BAB 2
LANDASAN TEORI

2.1 Covid-19

Penyakit *Coronavirus Disease* (COVID-19) adalah sebuah penyakit jenis baru yang diakibatkan oleh virus Sars-CoV-2. COVID-19 termasuk ke dalam jenis penyakit yan dapat menular ke manusia melalui hewan. Pertama kali ditemukannya COVID-19 adalah di Tiongkok, Wuhan. Virus ini pada awalnya diduga diakibatkan oleh pasar grosir makanan laut huanan yang menjual banyak spesies hewan hidup. Penyakit ini dengan cepat menyebar di dalam negeri ke bagian lain China [5].

Diagnosis COVID-19 pada umumnya dilakukan dengan melakukan tes Reverse Transcription Polymerase Chain Reaction (RT-PCR) pada saluran pernapasan untuk mendeteksi keberadaan material genetik dari sel, termasuk virus dan bakteri. Tes ini dapat dilakukan di laboratorium yang memiliki fasilitas yang mendukung dan lengkap.

2.2 Pre-Processing Data

Dalam pembelajaran *Machine Learning*, *Pre-processing* data termasuk hal penting dalam proses pembuatan *Machine Learning* karena data yang berkualitas harus didasarkan juga dari keputusan yang berkualitas. *Pre-processing* data seringkali digunakan untuk mengurangi kesalahan data dan sistematis bias dalam data mentah sebelum analisis apapun terjadi[6].

Data yang digunakan adalah dataset yang akan dinormalisasikan agar dapat diimplementasikan ke dalam algoritma yang digunakan dan data hasil wawancara dengan pakar. Tahap ini diperlukan untuk membersihkan data dari elemen yang tidak perlu dengan tujuan memproses data perhitungan yang lebih optimal ketika masuk ke tahap implementasi algoritma.

2.3 Swarm Intelligence

Swarm Intelligence adalah salah satu teknik kecerdasan buatan yang berlandaskan kepada perilaku kolektif (*collective behaviour*) pada sistem yang terdesentralisasi dan dapat mengatur dirinya sendiri (*self-organizing*)[7]. Pada umumnya, sistem dengan teknik *Swarm Intelligence* adalah sebuah populasi yang

terdiri atas anggota yang sederhana, yang berinteraksi secara lokal dengan sesama anggota, dan juga berinteraksi dengan lingkungan sekitar.

2.3.1 Sifat Dasar Swarm Intelligence

Sifat sifat dari *Swarm Intelligence* diantaranya:

1. Memiliki individu yang banyak.
2. Setiap individu bersifat homogen.
3. Interaksi antar individu yang berdasar pada aturan yang memanfaatkan informasi lokal baik secara langsung atau melalui lingkungannya.

Setiap individu dapat mengatur diri mereka sendiri dengan kemampuan interaksi antar sesama maupun dengan lingkungan *swarm intelligence* disebut dengan *selforganizes*.

2.3.2 Prinsip Dasar Swarm Intelligence

Prinsip-prinsip dasar *Swarm Intelligence* adalah sebagai berikut:

1. Prinsip kedekatan : Setiap individu harus dapat menyesuaikan ruang dan waktu dengan perhitungan sederhana.
2. Prinsip kualitas : setiap individu harus dapat menanggapi faktor lingkungan.
3. Prinsip respon beragam : setiap individu tidak seharusnya melakukan kegiatannya pada jalur yang terlalu sempit.
4. Prinsip stabilitas : populasi tidak harus mengubah modus perilakunya setiap kali perubahan lingkungan.
5. Prinsip adaptasi : populasi harus mampu mengubah modus perilaku ketika komputasi bernilai penting atau berharga [7].

2.3.3 Algoritma Swarm Intelligence

Adapun beberapa contoh algoritma atau metode yang populer dari *swarm intelligence* di antaranya [7]:

1. Particle Swarm Optimization (PSO)
   PSO adalah algoritma yang berdasar pada kawanan serangga, semut, rayap, lebah atau burung dengan metode pencarian acak berbasis populasi yang didasarkan pada perilaku perpindahan individu dalam kawanan.

2. Ant Colony System (ACS)
   ACS adalah algoritma yang berdasar pada semut merupakan algoritma yang digunakan untuk menyelesaikan masalah optimasi yang terinspirasi dari perilaku semut.

3. Artificial Bee Colony (ABC)
   ABC adalah kecerdasan buatan yang menirukan koloni lebah dalam mencari sumber nektahr (sari bunga).

4. Firefly Algorithm (FA)
   FA adalah sebuah algoritma metaheuristik yang terinspirasi dari perilaku kedip cahaya kunang-kunang.

2.4 Ant Colony System

ACS adalah sebuah metode yang dihasilkan melalui pengamatan terhadap semut. Pada algoritma ACS, semut berfungsi sebagai agen yang ditugaskan untuk mencari solusi terhadap suatu masalah optimisasi[8].

ACS meiliki tiga karakteristik utama, antara lain : aturan transisi status, aturan pembaharuan pheromone lokal, dan aturan pembaharuan *pheromone* global [9].

1. Aturan Transisi Status
   Aturan transisi status berlaku pada seekor semut yang awalnya berada pada titik t memilih untuk menuju ke titik v, kemudian diberikan bilangan pecahan acak q dimana $0 \leq q \leq 1$. Variabel $q0$ adalah sebuah parameter probabilitas pada saat semut melakukan eksplorasi pada setiap tahapan, dimana ($0 \leq q0 \leq 1$) dan $p_k(t,v)$ adalah probabilitas dimana semut k memilih untuk bergerak dari titik t ke titik v. Jika $q \leq q0$ maka pemilihan titik yang akan dituju menerapkan aturan yang ditunjukkan oleh Persamaan 2.1.

$$\begin{aligned} temporary(t,u) = [\tau(t,u_i)].[\eta(t,u_i)]^{\beta}, i = 1,2,3.....,n \\ v = max[\tau(t,u_i)].[\eta(t,u_i)^{\beta} \end{aligned} \quad (2.1)$$

dengan v = titik yang akan dituju sedangkan jika $q \geq q0$ digunakan Persamaan 2.2

$$v = p_k(t,v) = \frac{[\tau(t,v)].[\eta(t,v)]^{\beta}}{\sum_{i=1}^{n}[\tau(t,u_i)].[\eta(t,u_i)]^{\beta}} \tag{2.2}$$

dengan $\eta(t,u_i) = \frac{1}{Jarak(t,u_i)}$ dimana

$\tau(t,u)$ = Nilai dari jejak *pheromone* pada titik (t,u)

$\eta(t,u)$ = Fungsi heuristik yang dipilih sebagai *invers* jarak antara titik t dan u

β = Parameter kepentingan relatif dari informasi *heuristic*

Pembaruan pheromone pada algoritma ACS dibagi menjadi 2, antara lain : aturan pembaruan *pheromone* lokal dan aturan pembaruan *pheromone* global.

2. Aturan Pembaruan *Pheromone* Lokal

   Dalam melakukan perjalanan untuk mencari jalur terpendek, semut mengunjungi ruas-ruas dan mengubah tingkat pheromone pada ruas-ruas tersebut dengan menerapkan pheromone lokal yang ditunjukkan oleh Persamaan 2.3.

$$\tau(t,v) \leftarrow (1-\rho).\tau(t,v) + \rho.\triangle t(t,v)$$
$$\triangle t(t,v) = \frac{1}{L_{mn}.c} \tag{2.3}$$

   dimana

   L_{mn} = panjang tur yang diperoleh

   c = jumlah lokasi

   ρ = parameter dengan nilai 0 sampai 1

   $\triangle t(t,v)$ = perubahan *pheromone*

   ρ adalah sebuah parameter (koefisien evaporasi), yang merupakan besarnya suatu koefisien *pheromone evaporation*. Evaporasi pada *pheromone* menyebabkan semua semut tidak melewati jalan yang sama dengan semut sebelumnya sehingga menghasilkan adanya jalan lain baru yang lebih banyak.

3. Aturan pembaharuan *pheromone* global

   Pembaruan *pheromone* global dilakukan oleh semut yang membuat jalur terpendek sejak semut bergerak. Pada akhir iterasi, setelah semut menyelesaikan tur, jalan terbaik yang dilewati oleh semut akan ditaruh sejumlah *pheromone*. Tingkat *pheromone* kemudian akan diperbarui dengan

menerapakan aturan yang ditunjukkan oleh Persamaan 2.4

$$\tau(t,v) \leftarrow (1-\alpha).\tau(t,v) + \alpha.\triangle\tau(t,v)$$
$$\triangle\tau(t,v) = \begin{cases} L_{g_b}^{-}1 & \text{jika (t,v)} \in \text{tur terbaik} \\ 0 \end{cases} \quad (2.4)$$

dimana

$\tau(t,v)$ = nilai pheromone akhir setelah mengalami pembaruan lokal
L_{gb} = panjang jalur terpendek pada akhir siklus
α = parameter dengan nilai antara 0 sampai 1
$\triangle t(t,v)$ = perubahan *pheromone*

2.5 Simple Ant Colony Optimization

Algortima *Simple Ant Colony Optimization* (SACO) merupakan bagian dari algoritma metaheuristik yang didasarkan pada kinerja koloni semut saat mereka saling mencari makanan. Pada tahap awal, semut bergerak ke segala arah mencari sumber terbaik makanan tanpa tujuan. Setelah setiap perjalanan, semut meninggalkan *pheromone*. *Pheromone* ini dapat dideteksi oleh semut lain yang membuat berjalan ke arah yang sama dan meninggalkan *pheromone* yang semakin pekat [10]. Keuntungan dari algoritma ini adalah proses dilakukan dengan melewati banyak jalan dan akan memilih jalur yang optimal atau terpendek.

Algorithm 1 SACO Algorithm

```
begin
Initialize
while stopping criterion not satisfied do do
  Position each in a starting node
  repeat
    for each ant do
      Choose next node by applying the state transition rule
      Apply step by step pheromone update
    end for
  until every ant has built a solution
  Update best solution
  Apply offline pheromone update
end while
end
```

Pada umumnya, algoritma SACO adalah sebagai berikut : Seekor semut memulai jalan atau tour melalui sebuah titik secara acak. Kemudian semut akan mungunjungi titik lain dan diikuti oleh semut lain satu persatu dengan tujuan akhir menghasilkan sebuah tour. Pemilihan jalur atau titik yang akan dilalui didasarkan pada fungsi probabilitas, yang bernama (*state transition rule*), probabilitas ini mempertimbangkan *visibility* pada titik tersebut dan jumlah *pheromone* yang terdapat pada ruas yang menghubungkan titik tersebut karena semut lebih suka bergerak ke titik yang memiliki ruas yang pendek dan memiliki *pheromone* yang tinggi.

Tabulist merupakan sebuah memori yang dimiliki setiap semut, yang merupakan semua titik yang telah dilalui oleh semut. *Tabulist* berfungsi mencegah semut pergi ke titik yang pernah dikunjungi oleh semut tersebut sebelumnya. Setelah semua semut menyelesaikan tour, diterapkan aturan *pheromone* global pada setiap semut. Setelah dilakukan penguapan *pheromone* pada semua ruas dilakukan, setiap semut menghitung panjang dari tour dan meninggalkan *pheromone* pada bagian dari tour yang sebanding dengan kualitas dari solusi yang dihasilkan. Semakin pendek sebuah tour yang dihasilkan oleh setiap semut, maka semakin besar jumlah *heromone* yang ditinggalkan pada edge-edge yang dilalui.

Untuk membuat simulasi dengan menggunakan algortima *Simple Ant Colony Optimization* (ACO) diperlukan tiga hal atau rumus. Ketiga rumus tersebut antara lain untuk menghitung probabilitas, menghitung jumlah *pheromone* pada rute yang dilewati semut, dan menghitung penguapan pada *pheromone*.

$$P_{i,j} = \begin{cases} \frac{\tau_{i,j}^{\alpha}}{\sum_{j \notin N_i^{(k)}} \tau_{i,j}^{\alpha}}, & \text{jika } j \in N_i^{(k)} \\ 0, & \text{jika } j \notin N_i^{(k)} \end{cases} \tag{2.5}$$

Pada Persamaan 2.5 merupakan rumus untuk menghitung probabilitas dimana α sebagai derajat kepentingan *pheromone* dan $N_i^{(k)}$ merupakan pilihan yang dimiliki semut k lain pada saat sedang berada di simpul i. Semut k lain pada simpul i akan mengandung semua simpul yang tersambung ke simpul i secara langsung dan bisa dituju, kecuali simpul yang sebelumnya sudah pernah dikunjungi.

$$\tau_{i,j} \leftarrow \tau_{i,j} + \triangle \tau^{k} \tag{2.6}$$

Pada 2.6 merupakan rumus untuk menghitung jumlah *pheromone* pada rute yang dilewati semut. Ketika semut k melewati ruas, semut akan meninggalkan

pheromone yang terdapat pada ruas ij pada setiap ruas yang dilewati.

$$\tau_{i,j} \leftarrow (1-\rho)\tau_{i,j}, j; \forall (i,j) \in A \tag{2.7}$$

Pada 2.7 merupakan rumus untuk menghitung penguapan pada *pheromone*. Kemungkinan suatu ruas akan dipilih lagi pada iterasi berikutnya adalah dengan meningkatnya nilai *pheromone* pada ruas$i-j$. Setelah sejumlah simpul dilewati, akan terjadi proses pengeuapan *pheromone*.

2.6 Fuzzy Inference System

Algoritma FIS merupakan kerangka komputasi yang didasarkan pada dasar prinsip penelaran *fuzzy*. *Fuzzy Inference System* pada dasarnya terdiri atas 4 unit, yaitu:

1. Fuzzification
   Tahap fuzzifikasi merupakan suatu tahap yang mengubah suatu input dari bentuk tegas (crisp) menjadi *fuzzy* (variabel linguistik) yang biasa disajikan dalam bentuk himpunan-himpunan *fuzzy* dengan suatu fungsi kenggotaannya masing-masing. Langkah pertama adalah menentukan label *fuzzy* pada daerah batasan tegas dari setiap masukan. Fungsi keanggotaan dihasilkan dengan menuliskan bilangan satu per satu, yaitu derajat keanggotaan, untuk setiap masukan yang mungkin dari label yang diberikan [11].
2. Penalaran Logika *Fuzzy*
3. Basis Pengetahuan
   - Basis data, yang membentuk fungsi keanggotaan dari himpunan *fuzzy* yang terkait dengan nilai dari variabel linguistik yang dipakai.
   - Basis aturan, yang membuat aturan berupa implikasi *fuzzy*.
4. Defuzzification
   Defuzzifikasi berfungsi mengubah *output fuzzy* menjadi *value crisp* yang berdasarkan kepada fungsi keanggotaan yang telah ditentukan [12].

Fuzzy Inference System pada umumnya memiliki tiga metode yang digunakan dalam aturan logika *fuzzy* yaitu:

- Metode Tsukamoto

  Metode ini merupakan metode dimana himpunan *fuzzy* digunakan untuk merepresentasikan setiap aturan dengan fungsi keanggotaan yang monoton. Setiap aturan yang direpresentasikan berbentuk JIKA-MAKA atau *IF-THEN*. Hasil akhir didapatkan dengan hasil inferensi dari tiap aturan diberikan secara tegas (*crisp*) berdasarkan predikat.

- Metode Mamdani

  Metode ini memiliki himpunan aturan yang bersifat tidak saling bergantung Hal ini menyebabkan setiap aturan yang berbentuk implikasi (sebab-akibat) dengan bentuk konjungsi (AND) mempunyai nilai keanggotaan minimum (MIN) dan pada konsekuen gabungan berbentuk maksimuma (MAX).

- Metode Sugeno

  Metode ini memiliki penalaran yang hampir sama dengan metode Mamdani, akan tetapi tidak semua *output* sistem berupa himpunan *fuzzy*. *Output* dari metode ini dapat juga berupa konstanta atau persamaan linear.