

Analisis Parameter Laju Infeksi COVID-19 Menggunakan Algoritma *Particle Swarm Optimization*

1st Rifqi Baihaqi Putra Heriansyah
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

rifqibaihaqi@student.telkomuniversity.ac.id

2nd Meta Kalista
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

metakallista@telkomuniversity.ac.id

3rd Marisa W.Paryasto
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

marisaparyasto@telkomuniversity.ac.id

Abstrak—Virus Covid-19 merupakan virus baru yang berawal mula dari Wuhan, hingga saat ini virus tersebut masih menyebar di seluruh penjuru dunia, salah satunya Indonesia yang juga terkena dampak pada virus Covid-19. Laju penyebaran kasus Covid-19 secara sistematis dapat dianalisis menggunakan model matematika SIR *Susceptible (S), Infected (I) dan Removed (R)*. Penggunaan istilah *Removed* pada artikel ini dikarenakan populasi ini terdiri dari atas individu yang sembuh (*Recovered*) dan Meninggal (*Death*) yang dimana laju penyebaran tersebut dapat diperoleh menggunakan metode optimasi yaitu *Particle Swarm Optimization (PSO)* Dalam hal ini, Metode PSO dapat memperoleh nilai dari laju penyebaran dengan cukup optimal dan cepat. Algoritma *Particle Swarm Optimization (PSO)* memiliki berbagai fungsi sebagai optimasi laju dari Covid-19 harus memiliki fungsi dasar untuk menentukan penyebaran dari suatu nilai. PSO dapat memiliki beberapa bagian terpenting seperti N (jumlah populasi), C (kemampuan individu (*cognitive*)) dan pengaruh sosial (*group*) dan menunjukkan nilai dari posisi sebuah partikel terhadap memori dari kelompok), Maxit, W, WD. Untuk percobaan yang telah dilakukan bahwa suatu nilai dari setiap posisi partikel semakin tinggi nilai posisi partikel semakin tinggi nilai terhadap beta dan delta yang dihasilkan. Pada pengujian algoritma *Particle Swarm Optimization (PSO)* nilai jumlah populasi yang paling optimal adalah 10, dengan nilai beta 0,9581, delta 0,9453 dan time 12,45081.

Kata Kunci— particle swarm optimization (PSO), Covid-19, optimasi.

I. PENDAHULUAN

Wabah Covid-19 yang pertama muncul di Cina pada bulan Desember tahun 2019 kasus ini semakin membesar hingga seluruh penjuru di berbagai belahan dunia sehingga *World Health Organization (WHO)* mengatakan bahwa wabah tersebut sebagai pandemik global. Wabah ini bisa dikenal dengan sebutan nama *CoronaVirus* menjadi virus yang sangat berbahaya bahkan hingga mematikan dengan tingkat penyebaran yang sangat begitu cepat hingga meluas ke seluruh penjuru dunia salah satunya hingga masuk ke Indonesia. *World Health Organization (WHO)* mencatat data pada tanggal 5 juni 2021 sebanyak 173.357.945 kasus positif Covid-19, 156.121.170 kasus sembuh dan 3.728.668 kasus meninggal dunia. Pada kasus Covid-19 di Indonesia tercatat 1.850.206 kasus positif, 1.701.784 kasus sembuh dan 51.449

kasus meninggal dunia. Dalam dampak wabah Covid-19 yang ada di Indonesia selain bidang kesehatan juga dapat berdampak pada bidang ekonomi, pendidikan, politik dan juga kehidupan sosial [1]. Upaya pemerintah untuk menanggulangi dampak kasus Covid-19 yang ada di Indonesia dengan memberlakukan sesuai dengan kebijakan Pembatasan Berskala Besar (PSBB) dan hingga proses suatu vaksin untuk mengurangi dampak pada kasus Covid-19 yang semakin banyak di Indonesia. Dalam tindakan vaksinasi masyarakat dinilai sangat kontroversial dengan munculnya berbagai merek vaksin sehingga banyak di berbagai kalangan masyarakat yang ada di Indonesia [1].

Pada analisis sentiment berada di antara berbagai bidang penelitian lainnya seperti Data Mining, *Natural Language Processing (NLP)* dan *Machine Learning* yang untuk melakukan suatu pengelompokan data-data, proses pada analisis laju parameter Covid-19 atau juga pendapat yang memiliki suatu aspek yang positif, negatif atau netral [2].

Secara garis klarifikasi dengan analisis suatu sentiment dapat membantu memberikan masukan dan juga tanggapan dengan cepat [3].

II. KAJIAN TEORI

A. *Particle Swarm Optimization (PSO)*

Algoritma *Particle Swarm Optimization (PSO)* awalnya dapat diusulkan oleh Kennedy dan Eberhart pada tahun 1995, *Particle Swarm Optimization (PSO)* adalah salah satu teknik dalam evolusioner, yang populasi pada PSO berdasarkan pada penelusuran algoritma dan diawali dengan suatu populasi yang disebut dengan *particle*. Pada partikel tersebut dapat bergerak melalui berbagai penelusuran ruang dengan *velocity* yang dinamis yang dapat disesuaikan menurut perilaku pada historynya. Maka dari itu, suatu partikel-partikel kecenderungan untuk bergerak ke penelusuran yang lebih baik setelah melewati berbagai banyaknya proses penelusuran. Pada PSO memiliki nilai kesamaan dengan algoritma genetika yang dimana dengan suatu populasinya yang secara acak dalam bentuk matriks. Namun, PSO tidak mempunyai operator evolusi yaitu pada *crossover* dan mutasi seperti sama dengan algoritma genetika [4].

1. *Swarm*: Suatu populasi dari suatu algoritma, dengan rentang $x^{(B)}$ dan $x^{(A)}$ dengan secara *random* sehingga didapat x_1, x_2, \dots, x_N . Dalam partikel j dan hasil pada

- kecepatannya pada iterasi i dinotasikan sebagai $x_j^{(i)}$ dan $v_j^{(i)}$.
- Dalam menentukan nilai pada fungsi/fitness pada kasus seperti saat ini, dapat ditentukan dengan fitness terbaik dan ditetapkan sebagai G_{best} . Dengan menggunakan Rumus: $f(x) = (100-x)^2$.
 - Particle**: Pada anggota (individu) pada suatu *swarm* tersebut. Dengan setiap partikel mempresentasikan suatu solusi yang dibidang potensial pada suatu permasalahan yang telah diselesaikan. Pada posisi dari suatu *particle* adalah dapat ditentukan dengan suatu representasi suatu solusi seperti saat ini.
 - Pbest** (Personal best): Pada posisi P_{best} suatu partikel yang dapat menunjukkan pada posisi partikel yang telah dipersiapkan untuk mendapatkan suatu solusi yang terbaik pada P_{best} .
 - Gbest** (Global best): Pada posisi yang terbaik terhadap *particle-particle* pada *swarm*.
 - Velocity** (Vektor): Pada vektor yang dapat menggerakkan pada proses optimasi yang dapat menentukan arah dimana suatu partikel dapat diperlukan dengan cara berpindah (*move*) untuk memperbaiki suatu posisinya dengan semula.
 - Berikut merupakan suatu formulasi matematika yang dapat menggambarkan suatu posisi dan kecepatan partikel pada suatu dimensi terhadap ruang tertentu [5].

B. Implementasi Particle Swarm Optimization (PSO)

Dalam implemtasi PSO dapat dilihat seperti fungsi berikut ini:

$$\min f(x) \quad (2.1)$$

dimana $X^{(B)} \leq X \leq X^{(A)}$

Terdapat fungsi $X^{(B)}$ adalah batas bawah dapat disebut dengan istilah *Lower Bound* (LB) dan $X^{(A)}$ adalah batas atas *Upper Bound* (UP) dari nilai X . Pada prosedur PSO dapat dilampirkan dengan berbagai langkah-langkah sebagai berikut [11]:

- Mengasumsikan bahwa ukuran pada setiap kelompok atau kawan (jumlah partikel) adalah N . Untuk mengurangi jumlah evaluasi fungsi dari yang diperlukan untuk menemukan solusi, sebaiknya ukuran N tidak terlalu besar, tetapi juga tidak terlalu kecil, agar dapat banyak kemungkinan dalam posisi untuk menuju solusi terbaik atau menjadi optimal. Jika pada suatu nilai terlalu kecil, maka sedikit pun kemungkinan menemukan posisi pada suatu partikel yang terbaik. Jika nilai terlalu besar juga akan membuat sebuah perhitungan menjadi panjang. Biasanya nilai yang dapat digunakan pada ukuran kawan adalah 20 sampai 30 partikel.
- Pada suatu populasi awal yang meningkat X dengan rentang $X^{(B)}$ dan $X^{(A)}$ pada secara acak sehingga didapat X_1, X_2, \dots, X_N . Setelah itu, untuk memudahkannya pada partikel j dan pada kecepatannya pada iterasi I dapat diintonasikan sebagai $X^{(i)}$ dan $V^{(i)}$ sehingga pada setiap partikel-partikel awal ini akan menjadi sebuah $X_1(0), X_2(0), \dots, X_N(0)$. Pada vektor $X_j(0), (j = 1, 2, \dots, N)$ dapat disebut dengan partikel atau vektor koordinat dari setiap partikel (seperti pada kromosom dalam algoritma genetika). Evaluasi terhadap nilai fungsi tujuan untuk setiap partikel dan dapat dinyatakan dengan:

$$f[X_1(0)], f[X_2(0)], \dots, f[X_N(0)]$$

- Pada perhitungan sebuah kecepatan dari semua partikel. Setiap semua partikel dapat bergerak menuju ke titik optimal dengan suatu kecepatan. Awalnya untuk semua kecepatan dari partikel diasumsikan sama dengan nol. Set iterasi $I = 1$.
- Pada iterasi ke- I , temukan 2 parameter penting untuk setiap partikel j .
- Melakukan pengecekan terhadap solusi yang sekarang sudah konvergen. Jika suatu posisi semua pada setiap partikel dapat menuju ke satu nilai yang sama, maka ini dapat disebut dengan konvergen. Jika suatu konvergen maka langkah 4 dapat diulang dengan memperbarui iterasi $I = I + 1$, dengan cara menghitung nilai baru dari P_{best-j} dan G_{best} . Pada suatu proses iterasi ini dapat dilanjutkan sampai dengan semua partikel menuju ke satu titik solusi yang sama. Biasanya akan ditentukan dengan kriteria penghentian (*stopping criteria*). Misalnya jumlah selisih solusi sekarang dengan solusi sebelumnya sudah sangat kecil.

C. Algoritma Particle Swarm Optimization (PSO)

Dalam metode algoritma *Particle Swarm Optimization* (PSO), sebagai berikut[7]:

Nilai $\min f(x)$ yang dimana $X(B) \leq X \leq X(A)$

Berikut ini adalah prosedur pada *Particle Swarm Optimization* (PSO) yang dapat dilihat dengan berbagai langkah-langkah dibawah ini [4]:

- Memperkirakan bahwa ukuran pada tiap kasus (jumlah partikel) adalah N . Dalam mengurangi jumlah suatu evaluasi fungsi yang sangat diperlukan untuk menemukan solusi pada suatu kasus tersebut.
- Meningkatkan dan menurunnya suatu kasus awal X dengan rentang $X(B)$ dan $X(A)$ secara acak dalam data yang telah dilihat sehingga mendapatkan X_1, X_2, \dots, X_N .
- Menghitung kasus yang telah didapatkan dari semua partikel. Semua partikel-partikel kasus yang telah didapatkan akan bergerak setiap waktunya menuju titik optimal dengan melihat data yang masuk. Awal dari semua data yang telah masuk diperkirakan sama dengan nol. Set iterasinya adalah $i=1$.
- Pada tahap selanjutnya dalam iterasi ke- I , dapat ditemukan 2 parameter yang penting untuk setiap partikel-partikel kasus j .
- Dalam pengecekan apakah partikel-partikel kasus mempunyai solusi yang sudah konvergen.

D. Transform Data

Pada model SIR *Susceptible* (S), *Infected* (I) dan *Removed* (R) dapat diatur oleh tiga variabel: $I(t)$ merupakan jumlah individu yang terinfeksi sebagai fungsi waktu $S(t)$ adalah jumlah individu yang rentan terhadap penyakit dan mungkin terinfeksi pada waktu t , dan $R(t) = N - S(t) - I(t) - I(t)$ adalah jumlah individu yang pulih pada waktu t , dapat mengikuti dinamika $I(t)$ dan $S(t)$, karena N total populasi yang telah ditetapkan. Kemudian secara persamaan untuk memudahkan dan juga setiap individu yang terinfeksi dari model SIR *Susceptible* (S), *Infected* (I) dan *Removed* (R) dapat dilihat sebagai berikut:

$$\frac{dS}{dt} = -\alpha SI \quad (2.3)$$

$$\frac{dI}{dt} = \alpha SI - \beta I \quad (2.4)$$

$$\frac{dR}{dt} = \beta I \quad (2.5)$$

E. Ordinary Differential Equation 45 (ODE 45)

Metode ODE45 merupakan penyelesaian terhadap ODE dengan pendekatan *Embedded Runge Kutta*. Dengan pendekatan tersebut diharapkan estimasi error dapat diperoleh dari dua komputasi *Runge Kutta* dengan order berbeda. ODE 45 biasa dituangkan dalam persamaan 2.6.

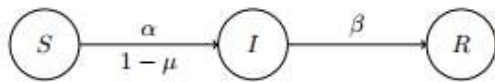
$$y' = f(t, y) \quad (2.6)$$

Keterangan:

- y' : solusi pada saat kejadian.
- f : nilai awal
- t : waktu kejadian awal
- y : nilai pertumbuhan

F. Susceptible, Infected dan Removed (SIR) Model

SIR (*Susceptible, Infected dan Removed*) model adalah model matematika yang digunakan dalam proses pemodelan ada, yang merupakan akronim dari *Susceptible (S), Infected (I)* dan *Removed (R)*. Penggunaan istilah *Removed* pada artikel ini dikarenakan populasi in terdiri atas individu yang sembuh (*Recovered*) dan Meninggal (*Death*). Model yang dibentuk berupa sistem persamaan diferensial biasa nonlinear. Proses pemodelan didasarkan pada alur penyebaran infeksi.



GAMBAR 1 SKEMA SIR UNTUK PERUBAHAN COVID-19

Keterangan:

- $S(t)$: Populasi *Susceptible*, populasi sehat yang rentan terkena virus.
- $I(t)$: Populasi *Infected*, populasi yang telah dikonfirmasi terinfeksi virus dengan menggunakan tes swab yang diuji melalui PCR.
- $R(t)$: Populasi *Removed*, populasi yang telah bebas dari virus, bisa diartikan sembuh maupun meninggal.
- α : Laju sukses infeksi dari populasi *infected* pada populasi *susceptible*.
- β : Proporsi kesembuhan (kematian) populasi *infected*.
- μ : Indeks efektivitas kontrol vaksinasi yang diberikan.

Berdasarkan skema dari Gambar 1 maka dapat dijelaskan dan asumsi yang digunakan untuk penyebaran COVID-19 maka dapat didefinisikan oleh persamaan 2.7 sebagai berikut.

$$\begin{cases} \frac{dS(t)}{dt} = -\alpha(1 - \mu)S(t) \frac{I(t)}{N} \\ \frac{dI(t)}{dt} = \alpha(1 - \mu)S(t) \frac{I(t)}{N} - \beta I(t) \\ \frac{dR(t)}{dt} = \beta I(t) \end{cases} \quad (2.7)$$

Dengan jumlah populasi setiap waktunya adalah $= S(t) + I(t) + R(t)$. Oleh karena populasi total tiap waktu bersifat konstan, model dapat ditransformasi menjadi model

tanpa dimensi. Didefinisikan $(t) = \frac{S(t)}{N}, i(t) = \frac{I(t)}{N}$ dan $r(t) = \frac{R(t)}{N}$.

III. METODE

A. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem merupakan analisis yang keperluan sistem guna mendukung pembuatan sistem. Terdapat dua jenis kebutuhan sistem yaitu analisis kebutuhan pada perangkat keras dan analisis kebutuhan pada perangkat lunak.

1. Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan dalam membuat sistem ini adalah laptop, dengan spesifikasi berikut:

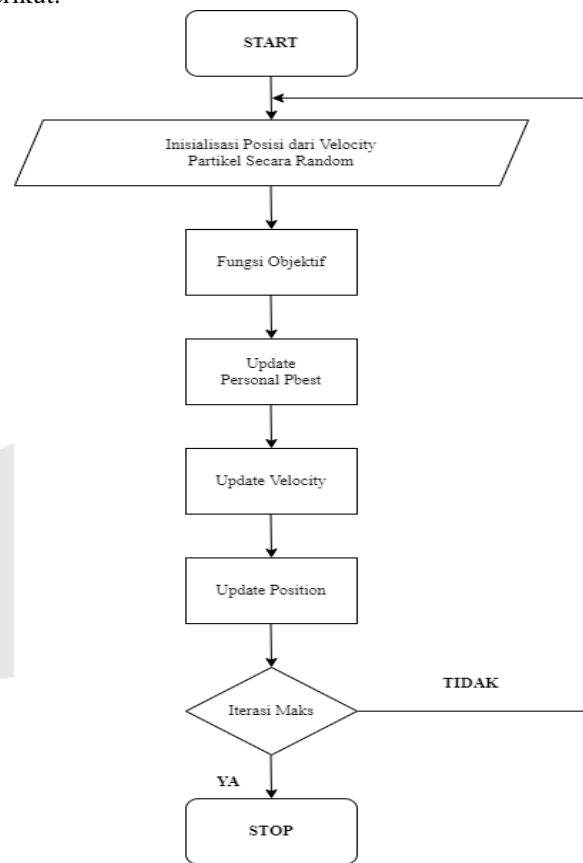
- Jenis laptop : ACER ASPIRE 5,
- Sistem Operasi : Windows 11,
- Prosesor : Intel Core i3 gen 11,
- RAM : 8 GB.

2. Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan dalam membuat sistem ini adalah aplikasi Matlab R2021.

B. Diagram Alir Algoritma Genetika

Pada Gambar 2 menunjukkan diagram alir cara kerja dari algoritma *Particle Swarm Optimization (PSO)* sebagai berikut:



GAMBAR 2 FLOWCHART ALGORITMA PSO.

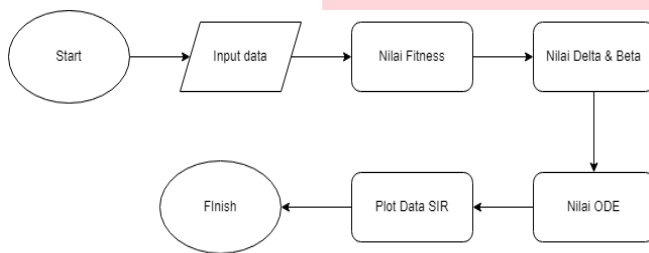
Langkah-langkah yang dilakukan pada algoritma genetika untuk penelitian ini yaitu:

1. Masukkan data pertumbuhan penyakit COVID-19 yang akan diolah.

2. Masukkan nilai populasi yang akan diproses kedalam pengujian.
3. Sistem akan memulai proses pengolahan pengujian.
4. Setelah sistem mengolah pengujiannya, akan menampilkan nilai beta, delta dan waktu berapa lama pengujian diproses berlangsung.
5. Sistem dapat mengeluarkan hasil dari ODE sesuai pengujian yang dibutuhkan.
6. Setelah itu, sistem akan menampilkan hasil GUI, yang terdapat sebuah plot data dari hasil pengujian dan menampilkan plot data SIR *Susceptible* (S), *Infected* (I) dan *Removed* (R).

C. Diagram Alir GUI

Pada gambar 3 menunjukkan gambaran umum dari cara kerja GUI pada Aplikasi Matlab.



GAMBAR 3
DIAGRAM ALIR GUI

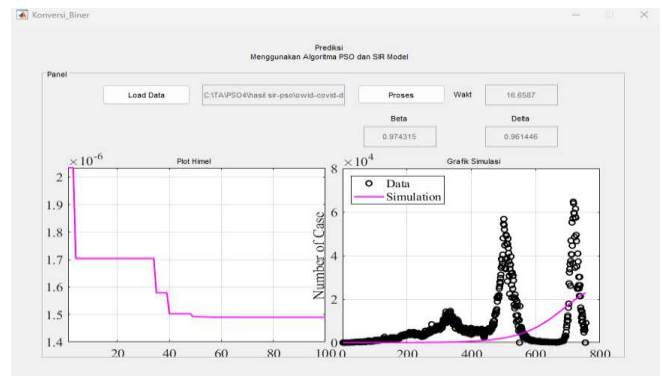
Langkah-langkah yang dilakukan dalam GUI sebagai berikut.

1. Masukkan data COVID-19 yang akan diolah.
2. Memasukkan nilai populasi yang akan diproses kedalam pengujian.
3. Mendapatkan nilai ODE bertujuan untuk mengetahui nilai *error* yang di dapat selama pengujian dari proses yang dilakukan sebelumnya.
4. Hasil GUI akan memprediksi nilai dari beta dan delta yang ditampilkan setelah proses pengujian.
5. Hasil GUI akan mengeluarkan nilai ODE setelah proses pengujian.
6. Setelah itu, hasil GUI yang terdapat sebuah plot data dari hasil pengujian dan menampilkan plot data SIR *Susceptible* (S), *Infected* (I) dan *Removed* (R).
7. Plot data SIR *Susceptible* (S), *Infected* (I) dan *Removed* (R). ini bertujuan menampilkan nilai SIR dalam bentuk grafik.

IV. HASIL DAN PEMBAHASAN

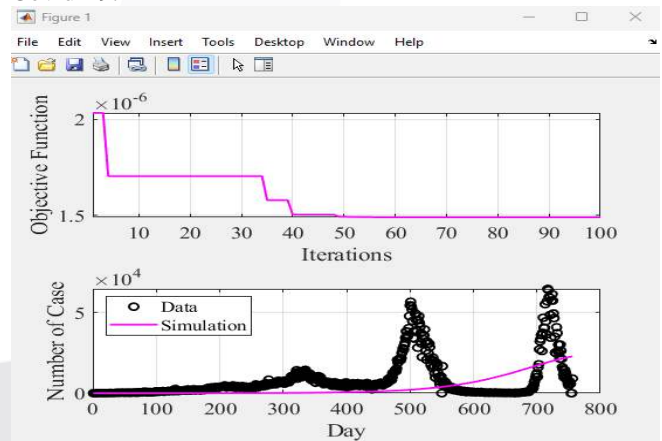
A. Analisis Parameter Infeksi Menggunakan PSO

Pada gambar 4 menjelaskan garis y merupakan angka populasi dan garis x merupakan waktu yang dihasilkan pada setiap populasi meningkat akan mendapatkan waktu yang semakin tinggi. Dari hasil tampilan data pada GUI setelah data berhasil dimasukkan dan dapat diproses, data akan diproses setelah menekan tombol proses, setelah data dimasukkan data akan dilakukan optimasi dan mengeluarkan nilai terhadap beta dan delta.



GAMBAR 4
TAMPILAN GUI.

Pada gambar 5 menjelaskan garis y bahwa semakin meningkat fungsi objektif/jumlah populasi pada kasus yang terkonfirmasi Covid-19, maka garis x menunjukkan bahwa jumlah iterasi yang digunakan oleh setiap partikel meningkat akan semakin meningkat juga nilai pada iterasi data dan waktu yang dihasilkan. Dari hasil pengujian yang telah dilakukan dapat dilihat bahwa dari nilai beta dan delta tersebut akan berubah, jika nilai terhadap populasi meningkat maka semakin besar nilai beta dan delitanya. Setelah nilai beta dan delta sudah didapatkan dalam proses pengujian, maka akan berlanjut pada tahap selanjutnya yaitu menganalisis pengujian dengan metode *Ordinary Differential Equation 45* (ODE45), pada ODE45 dapat memprediksi terhadap laju Covid-19.



GAMBAR 5
NILAI ODE.

Pada gambar 6 menjelaskan bahwa semakin banyak populasi yang terkonfirmasi Covid-19 maka akan semakin tinggi nilai terhadap alpa, beta dan waktu pengujian yang dilakukan semakin tinggi.

```

bd =

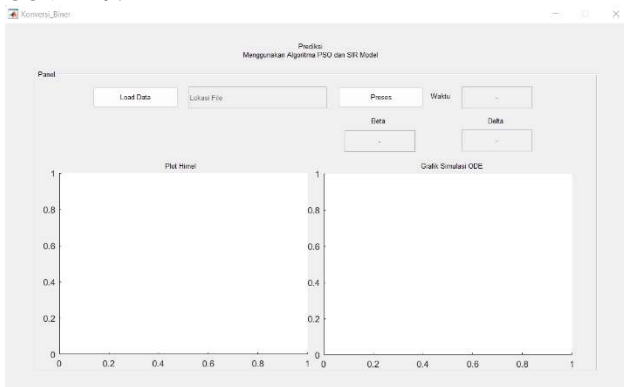
    0.9743    0.9614

Elapsed time is 11.875565 seconds.
fx >>
  
```

GAMBAR 6
NILAI DELTA DAN BETA.

B. Analisis pada *Graphical User Interface* (GUI)

Pada *Graphical User Interface* (GUI) a proses yang dilakukan oleh aplikasi ini dimulai dengan tampilan awal pada Gambar 7 pada gambar tersebut akan disajikan dengan nilai beta, nilai delta, grafik ODE45 dan grafik pertumbuhan COVID19.



GAMBAR 7
TAMPILAN AWAL PADA GUI.

Kemudian langkah selanjutnya adalah memasukan dataset yang akan digunakan. Setelah dataset dimasukan maka aplikasi akan melakukan proses algoritma *particle swarm optimization* dengan memunculkan nilai populasi yang akan digunakan. Setelah nilai populasi didapatkan maka nilai beta dan delta juga akan ditampilkan melalui aplikasi, setelah nilai beta dan delta didapatkan maka proses berikutnya membuat grafik ODE yang digunakan untuk analisis pertumbuhan COVID-19 dengan menampilkan data dengan bentuk grafik.

C. Pengujian Teknis

Pada pengujian ini bertujuan untuk melihat seberapa tinggi pengaruh perubahan jumlah populasi terhadap nilai beta, delta dan waktu yang telah didapatkan.

TABEL 1
TABEL PENGUJIAN TEKNIS ALGORITMA PSO

Populasi	C1	C2	W	WD	Maxit	Beta	Delta	Time
10	1	1	1	0,99	100	0,9581	0,9453	12,45081
20	1	1	1	0,99	100	0,7678	0,7554	13,80120
30	1	1	1	0,99	100	1,0220	1,0090	17,95629
40	1	1	1	0,99	100	0,9955	0,9826	20,11558
50	1	1	1	0,99	100	0,7276	0,7153	24,47838
60	1	1	1	0,99	100	0,7641	0,7517	26,32803

Pada Tabel 1 pengujian kali ini didapatkan bahwa jumlah populasi pada algoritma *Particle Swarm Optimization* (PSO) memiliki pengaruh terhadap nilai beta, delta dan waktu yang telah didapatkan. Semakin tinggi nilai populasi yang digunakan maka semakin berubah nilai beta dan delitanya. Namun, waktu yang didapatkan akan mengalami perubahan cukup tinggi yang didapatkan dari algoritma *Particle Swarm Optimization* (PSO) pada pengujian kali ini.

V. KESIMPULAN

Berdasarkan hasil implementasi dan proses pengujian yang telah dilakukan pada GUI Matlab dengan menggunakan metode algoritma *Particle Swarm Optimization* (PSO) dapat diambil kesimpulan bahwa:

1. Metode algoritma *Particle Swarm Optimization* (PSO) pada GUI Matlab mampu prediksi laju infeksi Covid-19 berdasarkan hasil dari pengujian alpha, beta dan kuesioner yang telah disebarakan.
2. Pada hasil pengujian *white box* juga didapatkan bahwa sebuah fungsi dapat berjalan 100% walaupun ada sedikit kendala dalam proses pengerjaan.
3. Parameter yang digunakan pada tugas akhir ini adalah model SIR atau (*Susceptible, Infected dan Removed*) model dengan menentukan jumlah kasus aktif, kasus kematian serta kasus sembuh COVID-19 dari 2020-2022.
4. Hasil pengujian beta pada GUI Matlab dari 10 responden dengan total 5 pertanyaan, sebesar 94% responden yang memilih sangat setuju dan setuju. Maka hasil kuesioner pada pengujian validitas dapat dinyatakan dengan status valid dan pada uji reliabilitas yaitu sangat tinggi.

REFERENSI

- [1] A. B. S. T. Ristasari Dwi Septiana, "Analisis Sentimen Vasinasi Covid-19 Pada Twitter Menggunakan Naive Bayes Classifier Dengan Feature Selection Chi-Squared Statistic Dan Particle Swarm Optimization," *Jurnal Sistem Komputer dan Kecerdasan Buatan*, vol. V, no. 1, pp. 49-50, September 2021.
- [2] A. A. K. Y. J. M. N.-K. Mahmoud Al-Ayyoub, "A comprehensive survey of arabic sentiment analysis," *Information Processing and Management*, vol. 56, no. 2, pp. 320-342, 2019.
- [3] I. P. E. N. K. T. B. O. Juen Ling, "Analisis Sentimen Menggunakan Metode Nalve Bayes Classifier Dengan Seleksi Fitur Chi Square," *E-Jurnal Matematika*, vol. 3, no. 3, pp. 92-99, 2014.
- [4] A. S. M. H. P. Maickel Tuegeh, "Optimal Generator Scheduling Based On Particle Swarm Optimization," *Seminar Nasional Informatika 2009 (semnasIF 2009)*, 2009.
- [5] S. P. S. A. P. Khozin Mu'tamar, "Analisis Penyebaran Covid-19 Dengan Menggunakan Model SIR dan Vaksinasi Serta Estimasi Parameter," *Jurnal Ilmu Matematika dan Terapan*, vol. 15, no. 2, pp. 323-334, June 2021.
- [6] I. F. Mukhlis, "Optimasi PSO untuk Peramalan Harga Emas Secara Rentet Waktu," *Jurnal GEMA AKTUALITA*, vol. 5, no. 1, Juni 2016.
- [7] M. H. S. M. S. N. Syahri Mu'min, "Pergerakan Otonom Pasukan Berbasis Algoritma Boids Menggunakan Metode Particle Swarm Optimization," *Journal of Animation and Games Studies*, vol. 1, no. 1, April 2015.
- [8] A. D. Syamsiah, "Analisis Particle Swarm Optimization Terhadap Kepuasan Taman Dadap

Merah Dengan Model SVM," *Seminar Nasional Riset dan Teknologi (SEMNAS RISTEK)*. Jakarta, 27 Januari 2020, 2020.

- [9] A. J. A. Z. A. Monica Widiastri, "Penerapan Particle Swarm Optimization untuk Penentuan Parameter Regularisasi pada Kernel Regularized Discriminant Analysis," *Industrial Electronics Seminar 2011 (IES 2011)*, Indonesia, October 26, 2011, 2011.
- [10] M. D. H. R. Ratna Nur Ayu, "Simulasi Animasi Tiga Dimensi Gerombolan Ikan dalam Akuarium Virtual Menggunakan Algoritma Particle Swarm Optimization," *Jurnal Aksara*, vol. 2, no. 2, 2013.

