BAB II
TINJAUAN PUSTAKA

A. Penelitian Terdahulu

Penelitian yang dilakukan oleh Septiani (2020) membahas tentang penyakit hepatitis. Penelitian ini menggunakan metode algoritma C4.5 dan dioptimasi dengan algoritma algoritma genetika. Evaluasi dalam pengujian menggunakan *algoritma C4.5* dengan seleksi fitur *algoritma genetika* menghasilkan akurasi 89,71%, sedangkan evaluasi tanpa fitur seleksi menghasilkan akurasi 77,29% dapat disimpulkan ketika menggunakan optimasi *algoritma genetika* akurasi mendapatkan peningkatan sebesar 12,42%. Perbedaaan dengan penelitian yang dilakukan penulis yaitu dari algoritma yang digunakan, penulis menggunakan *algoritma support vector machine* sedangkan penelitian terdahulu *algoritma C4.5,* selain itu data yang digunakan berbeda penelitian terdahulu didapatkan dari *machine learning repository UCI* (Universitas California Ivene) sedangkan penulis didapatkan dari *Kaggle.com*.

Penelitian klasifikasi penyakit gigi dan mulut dilakukan oleh Puspitasari *et al.* (2018). Penelitian ini menggunakan metode *support vector machine.* Hasil penelitian ini menghasilkan akurasi sebesar 94,442% dengan menggunakan nilai parameter SVM yaitu lamda 0,1 dengan gamma 0,1 dan C (*complexity*) 1, epsilon 1.10^{-10} dengan itermax 50 dan melakukan 5 kali iterasi. Proses klasifikasi ini menggunakan *dataset* terbatas sebanyak 122 data dalam empat kelas yaitu pulpitis, gingivitis, nekrosis pulpa dan periodontitis. Penelitian ini juga menggunakan kernel RBF. Perbedaan penelitian terdahulu dengan penelitian yang dilakukan yaitu terdapat pada penyakit yang diklasifikasi, jumlah data dan parameter yang digunakan.

Prediksi penyakit diabetes menggunakan metode *support vector machine* dan algoritma genetika dilakukan oleh Handayanna (2015). Hasil penelitian menggunakan model *support vector machine* dapat nilai *accuracy*

adalah 74,21 dengan nilai *precision* 74,75% dan nilai AUC 0,753. Pengujian menggunakan SVM-GA mendapatkan nilai *accuracy* 75,26 dengan nilai *precision* 76,25 dan nilai AUC adalah 0,771. Di dapatkan sebuah kesimpulan pengujian diabetes menggunakan *support vector machine* dan *algoritma genetika* lebih baik dari pada hanya menggunakan *support vector machine* saja. Dengan demikian SVM-GA dapat memberikan pemecahan untuk permasalahan prediksi penyakit diabetes lebih baik. Perbedaan dengan penelitian yang dilakukan yaitu pada hasil yang didapat pada model tidak mencari nilai AUC, study kasus yang digunakan yaitu *stroke.*

Prediksi penyakit systemic lupus erythematosus yang dilakukan oleh Putri & Setiawan (2021) menggunakan metode algoritma genetika, karena memiliki akurasi lebih besar dari 94%. Tujuan penelitian ini adalah prediksi apakah terindikasi penyakit lupus atau tidak sehingga menekan angka kematian akibat systemic lupus erythematosus. Data pada penelitian ini diambil dengan cara menyebar questioner pada pasien Rumah Sakit Umum daerah Arifin Achmad. Hasil penelitian ini membuktikan bahwa algoritma genetika dapat melakukan prediksi penyakit SLE, ditemui 14 pasien yang tidak terdeteksi dari 30 pasien, 9 orang pasien terdeteksi penyakit SLE menyerang otak, 6 orang terdeteksi SLE yang menyerang ginjal dan 1 orang terdeteksi penyakit SLE menyerang kulit. Perbedaan penelitian ini dan yang dilakukan penulis yaitu pada penyakit yang diklasifikasi dan metode yang digunakan pada penulis yaitu SVM-GA.

Penelitian menggunakan metode SVM-GA yang dilakukan oleh Kusumaningrum (2017) membahas klasifikasi *miccroaray* data dengan metode *support vector machine* dan algoritma genetika. Hasil pada penelitian ini menggunakan SVM-GA menghasilkan nilai lebih baik dibandingkan hanya menggunakan metode SVM. Hasil akurasi menggunakan SVM sebesar 75%, sedangkan SVM-GA menghasilkan akurasi sebesar 90%. Pada penelitian ini menggunakan Grid Search SVM yang menghasilkan sensitivitas sebesar 60%, sedangkan SVM-GA menghasilkan 77,78% lebih tinggi dari SVM. Perbedaan penelitian ini dengan penelitian yang dilakukan penulis

yaitu data pada penelitian terdahulu menggunakan 2 data, data Colon Cancer dan Leukemia sedangkan penelitian yang dilakukan penulis menggunakan 1 data yaitu *stroke*.

Prediksi penyakit *stroke* juga dilakukan oleh Amelia *et al*. (2022) menggunakan atribut yang berpengaruh dalam data penyakit *stroke*. Penelitian ini menggunakan metode *support vector machine* dengan *relief-f.* Penelitian ini menggunakan data dari Kaggle sebagai acuan yang memiliki 11 kolom dan 3426 baris. Hasil yang disimpulkan pada penelitian prediksi penyakit *stroke* menggunakan Algoritma SVM adalah model prediksi yang dibangun menggunakan algoritma SVM dengan *Relief-f* menghasilkan akurasi sebesar 100%. Perbedaan pada penelitian yang dilakukan yaitu melakukan pengembangan pada jumlah data, dan melakukan optimasi menggunakan *Algoritma Genetika.*

B. Landasan Teori

1. Machine Learning

Machine learning (ML) atau disebut juga Mesin Pembelajaran merupakan cabang AI yang berfokus pada pengembangan system tanpa harus diprogram manusia berulang-ulang. *Machine learning* (ML) membutuhkan data valid untuk bahan belajar saat proses training (pelatihan) sebelum dilakukan proses testing (pengujian) untuk hasil yang optimal (Cholissodin & Soebroto, 2021). *Machine learning* terbagi menjadi tiga kategori yaitu *supervised learning, unsupervised learning, reinforcement learning* (Roihan *et al*., 2020).

Supervised learning adalah metode klasifikasi yang memiliki data diberikan label untuk mengklasifikasi kelas yang tidak dikenal. Beberapa algoritma yang dimiliki pada *supervised learning* seperti *linear regression, random florest, back-propagation, support vector machine, naïve Bayesian, rocchio, decision tree, k-nearest neighbor, neural network, logistic regression,* dan *neural network*. Dalam Teknik ini dikelompokan menjadi dua masalah yaitu klasifikasi dan regresi. Masalah klasifikasi

adalah *variable output* berbentuk kategori atau pelabelan sedangkan masalah regeresi adalah ketika *variabel output* merupakan nilai real (Roihan *et al*., 2020).

2. Support Vector Machine

Pada tahun 1992 *Support Vector Machine* (SVM) dikembangkan oleh Benhard Boser, Isabelle Guyon dan Vapnik di *Annual Workshop on Computation Learning Theory*. *Support vector machine* (SVM) adalah sistem pembelajaran yang menggunakan ruang hipotesis sebagai fungsi linier dalam ruang fitur berdimensi tinggi, dilatih menggunakan algoritma pembelajaran berdasarkan teori optimasi dengan menerapkan bias pembelajaran yang berasal dari teori pembelajaran statistic (Cholissodin & Soebroto, 2021).

SVM adalah metode klasifikasi *supervised learning* yang konsep dasarnya merupakan kombinasi harmonis dari teori komputasi yang telah ada selama puluhan tahun, seperti margin *hyperplane* kernel yang diperkenalkan oleh Aronszajn pada tahun 1950, dan konsep pendukung lainnya. SVM membutuhkan data positif dan negatif untuk membuat keputusan terbaik saat memisahkan data positif dan negatif dalam ruang n-dimensi, ini disebut hyperplane. Penggunaan SVM banyak digunakan dalam permasalahan klasifikasi dan mampu menghasilkan performa yang baik seperti klasifikasi text, klasifikasi citra, klasifikasi biner dan lainnya.

Berdasarkan sifatnya, metode SVM dibagi menjadi dua bagian, yaitu SVM linear dan SVM nonlinear. SVM linear adalah dipemisahan kedua kelas berada di hyperplane dengan soft margin. Sedangkan SVM nonlinear diterapkan kernel trick terhadap ruang dimensi tinggi. SVM merupakan algoritma yang menggunakan pemetaan *nonlinear* mengubah data asli ke dimensi yang lebih tinggi, dan menjadi dimensi yang baru, kemudian mencari *linear* optimal pemisah *hyperplane.* SVM memiliki tujuan mencari *hyperplane* atau pemisah antara dua kelas pada

menggunakan fungsi linear secara matematis dapat didefinisikan pada persamaan 1 (Cholissodin & Soebroto, 2021).

$$f(x) = w.x_i + b \quad (1)$$

Di mana w merupakan bobot vector tegak lurus, dengan *hyperplane* yang dapat didefinisikan pada persamaan 2:

$$w = \sum_{i=1}^{n} a_i \, y_i x_i \quad (2)$$

Di mana:

Gambar 2.1 Ilustrasi SVM linear data

x^+ = kelas data positif

x^- = kelas data negatif

a_i = nilai bobot data ke-i

y_i = kelas data ke-i

x_i = data ke-i

Menghitung nilai bias menggunakan rumus persamaan 3.

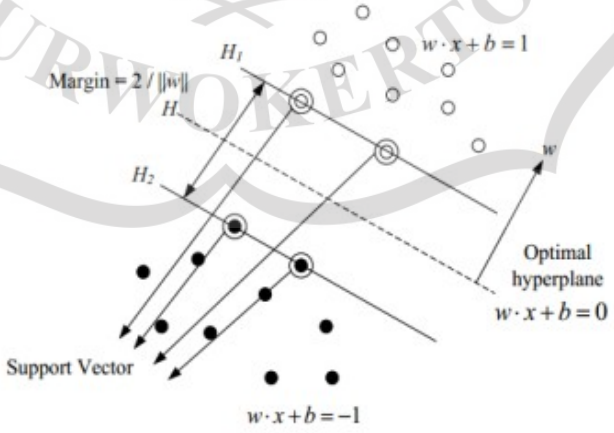
$$b = -\frac{1}{2}(w.x^+ + w.x^-) \quad (3)$$

Pengklasifikasian data pada kelas positif dan negatif seperti pada persamaan 4 dan 5.

sign(f(x)) = 1; kelas positif (4)

sign(f(x)) = -1; kelas negatif (5)

Ilustrasi *support vector machine* untuk *linear* data menurut (Cholissodin & Soebroto, 2021) ditunjukkan pada Gambar 2.1.



Dapat dilihat pada Gambar 2.1 ada dua dataset yang dipisahkan oleh *hyperplane* optimal, garis batas H1 adalah untuk kelas positif pada persamaan 1 dan H2 untuk kelas negatif dengan persamaan *hyperplane* w.x+b = -1, gunakan 2/||w|| untuk menghitung nilai margin (jarak) antara garis pemisah. Data terdekat *hyperplane* atau pada garis pemisah disebut *support vector*. Selanjutnya, tentukan hyperplane dari dua kelas maka margin perlu dimaksimalkan dengan menggunakan persamaan 6:

$$Minimize \int_{\mathrm{i}} [w] = \frac{1}{2}||w||^2 \quad (6)$$

Dengan syarat ditunjukkan pada persamaan 7:

$$yi(xi.w + b) - 1 \geq 0,\ \mathrm{i} = 1,2,3,\ldots,N \quad (7)$$

Beberapa kasus, beberapa data tidak dapat dipisahkan secara linear. Oleh karena itu perlu untuk menambah ξ_i (slack variable) untuk mengatasi situasi yang tidak layak. Sehingga secara matematis dapat dinyatakan pada persamaan 8:

$$Min\ \frac{1}{2}||w||^2 + C\ (\sum_{\mathrm{i}=1}^{n} (\mathrm{i}) \quad (8)$$

Dengan $yi(xi.w + b) - 1 + (\mathrm{i} \geq 0 \quad (9)$

$(\mathrm{i} \geq 0, \mathrm{i} = 1,2,3,\ldots,N$

Di mana:

ξ_i = slack variable (mengukur error dari data)

C = parameter bernilai positif (Batasan error)

Fungsi kernel yang digunakan pada SVM terdiri dari kernel linear dan *non* linear. Pada kernel linear digunakan ketika suatu data diklasifikasikan dapat terpisah dengan hyperplane. Sedangkan kernel *non* linear digunakan untuk data yang hanya bisa dipisahkan menggunakan garis lengkung atau sebuah bidang pada ruang dimensi tinggi. Pada penelitian ini menggunakan kernel *Gaussian Radial Basic Function* karena kernel RBF digunakan untuk klasifikasi data nonlinear dan memiliki performa yang baik dengan parameter tertentu.

3. Algoritma Genetika

Algoritma genetika atau *Genetic Algorithm* (GA) pertama kali dikenalkan oleh seorang professor University of Michigan, Amerika yang

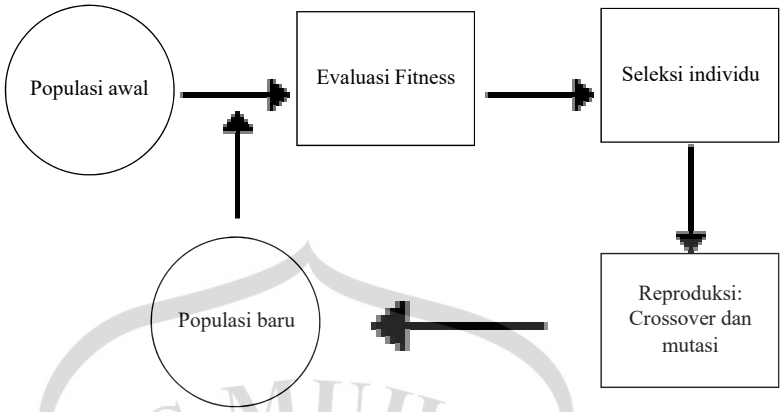
bernama John Holland pada tahun 1970. Dalam buku (Arkeman et al., 2018) algoritma genetika merupakan teknik pencarian dan teknik optimasi yang memiliki cara kerja meniru proses evolusi dan perubahan struktur gen pada makhluk hidup. GA menurut (Cholissodin & Soebroto, 2021) adalah Teknik identifikasi pendekatan solusi dalam permasalahan optimasi.

Optimasi GA menggunakan kriteria *fitness* untuk mendapatkan suatu solusi yang optimum. Dalam GA solusi optimum dihasilkan dalam proses seleksi, mutasi dan persilangan (crossover) dilakukan berulang-ulang. Untuk mendapatkan solusi terbaik dalam suatu permasalahan maka GA memanipulasi populasi struktur simbolis. Solusi yang dibangkitkan disebut kromosom, sedangkan kumpulan kromosom disebut populasi. Parameter terbaik yang di dapatkan pada proses SVM akan dilakukan optimasi GA sehingga akan meningkatkan akurasi klasifikasi pada SVM (Yenaeng *et al*., 2014).

Algoritma genetika memiliki beberapa kelebihan, yaitu :

1. Algoritma genetika hanya melakukan sedikit perhitungan matematis yang berhubungan dengan masalah yang akan diselesaikan.
2. Algoritma genetika menggunakan informasi fungsi tujuan, bukan informasi turunan dan lainnya.
3. Algoritma genetika menggunakan aturan perpindahan *probabilistik*, bukan *deterministik*.
4. Algoritma genetika bekerja dengan populasi titik, bukan satu titik.

   Siklus *algoritma genetika* menurut *David Goldberg* pada buku (Sutojo *et al*., 2011) dapat dilihat pada Gambar 2.2.



Gambar 2.2 Siklus Algoritma Genetika

Langkah-langkah *algoritma genetika* menurut (Hermawanto, 2013) sebagai berikut:

a. Inisialisasi Populasi Awal

Tahap pertama yaitu inisialisai populasi awal di mana tahapan ini akan membentuk N kromosom, dan nilai gen secara random.

b. Evaluasi Nilai *Fitness*

Tahap selanjutnya yaitu evaluasi nilai *fitness*. Tujuan dari algoritma genetika yaitu mencari nilai *fitness* maksimal (terbaik). Mencari nilai *fitness* dari setiap kromosom pada populasi pada penelitian ini menggunakan fungsi objektif, seperti pada persamaan 10.

$$f(x) = ((a + 2b + 3c + 4d + 5e) - 1) \quad (10)$$

c. Mencari Total Nilai *Fitness*

Mencari total nilai *fitness*, dapat dilihat pada persamaan 11.

$$total\ fitness = f[1] + f[2] + f[3] + f[4] \quad (11)$$

d. Mencari peluang untuk masing-masing kromosom

Dapat dilihat rumusnya pada persamaan 12.

$$P[\mathrm{i}] = Fitness[\mathrm{i}] \,/\, Total \quad (12)$$

e. Menghitung *cumulative probability*

Pada tahap ini nilai peluang akan dijumlahkan dengan peluang berikutnya, seperti pada persamaan 13.

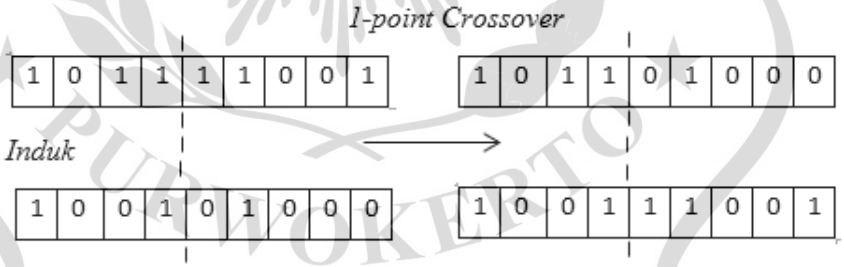
$$C[4] = P[1] + P[2] + P[3] + P[4] \quad (13)$$

f. Seleksi

Proses seleksi menggunakan roulete wheel, dengan membangkitkan nilai R secara random. Terdapat beberapa metode dalam seleksi yang dapat digunakan yaitu, *roulette wheel selection, tournament selection* dan *rank-based selection.* Metode yang sering digunakan yaitu, seleksi *roulette wheel selection.* Pada seleksi ini nilai R[i] dikategorikan ke dalam nilai cumulative dengan tujuan untuk mendapatkan *new* kromosom. Semakin besar nilainya maka semakin besar juga peluang kromosom tersebut untuk dipilih.

g. *Crossover*

Crossover (pindah silang) merupakan pertukaran antara kromosom sehingga akan membentuk kromosom baru dengan harapan lebih baik dari induknya. Proses ini semua kromosom mempunyai nilai gen secara random, nilai ≤ 0,25 akan dilakukan persilangan (Hermawanto, 2013). Teknik *crossover* ada 2 cara yaitu satu titik potong (*one point crossover*) dan dua titik potong (*n-point crossover*). Metode *Crossover* (pindah silang) yang sederhana yaitu metode pindah silang 1 titik. Dapat dilihat pada Gambar 2.3 (Prihastomo, 2018).



Gambar 2.3 Pindah Silang Satu Titik

h. Mutasi

Mutasi adalah proses penggantian gen dengan nilai terbaliknya. Gen 0 akan menjadi 1, dan gen 1 akan menjadi 0. Proses ini dilakukan secara acak pada lokasi gen tertentu pada individu yang dipilih untuk bermutasi. Sehingga akan menyebabkan terbentuknya kromosom baru (Arkeman *et al*., 2018).

i. Iterasi ke 2

Tahap ini akan menghitung nilai *fitness* kembali dengan fungsi objektif menggunakan kromosom baru yang telah di mutasi. Setelah tahap ini selesai maka akan dilakukan perbandingan kromosom pada iterasi 1 dan iterasi 2. Sehingga akan mendapatkan nilai *fitness* terbaik dan solusi terbaik.

j. Etilisme

Siklus algoritma genetika diperbaiki oleh Zbigniew Michalewicz menambahkan proses *etilisme* pada tahap akhir. Etilisme adalah teknik yang digunakan untuk pertahankan individu yang memiliki nilai *fitness* tertinggi agar tidak mengalami kerusakan karena proses genetik bagi individu untuk bertahan hidup untuk generasi berikutnya.

4. *Python*

Pyton merupakan bahasa pemrogram yang diciptakan oleh Guido van Rossum pada tahun 1991. Merujuk dari wikipedia, phyton merupakan bahasa pemrograman multiguna dengan perancangan yang berfokus pada tingkat keterbacaan kode. Salah satu fitur yang ada pada *python* adalah bahasa pemrograman dinamis dilengkapi oleh manajemen memori otomatis (Syahrudin & Kurniawan, 2018) . Salah satu notebook berbasis bahasa pemrograman python yaitu *google colaboratory*, sehingga pada penelitian ini menggunakan *google colaboratory* karena menyediakan library *machine learning* yang lebih banyak.