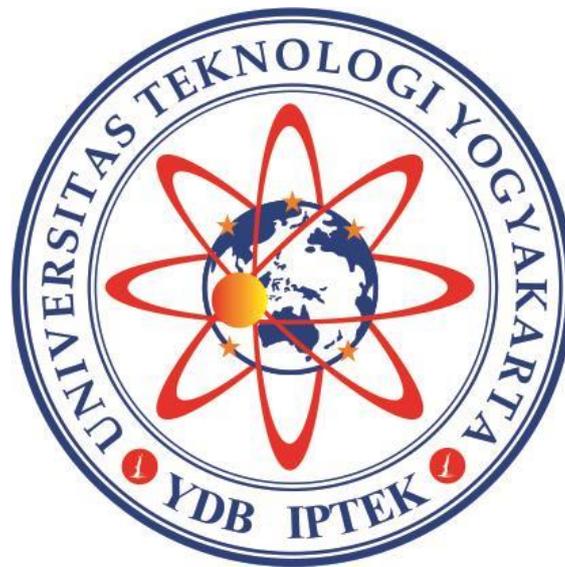


NASKAH PUBLIKASI

**SPKU: SISTEM PREDIKSI KUALITAS UDARA
(STUDI KASUS: DKI JAKARTA)**

Program Studi Teknik Informatika



Disusun oleh:

ADITYA HERMAWAN

5150411392

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN ELEKTRO
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2019**

NASKAH PUBLIKASI

PROYEK TUGAS AKHIR

**SPKU: SISTEM PREDIKSI KUALITAS UDARA
(STUDI KASUS: DKI JAKARTA)**

Program Studi Teknik Informatika
Fakultas Teknologi Informasi dan Elektro



Disusun oleh:

Aditya Hermawan

5150411392

Telah disetujui oleh pembimbing

Pembimbing,

A handwritten signature in black ink, appearing to be 'Enny Itje Sela', is written over the name of the supervisor.

Dr. Enny Itje Sela, S.Si., M.Kom

Tanggal: 201 Agustus 2019

SPKU: SISTEM PREDIKSI KUALITAS UDARA (STUDI KASUS: DKI JAKARTA)

Aditya Hermawan¹, Enny Itje Sela²

*Program Studi Teknik Informatika, Fakultas Bisnis dan Teknologi Informasi
Universitas Teknologi Yogyakarta*

Jl. Ringroad Utara Jombor Sleman Yogyakarta

E-mail : ¹okeaditya988@gmail.com, ²ennysela@uty.ac.id

ABSTRAK

Daerah Khusus Ibukota Jakarta adalah Ibu Kota Negara Republik Indonesia yang mempunyai jumlah penduduk yang semakin hari semakin meningkat yang tidak diimbangi dengan luas wilayah. Tingkat kepadatan kendaraan bermotor dan tingkat pembangunan infrastruktur semakin meningkat pula dengan hadirnya berbagai kendaraan dan pembangunan infrastruktur. Hal ini akan berakibat kepada tingkat pencemaran udara semakin tinggi. Oleh karena itu, dibutuhkan implementasi metode *support vector machine* (SVM) dengan kernel *radial basis function* (RBF) untuk memprediksi kualitas udara di Provinsi DKI Jakarta sehingga dapat memberikan peringatan dini kepada pemerintah dan masyarakat mengenai kualitas udara. Dengan menggunakan metode SVM kernel RBF, prediksi kualitas udara di DKI Jakarta berhasil dilakukan dengan tingkat akurasi prediksi pada semua data DKI Jakarta sebesar 96.03%, akurasi prediksi pada data Jakarta Pusat (DKI1) sebesar 93.97%, akurasi prediksi pada data Jakarta Utara (DKI2) sebesar 91.35%, akurasi prediksi pada data Jakarta Selatan (DKI3) sebesar 89.47%, akurasi prediksi pada data Jakarta Timur (DKI4) sebesar 89.27%, dan akurasi prediksi pada data Jakarta Barat sebesar 87.29% (DKI5). Hal ini dipengaruhi oleh pengaturan nilai parameter yang optimal yaitu nilai $cost = 3$, nilai $epsilon = 0.001$, nilai $sigma = 0.03$, nilai maksimal iterasi = 10, nilai toleransi = 0.0001 dan banyak nilai sama = 10.

Kata kunci: Prediksi, Kualitas Udara, DKI Jakarta, *Support Vector Machine*

1. PENDAHULUAN

Daerah Khusus Ibukota Jakarta adalah Ibu Kota Negara Republik Indonesia yang terdiri dari lima kota dan mempunyai jumlah penduduk yang semakin hari semakin meningkat yang tidak diimbangi dengan luas wilayah. Tingkat kepadatan kendaraan bermotor semakin meningkat pula dengan hadirnya berbagai kendaraan. Selain tingkat kepadatan kendaraan bermotor, tingkat pembangunan infrastruktur juga semakin bertambah. Hal ini akan berakibat kepada tingkat pencemaran udara semakin tinggi yang terdiri dari beberapa polutan yang dihasilkan dari pembuangan asap kendaraan bermotor, asap pabrik, dan polutan-polutan lainnya dari dampak pembangunan infrastruktur.

Berdasarkan situs pengamat kualitas udara aqicn.org tanggal 23 Juli 2019 pukul 17.00, DKI Jakarta menempati urutan 16 dari 100 negara teratas dengan polusi udara terburuk. Kualitas udara di DKI Jakarta mencapai level 155 *US Air Index Quality* (AIQ) yang mengindikasikan udara di DKI Jakarta tidak sehat. Kualitas udara dengan kategori tidak sehat dapat membuat setiap orang mengalami efek kesehatan, bahkan orang yang memiliki riwayat penyakit pernapasan dapat menyebabkan efek kesehatan yang lebih serius.

Penelitian sebelumnya yang dilakukan oleh [6] yang meneliti tentang penerapan metode klasifikasi *support vector machine* (SVM) pada data akreditasi Sekolah Dasar (SD) di Kabupaten Magelang menerapkan metode klasifikasi *support vector machine* (SVM) pada data akreditasi Sekolah Dasar (SD) di Kabupaten Magelang. Hasil yang didapat dari penelitian tersebut yaitu akurasi klasifikasi terbaik yaitu menggunakan *fungsi kernel Gaussian Radial Basic Function* (RBF) karena menghasilkan akurasi yang lebih besar.

Penelitian lainnya yang dilakukan oleh [3] yang meneliti tentang implementasi algoritme *support vector machine* (SVM) untuk penentuan seleksi atlet pencak silat membangun suatu sistem klasifikasi yang mampu mengklasifikasi penerimaan seleksi atlet pencak silat yang layak lolos. Hasil rata-rata akurasi menggunakan metode SVM pada klasifikasi penerimaan seleksi atlet pencak silat sebesar 69.09%. Sedangkan penelitian [11] yang meneliti tentang pengenalan jenis penyakit THT menggunakan jaringan LVQ mampu untuk mendiagnosa penyakit secara optimal dengan tingkat kebenaran pengujian pelatihan 94%.

Penelitian [2] yang meneliti tentang prediksi kadar polutan menggunakan jaringan syaraf tiruan

(JST) untuk pemantauan kualitas udara di Kota Surabaya membahas prediksi terhadap konsentrasi Ozon (O₃) dengan menggunakan metode jaringan syaraf tiruan (JST). Penelitian [1] yang meneliti tentang penerapan metode *adaptive neuro fuzzy inference system* (ANFIS) untuk memprediksi konsentrasi polutan udara di Provinsi Daerah Istimewa Yogyakarta membahas pembuatan model prediksi konsentrasi polutan udara di Provinsi Daerah Istimewa Yogyakarta menggunakan metode *adaptive neuro fuzzy inference system* (ANFIS).

Penelitian [14] yang meneliti tentang penerapan algoritme *support vector machine* terhadap klasifikasi tingkat risiko pasien gagal ginjal membangun sebuah sistem cerdas yang dapat melakukan klasifikasi tingkat risiko pasien gagal ginjal dengan menggunakan fitur dan masukan sesuai dengan penelitian di Laboratorium Klinik Sejahtera kota Probolinggo.

Berdasarkan uraian di atas, penelitian ini menggunakan metode *support vector machine* (SVM) dengan *kernel radial basis function* (RBF) untuk memprediksi kualitas udara di Provinsi DKI Jakarta yang diimplementasikan pada sistem prediksi kualitas udara (SPKU) DKI Jakarta.

2. LANDASAN TEORI

2.1. Kualitas Udara

Kualitas udara adalah ukuran atau tingkat baik buruknya suatu campuran gas yang terdapat pada lapisan troposfer yang dibutuhkan dan mempengaruhi kesehatan manusia, makhluk hidup, dan unsur lingkungan hidup lainnya yang komposisinya tidak selalu konstan [13].

2.2. Pencemaran Udara

Pencemaran udara adalah masuknya atau dimasukkannya zat, energi dan/atau komponen lain ke dalam udara ambien oleh kegiatan manusia, sehingga mutu udara ambien turun sampai ke tingkat tertentu yang menyebabkan udara ambien tidak dapat memenuhi fungsinya [7].

Ada berbagai macam jenis zat pencemar udara terhadap penurunan kualitas udara seperti gas pencemar yang terdiri dari:

- Nitrogen Dioksida (NO₂)
- Karbon Monoksida (CO)
- Sulfur Dioksida (SO₂)
- Ozon (O₃)
- Partikulat Debu (PM10)

2.3. Indeks Standar Pencemaran Udara (ISPU)

Indeks standar pencemar udara adalah angka yang tidak mempunyai satuan yang menggambarkan kondisi kualitas udara ambien di lokasi dan waktu tertentu yang didasarkan kepada dampak terhadap kesehatan manusia, nilai estetika dan makhluk hidup lainnya. Indeks standar pencemar udara ditetapkan dengan cara mengubah kadar pencemar udara yang

terukur menjadi suatu angka yang tidak berdimensi [12].

2.4. Support Vector Machine (SVM)

Support vector machine (SVM) dikembangkan pertama kali oleh Boser, Guyon, dan Vapnik. *Support vector machine* (SVM) memiliki prinsip dasar yaitu *linear classifier*, dan selanjutnya dikembangkan agar dapat bekerja pada masalah seperti *nonlinear* dengan memasukkan konsep *kernel trick* pada ruang kerja berdimensi tinggi. SVM sendiri merupakan usaha untuk mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas pada *input space* dengan cara menghitung nilai *margin* yang mana merupakan jarak antara *hyperplane* dengan pola terdekat dari setiap kelas, yang mana pola itu disebut sebagai *support vector* [4].

2.5. Kernel Trick

Kernel trick merupakan metode yang menggunakan algoritme *linier classifier* untuk menyelesaikan permasalahan *nonlinier* dengan cara memetakan dimensi *input* ke ruang dimensi yang lebih tinggi, sehingga membuat *linier classifier* di ruang dimensi yang baru setara dengan *nonlinear classifier* di ruang dimensi asli. Dengan *kernel*, fungsi pemetaan tidak pernah dihitung secara eksplisit, karena ruang dimensi tinggi yang digunakan memungkinkan pada dimensi yang tak terbatas [9].

Fungsi kernel yang umum digunakan adalah sebagai berikut [10]:

- Kernel Linear* ditunjukkan pada Persamaan 1.

$$K(x_i, x) = x_i^t x \quad (1)$$

- Polynomial Linear* ditunjukkan pada Persamaan 2.

$$K(x_i, x) = (\gamma \cdot x_i^t x + r)^p, \gamma > 0 \quad (2)$$

- Gaussian Radial Basis Function* ditunjukkan pada Persamaan 3.

$$K(x_i, x) = \exp\left(-\frac{\|x_i - x\|^2}{2\sigma^2}\right) \quad (3)$$

- Sigmoid Function* ditunjukkan pada Persamaan 4.

$$K(x_i, x) = \tanh(\sigma(x_i, x) + C) \quad (4)$$

dengan $K(x_i, x)$ sebagai fungsi *kernel*, x dan x_i sebagai pasangan dua data, t sebagai banyak data, dan C, p, σ sebagai nilai konstanta.

2.6. One Against All (OAA)

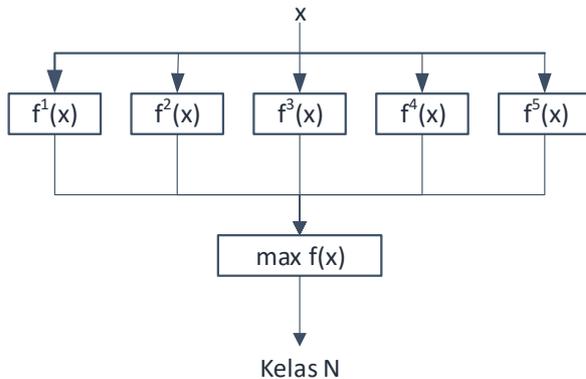
Metode OAA untuk kasus klasifikasi k-kelas, menemukan k *hyperplane* dimana k adalah banyak kelas dan p adalah *hyperplane*. Dalam metode ini $\rho(\ell)$ diujikan dengan semua data dari kelas ℓ dengan label +1, dan semua data dari kelas lain dengan label -1 [6].

Berikut ini merupakan ilustrasi untuk persoalan klasifikasi dengan lima buah jumlah kelas,

digunakan lima buah SVM *biner* pada Tabel 1 dan penggunaannya pada pengklasifikasian data baru pada Gambar 1 [5].

Tabel 1 Contoh Metode One Against All

$y_i=1$	$y_i=-1$	Hipotesis
Kelas 1	Bukan Kelas 1	$f^1(x) = (w^1)x + b^1$
Kelas 2	Bukan Kelas 2	$f^2(x) = (w^2)x + b^2$
Kelas 3	Bukan Kelas 3	$f^3(x) = (w^3)x + b^3$
Kelas 4	Bukan Kelas 4	$f^4(x) = (w^4)x + b^4$
Kelas 5	Bukan Kelas 5	$f^5(x) = (w^5)x + b^5$



Gambar 1 Metode Klasifikasi SVM One Against All

Berdasarkan Tabel 2.3 dan Gambar 2.1, konsep pada OAA yaitu dimisalkan pada kasus lima kelas, kelas 1, 2, 3, 4 dan 5. Bila akan diujikan $\rho(1)$, semua data dalam kelas 1 diberi label +1 dan data dari kelas 2, 3, 4 dan 5 diberi label -1. Pada $\rho(2)$, semua data dalam kelas 2 diberi label +1 dan data dari kelas 1, 3, 4 dan 5 diberi label -1. Pada $\rho(3)$, semua data dalam kelas 3 diberi label +1 dan data dari kelas 1, 2, 4 dan 5 diberi label -1. Pada $\rho(4)$, semua data dalam kelas 4 diberi label +1 dan data dari kelas 1, 2, 3 dan 5 diberi label -1. Begitu juga untuk $\rho(5)$, semua data dalam kelas 5 diberi label +1 dan data dari kelas 1, 2, 3 dan 4 diberi label -1. Kemudian dicari *hyperplane* dengan algoritme SVM dua kelas. Maka akan didapat *hyperplane* untuk masing-masing kelas di atas. Kemudian kelas dari suatu data baru x ditentukan berdasarkan nilai terbesar dari *hyperplane* melalui Persamaan 5 berikut [6]:

$$\text{kelas } x = \arg \max_{\ell=1..k} ((w^{(\ell)})^T \cdot \varphi(x) + b^{(\ell)}) \quad (5)$$

dimana x adalah data, ℓ menunjukkan kelas, w sebagai bidang normal pemisah antar bidang pemisah terhadap pusat koordinat, b sebagai posisi bidang relatif terhadap pusat koordinat, dan φ sebagai fungsi transformasi.

2.7. Sequential Minimal Optimization (SMO)

Sequential Minimal Optimization (SMO) adalah algoritme untuk proses pelatihan SVM yang dapat memberikan solusi pada masalah optimasi. Pada dasarnya penggunaan SVM hanya terbatas pada masalah yang kecil karena algoritme pelatihan SVM cenderung lambat, kompleks, dan sulit untuk diimplementasikan. Algoritme SMO lebih sederhana, lebih mudah diimplementasikan dan lebih cepat waktu komputasinya dari pada algoritme *Chunking* [8].

SMO memilih menyelesaikan masalah optimasi pada Persamaan 6 seminimal mungkin untuk setiap tahapnya. Pada setiap tahap, SMO memilih dua *lagrange multiplier* a_i , untuk dioptimasi bersama-sama, mencari nilai yang paling optimal untuk *lagrange multiplier* tersebut, dan memperbaharui SVM dengan nilai optimal yang baru.

$$\text{Maksimal} \quad \sum_{i=1}^n a_i - \frac{1}{2} \sum_{j=i}^n a_i a_j y_i y_j x_i x_j \quad (6)$$

dimana $a_i \geq 0 (i = 1, 2, \dots, n)$ dan $\sum_{i,j=1}^n a_i y_i = 0$

Algoritma SMO yang telah disederhanakan adalah sebagai berikut:

- Masukkan data latih $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, nilai parameter C , nilai *sigma*, nilai *epsilon*, maksimal iterasi, nilai toleransi, dan nilai maksimal *pass*.
- Inisialisasi nilai awal untuk *lagrange multiplier* α dan bias b serta iterasi dan nilai *pass* = 0.
- Lakukan iterasi selama nilai *pass* < maksimal *pass* dan iterasi < maksimal iterasi.
- Inisialisasi perubahan $\alpha = 0$.
- Lakukan perulangan sebanyak jumlah data latih.
- Hitung $E_i = f(x^{(i)}) - y_i$, dengan $f(x^{(i)})$ menggunakan Persamaan 7.

$$f(x_d) = \sum_{i=1}^m a_i y_i \varphi(x_i) \varphi(x_d) + b \quad (7)$$

- Jika $(y^{(i)} E_i < -tol \ \&\& \ \alpha_i < C) \ || \ (y^{(i)} E_i > tol \ \&\& \ \alpha_i > 0)$, lakukan:
 - Buat nilai j baru yang tidak sama dengan nilai i
 - Hitung $E_j = f(x^{(j)}) - y_j$, dengan $f(x^{(j)})$ menggunakan Persamaan 7
 - Simpan nilai α lama yaitu $\alpha_i^{lama} =$

- $\alpha_i, \alpha_j^{lama} = \alpha_j$
- Jika $y^{(i)} \neq y^{(j)}$, hitung nilai H dan L dengan Persamaan 8

$$L = \max(0, \alpha_j - \alpha_i),$$

$$H = \min(C, C + \alpha_j - \alpha_i)$$
 (8)
 - Jika $y^{(i)} = y^{(j)}$, hitung nilai H dan L dengan Persamaan 9

$$L = \max(0, \alpha_i + \alpha_j - C),$$

$$H = \min(C, \alpha_i + \alpha_j)$$
 (9)
 - Jika $L=H$, maka ulangi langkah 3 dengan $i=i+1$
 - Menghitung nilai eta (η) dengan Persamaan 10

$$\eta = 2\langle x^{(i)}, x^{(j)} \rangle - \langle x^{(i)}, x^{(i)} \rangle - \langle x^{(j)}, x^{(j)} \rangle$$
 (10)
 - Jika $\eta > 0$, maka ulangi langkah 3 dengan $i=i+1$
 - Menghitung nilai α_j baru dengan Persamaan 11 dan 12

$$\alpha_j = \alpha_j - \frac{y^{(j)}(E_i - E_j)}{\eta}$$
 (11)

$$\alpha_j = \begin{cases} H, & \text{jika } \alpha_j > H \\ \alpha_j, & \text{jika } L \leq \alpha_j \leq H \\ L, & \text{jika } \alpha_j < L \end{cases}$$
 (12)
 - Jika $|\alpha_j - \alpha_j^{lama}| < \epsilon$, maka ulangi langkah 3 dengan $i=i+1$
 - Menghitung nilai α_i baru dengan Persamaan 13

$$\alpha_i = \alpha_i + y^{(i)}y^{(j)}(\alpha_j^{lama} - \alpha_j)$$
 (13)
 - Hitung b_1 dan b_2 menggunakan Persamaan 14 dan Persamaan 15

$$b_1 = b - E_i - y^{(i)}(\alpha_i - \alpha_i^{lama})\langle x^{(i)}, x^{(i)} \rangle - y^{(j)}(\alpha_j - \alpha_j^{lama})\langle x^{(i)}, x^{(j)} \rangle$$
 (14)

$$b_2 = b - E_j - y^{(j)}(\alpha_j - \alpha_j^{lama})\langle x^{(i)}, x^{(j)} \rangle - y^{(i)}(\alpha_i - \alpha_i^{lama})\langle x^{(i)}, x^{(i)} \rangle$$
 (15)
 - Hitung b menggunakan Persamaan 16

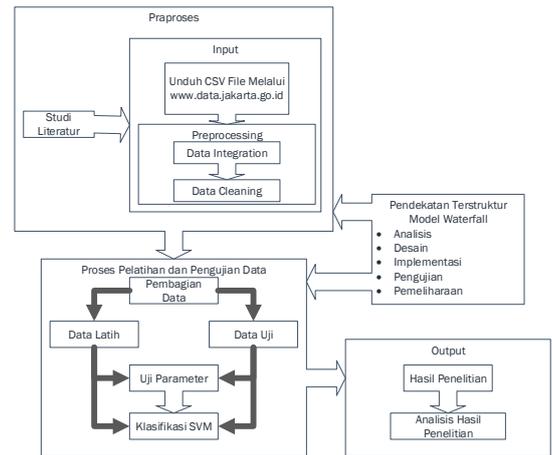
$$b = \begin{cases} b_1, & \text{jika } 0 < \alpha_i > C \\ b_2, & \text{jika } 0 < \alpha_j > C \\ \frac{(b_1+b_2)}{2}, & \text{jika tidak} \end{cases}$$
 (16)
 - Perubahan $\alpha = \alpha + \eta$
 - Jika ($\alpha = 0$) maka $pass=pass+1$, jika tidak $pass=0$
 - Ulangi langkah 2 sampai kondisi terpenuhi
 - Selesai

3. METODOLOGI PENELITIAN

3.1. Desain Penelitian

Desain penelitian ini akan menggambarkan proses implementasi *support vector machine* (SVM) dengan *kernel radial basis function* (RBF) untuk

memprediksi kualitas udara di Provinsi DKI Jakarta yang ditunjukkan pada Gambar 2 berikut:



Gambar 2 Desain Penelitian Sistem Prediksi Kualitas Udara (SPKU) DKI Jakarta

Penjelasan desain penelitian dari Gambar 2 sebagai berikut:

- Studi Literatur

Studi literatur merupakan tahap awal dalam pembuatan penelitian ini dengan cara mengumpulkan data berupa buku, artikel, jurnal, ataupun bahan lainnya yang menunjang dalam penelitian. Selain itu memahami juga teori tentang kualitas udara dan *support vector machine* (SVM).
- Data Penelitian
 - Menentukan sampel data yang akan digunakan dalam penelitian yaitu data ISPU DKI Jakarta tahun 2017 dan tahun 2018.
 - Mengumpulkan data ISPU DKI Jakarta Tahun 2017 dengan cara mengambil dari situs sumber terbuka DKI Jakarta yaitu www.data.jakarta.go.id dengan keterangan data yang ditunjukkan pada Tabel 2.
 - Mengumpulkan data ISPU DKI Jakarta Tahun 2018 dengan cara mengambil dari situs sumber terbuka DKI Jakarta yaitu www.data.jakarta.go.id dengan keterangan data yang ditunjukkan pada Tabel 3.

Tabel 2 Data Indeks Standar Pencemaran Udara DKI Jakarta Tahun 2017

Lokasi	Jumlah Data	
	Terisi	Kosong
Jakarta Pusat (DKI1)	353	12

Jakarta Utara (DKI2)	361	4
Jakarta Selatan (DKI3)	360	5
Jakarta Timur (DKI4)	362	3
Jakarta Barat (DKI5)	365	0
Total	1801	24

Tabel 3 Data Indeks Standar Pencemaran Udara DKI Jakarta Tahun 2018

Lokasi	Jumlah Data	
	Terisi	Kosong
Jakarta Pusat (DKI1)	353	12
Jakarta Utara (DKI2)	361	4
Jakarta Selatan (DKI3)	351	14
Jakarta Timur (DKI4)	360	5
Jakarta Barat (DKI5)	363	2
Total	1788	37

Tabel 4 Atribut Data Indeks Standar Pencemaran Udara DKI Jakarta

No	Atribut	Proses	Keterangan
1	Tanggal	Atribut Digunakan	Tanggal pengambilan data
2	Stasiun	Data Cleaning	Lokasi pengambilan data
3	PM10	Atribut Digunakan	Partikulat salah satu parameter yang diukur
4	SO ₂	Atribut Digunakan	Sulfida (dalam bentuk SO ₂)
5	CO	Atribut Digunakan	Carbon Monoksida
6	O ₃	Atribut Digunakan	Ozon
7	NO ₂	Atribut Digunakan	Nitrogen dioksida
8	Max	Atribut Tidak Digunakan	Nilai ukur paling tinggi dari seluruh parameter yang diukur dalam waktu yang sama
9	Critical	Atribut Tidak Digunakan	Parameter yang hasil pengukurannya paling tinggi
10	Categori	Data Cleaning	Kategori hasil perhitungan indeks standar pencemaran udara

Pada Tabel 4, atribut tanggal digunakan dan tidak terjadi perubahan, atribut stasiun dilakukan tahap *data cleaning* karena terjadi inkonsisten terhadap pengisian data ISPU tahun 2017 dan tahun 2018 yaitu penulisan nama lokasi, atribut PM10, SO₂, CO, O₃, dan NO₂ digunakan dan tidak terjadi perubahan, atribut *max* dan *critical* tidak digunakan karena tidak mempengaruhi pelatihan, dan atribut

categori dilakukan tahap *data cleaning* karena terjadi inkonsisten terhadap pengisian data ISPU tahun 2017 dan tahun 2018 yaitu penulisan nama kategori ada yang menggunakan huruf kapital dan ada yang menggunakan huruf kecil.

3. Pengembangan Perangkat Lunak

- Data Integration
- Data Cleaning
- Data Transformation

Tahap selanjutnya yaitu proses pelatihan dan pengujian data yang meliputi:

a. Pembagian Data

Tahapan ini akan membagi semua data ISPU menjadi dua bagian, yaitu data latih sebanyak 60% dan data uji sebanyak 40% yang ditunjukkan pada Tabel sebagai berikut:

Tabel 5 Data Pelatihan dan Pengujian

No	Lokasi	Jumlah Data Pelatihan	Jumlah Data Pengujian	Total
1	Seluruh DKI Jakarta	2153	1436	3589
2	Jakarta Pusat (DKI1)	424	282	706
3	Jakarta Utara (DKI2)	433	289	722
4	Jakarta Selatan (DKI3)	426	285	711
5	Jakarta Timur (DKI4)	433	289	722
6	Jakarta Barat (DKI5)	437	291	728

- Unggah *file* CSV "Data ISPU - Latih" ke sistem melalui fitur *upload* data pelatihan.
- Unggah *file* CSV "Data ISPU - Uji" ke sistem melalui fitur *upload* data prediksi.
- Pengujian Parameter SVM

Tahapan ini dilakukan untuk mencari parameter-parameter *support vector machine* (SVM) untuk dilakukan pelatihan data sehingga akan menghasilkan waktu komputasi yang optimal dan akurasi yang terbaik. Pengujian dilakukan terhadap parameter C (*cost*), *epsilon*, *sigma*, maksimal iterasi, toleransi dan banyak nilai yang sama. Data yang digunakan yaitu data semua DKI Jakarta.

e. Pengujian Support Vector Machine (SVM)

Tahapan ini merupakan proses yang sangat penting karena merupakan inti dari penelitian ini. Tahapan ini bertujuan untuk menghasilkan

pola keluaran (nilai α dan bias) yang sesuai dengan harapan.

4. Output

Tahapan ini bertujuan untuk menghasilkan hasil dari penelitian yang terdiri dari dua tahapan yaitu:

- a. Hasil Penelitian
- b. Analisis Hasil Penelitian
 - a) Waktu *Training* dan *Testing*
 - b) Tingkat Akurasi
 - c) Jumlah Data, Atribut, dan Inisialisasi Awal *Training*

3.2. Objek Penelitian

Objek penelitian yang diambil dalam penelitian ini adalah data indeks standar pencemar udara (ISPU) dengan paramater indeks berupa diameter debu (PM10), sulfur dioksida (SO₂), karbon monoksida (CO), ozon (O₃), dan nitrogen dioksida (NO₂) yang didapatkan dari Portal Data Terpadu Pemerintah Provinsi DKI Jakarta yang menyajikan data-data dari seluruh satuan dan unit kerja di Pemerintah Provinsi DKI Jakarta melalui situs www.data.jakarta.go.id.

Data tersebut diambil dari lima lokasi berbeda selama dua tahun (2017-2018) yaitu di Jakarta Pusat (DKI1), Jakarta Utara (DKI2), Jakarta Selatan (DKI3), Jakarta Timur (DKI4), dan Jakarta Barat (DKI5).

4. HASIL DAN PEMBAHASAN

4.1. Pengujian Parameter

Pengujian parameter merupakan pengujian yang dapat menunjukkan seberapa jauh pengaruh parameter-parameter yang digunakan terhadap akurasi dan lama waktu yang dibutuhkan. Pengujian parameter yang dilakukan menggunakan data kualitas udara dari semua stasiun DKI Jakarta.

a. Uji Parameter C (*Cost*)

Parameter *cost* digunakan dalam proses pelatihan yang memberikan batas α . Berdasarkan hasil pengujian pada Tabel 6 dengan menggunakan nilai $\epsilon = 0.0001$, nilai $\sigma = 0.02$, nilai iterasi maksimal = 10, nilai toleransi = 0.0001, dan banyak nilai yang sama = 10, nilai *cost* dapat memberikan pengaruh terhadap komputasinya, jika nilai *cost* semakin besar maka akan memberikan waktu komputasi yang lama dalam pemrosesan pelatihan data, tetapi memberikan akurasi yang kecil jika nilai *cost* terlalu kecil.

Tabel 6 Uji Parameter C (*Cost*)

Uji Ke	<i>Cost</i>	Waktu <i>Training</i>	Waktu <i>Testing</i>	Akurasi <i>Training</i>	Akurasi <i>Testing</i>
1	0.001	38 Menit 37 Detik	4 Menit 3 Detik	79.87%	58.91%

2	0.01	39 Menit 29 Detik	4 Menit 13 Detik	80.55%	62.19%
3	0.1	39 Menit 4 Detik	3 Menit 53 Detik	94.62%	88.09%
4	0.5	35 Menit 22 Detik	3 Menit 58 Detik	97.21%	92.55%
5	1	37 Menit 34 Detik	4 Menit 5 Detik	98.15%	93.87%
6	1.5	36 Menit 52 Detik	3 Menit 47 Detik	97.91%	93.87%
7	2	37 Menit 51 Detik	4 Menit 7 Detik	98.34%	93.87%
8	3	42 Menit 8 Detik	4 Menit 31 Detik	98.49%	94.29%
9	5	37 Menit 52 Detik	4 Menit 36 Detik	96.27%	92.27%
10	10	47 Menit 25 Detik	4 Menit 6 Detik	97.11%	92.97%

b. Uji Parameter ϵ

Parameter ϵ digunakan dalam membatasi kesalahan / *error* pada klasifikasi. Berdasarkan hasil pengujian pada Tabel 7 dengan menggunakan nilai $\text{cost} = 3$, nilai $\sigma = 0.02$, nilai iterasi maksimal = 10, nilai toleransi = 0.0001, dan banyak nilai yang sama = 10, nilai ϵ dapat memberikan pengaruh terhadap komputasinya, jika nilai ϵ semakin besar maka akan memberikan akurasi yang kecil terhadap data *training*. Bahkan nilai ϵ yang tidak mendekati nilai 0 memiliki hasil akurasi yang konstan yaitu hanya mampu memprediksi secara benar 11 data *testing*.

Tabel 7 Uji Parameter ϵ

Uji Ke	ϵ	Waktu <i>Training</i>	Waktu <i>Testing</i>	Akurasi <i>Training</i>	Akurasi <i>Testing</i>
1	0.0001	42 Menit 8 Detik	4 Menit 31 Detik	98.49%	94.29%
2	0.001	37 Menit 43 Detik	4 Menit 49 Detik	98.34%	94.57%
3	0.01	44 Menit 18 Detik	3 Menit 49 Detik	97.84%	92.48%
4	0.05	36 Menit 58 Detik	3 Menit 50 Detik	96.60%	93.87%
5	0.1	41 Menit 36 Detik	4 Menit 3 Detik	97.12%	92.83%
6	0.5	38 Menit 35 Detik	4 Menit 22 Detik	98.41%	94.15%
7	1	45 Menit 21 Detik	3 Menit 42 Detik	92.66%	94.50%
8	5	50 Menit 41 Detik	3 Menit 45 Detik	75%	0.76%
9	10	47 Menit 24 Detik	4 Menit 22 Detik	75%	0.76%

c. Uji Parameter σ

Parameter σ digunakan pada *kernel* RBF untuk menentukan nilai optimal dalam setiap *dataset*. Berdasarkan hasil pengujian pada Tabel 8 dengan menggunakan nilai $\text{cost} = 3$, nilai $\epsilon = 0.001$, nilai iterasi maksimal = 10, nilai toleransi = 0.0001, dan banyak nilai yang sama = 10, nilai ϵ dapat memberikan pengaruh terhadap komputasinya, jika nilai σ semakin kecil maka akan memberikan waktu komputasi yang cepat

dalam pemrosesan pelatihan data dengan akurasi pelatihan semakin tinggi tetapi membuat akurasi prediksi semakin rendah. Akan tetapi sigma yang terlalu kecil dapat memberikan akurasi yang rendah terhadap data *training* dan data *testing*.

Tabel 8 Uji Parameter Sigma

Uji Ke	Sigma	Waktu Training	Waktu Testing	Akurasi Training	Akurasi Testing
1	0.001	45 Menit 18 Detik	3 Menit 51 Detik	99.97%	74.09%
2	0.01	40 Menit 38 Detik	3 Menit 53 Detik	99.29%	94.01%
3	0.02	37 Menit 43 Detik	4 Menit 49 Detik	98.34%	94.57%
4	0.03	38 Menit 2 Detik	4 Menit 4 Detik	97.04%	96.03%
5	0.1	37 Menit 13 Detik	4 Menit 4 Detik	93.11%	91.92%
6	0.5	33 Menit 58 Detik	3 Menit 48 Detik	93.46%	89.26%
7	1	37 Menit 43 Detik	4 Menit 2 Detik	91.95%	87.53%
8	2	35 Menit 48 Detik	3 Menit 48 Detik	90.85%	85.86%
9	5	36 Menit 42 Detik	4 Menit 24 Detik	89.14%	84.19 %
10	10	41 Menit 14 Detik	3 Menit 51 Detik	87.03%	77.85%

d. Uji Parameter Maksimal Iterasi

Parameter maksimal iterasi digunakan untuk memberikan batasan iterasi yang dapat dilakukan selama pelatihan data. Berdasarkan hasil pengujian pada Tabel 9 dan Tabel 10 dengan menggunakan nilai $cost = 3$, nilai $epsilon = 0.001$, nilai $sigma = 0.03$, nilai toleransi = 0.0001, dan banyak nilai yang sama = 10, nilai maksimal iterasi dapat memberikan pengaruh besar terhadap waktu komputasi dan akurasi. Jika nilai maksimal iterasi sangat besar maka waktu komputasinya sangat lama, bahkan jika data yang digunakan sangat besar menyebabkan pemrosesan pelatihan data *error*.

Tabel 9 Uji Parameter Maksimal Iterasi

Uji Ke	Max Iterasi	Waktu Training	Waktu Testing	Akurasi Training	Akurasi Testing
1	1	16 Menit 26 Detik	4 Menit 3 Detik	93.74 %	90.46 %
2	5	26 Menit 5 Detik	3 Menit 57 Detik	96.28 %	91.92 %
3	10	38 Menit 2 Detik	4 Menit 4 Detik	97.04 %	96.03 %
4	15	50 Menit 1 Detik	4 Menit 30 Detik	97.94 %	95.20 %
5	20	60 Menit 10 Detik	3 Menit 43 Detik	98.78 %	95.26 %

Tabel 10 Lanjutan Uji Parameter Maksimal Iterasi

6	50	129 Menit 58 Detik	3 Menit 53 Detik	98.79 %	95.68 %
7	100	253 Menit 0 Detik	4 Menit 4 Detik	98.91 %	95.40 %

8	1000	Error	Error	Error	Error
9	5000	Error	Error	Error	Error
10	10000	Error	Error	Error	Error

e. Uji Parameter Toleransi

Parameter toleransi nilai digunakan untuk pengaturan kecepatan pelatihan data. Berdasarkan hasil pengujian pada Tabel 11 dengan menggunakan nilai $cost = 3$, nilai $epsilon = 0.001$, nilai $sigma = 0.03$, nilai maksimal iterasi = 10, dan banyak nilai yang sama = 10, nilai toleransi dapat memberikan pengaruh terhadap waktu komputasinya, jika nilai toleransi semakin besar maka waktu pelatihan yang dibutuhkan semakin cepat tetapi akan membuat akurasi prediksi menjadi lebih rendah.

Tabel 11 Uji Parameter Toleransi

Uji Ke	Tol	Waktu Training	Waktu Testing	Akurasi Training	Akurasi Testing
1	0.0001	38 Menit 2 Detik	4 Menit 4 Detik	97.04%	96.03%
2	0.0005	37 Menit 4 Detik	3 Menit 49 Detik	97.60%	94.29%
3	0.001	37 Menit 12 Detik	3 Menit 58 Detik	96.79%	92.34%
4	0.005	38 Menit 43 Detik	4 Menit 10 Detik	95.68%	92.55%
5	0.01	39 Menit 12 Detik	4 Menit 3 Detik	98.05%	93.94%
6	0.05	37 Menit 39 Detik	3 Menit 50 Detik	97.03%	94.01%
7	0.1	37 Menit 45 Detik	3 Menit 48 Detik	97.09%	94.57%
8	1	32 Menit 43 Detik	3 Menit 54 Detik	75%	0.77%
9	3	32 Menit 31 Detik	3 Menit 55 Detik	75%	0.77%
10	5	34 Menit 14 Detik	3 Menit 57 Detik	75%	0.77%
11	10	33 Menit 59 Detik	3 Menit 57 Detik	75%	0.77%

f. Uji Parameter Banyak Nilai Sama

Parameter banyak nilai digunakan untuk memberikan batasan banyaknya data yang sama tanpa perubahan. Berdasarkan hasil pengujian pada Tabel 12 dengan menggunakan nilai $cost = 3$, nilai $epsilon = 0.001$, nilai $sigma = 0.03$, nilai toleransi = 0.0001, dan maksimal iterasi = 10, nilai maksimal iterasi dapat memberikan sedikit pengaruh terhadap waktu komputasinya, jika banyak nilai sama semakin besar maka waktu pelatihan yang dibutuhkan semakin lama dan sedikit mempengaruhi akurasi prediksi.

Tabel 12 Uji Parameter Banyak Nilai Sama

Uji Ke	Banyak Nilai	Waktu Training	Waktu Testing	Akurasi Training	Akurasi Testing
1	15	38 Menit 42 Detik	4 Menit 1 Detik	96.15%	92.27%

2	11	39 Menit 8 Detik	3 Menit 52 Detik	97.61%	93.66%
3	10	38 Menit 2 Detik	4 Menit 4 Detik	97.04%	96.03%
4	9	38 Menit 3 Detik	3 Menit 53 Detik	97.25%	92.97%
5	5	39 Menit 24 Detik	4 Menit 6 Detik	98.35%	94.71%
6	4	38 Menit 7 Detik	3 Menit 58 Detik	97.81%	93.87%
7	3	39 Menit 1 Detik	4 Menit 8 Detik	98.26%	94.50%
8	2	37 Menit 57 Detik	3 Menit 38 Detik	97.98%	93.87%
9	1	37 Menit 57 Detik	3 Menit 55 Detik	98.11%	94.22%

Sehingga nilai parameter yang optimal didapatkan berdasarkan uji parameter di atas yaitu nilai $cost = 3$, nilai $epsilon = 0.001$, nilai $sigma = 0.03$, nilai maksimal iterasi = 10, nilai toleransi = 0.0001 dan banyak nilai sama = 10.

4.2. Hasil Pengujian Data

Pengujian data dilakukan untuk mengetahui hasil dari penelitian implementasi *support vector machine* (SVM) untuk memprediksi kualitas udara di Provinsi DKI Jakarta yang dibuat berhasil atau tidak berhasil. Pengujian dilakukan sebanyak satu kali di setiap lokasi ISPU. Parameter yang digunakan dalam pelatihan data sebanyak enam parameter yang terdiri dari nilai $cost = 3$, nilai $epsilon = 0.001$, nilai $sigma = 0.03$, nilai maksimal iterasi = 10, nilai toleransi = 0.0001 dan banyak nilai sama = 10.

Pengujian pertama dilakukan menggunakan data ISPU semua DKI Jakarta dengan hasil akurasi *training* sebesar 97.04%, akurasi *testing* sebesar 96.03%, lama *training* sebesar 38 menit 2 detik, dan lama *testing* sebesar 4 menit 4 detik serta hasil *confusion matrix* yang ditunjukkan pada Tabel 13.

Tabel 13 Confusion Matrix Data Uji DKI Jakarta

	Prediksi Baik	Prediksi Sedang	Prediksi Tidak Sehat	Prediksi Sangat Tidak Sehat	Recall
True Baik	323	0	0	0	100.00%
True Sedang	23	842	23	0	94.82%
True Tidak Sehat	1	4	203	0	97.60%
True Sangat Tidak Sehat	0	0	6	11	64.71%
Precision	93.08%	99.53%	87.50%	100.00%	
Akurasi					96.03%

Pengujian kedua dilakukan menggunakan data ISPU Jakarta Pusat (DKI1) dengan hasil akurasi *training* sebesar 99.37%, akurasi *testing* sebesar 93.97%, lama *training* sebesar 3 menit 9 detik, dan lama *testing* sebesar 0 menit 33 detik serta hasil *confusion matrix* yang ditunjukkan pada Tabel 14.

Tabel 14 Confusion Matrix Data Uji Jakarta Pusat (DKI1)

	Prediksi Baik	Prediksi Sedang	Prediksi Tidak Sehat	Recall
True Baik	112	10	0	91.80%
True Sedang	2	140	5	95.24%
True Tidak Sehat	0	0	13	100.00%
Precision	98.25%	93.33%	72.22%	
Akurasi	93.97%			

Pengujian ketiga dilakukan menggunakan data ISPU Jakarta Utara (DKI2) dengan hasil akurasi *training* sebesar 99.13%, akurasi *testing* sebesar 91.35%, lama *training* sebesar 3 menit 45 detik, dan lama *testing* sebesar 0 menit 49 detik serta hasil *confusion matrix* yang ditunjukkan pada Tabel 15.

Tabel 15 Confusion Matrix Data Uji Jakarta Utara (DKI2)

	Prediksi Baik	Prediksi Sedang	Prediksi Tidak Sehat	Prediksi Sangat Tidak Sehat	Recall
True Baik	47	8	0	0	85.45%
True Sedang	5	162	3	0	95.29%
True Tidak Sehat	0	5	54	1	90.00%
True Sangat Tidak Sehat	0	0	3	1	25.00%
Precision	90.38%	92.57%	90.00%	50.00%	
Akurasi	91.35%				

Pengujian keempat dilakukan menggunakan data ISPU Jakarta Selatan (DKI3) dengan hasil akurasi *training* sebesar 96.40%, akurasi *testing* sebesar 89.47%, lama *training* sebesar 3 menit 25 detik, dan lama *testing* sebesar 0 menit 34 detik serta hasil *confusion matrix* yang ditunjukkan pada Tabel 16.

Tabel 16 Confusion Matrix Data Uji Jakarta Selatan (DKI3)

	Prediksi Baik	Prediksi Sedang	Prediksi Tidak Sehat	Recall
True Baik	64	15	0	81.0%
True Sedang	2	144	0	98.63%
True Tidak Sehat	1	12	47	78.33%
Precision	95.52%	84.21%	100.00%	
Akurasi	89.47%			

Pengujian kelima dilakukan menggunakan data ISPU Jakarta Timur (DKI4) dengan hasil akurasi *training* sebesar 99.38%, akurasi *testing* sebesar 89.27%, lama *training* sebesar 3 menit 7 detik, dan lama *testing* sebesar 0 menit 38 detik serta hasil *confusion matrix* yang ditunjukkan pada Tabel 17.

Tabel 17 Confusion Matrix Data Uji Jakarta Timur (DKI4)

	Prediksi Baik	Prediksi Sedang	Prediksi Tidak Sehat	Recall
--	---------------	-----------------	----------------------	--------

True Baik	42	3	0	93.33%
True Sedang	17	210	5	90.52%
True Tidak Sehat	0	6	6	50.00%
Precision	71.19%	95.89%	54.55%	
Akurasi	89.27%			

Pengujian terakhir dilakukan menggunakan data ISPU Jakarta Barat (DKI5) dengan hasil akurasi *training* sebesar 98.74%, akurasi *testing* sebesar 87.29%, lama *training* sebesar 3 menit 48 detik, dan lama *testing* sebesar 0 menit 38 detik serta hasil *confusion matrix* yang ditunjukkan pada Tabel 18.

Tabel 18 Confusion Matrix Data Uji Jakarta Barat (DKI5)

	Prediksi Baik	Prediksi Sedang	Prediksi Tidak Sehat	Prediksi Sangat Tidak Sehat	Recall
True Baik	47	6	0	0	88.68%
True Sedang	8	119	11	0	86.23%
True Tidak Sehat	0	6	79	0	92.94%
True Sangat Tidak Sehat	0	0	6	9	60.00%
Precision	85.45%	90.84%	82.29%	100%	
Akurasi	87.29%				

Berdasarkan hasil enam pengujian di atas, dapat diringkas ke dalam tabel yang ditunjukkan pada Tabel 19 dan Tabel 20 sebagai berikut:

Tabel 19 Hasil Pengujian Data

Lokasi	Waktu Training	Waktu Testing	Akurasi Training	Akurasi Testing
DKI Jakarta	38 Menit 2 Detik	4 Menit 4 Detik	97.04%	96.03%
Jakarta Pusat (DKI1)	3 Menit 9 Detik	0 Menit 33 Detik	99.37%	93.97%
Jakarta Utara (DKI2)	3 Menit 45 Detik	0 Menit 49 Detik	99.13%	91.35%
Jakarta Selatan (DKI3)	3 Menit 25 Detik	0 Menit 34 Detik	96.40%	89.47%
Jakarta Timur (DKI4)	3 Menit 7 Detik	0 Menit 38 Detik	99.38%	89.27%
Jakarta Barat (DKI5)	3 Menit 48 Detik	0 Menit 38 Detik	98.74%	87.29%

Tabel 20 Lanjutan Hasil Pengujian Data

Lokasi	Recall	Precision	F1 Score
DKI Jakarta	89.28%	95.03%	92.06%
Jakarta Pusat (DKI1)	95.68%	87.93%	91.64%
Jakarta Utara (DKI2)	73.94%	80.74%	77.19%
Jakarta Selatan (DKI3)	85.99%	93.24%	89.47%
Jakarta Timur (DKI4)	77.95%	73.87%	75.86%
Jakarta Barat (DKI5)	81.96%	89.65%	85.63%

Berdasarkan hasil pengujian data di atas, didapatkan hasil akurasi prediksi pada semua data DKI Jakarta sebesar 96.03% dengan lama pelatihan data 38 menit 2 detik, akurasi prediksi pada data Jakarta Pusat (DKI1) sebesar 93.97% dengan lama pelatihan data 3 menit 9 detik, akurasi prediksi pada data Jakarta Utara (DKI2) sebesar 91.35% dengan

lama pelatihan data 3 menit 45 detik, akurasi prediksi pada data Jakarta Selatan (DKI3) sebesar 89.47% dengan lama pelatihan data 3 menit 25 detik, akurasi prediksi pada data Jakarta Timur (DKI4) sebesar 89.27% dengan lama pelatihan data 3 menit 7 detik, dan akurasi prediksi pada data Jakarta Barat (DKI5) sebesar 87.29% dengan lama pelatihan data 3 menit 48 detik.

Berdasarkan pembahasan di atas, penelitian yang dilakukan penulis terbukti dapat memprediksi kualitas udara di DKI Jakarta dengan akurasi tertinggi yaitu 96.03% dimana data yang tidak dapat diprediksi dengan benar sebanyak 57 data dari 1436 data.

5. PENUTUP

5.1. Kesimpulan

Setelah melakukan penelitian mengenai metode *support vector machine* (SVM) dengan *kernel radial basis function* (RBF) untuk memprediksi kualitas udara di Provinsi DKI Jakarta yang diimplementasikan pada sistem prediksi kualitas udara (SPKU) DKI Jakarta, peneliti dapat menarik kesimpulan bahwa metode *Support Vector Machine* dengan menggunakan kernel *Radial Basis Function* (RBF) dapat digunakan untuk memprediksi kualitas udara di DKI Jakarta dengan akurasi prediksi pada data semua DKI Jakarta sebesar 96.03%, akurasi prediksi pada data Jakarta Pusat (DKI1) sebesar 93.97%, akurasi prediksi pada data Jakarta Utara (DKI2) sebesar 91.35, akurasi prediksi pada data Jakarta Selatan (DKI3) sebesar 89.47%, akurasi prediksi pada data Jakarta Timur (DKI4) sebesar 89.27%, dan akurasi prediksi pada data Jakarta Barat (DKI5) sebesar 87.29% sehingga dapat memberikan peringatan dini kepada pemerintah dan masyarakat mengenai kualitas udara.

Parameter SVM yang digunakan pada pelatihan data sangat mempengaruhi waktu komputasi dan akurasi pelatihan serta akurasi prediksi. Parameter yang optimal yaitu nilai *cost* = 3, nilai *epsilon* = 0.001, nilai *sigma* = 0.03, nilai maksimal iterasi = 10, nilai toleransi = 0.0001 dan banyak nilai sama = 10.

5.2. Saran

Berdasarkan kesimpulan yang telah diuraikan di atas maka penulis dapat memberikan saran yang dapat membantu mengatasi beberapa kekurangan yang ada dalam penelitian ini yaitu untuk penelitian selanjutnya, hendaknya membandingkan antara dua atau lebih data kualitas udara dari provinsi yang berbeda. *Dataset* yang berbeda dapat memberikan waktu komputasi dan hasil akurasi yang berbeda.

DAFTAR PUSTAKA

- [1] Anugerah, A.R. (2017), *Penerapan Metode Adaptive Neuro Fuzzy Inference System (ANFIS) untuk Memprediksi Konsentrasi Polutan Udara di Provinsi Daerah Istimewa*

- Yogyakarta, Tugas Akhir, Universitas Islam Indonesia, Yogyakarta.
- [2] Arifien, N.F., Arifin, S. dan Widjiantoro, B.L. (2013), *Prediksi Kadar Polutan Menggunakan Jaringan Syaraf Tiruan (JST) untuk Pemantauan Kualitas Udara di Kota Surabaya*, Skripsi, Institut Teknologi Sepuluh Nopember, Surabaya.
- [3] Harahap, E.H., Muflikhah, L. dan Rahayudi, B. (2018), *Implementasi Algoritma Support Vector Machine (SVM) Untuk Penentuan Seleksi Atlet Pencak Silat*, Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIHK) Universitas Brawijaya, 2(10), 3843–3848.
- [4] Kurnia, A.I., Furqon, M.T. dan Rahayudi, B. (2018), *Klasifikasi Kualitas Susu Sapi Menggunakan Algoritme Support Vector Machine (SVM) (Studi Kasus: Perbandingan Fungsi Kernel Linier dan RBF Gaussian)*, Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIHK) Universitas Brawijaya, 2(11), 4453–4461.
- [5] Mauludiya, R.A. (2015), *Simulasi dan Analisis Klasifikasi Genre Musik Berbasis FFT dan Support Vector Machine*, Skripsi, Universitas Telkom, S1 Teknik Telekomunikasi, Bandung.
- [6] Octaviani, P.A., Wilandari, Y. dan Ispriyanti, D. (2014), *Penerapan Metode Klasifikasi Support Vector Machine (SVM) pada Data Akreditasi Sekolah Dasar (SD) di Kabupaten Magelang*, Jurnal Gaussian, 3(8), 811–820.
- [7] Pemerintah Republik Indonesia, (1999), *Peraturan Pemerintah Republik Indonesia Nomor 41 Tahun 1999 tentang Pengendalian Pencemaran Udara*, Jakarta: Sekretaris Kabinet Republik Indonesia.
- [8] Platt, J.C. (1998), *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*, Microsoft Research, 1–21.
- [9] Pratama, E.E. dan Trilaksono, B.R. (2017), *Klasifikasi Topik Keluhan Pelanggan Berdasarkan Tweet dengan Menggunakan Penggabungan Feature Hasil Ekstraksi pada Metode Support Vector Machine (SVM)*, Jurnal Edukasi dan Penelitian Informatika (JEPIN), 1(2).
- [10] Reyhana, Z. (2018), *Analisis Sentimen Pendapat Masyarakat Terhadap Pembangunan Infrastruktur Kota Surabaya Melalui Twitter dengan Menggunakan Support Vector Machine dan Neural Network*, Tesis, Program Magister Jurusan Statistika, Institut Teknologi Sepuluh Nopember, Surabaya.
- [11] Sela, E.I., dan Hartati, S. (2005), *Pengenalan Jenis Penyakit THT Menggunakan Jaringan Learning Vector Quantization*, Yogyakarta: STMIK Akakom, Universitas Gajah Mada.
- [12] Tama, P.W.S. (2016), *Aplikasi Sensor Asap MQ9 Dan Sensor Suhu SHT10 pada Robot Terbang sebagai Detektor Asap Berbasis Arduino*, Laporan Akhir, Pendidikan Diploma III Teknik Elektro, Politeknik Negeri Sriwijaya, Palembang.
- [13] Vallero, D. (2014), *Fundamentals of Air Pollution - Fifth Edition*, Cambridge: Elsevier Academic Pres.
- [14] Wijayanti, R.A., Furqon, M.T. dan Adinugroho, S. (2018), *Penerapan Algoritme Support Vector Machine Terhadap Klasifikasi Tingkat Risiko Pasien Gagal Ginjal*, Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIHK) Universitas Brawijaya, 2(10), 3500–3507.