

Penaksiran Parameter Distribusi Weibull Menggunakan Algoritma Genetika dan *Particle Swarm Optimization*

Sarah Ayatun Nufus¹, Sutarman²

^{1,2}Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sumatera Utara

Email: ¹sarahabdullah896@gmail.com

ABSTRAK

Penelitian ini bertujuan menaksir dan membandingkan hasil penaksiran parameter distribusi Weibull menggunakan Algoritma Genetika (AG) *Particle Swarm Optimization* (PSO). Distribusi Weibull yang digunakan adalah distribusi Weibull dua parameter dan distribusi Weibull tiga parameter. Pada algoritma AG terdapat operator evolusi seperti *Crossover* dan mutasi. Sedangkan pada algoritma PSO tidak. Operator evolusi tersebut yang membuat Ag dapat melakukan optimasi pada masalah kompleks dan ruang pencarian yang sangat luas. Evaluasi terhadap kedua metode dilakukan melalui pengamatan terhadap selisih nilai fitness yang dihasilkan. Berdasarkan simulasi data dari estimator yang diperoleh dengan menggunakan program R, diperoleh bahwa penaksiran parameter distribusi Weibull dua parameter dan distribusi Weibull tiga parameter menggunakan AG baik digunakan pada distribusi ini. Hal ini didukung dengan kecilnya selisih nilai fitness Algoritma AG yang diperoleh dibandingkan Algoritma PSO. *Sample space* juga mempengaruhi selisih fitness. Semakin besar *sample space* maka fitnessnya semakin bertambah besar.

Kata kunci: Penaksiran Parameter, Distribusi Weibull, Algoritma Genetika, *Particle Swarm Optimization*

ABSTRACT

This study aims to estimate and compare the results of the estimated parameters of the Weibull distribution using the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). The Weibull distribution used is a two-parameter Weibull distribution and a three-parameter Weibull distribution. In the GA Algorithm, there are evolution operators such as crossover and mutation. Meanwhile, in PSO algorithm doesn't. The evolution operators that can optimize complex problems and a very wide search space. Evaluation of the two method is carried out by observing the difference in the resulting fitness values. Based on the simulation data from the estimator obtained using the R program, it is found that the two-parameter Weibull distribution parameter and a three-parameter Weibull distribution using a GA algorithm are both used in this distribution. This is supported by the small difference in the fitness value of the GA algorithm obtained compared to the PSO algorithm. The sample space also affects the difference in fitness. The large the sample space, the greater the fitness.

Keywords: Parameter Estimation, Weibull Distribution, Genetic Algorithm, Particle Swarm Optimization

I. Pendahuluan

Penaksiran adalah bagian dari data yang berhubungan dengan perhitungan nilai parameter populasi (parameter) terhadap suatu karakter sampel [1]. Penaksiran parameter dibagi menjadi dua yaitu estimasi titik (*point estimation*) dan estimasi interval (*interval estimation*).

Salah satu analisis yang melibatkan penaksir parameter adalah analisis kelangsungan hidup (*survival*) merupakan suatu prosedur statistik yang digunakan untuk menganalisis data dimana variabel yang diteliti adalah waktu terjadinya peristiwa. Distribusi yang paling

sering digunakan dalam penanganan masalah kelangsungan hidup adalah distribusi Weibull. Distribusi ini memiliki parameter skala (*scale*) dan bentuk (*shape*) yang dapat menunjukkan ketepatan pemecahan pada sampel yang kecil. Untuk mendapatkan model survival Weibull, perlu dilakukan estimasi kedua parameter tersebut.

Distribusi Weibull dikenal sebagai distribusi yang fleksibel [2]. Salah satu fleksibilitasnya dapat dilihat dari perubahan distribusi ini menjadi distribusi lainnya seperti distribusi eksponensial. Distribusi ini bergantung pada perubahan nilai parameter skala dan bentuknya.

Kelebihan distribusi Weibull yaitu bentuk fungsional distribusi ini yang mudah sehingga tidak sukar diaplikasikan pada beberapa kejadian tertentu.

Salah satu algoritma metode optimasi paling populer adalah Algoritma Genetika (AG). AG merupakan suatu algoritma heuristik yang berbasis pada prinsip seleksi alam dan prinsip genetika. Algoritma genetika adalah salah satu metode optimasi yang memiliki 3 operator genetik utama yaitu seleksi, crossover dan mutasi, yang mana masing-masing operator tersebut memiliki perannya masing-masing guna dihasilkannya suatu solusi yang mendekati optimal [3].

AG dapat melakukan optimasi pada masalah yang kompleks. AG dapat digunakan dalam penyelesaian optimasi kombinatorial seperti *traveling salesman problem*, *crew scheduling* untuk penerbangan sampai masalah kontrol. AG merupakan algoritma penting dalam proses optimasi, yang dimana pada suatu algoritma dibuat dengan menerapkan proses evolusi dalam perkembangan pada makhluk hidup. Dalam AG proses pencarian solusi hanya didasarkan dari nilai fungsi tujuan, tidak ada penggunaan gradient ataupun kalkulus [4].

Selain AG, *Particle Swarm Optimization* (PSO) juga dapat digunakan dalam Algoritma optimasi. PSO adalah suatu teknik komputasi evolusioner, yang dimana populasi pada algoritma ini didasarkan pada pencarian algoritma tersebut dan dimulai dengan suatu populasi yang dibuat secara random yang disebut dengan istilah partikel. Tidak seperti metode komputasi evolusioner lainnya, setiap partikel dalam PSO juga terkait dengan *velocity*. Partikel bergerak di ruang penelusuran dengan kecepatan dinamis yang disesuaikan dengan perilaku historisnya. Oleh karena itu, setelah melalui proses penelusuran, partikel cenderung berpindah ke wilayah penelusuran yang lebih baik. PSO mirip dengan AG, yang dimulai dengan populasi acak dalam bentuk matriks. Namun kelemahan dari PSO adalah tidak adanya operator evolusi yaitu pindah silang dan mutasi seperti pada AG.

1.1 Distribusi Weibull

Distribusi Weibull merupakan distribusi yang sangat penting. Distribusi ini digunakan untuk menganalisis masalah kendala (Reliability) dan pemeliharaan (*maintainability*). Distribusi ini biasanya digunakan sebagai

pendekatan untuk mengkarakterisasi fungsi kerusakan karena perubahan nilai akan mengakibatkan distribusi ini memiliki sifat tertentu ataupun ekuivalen pada distribusi tertentu. Distribusi ini merupakan distribusi generik yang dapat diturunkan dari distribusi lain berdasarkan nilai dari bentuk parameternya [2]. Distribusi Weibull diperkenalkan pertama kali oleh Weibull pada tahun 1936 dengan 3 parameter, kemudian ada distribusi Weibull dengan 2 dan 1 parameter, di setiap distribusi [5]:

1. Distribusi Weibull Tiga Parameter

Distribusi weibull ini mempunyai 3 parameter dengan α adalah parameter lokasi, β adalah parameter skala dan λ parameter bentukannya, dengan fungsi densitasnya yaitu:

$$f(x|\alpha, \beta, \lambda) = \frac{\lambda}{\beta} \left(\frac{x-\alpha}{\beta}\right)^{\lambda-1} \exp\left\{-\left(\frac{x-\alpha}{\beta}\right)^\lambda\right\}; x \geq \alpha \quad (1)$$

2. Distribusi Weibull Dua Parameter

Berbeda dengan distribusi Weibull tiga parameter, pada distribusi Weibull dua parameter hanya memuat dua buah parameter yaitu β sebagai parameter skala dan λ adalah parameter bentukannya, dengan fungsi distribusi kumulatifnya adalah:

$$F(x|\beta, \lambda) = 1 - \exp\left[-\left(\frac{x}{\beta}\right)^\lambda\right] \quad (2)$$

Ekspektasi dan variansinya adalah:

$$\mu = \lambda \Gamma\left(1 + \frac{1}{\beta}\right) \quad (3)$$

$$\sigma^2 = \lambda^2 \left\{ \Gamma\left(1 + \frac{1}{\beta}\right) - \left[\Gamma\left(1 + \frac{1}{\beta}\right)\right]^2 \right\} \quad (4)$$

3. Distribusi Weibull Satu Parameter

Pada distribusi ini hanya ada satu parameter yaitu α sebagai parameter lokasi, β parameter skala dan λ adalah parameter bentukannya, dengan fungsi densitasnya yaitu:

$$f(x|0, 1, \lambda) = (\lambda)(x)^{\lambda-1} \exp\{-(x)\}^\lambda \quad (5)$$

$$f(x|0, \beta, 1) = \left(\frac{1}{\beta}\right) \exp\left\{-\left(\frac{x}{\beta}\right)\right\} \quad (6)$$

$$f(x|\alpha, 1, 1) = \exp\{-(x - \alpha)\} \quad (7)$$

1.2 Fungsi Likelihood pada Distribusi Weibull

Misalkan terdapat hasil pengamatan terhadap suatu variabel X sebanyak n . Dengan parameter dari fungsi likelihood dalam distribusi Weibullnya yaitu α , β dan λ . Dalam menentukan fungsi log-likelihood diawali dengan menentukan fungsi Likelihood dan fungsi densitas dari distribusi weibull,

$$L(\beta, \lambda) = \prod_{i=1}^n f(X_i | \beta, \lambda) \quad (8)$$

Fungsi Likelihood diperoleh dari fungsi kepekatan peluang distribusi Weibull dapat dituliskan dengan:

$$L(\beta, \lambda) = \prod_{i=1}^n \frac{\lambda}{\beta} \left(\frac{x_i}{\beta}\right)^{\lambda-1} \exp\left\{-\left(\frac{x_i}{\beta}\right)^\lambda\right\}$$

$$L(\beta, \lambda) = \left(\prod_{i=1}^n \frac{\lambda}{\beta^\lambda} x_i^{\lambda-1}\right) \exp\left(-\left(\frac{1}{\beta^\lambda}\right) \sum_{i=1}^n x_i^\lambda\right) \quad (9)$$

Berdasarkan persamaan (9) tersebut maka diperoleh fungsi logaritma natural :

$$\ln L(\beta, \lambda) = \ln\left\{\left(\prod_{i=1}^n \frac{\lambda}{\beta^\lambda} x_i^{\lambda-1}\right) \exp\left(-\left(\frac{1}{\beta^\lambda}\right) \sum_{i=1}^n x_i^\lambda\right)\right\}$$

$$\ln L(\beta, \lambda) = n \ln \lambda - n \lambda \ln \beta + (\lambda - 1) \sum_{i=1}^n \ln x_i - \frac{1}{\beta^\lambda} \sum_{i=1}^n x_i^\lambda \quad (10)$$

Persamaan (10) merupakan fungsi log likelihood dari distribusi Weibull dua parameter skala bentuk.

Untuk fungsi log-likelihood tiga parameter diawali dengan menentukan fungsi Likelihood dan fungsi densitas dari distribusi Weibull. Fungsi kepadatan peluang dari distribusi Weibull tiga parameter pada persamaan (1) dapat dibentuk fungsi Likelihood sebagai berikut:

$$L(\alpha, \beta, \lambda) = \prod_{i=1}^n \frac{\lambda}{\beta} \left(\frac{x - \alpha}{\beta}\right)^{\lambda-1} \exp\left\{-\left(\frac{x - \alpha}{\beta}\right)^\lambda\right\}$$

$$L(\alpha, \beta, \lambda) = \prod_{i=1}^n \frac{\lambda}{\beta^\lambda} (x_i - \alpha)^{\lambda-1} \exp\left(-\left(\frac{1}{\beta^\lambda}\right) \sum_{i=1}^n (x_i - \alpha)^\lambda\right) \quad (11)$$

ngsi logaritma natural, sebagai berikut:

$$\ln L(\alpha, \beta, \lambda) = \ln\left\{\prod_{i=1}^n \frac{\lambda}{\beta^\lambda} (x_i - \alpha)^{\lambda-1} \exp\left(-\left(\frac{1}{\beta^\lambda}\right) \sum_{i=1}^n (x_i - \alpha)^\lambda\right)\right\}$$

$$\ln L(\alpha, \beta, \lambda) = n \ln \lambda - n \lambda \ln \beta + (\lambda - 1) \sum_{i=1}^n \ln(x_i - \alpha) - \frac{1}{\beta^\lambda} \sum_{i=1}^n (x_i - \alpha)^\lambda \quad (12)$$

Persamaan (12) merupakan fungsi log likelihood dari distribusi Weibull tiga parameter.

1.3 Sifat-Sifat Penduga Parameter

Dalam statistika, penaksiran adalah cara untuk mencari nilai taksiran suatu populasi dengan menggunakan nilai lokasi sampel. Nilai-nilai populasi sering disebut sebagai parameter populasi. Estimasi adalah proses yang menggunakan model statistik atau untuk memperkirakan korelasi relatif dari populasi yang tidak diketahui. Dalam hal ini, variabel acak diambil dari populasi yang dituju, sehingga ukuran populasi dapat diperkirakan [1]. Sifat-sifat penaksir adalah sebagai berikut[2]:

1. Tak Bias

$$E(\hat{\theta}) = \theta \quad (13)$$

2. Efisien

Suatu penaksiran disebut efisien untuk parameter θ apabila penduganya $\hat{\theta}$ mempunyai nilai varian yang kecil. Apabila penduganya lebih dari satu, penaksiran dikatakan efisien jika penaksirnya memiliki varian yang kecil.

3. Konsisten

Penaksir parameter $\hat{\theta}$ dikatakan konsisten, jika penaksir mendekati nilai parameter yang sebenarnya terlepas dari ukuran sampel yang semakin besar. Suatu statistik $\hat{\theta}$ disebut penaksir yang konsisten dari parameter θ jika dan hanya jika $\hat{\theta}$ konvergen dalam probabilitas ke parameter θ atau dapat dituliskan sebagai berikut:

$$\lim_{p \rightarrow 0} P(|\hat{\theta} - \theta| > p) = 0 \quad (14)$$

Dimana p merupakan probabilitas.

1.4 Algoritma Genetika

Algoritma Genetika (AG) merupakan algoritma pencarian heuristik berdasarkan seleksi alam dan seleksi evolusioner. Teori seleksi alam dan evolusi pertama kali diperkenalkan oleh Charles Darwin. Algoritma ini didasarkan pada konsep evaluasi biologis dan dapat memberikan solusi alternatif untuk masalah yang akan dipecahkan. Algoritma genetika menggunakan metode seleksi, crossover, dan mutasi untuk memberikan solusi pemecahan masalah yang terbaik. Dengan terus-menerus mengulangi pencarian keturunan, maka dapat mencapai solusi yang paling diinginkan[7]. Algoritma Genetika standar adalah sebagai berikut:

1. Menentukan populasi awal
2. Set iterasi $t=1$
3. Pilih individu yang terbaik untuk disalin sejumlah tertentu untuk menggantikan individu lain
4. Lakukan seleksi
5. Lakukan *Crossover* untuk induk yang terpilih.
6. Menentukan beberapa individu dalam populasi untuk proses mutasi.
7. Jika belum mencapai konvergensi set iterasi $t=t+1$.
8. Kembali ke langkah 2.

Operator dalam Algoritma Genetika adalah sebagai berikut:

1. Kromosom

Dalam AG, kromosom adalah bagian penting dari algoritma. Suatu kromosom atau individu mewakili satu vektor keputusan. Terkadang vektor keputusan dapat digunakan dalam implementasi AG atau dapat dikodekan. Pengkodean dilakukan untuk merepresentasikan solusi menggunakan bilangan biner. Itu tergantung pada optimasi yang ada. Setiap anggota kromosom terdiri dari gen, di mana setiap gen mewakili elemen vektor keputusan. Dengan peningkatan populasi seperti itu, akan ada banyak solusi.

2. Fungsi Fitness

Fungsi kesesuaian berfungsi untuk mengukur tingkat kualitas atau kesesuaian suatu solusi terhadap solusi yang dicari. Fungsi fitness dapat berhubungan langsung dengan fungsi tujuan atau dapat berupa perubahan kecil pada fungsi tujuan. Himpunan solusi yang

dihasilkan dalam populasi dievaluasi terhadap nilai fitness. Pada proses minimasi nilai tujuan (x) digunakan fitness $\frac{1}{f(x)}$ dengan nilai terkecil untuk memperoleh nilai optimal. Sebaliknya, proses maksimasi digunakan nilai tujuan (x) dengan nilai terbesar untuk memperoleh nilai optimal. Untuk mengetahui fitness terbaik akan digunakan selisih fitnessnya dengan persamaan sebagai berikut:

$$Fitness = \min \left\{ \ln \left\{ \prod_{i=1}^n f(X_i | \theta) \right\} - \ln \left\{ \prod_{i=1}^n f(X_i | \hat{\theta}) \right\} \right\} \quad (15)$$

3. Seleksi

Metode seleksi roda roulette yaitu metode yang paling sederhana atau sering juga disebut dengan sampling substitusi stokastik. Dengan demikian, orang-orang ditampilkan pada segmen garis secara berurutan, sehingga setiap segmen berukuran sama dengan bilah kebugaran. Nomor acak dihasilkan dan orang-orang yang bagiannya berada di wilayah nomor acak akan dipilih. Proses ini diulang sampai jumlah individu yang diinginkan tercapai. Adapun Peluang dari setiap Kromosom adalah sebagai berikut:

$$P(\text{Cromosom}) = \frac{\text{Selisih Nilai Fitness}}{\text{Total Selisih Fitness}} \quad (16)$$

Selanjutnya untuk nilai peluang kumulatifnya adalah sebagai berikut:

$$P_{cum}(\text{Cromosom ke } n) = \sum_{i=1}^n P_i(\text{Cromosom}) \quad (17)$$

Kemudian akan dibangkitkan nilai acak antara 0 sampai 1 sebanyak jumlah kromosom. Nilai random pertama menunjukkan nilai dan posisi kromosom pertama, dan seterusnya

4. *Crossover* atau Pindah Silang

Salah satu komponen terpenting dari AG adalah crossover. Kromosom yang mengarah ke solusi baru dapat diperoleh dengan menyilangkan dua kromosom. Pada tahap ini akan dilakukan *crossover* pada kromosom baru yang diperoleh tadi. Hal yang pertama yang dilakukan adalah menentukan *crossover rate* misalnya 10%, 25% dan lainnya. Setelah itu dibangkitkan nilai acak antara 0 sampai 1 sebanyak jumlah kromosom. Jika nilai acak pada kromosom ke $i < crossover\ rate$ maka kromosom tersebut akan disilangkan. Misalkan terdapat kromosom 1, kromosom 2 dan kromosom 3 yang nilai acaknya lebih kecil dari

crossover rate. Maka dilakukan *crossover* sebagai berikut:

Kromosom 1baru = Kromosom 1 × Kromosom 2
 Kromosom 2baru = Kromosom 2 × Kromosom 3

Untuk jumlah gen yang *crossover* akan dibangkitkan angka acak dari 1 sampai panjang gen dikurangi satu.

5. Melakukan Mutasi

Operasi mutasi dilakukan pada kromosom dengan tujuan untuk mendapatkan kromosom baru sebagai kandidat solusi untuk generasi mendatang dengan morfologi fisik yang lebih baik, yang pada akhirnya mengarah pada solusi optimal yang diinginkan. Tahap pertama adalah menghitung jumlah gen. Karena parameter (gen) distribusi Weibull adalah 3 maka diperoleh persamaan sebagai berikut:

$$\text{Jumlah seluruh gen} = \text{jumlah gen kromosom} * \text{Populasi}$$

$$\text{Jumlah selirih gen} = 3 * \text{Populasi} \quad (18)$$

Setelah ditentukan itu akan ditentukan jumlah gen yang dimutasi. Kemudian ditentukan nilai *mutation rate*-nya. Setelah *mutation rate* ditentukan maka ibangkitkan nilai random antara 1 sampai jumlah seluruh gen. maka diperoleh persamaan mencari jumlah gen yang akan dimutasi sebagai berikut:

$$\text{Jumlah gen mtasi} = \text{mutation rate} * \text{Jumlah seluruh gen} \quad (19)$$

Untuk menentukan gen yang dimutasi akan dibangkitkan bilangan random antara 1 sampai jumlah gen sebanyak jumlah gen mutasi. Untuk nilai gen yang diganti diperoleh dari pembangkitan nilai acak antara suatu interval yang diyakini terdapat nilai parameter didalamnya pada tahap menentukan kromosom.

1.5 Particle Swarm Optimization

Particle Swarm Optimization (PSO) adalah salah satu teknik komputasi *evolusioner* yang mana populasinya didasarkan pada algoritma pencarian dan dimulai dengan populasi acak yang disebut partikel [8]. Tidak seperti metode komputasi *evolusioner* lainnya, setiap partikel PSO juga dikaitkan dengan kecepatan. Partikel bergerak di ruang pencarian dengan *velocity* dinamis yang diatur sesuai dengan perilaku

historisnya. Oleh karena itu, partikel cenderung berpindah ke area penelusuran yang lebih sesuai setelah melalui proses pencarian. PSO mirip dengan algoritma genetika yang dimulai dengan populasi acak dalam bentuk matriks. Namun, PSO tidak memiliki operator *evolusioner* seperti persilangan dan mutasi pada algoritma genetika. Baris-baris matriks disebut partikel sedangkan pada algoritma genetika disebut kromosom yang terdiri dari nilai-nilai variabel. Setiap partikel bergerak cepat dari posisi semula ke posisi yang lebih baik [9].

Studi lebih lanjut mengenai optimasi berbasis *particle swarm* di tahun 1999 menambahkan inerti (ω) dalam persamaan pembaharuan *velocity*. Persamaan ini kemudian menjadi algoritma optimasi berbasis *particle swarm* yang paling umum digunakan [10]:

$$v_i(t) = v_i(t - 1) + c_1 r_1 (x_{pi} - x_i) + c_2 r_2 (x_{si} - x_i) \quad (20)$$

$$x_i(t) = x_i(t - 1) + v_i(t) = v_i \quad (21)$$

Dimana:

- i = Indeks Partikel
- v_i = *Velocity* Partikel ke- i
- x_i = Posisi Partikel ke- i
- x_{pi} = Posisi terbaik dari semua partikel (gbest)
- x_{si} = Posisi terbaik dari partikel ke- i (pbest)
- $c_{1,2}$ = *Learning rate*
- $r_{1,2}$ = Bilangan acak U [0,1]

Adapun cara-cara melakukan penaksiran parameter distribusi Weibull tiga parameter menggunakan algoritma PSO adalah sebagai berikut:

1. Menentukan Partikel-Partikel

Dalam PSO, partikel merupakan taksiran awal dari parameter. Distribusi Weibull tiga parameter merupakan elemen pada partikel yaitu α , β dan λ . Nilai *velocity* pada iterasi pertama digunakan nilai 0. Nilai elemen dari partikel adalah bilangan acak antara suatu interval yang diyakini terdapat nilai parameter didalamnya.

2. Menentukan Posisi dan *Velocity*

Posisi awal partikel dapat ditentukan dengan melihat interval batas yang sudah ditentukan sebelumnya. Adapun persamaan menentukan posisi awalnya yaitu:

$$\text{Posisi} = (\text{B. atas} - \text{B. bawah}) + \text{angka random} * \text{B. bawah} \quad (22)$$

II. Hasil dan Pembahasan

1.1 Penaksiran Parameter Distribusi Weibull Dua Parameter Pada Data Kecepatan Angin Menggunakan Algoritma Genetika

Pada bagian ini akan diselidiki satu contoh menaksir parameter pada data Tabel 4.1. Diasumsikan bahwa data pada tabel 4.1 berdistribusi Weibull. Terdapat 66 data yang akan Data Kecepatan Angin Terbesar Per Bulan di Wilayah Nusa Tenggara Barat Tahun 2009-2014. Adapun data Data Kecepatan Angin Terbesar Per Bulan di Wilayah Nusa Tenggara Barat Tahun 2009- 2014 adalah sebagai berikut:

Tabel 1. Data Kecepatan Angin Terbesar Per Bulan di Wilayah Nusa Tenggara Barat Tahun 2009-2014

Bulan	Kecepatan Angin Knot					
	2009	2010	2011	2012	2013	2014
Januari	13	17	30	26	24	18
Februari	20	14	20	18	45	38
Maret	15	12	13	29	20	16
April	13	13	14	22	13	11
Mei	13	16	24	18	14	14
Juni	9	40	12	20	9	14
Juli	15	16	24	18	16	
Agustus	10	13	17	20	14	
September	13	12	18	18	14	
Oktober	15	13	16	17	13	
November	15	16	18	15	22	
Desember	18	14	15	15	18	

Selanjutnya dilakukan penaksiran parameter dalam distribusi Weibull menggunakan algoritma genetika sebagai berikut:

1. Menentukan Kromosom

Pada tahap ini digunakan populasi yang berjumlah 6 Kromosom. Nilai gen dari kromosom adalah bilangan acak antara 0 sampai 20. Adapun bentuk kromosomnya adalah sebagai berikut:

Tabel 2. Kromosom awal

Kromosom	β	λ
Kromosom 1	2,6782	18,0799
Kromosom 2	19,3950	0,8235
Kromosom 3	18,3234	18,6158
Kromosom 4	17,7541	16,6879
Kromosom 5	3,6793	4,8525
Kromosom 6	1,2910	0,8421

2. Menentukan Nilai Fitness

Fungsi fitness digunakan untuk mengukur tingkat kesesuaian dengan solusi yang dicari. Adapun nilai fitness dari tiap kromosom adalah sebagai berikut:

Tabel 3. Nilai Fitness Kromosom Data Kecepatan Angin pada Iterasi 1

Kromosom	β	λ	Fitness	Selisih Fitness
Kromosom 1	2,6782	18,0799	2,6004	14,8995
Kromosom 2	19,3950	0,8235	21,5380	4,0380
Kromosom 3	18,3234	18,6158	17,8051	0,3051
Kromosom 4	17,7541	16,6879	17,1998	0,3001
Kromosom 5	3,6793	4,8525	3,3723	14,1276
Kromosom 6	1,2910	0,8421	1,4128	16,0871

3. Melakukan Seleksi

Pada proses ini dipilih kromosom yang digunakan sebagai *parent*. Dalam penelitian ini digunakan seleksi roda roulette. Adapun Peluang dari setiap Kromosom adalah sebagai berikut:

Tabel 4. Peluang Kumulatif Kromosom Data Kecepatan Angin Iterasi 1

Kromosom	β	λ	Selisih Fitness	P	P_{cum}
Kromosom 1	2,6782	18,0799	14,8995	0,2994	0,2994
Kromosom 2	19,3950	0,8235	4,0380	0,0811	0,3805
Kromosom 3	18,3234	18,6158	0,3051	0,0061	0,3867
Kromosom 4	17,7541	16,6879	0,3001	0,0060	0,3927
Kromosom 5	3,6793	4,8525	14,1276	0,2839	0,6766
Kromosom 6	1,2910	0,8421	16,0871	0,3233	1

Selanjutnya dibangkitkan nilai random antara 0 sampai 1 sebanyak 6. Nilai tersebut digunakan untuk mengetahui kromosom baru.

4. Melakukan Crossover

Pada tahap ini akan dilakukan *crossover* pada kromosom baru yang diperoleh tadi. Hal yang pertama yang dilakukan adalah menentukan *crossover rate*. Pada penelitian ini digunakan *crossover rate* 25% . Setelah itu

dibangkitkan nilai acak antara 0 sampai 1 sebanyak jumlah kromosom. Jika nilai acak pada kromosom ke $i < crossover\ rate$ maka kromosom tersebut akan disilangkan.

5. Mutasi

Pada tahap ini dihitung jumlah seluruh gen dengan Persamaan (4.3). Diperoleh jumlah seluruh gen nya adalah 12. Selanjutnya ditentukan jumlah gen yang dimutasi dengan Persamaan (4.4). *mutation rate* yang digunakan adalah 0.1. maka diperoleh hasil 1 gen yang akan dimutasi. Posisi gen yang dimutasi dengan membangkitkan bilangan acak antara 1 sampai 12. Bilangan acak yang diperoleh adalah 6 dan nilai gennya antara 0 sanpai 12. Sehingga populasi barunya yaitu:

Tabel 5. Kromosom Baru Iterasi Baru

Kromosom	β	λ	Selisih Fitness
Kromosom 1 Baru	3,6793	4,8525	14,1276
Kromosom 2 Baru	2,6782	18,0799	14,8995
Kromosom 3 Baru	17,7541	3,9720	1,4140
Kromosom 4 Baru	18,3234	18,6158	0,3051
Kromosom 5 Baru	19,3950	0,8235	4,0380
Kromosom 6 Baru	1,2910	0,8421	16,0871

Setelah Proses mutasi selesai maka diperoleh estimasi parameter terbaik dengan selisih fitness terkecil pada iterasi pertama dengan $\beta=18,3234$ dan $\lambda =18,6158$. Karena iterasi belum terpenuhi maka iterasi akan dilanjutkan ke iterasi 2.

Selanjutnya perhitungan iterasi selanjutnya digunakan *software R* untuk melakukan simulasi dengan jumlah iterasi 100, diperoleh hasil estimatornya yaitu $\hat{\beta} = 18,8534$ dan $\hat{\lambda} = 6,0577$ serta selisih nilai fitnesnya adalah 0,00004.

1.2 Penaksiran Parameter Distribusi Weibull Dua Parameter Pada Data Kecepatan Angin Menggunakan Particle Swarm Optimization

Pada bagian ini dilakukan penaksiran parameter dalam distribusi Weibull dengan menggunakan Algoritma PSO. Selanjutnya

penaksiran parameternya adalah sebagai berikut:

1. Menentukan Partikel dan *Velocity*

Pada tahap ini digunakan populasi yang berjumlah 6 partikel. Nilai elemen dari partikel adalah bilangan acak antara 0 sampai 20. Untuk nilai *velocity* awalnya adalah 0. Adapun bentuk partikelnya adalah sebagai berikut:

Tabel 6. Partikel Awal

Partikel	β	λ
Partikel 1	2,6782	18,0799
Partikel 2	19,3950	0,8235
Partikel 3	18,3234	18,6158
Partikel 4	17,7541	16,6879
Partikel 5	3,6793	4,8525
Partikel 6	1,2910	0,8421

2. Menentukan Nilai Fitness

Fungsi fitness digunakan untuk mengukur grade atau mencocokkan solusi dengan solusi yang diinginkan. Nilai fitness dari setiap kromosom adalah sebagai berikut:

Tabel 7. Nilai Fitness Partikel Data Kecepatan Angin Iterasi 1

Partikel	β	λ	Fitness	Selisih Fitness
Partikel 1	2,6782	18,0799	2,6004	14,8995
Partikel 2	19,3950	0,8235	21,5380	4,0380
Partikel 3	18,3234	18,6158	17,8051	0,3051
Partikel 4	17,7541	16,6879	17,1998	0,3001
Partikel 5	3,6793	4,8525	3,3723	14,1276
Partikel 6	1,2910	0,8421	1,4128	16,0871

3. Menentukan Pbest dan Gbest

Pada langkah 2 dapat dilihat bahwa Gbest terbaik berada pada partikel 4 dengan elemen 17,7541 dan 16,6879. Dikarenakan tahap ini merupakan iterasi 1 maka setiap partikel adalah Pbest.

4. Memperbarui posisi partikel *velocity*

Pada tahap ini akan dilakukan perhitungan posisi partikel dan *velocity* yang baru. Karena PSO yang digunakan adalah PSO dengan inertia weight maka untuk menghitung *velocity* baru akan digunakan Persamaan (20) dengan

membangkitkan nilai *learning rate* secara acak. Untuk perhitungan nilai posisi partikel yang baru menggunakan Persamaan (21). Dengan *inersia weight* 0.9 dan *learning rate* 0.2. Adapun Posisi dan *Velocity* partikel baru yaitu:

Posisi dan *Velocity* partikel 1

$$v_1(2) = w \cdot v_1(1) + c_1 r_1 (Gbest - x_1) + c_2 r_2 (Pbest - x_1)$$

$$v_1(2) = 0,9(0; 0) + 0,2(0,9461)(17,7541; 16,6879 - 2,6782; 18,0799) + 0,2(0,667467)(2,6782; 18,0799 - 2,6782; 18,0799)v_2(1) = 2,8529; -0,2634$$

$$x_i(t) = x_i(t - 1) + v_i(t)$$

$$x_1(1) = 2,6782; 18,0799 + (2,8529; -0,2634)$$

$$x_1(1) = 5,5311; 17,8164$$

Untuk partikel selanjutnya posisi dan *velocity* barunya yaitu:

Tabel 8. Nilai Fitnes Partikel Baru Data Kecepatan Angin Iterasi 1

Partikel	posisi 1	posisi 2	velocity 1	velocity 2	Selisi h Fitnes
Partikel 1	5,5311	17,8164	2,8529	-0,2634	12,1316
Partikel 2	19,3013	1,7285	-0,0936	0,9050	0,2971
Partikel 3	18,2523	18,375	-0,0710	-0,2407	0,2299
Partikel 4	17,7541	16,6879	0	0	0,3001
Partikel 5	4,5769	5,6072	0,8976	0,7547	13,2699
Partikel 6	3,3158	2,7910	2,0248	1,9489	14,5477

Setelah Proses pembaruan posisi dan *velocity* selesai maka diperoleh estimasi parameter terbaik dengan selisih fitness terkecil pada iterasi pertama dengan $\beta=18,2523$ dan $\lambda=18,375$. Karena iterasi belum terpenuhi maka iterasi akan dilanjutkan ke iterasi 2.

Selanjutnya dilakukan dengan cara yang sama untuk iterasi selanjutnya sampai dengan jumlah iterasi yang ditentukan. Untuk mempermudah penelitian ini, digunakan *software R* untuk melakukan simulasi. Setelah dilakukan penaksiran parameter dengan jumlah iterasi 100, diperoleh hasil penaksirannya yaitu

$\beta = 18,4188$ dan $\lambda = 9,6506$ serta selisih nilai fitnessnya adalah 0,004.

Setelah itu dilakukan penaksiran menggunakan metode Algoritma Genetika dan Algoritma PSO. Fitness yang digunakan adalah selisih mean data dengan mean estimator dengan persamaan (2.11). Perbandingan penaksiran parameter distribusi Weibull dengan Algoritma Genetika dan algoritma PSO adalah sebagai berikut:

Tabel 9. Perbandingan Penaksiran Parameter Distribusi Weibull Dengan Algoritma Genetika dan Algoritma PSO Pada Data Kecepatan Angin Terbesar

Algoritma Genetika			Algoritma PSO		
β	λ	Selisi h Fitnes	β	λ	Selisi h Fitnes
18,8534	6,05779	0,00004	18,4188	9,6506	0,004

1.3 Perbandingan AG dan PSO Pada Distribusi Weibull Menggunakan Data Bangkitan

Selanjutnya dilakukan simulasi dengan total sampel 200, 500, dan 1000. Hasil simulasi dapat dilihat pada lampiran 8. Perbandingan penaksiran parameter distribusi Weibull dengan AG dan PSO adalah sebagai berikut:

Tabel 10 Perbandingan Penaksiran Parameter Distribusi Weibull Dengan Algoritma Genetika dan PSO

N	Parameter			Algoritma Genetika				Algoritma PSO			
	α	β	λ	$\hat{\alpha}$	$\hat{\beta}$	$\hat{\lambda}$	Selisih Fitness	$\hat{\alpha}$	$\hat{\beta}$	$\hat{\lambda}$	Selisih Fitness
100	-	1	1	-	1,0	1,328	0,0063	-	1,197	1,416	1,5764
200	3	4	5	2,6106	4,0914	4,964	0,0000	2,5201	4,4130	4,8591	0,19469
500	3	4	5	2,5860	4,3749	4,5115	0,0000	2,5987	4,4757	4,6776	3,9754
1000	3	4	5	2,5333	4,4344	4,5676	2,2431	2,5000	4,4999	4,7156	155,868

Dari tabel 10 dapat dilihat nilai selisih fitness dari kedua metode menunjukkan nilai fitness yang cenderung bertambah besar dengan ukuran sampel yang semakin besar. Hal ini disebabkan karena fungsi fitness yang digunakan pada penelitian ini adalah selisih dari nilai loglikelihood data bangkitan dari parameter tertentu dengan selisih nilai loglikelihood estimatornya. Sehingga semakin besar ukuran sampel maka semakin besar nilai log likelihoodnya. Hal tersebut menimbulkan nilai fitness yang semakin besar pula.

Penaksiran parameter distribusi Weibull dua parameter dan distribusi Weibull tiga parameter menggunakan algoritma genetika baik digunakan pada distribusi ini. Hal ini didukung dengan kecilnya selisih nilai fitness algoritma genetika yang diperoleh dibandingkan dengan algoritma *Particle Swarm Optimization* pada setiap *sample space* yang digunakan. Hal tersebut dikarenakan pada Algoritma Genetika terdapat operator evolusi yang dapat melakukan optimasi pada masalah kompleks dan ruang pencarian yang sangat luas.

III. Kesimpulan

Berdasarkan pembahasan yang dilakukan pada bab sebelumnya diperoleh hasil pada Penaksiran parameter distribusi Weibull dua parameter dan distribusi Weibull tiga parameter

menggunakan algoritma genetika dan algoritma PSO tidak melibatkan turunan. Pada algoritma PSO tidak terdapat operator evolusi seperti *crossover* dan mutasi. Sedangkan pada

N	Parameter			Algoritma Genetika				Algoritma PSO			
	α	β	λ	$\hat{\alpha}$	$\hat{\beta}$	$\hat{\lambda}$	Selisih Fitness	$\hat{\alpha}$	$\hat{\beta}$	$\hat{\lambda}$	Selisih Fitness
100	-	1	1	-	1,0	1,328	0,0063	-	1,197	1,416	1,5764
200	-	1	1	-	0,9994	0,9992	0,0002	-	0,9498	0,8899	0,0651
500	-	1	1	-	0,9832	1,0553	0,0085	-	1,0167	0,9809	0,5165
1000	-	1	1	-	0,2262	1,0183	0,0033	-	1,0474	0,9432	8,9963
1000	3	4	5	3,4197	4,8270	4,8270	0,0004	3,0187	3,5011	4,5001	0,1255

Algoritma Genetika memiliki operator evolusi yang dapat melakukan optimasi pada masalah kompleks dan ruang pencarian yang sangat luas. Penaksiran parameter distribusi Weibull dua parameter dan distribusi Weibull tiga parameter menggunakan algoritma genetika baik digunakan pada distribusi ini. Hal ini didukung dengan kecilnya selisih nilai fitness algoritma genetika yang diperoleh dibandingkan dengan algoritma *Particle Swarm Optimization*. Ukuran sampel juga mempengaruhi selisih fitness. Semakin besar ukuran sampel maka fitnessnya semakin bertambah besar.

IV. Daftar Pustaka

[1] Y. Bard, *Nonlinear parameter estimation*. New York: Academic Press, 1974.
 [2] L. G. Oyata, “Distribusi probabilitas weibull dan aplikasinya (pada persoalan keandalan (reliability) dan analisis rawatan (mantainability),” *Tadbir: Jurnal Manajemen Pendidikan Islam*, vol. 4, no. 2, pp. 44–66, 2016.
 [3] Y. Arkeman, *Algoritma genetika: teori dan aplikasinya untuk bisnis dan industri*. PT Penerbit IPB Press, 2012.

- [4] B. Santosa and P. Willy, *Metoda Metaheuristik konsep dan implementasi*. Guna Widya, Surabaya, 2011.
- [5] H. Rinne, *The Weibull distribution: a handbook*. Chapman and Hall/CRC, 2008.
- [6] L. Bain and E. M., *Introduction to Probability and Mathematical Statistics*. Duxbury Press, 1992.
- [7] T. Ferdian, A. Afriyudi, and B. Mutakin, "Optimasi penjadwalan perkuliahan di universitas tridinanti palembang," *JURNAL MAHASISWA TI S1*, 2013.
- [8] R. L. Haupt and S. E. Haupt, *Practical genetic algorithms*. John Wiley & Sons, 2004.
- [9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95- international conference on neural networks*, vol. 4. IEEE, 1995, pp. 1942–1948.
- [10] R.-M. Chen and H.-F. Shih, "Solving university course timetabling problems using constriction particle swarm optimization with local search," *Algorithms*, vol. 6, no. 2, pp. 227–244, 2013.
- [11] Q. Bai, "Analysis of particle swarm optimization algorithm," *Computer and information science*, vol. 3, no. 1, p. 180, 2010.