatan sudut, dan torsi. Model ini kemudian melakukan prediksi daya turbin menggunakan $\text{model.predict}(X_{\text{candidate}})$, dan hasilnya dibandingkan dengan target 100 W menggunakan kuadrat selisih sebagai metrik kesalahan. Selain itu, kode menerapkan penalti untuk parameter tertentu, termasuk kecepatan angin, ukuran blade, dan torsi, dengan cara normalisasi agar nilai penalti tetap dalam skala yang sesuai. Model matematis, fungsi objektif komposit dapat diekspresikan sebagai berikut:

$\text{error_power} = (\text{predicted_power} - 100)^2$	(3)
---	-----

$\text{penalty} = \frac{x[:,0] - 2}{12 - 2} + \frac{x[:,1] - 0.3}{0.65 - 0.3} + \frac{x[:,3] - 0.14}{2 - 0.14}$	(4)
---	-----

$\text{fitness} = \text{error_power} + \lambda \text{penalty} * \text{penalty}$	(5)
--	-----

Nilai λ_{penalty} adalah 100 digunakan untuk memberikan bobot signifikan pada penalti parameter, sehingga model tidak hanya fokus pada minimisasi error daya tetapi juga pada efisiensi sumber daya. Jika nilai λ terlalu rendah, optimasi akan lebih banyak berfokus pada prediksi daya, yang bisa menyebabkan solusi dengan konsumsi energi yang tidak optimal. Sebaliknya, jika nilai λ terlalu tinggi, model akan lebih memprioritaskan penalti parameter, yang bisa mengorbankan akurasi daya prediksi. Pendekatan ini memungkinkan keseimbangan antara ketepatan daya dan penghematan sumber daya, sehingga model dapat mencari solusi optimal yang mempertimbangkan aspek ekonomi dan teknis secara bersamaan.

2.3 Metode GA

Dalam kode PSO, dataset disiapkan dan dibagi untuk membangun model prediksi daya turbin dengan XGBoost Regressor. Hal ini memungkinkan analisis hubungan antara parameter seperti kecepatan angin, ukuran blade, kecepatan sudut, dan torsi terhadap output daya yang dihasilkan. Model tersebut bertindak sebagai fungsi evaluasi yang digunakan dalam proses optimasi berikutnya. Namun, dalam kode kedua, optimasi dilakukan dengan prinsip GA, yaitu berbasis seleksi alam. GA bekerja dengan populasi individu yang merepresentasikan solusi potensial. Setiap generasi mengalami seleksi, crossover, dan mutasi untuk menghasilkan kombinasi parameter yang semakin optimal. Dalam konteks ini, GA digunakan untuk menemukan konfigurasi terbaik bagi turbin agar daya yang dihasilkan sesuai kebutuhan dengan desain turbin angin seminimal mungkin. Perbedaan utama antara PSO dan GA adalah mekanisme pencarian solusi. PSO berbasis pergerakan partikel dalam ruang solusi, sedangkan GA bekerja dengan evolusi populasi. GA biasanya lebih eksploratif dalam mencari solusi global, sementara PSO cenderung lebih cepat dalam konvergensi ke solusi terbaik. Dengan latar belakangmu dalam optimasi dan machine learning, mungkin menarik untuk membandingkan keduanya dalam meningkatkan performa turbin angin.

Pendekatan algoritma genetika dalam kode ini dapat diekspresikan sebagai berikut:

1. Inisialisasi Populasi: Setiap individu dalam populasi memiliki empat variabel utama yang dihasilkan secara acak dalam rentang batasnya;

$$\text{individual}=[v,r,\omega,\tau]$$

Dengan batas parameter: $2 \leq v \leq 12$, $0.3 \leq r \leq 0.65$, $10 \leq \omega \leq 150$, $0.4 \leq \tau \leq 2.0$

2. Fungsi Objektif (Fitness Function): tujuan optimasi adalah meminimalkan error daya output dan penalti terhadap parameter sesuai dengan persamaan 6-8.
3. Operator Evolusi: setiap generasi menggunakan operator crossover dan mutase.
4. Seleksi Generasi Berikutnya: proses seleksi dilakukan dengan turnamen seleksi, yaitu memilih individu terbaik dalam kelompok.

Fungsi fitness dalam metode optimasi GA digunakan untuk mengevaluasi kandidat solusi dengan dua tujuan utama: (1) meminimalkan error daya output agar mendekati 100 W, (2) mengurangi penggunaan sumber daya seperti kecepatan angin, ukuran blade, dan torsi. Kode GA mengambil daftar parameter individu ([kecepatan_angin, ukuran_blade, kecepatan_sudut, torsi]), lalu melakukan transformasi skala menggunakan `scaler.transform()`. Hasilnya digunakan oleh model XGBoost untuk memprediksi daya output turbin. Fungsi objektif (yang ingin diminimalkan) dihitung sebagai jumlah dari error daya output dan penalti parameter. Secara matematis, fungsi objektifnya dapat ditulis sebagai berikut:

$error_power = (predicted_power - 100)^2$	(6)
---	-----

$penalty = \frac{individual[0] - 2}{12 - 2} + \frac{individual[1] - 0.3}{0.65 - 0.3} + \frac{individual[2] - 0.14}{2 - 0.14}$	(7)
---	-----

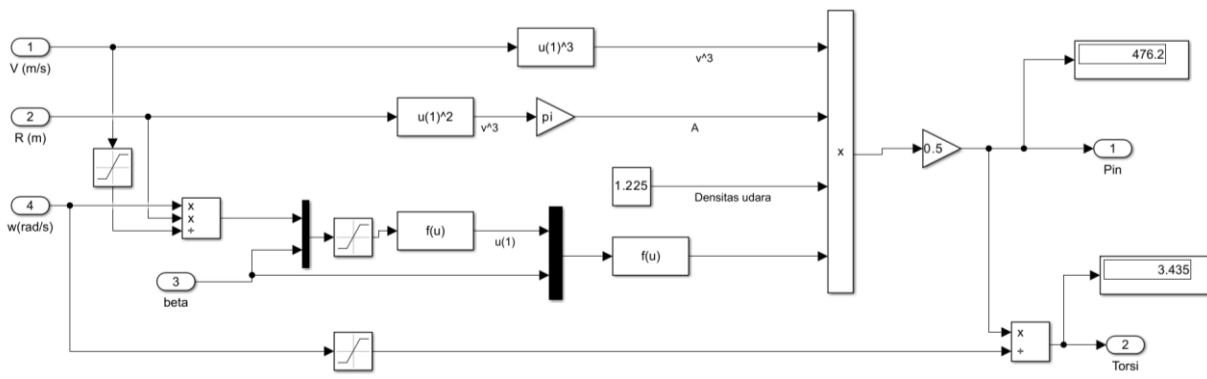
$fitness = error_power + 100 * penalty$	(8)
--	-----

Fungsi fitness mengembalikan nilai dalam bentuk tuple, yang akan digunakan oleh algoritma genetika untuk menentukan kelayakan suatu solusi. Dengan pendekatan seleksi alam, populasi individu akan mengalami seleksi, crossover, dan mutasi, menghasilkan solusi yang semakin optimal dalam setiap generasi.

Dalam konteks optimasi berbasis Genetic Algorithm (GA), fungsi fitness berperan sebagai mekanisme evaluasi yang menentukan kelayakan suatu individu dalam populasi. Tujuan utama dari algoritma ini adalah menemukan konfigurasi optimal dari parameter kecepatan angin, ukuran blade, kecepatan sudut, dan torsi yang dapat menghasilkan daya turbin mendekati target yang diinginkan, yaitu 100 W. Dengan pendekatan evolusi, setiap individu dalam populasi mengalami seleksi berdasarkan nilai fitness mereka, sehingga individu dengan nilai fitness terbaik memiliki peluang lebih besar untuk bertahan dan berkontribusi pada generasi berikutnya. Model ini meniru mekanisme seleksi alam, di mana solusi yang lebih adaptif terhadap tujuan optimasi akan berkembang seiring iterasi. Bentuk matematis dari fungsi objektif yang digunakan dalam algoritma genetika ini terdiri dari dua komponen utama: error daya output dan penalti parameter. Error daya dihitung sebagai selisih kuadrat antara daya yang diprediksi oleh model dan nilai targetnya (100 W), sedangkan penalti parameter berfungsi untuk menjaga agar nilai parameter tetap dalam rentang yang wajar. Strategi ini memungkinkan sistem untuk tidak hanya mencapai nilai daya yang optimal, tetapi juga mempertimbangkan efisiensi penggunaan sumber daya. Konsep penalti dalam optimasi sering digunakan untuk menghindari eksplorasi solusi yang tidak realistis dan memastikan keseimbangan antara kualitas solusi dan batasan teknisnya. Pendekatan berbasis GA menawarkan keunggulan dibandingkan metode optimasi konvensional, terutama dalam menangani masalah dengan ruang pencarian yang luas dan kompleksitas non-linear. Dibandingkan dengan metode seperti L-BFGS-B, yang lebih bergantung pada gradien untuk menemukan titik optimum, GA menggunakan mekanisme pencarian berbasis populasi yang lebih eksploratif.

3. Hasil dan Diskusi

Sebelum melakukan optimasi parameter mini pembangkit listrik tenaga angin (mini PLTB) terlebih dahulu menjalankan simulasi model di Matlab Simulink untuk memperoleh beberapa parameter optimasi seperti kecepatan angin, ukuran blade, kecepatan sudut, dan torsi.



Gambar 1. Model Mini PLTB di Matlab Simulink

Gambar 1 menunjukkan diagram blok dari sistem kontrol yang bertujuan untuk menghitung dan mengoptimalkan parameter turbin angin. Diagram ini menghubungkan berbagai variabel fisik, seperti kecepatan angin (V), radius blade (R), kecepatan sudut (ω), dan sudut beta (β), dengan model matematika yang digunakan untuk menentukan daya input (P_{in}) dan torsi (τ). Dengan menggunakan persamaan yang melibatkan luas penampang, densitas udara, serta faktor efisiensi, diagram ini memvisualisasikan bagaimana energi kinetik dari aliran udara dikonversi menjadi daya mekanik melalui interaksi parameter tersebut.

Secara matematis, daya yang dihasilkan dihitung berdasarkan persamaan fundamental dari konversi energi aerodinamika, di mana daya input (P_{in}) ditentukan melalui hubungan antara kecepatan angin dan luas penampang blade. Variabel $u(1)$ merupakan nilai kecepatan relatif yang melalui fungsi aerodinamika $f(u)$, yang kemudian digunakan untuk menghitung daya dengan faktor π , luas area (A), dan densitas udara ($\rho = 1.225 \text{ kg/m}^3$). Hasil akhir perhitungan ini dikalikan dengan faktor efisiensi untuk mendapatkan daya yang lebih realistis, dalam hal ini $P_{in} = 476.2 \text{ W}$. Selain itu, torsi (τ) yang dihasilkan dari rotasi blade juga dipengaruhi oleh interaksi antara kecepatan sudut dan geometri blade. Pendekatan diagram ini memungkinkan pemahaman sistematis terhadap bagaimana parameter desain dan kondisi operasi memengaruhi efisiensi konversi energi dalam turbin angin. Salah satu aspek penting dalam diagram ini adalah peran $f(u)$ sebagai fungsi non-linear yang menangkap efek kompleks dari sudut serang (β) terhadap performa aerodinamika blade. Interaksi ini sering dimodelkan dalam optimasi turbin untuk meningkatkan efisiensi daya dan mengurangi kerugian mekanis akibat torsi yang tidak optimal. Dengan menggunakan metode numerik, seperti Genetic Algorithm (GA) atau Particle Swarm Optimization (PSO), parameter-parameter ini dapat disesuaikan untuk mencapai konfigurasi paling efisien.

Tabel 1. Hasil simulasi model mini PLTB

Kecepatan Angin (m/s)	R (m)	ω (rad/s)(Kusnadi & Ismail, 2023; Slah dkk., 2016b)	Torsi (Nm)	P_m (watt)
2	0,606	8,69	0,0082	0,071
3		22,39	0,14	3,18
4		30,46	0,26	8
5		39,15	0,43	16,96
6		69,31	0,86	59,52
7		80,86	1,17	94,52
7,7		88,94	1,41	125,8
8		92,41	1,53	141,1
9		103,96	1,93	200,9
10		115,51	2,39	275,76
11		127,06	2,89	366,8
12		138,61	3,44	476,2

3.1. Metode PSO

Setelah mengidentifikasi parameter-parameter awal untuk konfigurasi mini Pembangkit Listrik Tenaga Bayu (PLTB) melalui simulasi menggunakan Matlab Simulink, sebagaimana tercantum pada Tabel 1, langkah selanjutnya adalah melakukan optimasi terhadap parameter-parameter tersebut. Proses optimasi ini krusial untuk mencapai kinerja optimal dengan target daya spesifik sebesar 100Watt, yang disesuaikan dengan kebutuhan penerangan pada perahu nelayan. Tujuan utama dari optimasi ini tidak hanya untuk memenuhi target daya, tetapi juga untuk meminimalisir dimensi fisik, khususnya jari-jari blade turbin. Dengan demikian, diharapkan desain mini PLTB akan menjadi lebih kompak dan ideal untuk integrasi pada struktur perahu nelayan yang seringkali memiliki keterbatasan ruang dan membutuhkan solusi yang efisien dalam hal ukuran dan berat. Untuk mencapai tujuan tersebut, penelitian ini mengadopsi PSO sebagai pendekatan utama dalam menemukan *set parameter* yang paling ideal. Pemilihan PSO didasari oleh kemampuannya yang terbukti dalam menangani masalah optimasi multivariabel yang kompleks, serta efisiensinya dalam mengeksplorasi ruang pencarian solusi yang luas. Algoritma PSO akan secara iteratif menyesuaikan parameter desain turbin—seperti jari-jari blade, kecepatan angin efektif, dan parameter aerodinamis lainnya—dengan tujuan meminimalkan fungsi biaya yang merepresentasikan deviasi dari target daya 100 Watt, sembari mempertimbangkan batasan dimensi dan efisiensi. Pendekatan ini memungkinkan penemuan konfigurasi optimal yang secara simultan memenuhi kebutuhan daya listrik untuk penerangan perahu nelayan dan persyaratan desain yang ringkas.

```
# --- Batasan Parameter ---
# x = [kecepatan_angin, ukuran_blade, kecepatan_sudut, torsi]
lb = [2, 0.3, 10, 0.14]
ub = [12, 0.65, 150, 2.0]
bounds = (np.array(lb), np.array(ub))

# --- Konfigurasi dan Inisialisasi PSO ---
options = {'c1': 2.0, 'c2': 2.0, 'w': 0.9}
optimizer = ps.single.GlobalBestPSO(n_particles=50, dimensions=4, options=options, bounds=bounds)

# Jalankan optimasi selama 100 iterasi
best_cost, best_pos = optimizer.optimize(composite_objective, iters=100)

# --- Evaluasi dan Tampilkan Hasil ---
optimal_input = best_pos
optimal_input_scaled = scaler.transform([optimal_input])
optimal_power = model.predict(optimal_input_scaled)[0]
```

Gambar 2. Batasan parameter di PSO

Gambar 2 menunjukkan implementasi PSO dalam kode Python, menggunakan *library* pyswarm, untuk mencari solusi optimal pada suatu masalah. Pada bagian awal, kode mendefinisikan batasan parameter untuk variabel optimasi *x* yang terdiri dari *kecepatan_angin*, *ukuran_blade*, *kecepatan_sudut*, dan *torsi*. Batasan bawah (*lb*) dan batasan atas (*ub*) untuk setiap variabel ditetapkan, membentuk rentang pencarian yang valid bagi algoritma PSO. Selanjutnya, bagian konfigurasi dan inisialisasi PSO menunjukkan bagaimana parameter algoritma diatur. Di sini, koefisien akselerasi *c1* dan *c2* keduanya ditetapkan pada 2.0, sementara faktor inersia *w* diatur ke 0.9. Sebuah objek optimizer *ps.single.GlobalBestPSO* diinisialisasi dengan 50 partikel (*n_particles*), 4 dimensi (sesuai dengan jumlah variabel), dan opsi serta batasan yang telah didefinisikan. Proses optimasi kemudian dijalankan selama 100 iterasi, di mana PSO akan berusaha meminimalkan (atau memaksimalkan, tergantung pada fungsi objektif *composite_objective*) nilai *best_cost* dan menemukan posisi terbaik *best_pos* yang sesuai. Bagian evaluasi dan tampilan hasil menunjukkan bagaimana solusi optimal yang ditemukan oleh PSO digunakan. *best_pos* yang merupakan input optimal kemudian diubah skalanya menggunakan *scaler.transform* sebelum dimasukkan ke dalam model (*model.predict*) untuk memprediksi *optimal_power*. Ini mengindikasikan bahwa masalah yang dioptimalkan kemungkinan berkaitan dengan estimasi daya atau kinerja berdasarkan input-input tersebut, dan PSO berhasil menemukan kombinasi input terbaik yang menghasilkan *optimal_power* yang diharapkan.

```
print("\nOptimal Input yang Ditemukan:")
print(f"Kecepatan Angin : {optimal_input[0]:.2f} m/s")
print(f"Ukuran Blade : {optimal_input[1]:.2f}")
print(f"Kecepatan Sudut : {optimal_input[2]:.2f}")
print(f"Torsi : {optimal_input[3]:.2f}")
print(f"\nPrediksi Daya Output: {optimal_power:.2f} W")
```

Optimal Input yang Ditemukan:
Kecepatan Angin : 2.06 m/s
Ukuran Blade : 0.45
Kecepatan Sudut : 12.63
Torsi : 0.50

Prediksi Daya Output: 102.55 W

Gambar 3. Output Optimasi PSO

Tampilan output dari Optimasi PSO menyajikan solusi optimal yang berhasil ditemukan untuk masalah desain mini turbin angin, yang secara spesifik ditujukan untuk kebutuhan penerangan perahu nelayan. Data yang disajikan—meliputi kecepatan angin, ukuran blade, kecepatan sudut, dan torsi yang merepresentasikan konfigurasi parameter yang paling efektif yang dicapai oleh swarm setelah 100 iterasi. Secara substansial, hasil optimasi ini menunjukkan bahwa pada kecepatan angin 2.06 m/s, dengan ukuran blade optimal 0.45 meter, kecepatan sudut putar turbin sebesar 12.63 rad/s, dan torsi sebesar 0.50 Nm, sistem mampu memprediksi daya output sebesar 102.55 Watt. Angka ini secara impresif melampaui target daya minimum 100 Watt yang telah ditetapkan, mengkonfirmasi keberhasilan PSO dalam mengidentifikasi titik optimum pada ruang pencarian multidimensional yang kompleks.

Konfigurasi parameter yang spesifik ini tidak hanya menunjukkan kemampuan PSO untuk secara akurat menargetkan nilai output yang diinginkan, tetapi juga memberikan wawasan krusial bagi implementasi praktis. Misalnya, ukuran blade 0.45 meter mengindikasikan desain yang relatif kompak, ideal untuk integrasi pada perahu nelayan yang memiliki keterbatasan ruang. Kecepatan angin 2.06 m/s merefleksikan efisiensi turbin pada kondisi angin yang relatif rendah, yang sering dijumpai di lingkungan pesisir atau lautan tenang. Kombinasi parameter ini menunjukkan bahwa PSO tidak hanya menemukan solusi yang memenuhi target daya, tetapi juga mengidentifikasi konfigurasi yang secara aerodinamis optimal dalam batasan desain yang praktis.

3.2. Metode GA

Mini PLTB yang optimal memerlukan penyesuaian parameter desain yang presisi untuk mencapai efisiensi maksimum dan memenuhi kebutuhan daya spesifik. Berdasarkan kode yang diberikan, terdapat empat atribut kunci yang dioptimalkan: kecepatan angin, ukuran blade, kecepatan sudut, dan torsi. Setiap atribut ini diregistrasikan dengan rentang nilai uniform yang telah ditentukan, mencerminkan batasan fisik dan operasional yang realistis untuk mini PLTB. Misalnya, kecepatan angin dibatasi antara 2 hingga 12 m/s, sementara ukuran blade berada dalam rentang 0.3 hingga 0.65 meter. Definisi rentang parameter ini menjadi fondasi bagi algoritma optimasi untuk mengeksplorasi ruang solusi secara sistematis dan efisien, memastikan bahwa hasil yang diperoleh tetap berada dalam domain yang valid dan layak secara teknis.

Metode optimasi yang diterapkan adalah Algoritma Genetika, sebagaimana ditunjukkan oleh penggunaan `deap.tools` dan konfigurasi operasi genetik seperti *crossover* (`cxBlend`) dan mutasi (`mutGaussian`). Populasi awal yang terdiri dari 50 individu ($n=50$) diinisialisasi, dan proses evolusi akan berjalan selama 100 generasi ($NGEN=100$). Pada setiap generasi, individu-individu dalam populasi akan menjalani proses seleksi, rekombinasi genetik melalui *crossover* dengan probabilitas 0.5 ($CXPB=0.5$), dan mutasi dengan probabilitas 0.2 ($MUTPB=0.2$). Operasi-operasi ini bertujuan untuk menghasilkan offspring baru yang memiliki karakteristik gabungan dari induknya, sekaligus memperkenalkan variasi genetik yang diperlukan untuk menghindari terjebak pada optimum lokal dan mengeksplorasi solusi-solusi baru yang lebih baik. Setelah offspring baru dihasilkan, batasan parameter yang telah didefinisikan sebelumnya (*lower_bounds* dan *upper_bounds*) diterapkan untuk memastikan setiap individu tetap valid. Selanjutnya, fitness dari setiap individu dievaluasi, yang mengukur seberapa baik solusi tersebut memenuhi fungsi objektif (misalnya, mencapai target daya 100 Watt). Berdasarkan nilai fitness ini, individu-individu terbaik dipilih (`selTournament`) untuk membentuk populasi generasi berikutnya, memastikan bahwa karakteristik yang mengarah pada solusi optimal secara progresif diperkuat dari generasi ke generasi. Proses iteratif ini berlanjut hingga jumlah generasi yang ditentukan tercapai, dengan harapan

individu terbaik (best_ind) yang ditemukan pada akhirnya merepresentasikan konfigurasi parameter optimal untuk mini PLTB yang efektif dan efisien.

```
# --- Setup Toolbox GA ---
toolbox = base.Toolbox()
# Mendaftarkan atribut dengan rentang masing-masing parameter
toolbox.register("attr_kecepatan_angin", np.random.uniform, 2, 12)           # kecepatan angin
toolbox.register("attr_ukuran_blade", np.random.uniform, 0.3, 0.65)         # ukuran blade
toolbox.register("attr_kecepatan_sudut", np.random.uniform, 10, 150)        # kecepatan sudut
toolbox.register("attr_torsi", np.random.uniform, 0.14, 2.0)                # torsi

toolbox.register("individual", tools.initCycle, creator.Individual,
                 (toolbox.attr_kecepatan_angin, toolbox.attr_ukuran_blade,
                  toolbox.attr_kecepatan_sudut, toolbox.attr_torsi), n=1)
toolbox.register("population", tools.initRepeat, list, toolbox.individual)

toolbox.register("mate", tools.cxBlend, alpha=0.5)
toolbox.register("mutate", tools.mutGaussian, mu=0, sigma=0.1, indpb=0.2)
toolbox.register("select", tools.selTournament, tournsize=3)
toolbox.register("evaluate", fitness)

# --- Batasan parameter ---
lower_bounds = [2, 0.3, 10, 0.14]
upper_bounds = [12, 0.65, 150, 2.0]

# --- Jalankan Genetic Algorithm ---
population = toolbox.population(n=50)
NGEN = 100 # Jumlah generasi
CXPB, MUTPB = 0.5, 0.2 # Probabilitas crossover dan mutasi

for gen in range(NGEN):
    # Hasilkan offspring melalui crossover dan mutasi
    offspring = algorithms.varAnd(population, toolbox, cxpb=CXPB, mutpb=MUTPB)

    # Perbaiki setiap individu agar tetap berada dalam batas yang sesuai
    for ind in offspring:
        check_bounds(ind, lower_bounds, upper_bounds)

    # Evaluasi offspring
    fits = list(map(toolbox.evaluate, offspring))
    for ind, fit in zip(offspring, fits):
        ind.fitness.values = fit

    # Seleksi populasi untuk generasi berikutnya
    population = toolbox.select(offspring, k=len(population))

    # (Optional) Cetak nilai rata-rata fitness tiap generasi
    mean_fit = np.mean([ind.fitness.values[0] for ind in population])
    print(f"Generasi {gen+1} - Rata-rata Fitness: {mean_fit:.2f}")

best_ind = tools.selBest(population, k=1)[0]
```

Gambar 4. Set up Optimasi GA

Pada sisi lain, Optimasi Partikel Swarm (PSO) menyajikan pendekatan komputasi yang terinspirasi dari perilaku sosial kawanan burung atau sekolah ikan, menawarkan metode yang komplementer untuk mencari solusi optimal. Berbeda dengan GA yang beroperasi pada populasi genetik, PSO memanfaatkan 'partikel-partikel' yang bergerak dalam ruang pencarian multidimensional, menyesuaikan lintasan mereka berdasarkan pengalaman terbaik yang pernah dicapai secara individu (pbest) dan pengalaman terbaik yang ditemukan oleh seluruh swarm (gbest). Kode menunjukkan inisialisasi 50 partikel (n_particles=50) dengan 4 dimensi, merepresentasikan parameter yang sama dengan GA (kecepatan angin, ukuran blade, kecepatan sudut, dan torsi). Konfigurasi parameter PSO, seperti koefisien akselerasi kognitif (c1=2.0), koefisien akselerasi sosial (c2=2.0), dan faktor inersia (w=0.9), sangat krusial dalam menyeimbangkan eksplorasi (pencarian di area baru) dan eksploitasi (penyempurnaan di area yang menjanjikan) dalam proses optimasi.

```
# --- Evaluasi Solusi Optimal ---
optimal_input = best_ind
optimal_input_scaled = scaler.transform([optimal_input])
optimal_power = model.predict(optimal_input_scaled)[0]

print("\nOptimal Input yang Ditemukan:")
print(f"Kecepatan Angin : {optimal_input[0]:.2f} m/s")
print(f"Ukuran Blade : {optimal_input[1]:.2f}")
print(f"Kecepatan Sudut : {optimal_input[2]:.2f}")
print(f"Torsi : {optimal_input[3]:.2f}")
print(f"\nPrediksi Daya Output: {optimal_power:.2f} W")

Optimal Input yang Ditemukan:
Kecepatan Angin : 2.00 m/s
Ukuran Blade : 0.45
Kecepatan Sudut : 32.87
Torsi : 0.14

Prediksi Daya Output: 102.55 W
```

Gambar 5. Output Optimasi GA

Gambar 5 secara spesifik menampilkan bagian Evaluasi Solusi Optimal dari sebuah kode Python dalam Optimasi GA. Di sini, `optimal_input` diinisialisasi dengan nilai `best_ind`, yang mengindikasikan bahwa input optimal ini berasal dari hasil akhir dari GA sebelumnya (karena `best_ind` sering merujuk pada individu terbaik yang ditemukan oleh GA). Kemudian, `optimal_input` ini diubah skalanya menggunakan `scaler.transform` untuk memastikan bahwa nilai-nilai tersebut sesuai dengan format yang diharapkan oleh model prediksi. Terakhir, `model.predict` digunakan untuk menghitung `optimal_power` berdasarkan input yang telah diubah skalanya. Hasil akhir menunjukkan bahwa kombinasi parameter optimal yang ditemukan adalah kecepatan angin 2.00 m/s, ukuran blade 0.45, kecepatan sudut 32.87, dan torsi 0.14. Dengan input ini, model memprediksi daya output sebesar 102.55 W, yang kemungkinan besar sesuai atau melampaui target daya yang diinginkan untuk aplikasi mini PLTB ini.

4. Kesimpulan

Penelitian ini secara komprehensif berhasil mengidentifikasi konfigurasi optimal mini turbin angin untuk aplikasi penerangan pada perahu nelayan, mengatasi tantangan efisiensi energi dan keterbatasan ruang melalui penerapan Algoritma Genetika (GA) dan Particle Swarm Optimization (PSO). Dengan memodelkan sistem dalam Matlab Simulink dan menetapkan fungsi objektif yang meminimalkan deviasi daya dari target 100 Watt sekaligus mengoptimalkan dimensi blade, studi ini secara sistematis mengeksplorasi ruang parameter yang kompleks. Temuan kunci dari kedua algoritma optimasi secara konsisten menunjukkan kemampuan untuk mencapai daya output yang melampaui target, dengan PSO menghasilkan 102.55 W pada kecepatan angin 2.06 m/s, ukuran blade 0.45 m, kecepatan sudut 12.63 rad/s, dan torsi 0.50 Nm. Sementara itu dengan menggunakan Optimasi GA menghasilkan daya sama dengan PSO dan ukuran blade yang sama juga sehingga dimensi yang cocok untuk mini PLTB bilah turbin adalah 0.45 m. Hasil ini memberikan solusi untuk peningkatan kinerja sistem dan operasional turbin angin skala kecil, serta berkontribusi pada kemajuan teknologi energi angin skala kecil yang berkelanjutan.

4. Ucapan Terima Kasih

Penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada Politeknik Perkapalan Negeri Surabaya (PPNS) atas dukungan fasilitas, sumber daya, dan kesempatan yang telah diberikan sehingga penelitian ini dapat terlaksana dengan baik. Apresiasi setinggi-tingginya juga kami sampaikan kepada Tim DIPA Internal PPNS yang telah memberikan kontribusi signifikan, baik berupa dukungan finansial maupun masukan konstruktif selama proses penelitian dan penulisan paper ini. Dukungan yang diberikan sangat membantu dalam kelancaran dan penyelesaian studi ini.

5. Daftar Pustaka

Battacharjee, P., & Bhattacharya, S. (2023). Optimizing the wind power generation cost in the Tirumala Region of India. *International Advanced Researches and Engineering Journal*, 7(1), 8–12. <https://doi.org/10.35860/iarej.1137173>

- Belgaid, Y., Helaimi, M., Taleb, R., & Benali Youcef, M. (2020). Optimal tuning of PI controller using genetic algorithm for wind turbine application. *Indonesian Journal of Electrical Engineering and Computer Science*, 18(1), 167. <https://doi.org/10.11591/ijeecs.v18.i1.pp167-178>
- BMKG. (2025). *Saran-Keselamatan-Berlayar*. BMKG. <https://maritim.bmkg.go.id/info/37/Saran-Keselamatan-Berlayar>
- IEA. (2025). *Wind*. International Energy Agency. <https://www.iea.org/energy-system/renewables/wind>
- Kusnadi, & Ismail. (2023). Eksperimental Turbin Zephyr dengan Pengaruh Sudu Rotor dan Sudu Statis. *Rotasi*, 25, 33–39. <https://ejournal.undip.ac.id/index.php/rotasi/article/download/58171/24528>
- Li, Y., Wei, K., Yang, W., & Wang, Q. (2020). Improving wind turbine blade based on multi-objective particle swarm optimization. *Renewable Energy*, 161, 525–542. <https://doi.org/10.1016/j.renene.2020.07.067>
- Perpustakaan Emil Salim. (2018). *Indonesia Darurat Energi*. BPPT. http://perpustakaan.menlhk.go.id/pustaka/home/index.php?page=detail_news&newsid=588
- Prasetyo, A. A., Rafinda, F. A., & Nugroho, H. (2022). Perbandingan Metode Optimasi Non-Linear Partical Swarm Optimization (PSO) Dengan Metode Interior Point Untuk Optimasi Daya Pada Turbin Angin Dengan Menentukan Nilai Optimum Pitch Angle. *KILAT*, 11(1), 103–110. <https://doi.org/10.33322/kilat.v11i1.1324>
- Prmod Jain. (2011). *Wind Energy Engineering* (1st ed.). McGraw-Hil.
- Slah, H., Mehdi, D., & Lassaad, S. (2016a). Advanced Control of a PMSG Wind Turbine. *International Journal of Modern Nonlinear Theory and Application*, 05(01), 1–10. <https://doi.org/10.4236/ijmnta.2016.51001>
- Slah, H., Mehdi, D., & Lassaad, S. (2016b). Advanced Control of a PMSG Wind Turbine. *International Journal of Modern Nonlinear Theory and Application*, 05(01), 1–10. <https://doi.org/10.4236/ijmnta.2016.51001>
- Zorić, J. (2023). Optimizing Wind Farm Layouts with Genetic Algorithms (Enhancing Efficiency in Wind Energy Planning and Utilization in Bosnia and Herzegovina). *Academic Journal of Research and Scientific Publishing*, 5(51), 51–73. <https://doi.org/10.52132/Ajrsp.en.2023.51.3>