

## DAFTAR GAMBAR

2-1	Update Posisi dan Kecepatan PSO .....	12
2-2	Diagram Alir Optimasi Berbasis Particle Swarm .....	14
2-3	Mencari Fungsi Pemisah yang Optimal untuk Objek yang bisa Dipisahkan secara Linier .....	17
2-4	Memperbesar Margin untuk Meningkatkan Probabilitas suatu Data secara benar .....	18
2-5	Data Spiral yang Menggambar Ketidaklinieran .....	24
2-6	Suatu Kernel Map Mengubah Problem yang tidak Linier menjadi Linier dalam Space Baru .....	25
3-1	Bagan Tahapan Penelitian .....	33
3-2	Implementasi Algoritma PSO .....	35
3-3	Implementasi Algoritma SVM .....	37
3-4	Implementasi Algoritma PSO-SVM .....	41
4-1	Bagan Lingkup Kerja Sistem Otomatisasi Kelayakan Buang Limbah Pertambangan Nikel Menggunakan PSO dan SVM .....	46
4-2	Diagram Konteks Aplikasi Analisis Pembuangan Limbah .....	47
4-3	DFD Level 0 Aplikasi Analisis Pembuangan Limbah .....	49
4-4	Use Case Diagram Aplikasi Simulator .....	51
4-5	Activity Diagram pada Case View Menu .....	53
4-6	Activity Diagram Case Buka Tutup Pintu Pembuangan .....	55

4-7	Sequance Diagram Koneksi Client Server .....	58
4-8	Sequance Diagram Buka Tutup Pintu Pembuangan Limbah .....	59
4-9	Sequance Diagram Pengubah Nilai .....	61
4-10	Use Case Diagram Aplikasi Analisis .....	64
4-11	Activity Diagram Case View Menu, Menerima Data, Analisis dan Eksekusi Hasil Analisis .....	66
4-12	Activity Diagram Case Laporan .....	67
4-13	Rancangan Antar muka Aplikasi Simulator .....	69
4-14	Rancangan Antarmuka Aplikasi Analisis Kandungan Unsur .....	71
4-15	Implementasi Aplikasi Simulator .....	74
4-16	Implementasi Aplikasi Analisis .....	80
4-17	Data Awal Aplikasi Analisis .....	81
4-18	Hasil Proses Proses Nilai Menurun dalam Range Baku Mutu.....	84
4-19	Data Awal Kondisi Menaikkan Nilai .....	90
4-20	Hasil Proses Nilai Naik dalam Range Baku Mutu .....	92
4-21	Data Awal Aplikasi Analisis di Atas Toleransi Batas Atas .....	98
4-23	Hasil Proses Nilai di Atas Toleransi Batas Atas .....	99
4-24	Data Awal Aplikasi Analisis di Atas Toleransi Batas Bawah .....	106
4-25	Hasil Proses Nilai di Bawah Tolerani Batas Bawah .....	107
4-26	Flowchart Aplikasi Simulator .....	116
4-27	Flowgraph Aplikasi Simulator .....	116
4-28	Flowchart Aplikasi Analisis .....	118

4-29	Flowgraph Aplikasi Analisis .....	119
4-30	Bagan Real Time Measurement .....	124

## DAFTAR TABEL

2-1	Baku Mutu Air Limbah Bagi Usaha dan/Kegiatan Pertambangan Biji Nikel .....	9
4-1	Asumsi Toleransi Batas Bawah dan Batas Atas Baku Mutu .....	79
4-2	Bobot Hasil Analisis Kondisi Pertama .....	89
4-3	Bobot Hasil Analisis Kondisi Kedua .....	97
4-4	Bobot Hasil Analisis Kondisi Ketiga .....	105
4-5	Bobot Hasil Analisis Kondisi Keempat .....	113
4-6	Rekapitulasi Nilai Bobot pada Empat Kondisi .....	114
4-7	Pengujian Aplikasi Simulator .....	121
4-8	Pengujian Aplikasi Analisis .....	122

## DAFTAR GRAFIK

4-1	Data Awal pada Kondisi Pertama .....	82
4-2	Hasil Analisis PSO Data Kondisi Pertama .....	88
4-3	Hasil Analisis SVM Data Kondisi Pertama .....	88
4-4	Hasil Analisis PSO-SVM Data Kondisi Pertama .....	88
4-5	Data Awal pada Kondisi Kedua .....	91
4-6	Hasil Analisis PSO Data Kondisi Kedua .....	96
4-7	Hasil Analisis SVM Data Kondisi Kedua .....	96
4-8	Hasil Analisis PSO-SVM Data Kondisi Kedua .....	96
4-9	Data Awal pada Kondisi Ketiga .....	99
4-10	Hasil Analisis PSO Data Kondisi Ketiga .....	104
4-11	Hasil Analisis SVM Data Kondisi Ketiga .....	104
4-12	Hasil Analisis PSO-SVM Data Kondisi Ketiga .....	104
4-13	Data Awal pada Kondisi Keempat .....	107
4-14	Hasil Analisis PSO Data Kondisi Keempat .....	112
4-15	Hasil Analisis SVM Data Kondisi Keempat .....	112
4-16	Hasil Analisis PSO-SVM Data Kondisi Keempat .....	112

# **BAB I**

## **PENDAHULUAN**

### **I.1. Latar Belakang**

Limbah industri pada umumnya merupakan bahan beracun dan berbahaya (B3) yang berasal dari proses produksi sebuah industri. Limbah dapat mencemari dan merusak lingkungan serta dapat membahayakan kelangsungan hidup manusia. Berbagai upaya dilakukan untuk mencegah terjadinya pencemaran dan rusaknya lingkungan hidup akibat B3 melalui peraturan pemerintah (PP) tentang pengelolaan limbah, penetapan standar baku mutu air limbah, peneliti-penelitian tentang pengelolaan limbah dan usaha-usaha lainnya yang dilakukan untuk menurunkan kandungan B3 sehingga pengelolaan dan pemanfaatan sumber daya alam tidak memberikan dampak negatif terhadap manusia dan lingkungannya.

Beberapa penelitian telah dilakukan dibidang pengelolaan limbah, yaitu menurunkan kadar total suspended solid (TSS), total Fe, total Mn menggunakan biji kelor pada pertambangan batu bara [1], menurunkan kadar Cu, Cr dan Ag melalui adsorpsi (penyerapan) menggunakan tanah liat pada industri perak [2], pengaruh pH dan penggunaan biomassa *Aspergillus niger* van Tieghem dalam penyerapan logam Zn dari limbah pertambangan nikel melalui proses biosorpsi [3]. Dan masih banyak lagi penelitian-penelitian yang

dilakukan untuk mencari cara dan metode sehingga limbah yang dihasilkan oleh industri pertambangan tidak merusak lingkungan.

Dari banyak penelitian yang dilakukan, telah ditemukan banyak metode pengelolaan limbah industri pertambangan sehingga kandungan B3 pada limbah dapat dikendalikan sesuai standar baku mutu air limbah yang tidak membahayakan kehidupan manusia dan lingkungan. Tetapi bagaimana cara membuang limbah tersebut merupakan hal yang penting diteliti, untuk menemukan metode baru yang lebih cepat dan tepat menentukan kelayakan buang limbah yang mengandung B3.

Proses pembuangan limbah pertambangan nikel pada umumnya dilakukan melalui uji laboratorium untuk mengetahui kandungan unsur-unsur kimia yang terkandung dalam limbah sebelum proses pembuangan dilakukan. Jika hasil pengujian laboratorium terhadap kandungan unsur-unsur dalam limbah telah memenuhi standar kelayakan buang, maka proses pembuangan limbah dilakukan dengan cara membuka pintu penampungan limbah. Metode buka tutup pintu pembuangan limbah berdasarkan hasil uji laboratorium membutuhkan waktu yang relatif lama. Karena untuk menutup kembali pintu pembuangan limbah harus menunggu hasil uji labaratorium, sementara limbah terus mengalir. Bisa jadi ketika hasil uji laboratorium menyatakan proses pembuangan harus dihentikan, limbah yang tidak memenuhi standar kelayakan sudah ikut terbuang. Hasil uji kelayakan buang menjadi tidak valid karena terdapat rentang waktu dari pengambilan sampel

limbah yang akan diuji sampai dengan keputusan penghentian pembuangan limbah.

Untuk itu perlu dilakukan otomatisasi pada proses pembuangan limbah tersebut melalui penerapan teknologi informasi yang dapat memberikan informasi secara real time tentang kandungan unsur dalam limbah yang akan dibuang dan selanjutnya teknologi tersebut secara otomatis menentukan apakah limbah akan dibuang atau tidak..

Teknologi informasi yang dimaksud adalah penggunaan aplikasi untuk menganalisis kelayakan buang limbah pertambangan nikel dengan menggunakan algoritma Support Vektor Machine (SVM) untuk mengklasifikasikan data-data kandungan unsur limbah yang layak dibuang dengan data-data kandungan unsur yang tidak layak dibuang dimana data-data tersebut terlebih dahulu dioptimalisasikan dengan metode Particle Swarm Optimization (PSO).

Proses otomatisasi yang dilakukan dengan metode yang telah dijelaskan di atas, memperoleh input dari detektor/sensor yang bekerja secara real time untuk mendeteksi kandungan unsur-unsur dalam limbah yang akan dianalisis oleh aplikasi, dan selanjutnya hasil analisis dikirim ke peralatan secara real time untuk membuka atau menutup pintu pembuangan limbah.



## **I.2. Rumusan Masalah**

Berdasarkan uraian latar belakang di atas, maka permasalahan dirumuskan sebagai berikut :

1. Bagaimana mengetahui secara otomatis kelayakan buang limbah pertambangan nikel, sehingga limbah yang terbuang sesuai dengan baku mutu air limbah pertambangan nikel.
2. Bagaimana pembuangan limbah pertambangan nikel dilakukan secara otomatis berdasarkan data real time tentang kandungan unsur limbah aktual.
3. Bagaimana otomatisasi pembuangan limbah pertambangan nikel dilakukan menggunakan metode Particle Swarm Optimization (PSO) dan Support Vector Machine (SVM).

## **I.3. Tujuan Penelitian**

Dari penelitian yang dilakukan diharapkan dapat dilakukan otomatisasi dalam penentuan kelayakan buang limbah pertambangan. Untuk menemukan metode yang dimaksud maka dirumuskan tujuan penelitian ini sebagai berikut :

1. Merancang simulasi untuk menganalisis kelayakan buang limbah pertambangan nikel secara otomatis.

2. Membuat database untuk menyimpan data pemantauan kandungan unsur-unsur dalam limbah pertambangan nikel.

#### **I.4. Manfaat Penelitian**

Manfaat dari hasil penelitian yang dilakukan ini dapat menjadi masukan bagi pihak pertambangan nikel. Masukan tersebut adalah :

1. Pembuangan limbah dapat dilakukan secara otomatis saat kondisi limbah mencapai kelayakan buang.
2. Penentuan kelayakan buang limbah dapat dilakukan secara otomatis melalui analisis kesesuaian kadar kandungan unsur dalam limbah dengan standar baku mutu air limbah pertambangan nikel.
3. Pertambangan nikel dapat memantau kadar/konsentrasi bahan beracun dan berbahaya dalam limbah, sehingga dapat mengambil kebijakan dalam proses pengelolaan limbah.

#### **I.5. Batasan Masalah**

Untuk mencapai tujuan yang diharapkan, maka permasalahan pada penelitian ini dibatasi sesuai bidang keilmuan peneliti, yaitu teknik informatika sehingga batas sebagai berikut :

1. Otomatisasi yang dimaksud adalah menentukan kelayakan buang tidak dilakukan melalui uji laboratorim, tetapi melalui aplikasi yang

menggunakan metode PSO dan SVM berdasarkan data kandungan unsur-unsur yang diterima dari peralatan detektor (sensor).

2. Limbah pertambangan nikel yang dimaksud adalah limbah yang sedang dalam proses untuk dibuang.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Limbah Pertambangan Nikel**

Pertambangan adalah kegiatan mengambil bahan galian potensial dari lapisan bumi untuk dimanfaatkan secara optimal untuk kepentingan dan kemakmuran masyarakat, melalui serangkaian kegiatan eksplorasi, pengusahaan dan pemanfaatan hasil tambang. Kegiatan tersebut bertumpu pada pendayagunaan sumber daya terutama sumber daya alam dan energi, didukung oleh sumber daya manusia yang berkualitas, penguasaan ilmu pengetahuan dan teknologi serta kemampuan manajemen mengakibatkan semakin luas dan dalam lapisan bumi yang harus digali [4].

Proses pertambangan nikel sampai menghasilkan biji nikel melalui beberapa tahapan antara lain, pengupasan, penggalian, pemisahan dan penyimpanan. Salah satu sumber limbah pertambangan nikel berasal dari tahapan pemisahan partikel biji nikel dengan butiran pengotor yang tidak dibutuhkan yang terikut pada saat penggalian. Tahapan pemisahan menggunakan air dan bahan kimia lainnya menghasilkan limbah cair [5].

Kegiatan pertambangan, selain menimbulkan dampak lingkungan, ternyata menimbulkan dampak sosial yang kompleks. Oleh sebab itu, analisis dampak lingkungan (AMDAL) suatu kegiatan pertambangan harus dapat menjawab dua tujuan pokok [6] :

1. Memastikan bahwa biaya lingkungan, sosial dan kesehatan dipertimbangkan dalam menentukan kelayakan ekonomi dan penentuan alternatif kegiatan yang akan dipilih.
2. Memastikan bahwa pengendalian, pengelolaan, pemantauan serta langkah-langkah perlindungan telah terintegrasi di dalam desain dan implementasi proyek serta rencana penutupan tambang.

## **II.2. Pembuangan Limbah Pertambangan Nikel**

Menurut Peraturan Menteri Negara Lingkungan Hidup No. 09 Tahun 2006 baku mutu air limbah usaha dan atau kegiatan pertambangan biji nikel adalah ukuran batas atau kadar maksimum unsur pencemar dan atau jumlah unsur pencemar yang ditenggang keberadaannya dalam air limbah yang akan dibuang atau dilepas ke sumber air dari usaha dan atau kegiatan pertambangan biji nikel ditetapkan dengan metode analisis Standar Nasional Indonesia (SNI), sehingga mutu air limbah yang ke badan air tidak melampaui mutu air limbah yang telah ditetapkan dalam peraturan menteri. Pemantauan terhadap kadar parameter baku mutu air limbah dilakukan setiap hari paling sedikit terhadap *power of hydrogen* (pH) dan *total suspended solid* (TSS) air limbah [7].

Secara keseluruhan terdapat 11 unsur yang menjadi parameter baku mutu air limbah pertambangan nikel yang ditetapkan dalam peraturan menteri sebagai berikut :

Tabel 2-1. Baku Mutu Air Limbah Bagi Usaha dan/Kegiatan Pertambangan Biji Nikel [4]

Parameter	Satuan	Kadar Maksimum
pH	-	6-9
TSS	mg/L	100
Cu	mg/L	2
Cd	mg/L	0,05
Zn	mg/L	5
Pb	mg/L	0,1
Ni	mg/L	0,5
Cr <sup>(6+)</sup>	mg/L	0,1
Cr total	mg/L	0,5
Fe	mg/L	5
Co	mg/L	0,4

### II.3. Particle Swarm Optimization

Algoritma *Particle Swarm Optimization* (PSO) pertama kali diperkenalkan oleh oleh Dr. Eberhart dan Dr. Kennedy ditahun 1995 dalam sebuah konferensi jaringan syaraf di Perth, Australia. Algoritma PSO merupakan teknik optimasi berbasis *stochastic* yang diinspirasi oleh tingkah laku sosial sekawanan burung atau sekumpulan ikan [8].

Algoritma PSO menggambarkan tingkah laku sosial sekawanan burung yang sedang menuju ke sumber makan. Untuk memahami algoritma PSO dapat dianalogikan sebagai berikut :

Terdapat sekawanan burung pada suatu kawasan yang terbang secara random mencari makanan. Di kawasan tersebut hanya terdapat sepotong makanan yang akan di cari dan seluruh burung tidak mengetahui letak

makanan yang akan dicari. Tetapi kawanan burung tersebut mengetahui jarak makanan di setiap iterasi. Bagaimana strategi terbaik untuk menemukan makanan tersebut ? salah satu yang efektif adalah mengikuti burung yang lebih dekat dengan makanan.

Algoritma PSO adalah metode pencarian yang didasarkan pada populasi dan merupakan algoritma optimasi global yang dihubungkan dengan masalah yang mana solusi terbaik dapat direpresentasikan sebagai titik atau *surface* diarea *n-dimensional* [9].

### **Algoritma Particle Swarm Optimization**

Algoritma dasar PSO terdiri dari tiga tahap [10], yaitu pembangkitan posisi serta kecepatan partikel, *update velocity* (*update* kecepatan), *update position* (*update* posisi). Partikel berubah posisinya dari suatu perpindahan (*iterasi*) ke posisi lainnya berdasarkan pada *update velocity*. Pertama posisi  $X_k^i$ , dan kecepatan  $V_k^i$  dari kumpulan partikel dibangkitkan secara random menggunakan batas atas ( $X_{max}$ ) dan batas bawah ( $X_{min}$ ) dari *design variable*, seperti yang ditunjukkan pada persamaan berikut :

$$X_0^i = X_{min} + rand(X_{max} - X_{min}) \quad (2.1)$$

$$V_0^i = X_{min} + rand(X_{max} - X_{min}) \quad (2.2)$$

Posisi dan kecepatan direpresentasikan dalam bentuk vektor dimana  $n$  dimensi vektor merepresentasikan jumlah dari desain variabel partikel, dengan superscript dan subscript menotasikan partikel ke  $i$  pada waktu ke  $k$ .

Dengan proses inialisasi ini maka kumpulan partikel dapat terdistribusi secara random pada *design space*. Vektor seperti ditunjukkan di bawah ini :

$$X_k^i = (X_k^{i1}, X_k^{i2}, \dots, X_k^{in})^T \quad (2.3)$$

$$V_k^i = (V_k^{i1}, V_k^{i2}, \dots, V_k^{in})^T \quad (2.4)$$

Langkah kedua adalah *update velocity* (kecepatan) untuk semua partikel pada waktu  $k + 1$  menggunakan fungsi objektif atau nilai fitness posisi partikel saat ini pada *design space* saat waktu ke  $k$ . Dari nilai fitness dapat ditentukan partikel mana yang memiliki nilai global terbaik (*global best*) pada *swarm* saat ini,  $P_k^g$  dan juga dapat ditentukan posisi terbaik dari tiap partikel pada semua waktu yang sekarang dan sebelumnya,  $P_k^i$ . Perumusan *update velocity* menggunakan dua informasi tersebut untuk semua partikel pada kumpulan dengan pengaruh perpindahan yang sekarang,  $V_k^i$ , untuk memberikan arah pencarian,  $V_{k+1}^i$ , untuk generasi selanjutnya. Perumusan *update velocity* mencakup beberapa parameter random (*rnd*), untuk mendapatkan cakupan yang baik pada *design space*, tiga parameter yang mempengaruhi arah pencarian, yaitu *inertia factor* ( $w$ ), *self confidence* ( $c_1$ ), *swarm confidence* ( $c_2$ ) akan digabungkan dalam satu penyajian, seperti yang ditunjukkan persamaan berikut :

$$V_{k+1}^i = w V_k^i + c_1 \text{rnd} (P_k^i - X_k^i) + c_2 \text{rnd} (P_k^g - X_k^i) \quad (2.5)$$

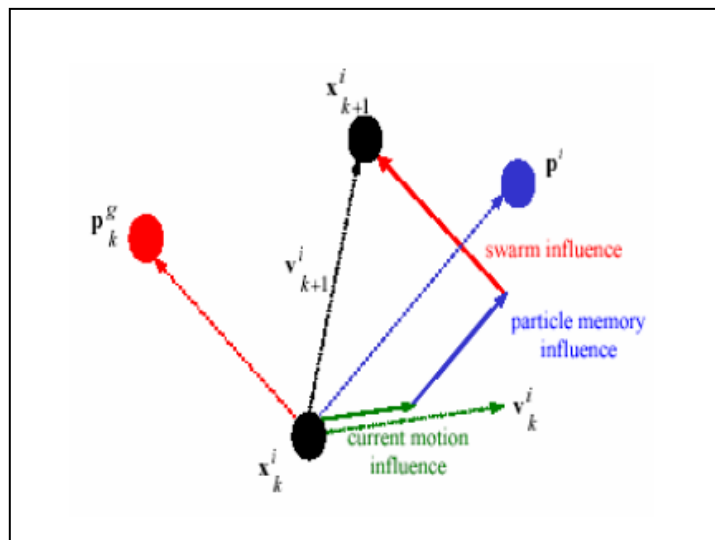
Dengan range  $w = 0.4 - 1.4$ ,  $c_1 = 1.5 - 2.0$ ,  $c_2 = 2.0 - 2.5$



Langkah terakhir dari setiap iterasi adalah *update* posisi tiap partikel dengan vektor *velocity*, seperti yang ditunjukkan pada persamaan berikut ini :

$$X_{k+1}^j = X_k^j + V_{k+1}^j \quad (2.6)$$

Tiga tahapan diatas akan diulang sampai kriteria kekonvergenan terpenuhi, kriteria kekonvergenan sangat penting dalam menghindari penambahan fungsi evaluasi setelah solusi optimum didapatkan, namun kriteria kekonvergenan tidak selalu mutlak diperlukan, penetapan jumlah iterasi maksimal juga dapat digunakan sebagai *stopping condition* dari algoritma.



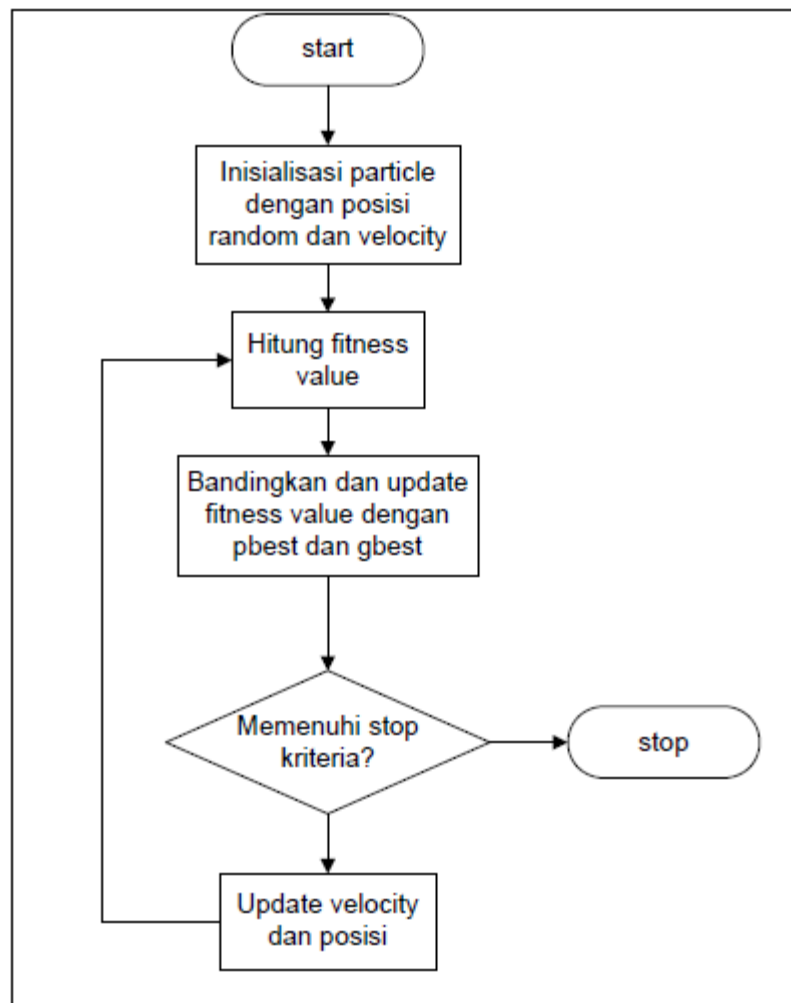
Gambar 2-1. Update Posisi dan Kecepatan PSO [10]

Pseudo-code untuk prosedur optimasi berbasis *particle swarm* adalah sebagai berikut:

1. Bentuk populasi awal dari partikel

2. Hitung nilai kecocokan tiap partikel, jika lebih baik dari nilai kecocokan selama proses berjalan, set nilai tersebut sebagai pbest.
3. Pilih partikel dengan nilai kecocokan terbaik, set nilai partikel ini sebagai gbest.
4. Untuk setiap partikel:
  - a. hitung velocity berdasarkan fungsi update velocity
  - b. perbaharui posisi partikel berdasarkan fungsi update posisi
5. Berhenti jika kondisi yang diharapkan tercapai. Jika belum, kembali ke langkah 2.

Prosedur di atas juga dapat digambarkan dalam bentuk lain, Gambar 2 menggambarkan prosedur optimasi berbasis particle swarm dengan bentuk diagram alir. Aliran diagram dimulai dengan menginisialisasi partikel dengan posisi dan velocity secara acak. Kemudian nilai kecocokan dari populasi tersebut dihitung, lalu nilai kecocokan tersebut dibandingkan dan dibaharui. Jika memenuhi kriteria untuk berhenti, maka prosedur ini akan berhenti. Namun jika tidak, nilai velocity dan posisi akan dibaharui dan kembali pada langkah menghitung nilai kecocokan tiap-tiap partikel pada populasi.



Gambar 2-2. Diagram Alir Optimasi Berbasis Particle Swam [9]

#### II.4. Support Vector Machine

Support vector machine (SVM) adalah suatu teknik yang relatif baru (1995) untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi, yang sangat populer belakangan ini. SVM berada dalam satu kelas dengan ANN dalam hal fungsi dan kondisi permasalahan yang bisa

diselesaikan. Keduanya masuk dalam kelas *supervised learning*. Baik para ilmuwan maupun praktisi telah banyak menerapkan teknik ini dalam menyelesaikan masalah-masalah nyata dalam kehidupan sehari-hari. Baik dalam masalah gene expression analysis, finansial, cuaca hingga di bidang kedokteran. Terbukti dalam banyak implementasi, SVM memberi hasil yang lebih baik dari ANN, terutama dalam hal solusi yang dicapai. ANN menemukan solusi berupa *local optimal* sedangkan SVM menemukan solusi yang *global optimal*. Tidak heran bila kita menjalankan ANN solusi dari setiap training selalu berbeda. Hal ini disebabkan solusi *local optimal* yang dicapai tidak selalu sama [11].

SVM selalu mencari solusi yang sama untuk setiap running. Dalam teknik ini, kita berusaha untuk menemukan *fungsi pemisah*(klasifier) yang optimal yang bisa memisahkan dua set data dari dua kelas yang berbeda. Teknik ini menarik orang dalam bidang data mining maupun machine learning karena performansinya yang meyakinkan dalam memprediksi kelas suatu data baru. Kita akan memulai pembahasan dengan kasus klasifikasi yang secara linier bisa dipisahkan. Dalam hal ini fungsi pemisah yang dicari adalah fungsi linier. Fungsi ini didefinisikan sebagai berikut :

$$g(x) := \text{sgn}(f(x)) \quad (2.7)$$

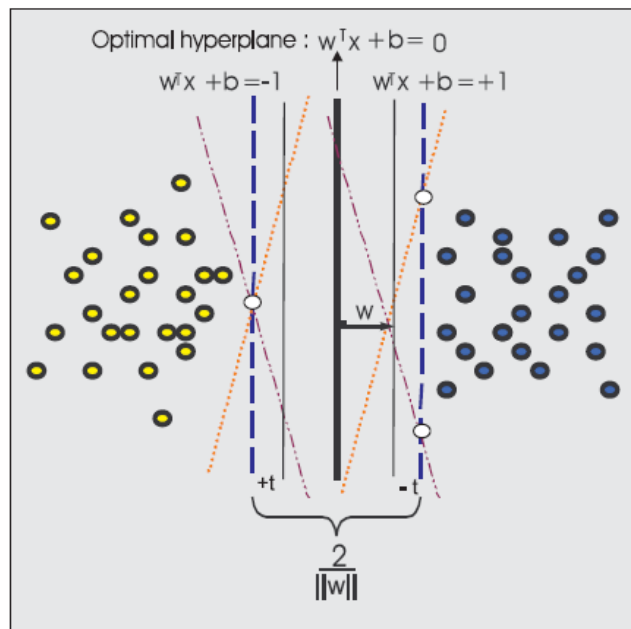
$$\text{dengan } f(x) = w^T x + b$$

dimana  $x, w \in R^n$  dan  $b \in R$ . Masalah klasifikasi ini bisa dirumuskan sebagai berikut:

Kita ingin menemukan set parameter  $(w, b)$  sehingga  $f(x) = \langle w, x \rangle + b = y_i$  untuk semua  $i$ . Dalam teknik ini kita berusaha menemukan fungsi pemisah (klasifier/hyperplane) terbaik diantara fungsi yang tidak terbatas jumlahnya untuk memisahkan dua macam obyek. Hyperplane terbaik adalah hyperplane yang terletak di tengah-tengah antara dua set obyek dari dua kelas. Mencari hyperplane terbaik ekuivalen dengan memaksimalkan margin atau jarak antara dua set obyek dari kelas yang berbeda. Jika  $wx_1 + b = +1$  adalah hyperplane-pendukung (supporting hyperplane) dari kelas  $+1$  ( $wx_1 + b = +1$ ) dan  $wx_2 + b = -1$  hyperplane-pendukung dari kelas  $-1$  ( $wx_2 + b = -1$ ), margin antara dua kelas dapat dihitung dengan mencari jarak antara kedua hyperplane-pendukung dari kedua kelas. Secara spesifik, margin dihitung dengan cara berikut  $(wx_1 + b = +1) - (wx_2 + b = -1) \Rightarrow w(x_1 - x_2) = 2 \Rightarrow \left( \frac{w}{\|w\|} (x_1 - x_2) \right) = \frac{2}{\|w\|}$ . Gambar 3 memperlihatkan bagaimana SVM bekerja untuk menemukan suatu fungsi pemisah dengan margin yang maksimal. Untuk membuktikan bahwa memaksimalkan margin antara dua set obyek akan meningkatkan probabilitas pengelompokkan secara benar dari data testing. Pada dasarnya jumlah fungsi pemisah ini tidak terbatas banyaknya. Misalkan dari jumlah yang tidak terbatas ini kita ambil dua saja, yaitu  $f_1(x)$

and  $f_2(x)$  (lihat gambar 2). Fungsi  $f_1$  mempunyai margin yang lebih besar dari pada fungsi  $f_2$ .

Setelah menemukan dua fungsi tersebut, sekarang suatu data baru masuk dengan keluaran  $-1$ . Kita harus mengelompokkan apakah data ini ada dalam kelas  $-1$  atau  $+1$  menggunakan fungsi pemisah yang sudah kita temukan. Dengan menggunakan  $f_1$ , kita akan kelompokkan data baru ini di kelas  $-1$  yang berarti kita benar mengelompokkannya.



Gambar 2-3. Mencari Fungsi Pemisah Yang Optimal Untuk Objek Yang Bisa Dipisahkan Secara Linier [11].

Sekarang coba kita gunakan  $f_2$ , kita akan menempatkannya di kelas  $+1$  yang berarti salah. Dari contoh sederhana ini kita lihat bahwa

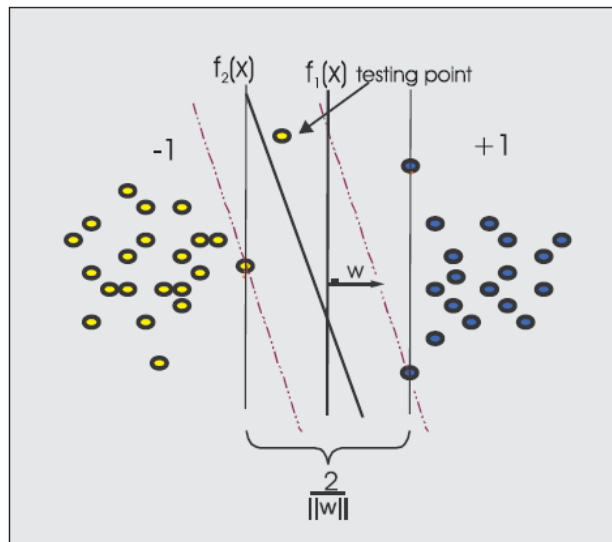
memperbesar margin bisa meningkatkan probabilitas pengelompokkan suatu data secara benar.

### Formulasi Matematis SVM

Secara matematika, formulasi problem optimisasi SVM untuk kasus klasifikasi linier di dalam *primal space* adalah :

$$\min \frac{1}{2} \|w\|^2 \quad (2.8)$$

$$y_i(wx_i + b) \geq 1, i = 1, \dots, l \quad (2.9)$$



Gambar 2-4 Memperbesar Margin Untuk Meningkatkan Probabilitas Suatu Data Secara Benar [11].

dimana  $x_i$  adalah data input,  $y_i$  adalah keluaran dari data  $x_i$ ,  $w$ ,  $b$  adalah parameter-parameter yang kita cari nilainya. Dalam formulasi di atas, kita ingin meminimalkan fungsi tujuan (obyektif function)  $\frac{1}{2} \|w\|^2$  atau

memaksimalkan kuantitas  $\|w\|^2$  atau  $w^T w$  dengan memperhatikan pembatas  $y_i(w x_i + b) \geq 1$ . Bila output data  $y_i = +1$ , maka pembatas menjadi  $(w x_i + b) \geq 1$ . Sebaliknya bila  $y_i = -1$ , pembatas menjadi  $(w x_i + b) \leq -1$ . Di dalam kasus yang tidak feasible (infeasible) dimana beberapa data mungkin tidak bisa dikelompokkan secara benar, formulasi matematikanya menjadi berikut :

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l t_i \quad (2.10)$$

$$y_i(w x_i + b) + t_i \geq 1 \quad (2.11)$$

$$t_i \geq 0, i = 1, \dots, l$$

dimana  $t_i$  adalah variabel *slack*. Dengan formulasi ini kita ingin memaksimalkan margin antara dua kelas dengan meminimalkan  $\|w\|^2$ . Dalam formulasi ini kita berusaha meminimalkan kesalahan klasifikasi (misclassification error) yang dinyatakan dengan adanya variabel *slack*  $t_i$ , sementara dalam waktu yang sama kita memaksimalkan margin,  $\frac{1}{\|w\|}$ . Penggunaan variabel *slack*  $t_i$  adalah untuk mengatasi kasus ketidaklayakan (infeasibility) dari pembatas (constraints)  $y_i(w x_i + b) \geq 1$  dengan cara memberi pinalti untuk data yang tidak memenuhi pembatas tersebut. Untuk meminimalkan nilai  $t_i$  ini, kita berikan pinalti dengan menerapkan konstanta ongkos  $C$ . Vektor  $w$  tegak lurus terhadap fungsi pemisah  $w x + b = 0$ . Konstanta  $b$  menentukan lokasi fungsi pemisah relatif terhadap titik asal (origin).



Persamaan (2.10) adalah masalah nonlinear. Ini bisa dilihat dari fungsi tujuan (objective function) yang berbentuk kuadrat. Untuk menyelesaikannya, secara komputasi agak sulit dan perlu waktu lebih panjang. Untuk membuat masalah ini lebih mudah dan efisien untuk diselesaikan, masalah ini bisa kita transformasikan ke dalam *dual space*. Untuk itu, pertama kita ubah persamaan (2.10) menjadi fungsi Lagrangian :

$$J(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^l \alpha_i [y_i (w^T x_i + b) - 1] \quad (2.12)$$

dimana variabel *non-negatif*  $\alpha_i$ , dinamakan *Lagrange multiplier*. Solusi dari problem optimisasi dengan pembatas seperti di atas ditentukan dengan mencari *saddle point* dari fungsi Lagrangian  $J(w, b, \alpha)$ . Fungsi ini harus diminimalkan terhadap variabel  $w$  dan  $b$  dan harus dimaksimalkan terhadap variabel  $\alpha$ . Kemudian kita cari turunan pertama dari fungsi  $J(w, b, \alpha)$  terhadap variabel  $w$  dan  $b$  dan kita samakan dengan 0. Dengan melakukan proses ini, kita akan mendapatkan dua kondisi optimalitas berikut:

1. Kondisi 1 :  $\frac{\partial J(w, b, \alpha)}{\partial w} = 0$
2. Kondisi 2 :  $\frac{\partial J(w, b, \alpha)}{\partial b} = 0$

Penerapan kondisi optimalitas 1 pada fungsi Lagrangian (4) akan menghasilkan :

$$w = \sum_{i=1}^l \alpha_i y_i x_i \quad (2.13)$$

Penerapan kondisi optimalitas pada fungsi Lagrangian (2.12) akan menghasilkan :

$$\sum_{i=1}^l \alpha_i y_i = 0 \quad (2.14)$$

Menurut duality theorem :

1. Jika problem primal mempunyai solusi optimal, maka problem dual juga akan mempunyai solusi optimal yang nilainya sama.
2. Bila  $w_0$  adalah solusi optimal untuk problem primal dan  $\alpha_0$  untuk problem dual, maka *perlu* dan *cukup* bahwa  $w_0$  solusi layak untuk problem primal dan  $\Phi(w_0) = J(w_0, b_0, \alpha_0) = \min_w J(w, b, \alpha)$ .

Untuk mendapatkan problem dual dari problem kita, kita jabarkan persamaan (2.11) sebagai berikut :

$$J(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^l \alpha_i y_i w^T x_i - b \sum_{i=1}^l \alpha_i y_i + \sum_{i=1}^l \alpha_i \quad (2.15)$$

Menurut kondisi optimalitas ke dua dalam (2.14), term ketiga sisi sebelah kanan dalam persamaan di atas sama dengan 0. Dengan memakai nilai-nilai  $w$  di (2.13), kita dapatkan

$$w^T w = \sum_{i=1}^l \alpha_i y_i w^T x_i = \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (2.16)$$

maka persamaan (2.15) menjadi :

$$Q(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l y_i y_j \alpha_i \alpha_j x_i^T x_j \quad (2.17)$$

Selanjutnya kita dapatkan formulasi dual dari persamaan (2.10) :

$$\text{Max } \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j x_i^T x_j \quad (2.18)$$

$$\sum_{i=1}^l \alpha_i y_i = 0$$

$$0 \leq \alpha_i, i = 1, \dots, l$$

Dengan dot product  $x_i x_j$  sering diganti dengan simbol  $K$ .  $K$  adalah matrik kernel yang dijelaskan dalam bagian 3. Formulasi (2.18) adalah quadratic programming (QP) dengan pembatas (constraint) linier. Melatih SVM ekuivalen dengan menyelesaikan problem *convex optimization*. Karena itu solusi dari SVM adalah unik (dengan asumsi bahwa  $k$  adalah positive definite) dan global optimal. Hal ini berbeda dengan solusi neural networks yang ekuivalen dengan problem *nonconvex optimization* dengan akibat solusi yang ditemukan adalah *local optima*. Ambil  $f(x) = \sum_{i=1}^l y_i \alpha_i^* k(x_i, x) + b^*$ .

Fungsi pemisah optimal adalah  $g(x) = \text{sign} (\sum_{i=1}^l y_i \alpha_i^* k(x_i, x)) + b^*$ , dimana  $\alpha_i^*, i = 1, \dots, l$  adalah solusi optimal dari persamaan (2.18) dan  $b^*$  dipilih sehingga  $y_i f(x_i) = 1$  untuk sembarang  $i$  dengan  $C > \alpha_i^* > 0$ . Data  $x_i$  dimana  $\alpha_i^* > 0$  dinamakan *support vector* dan menyatakan data training yang diperlukan untuk mewakili fungsi keputusan yang optimal. Dalam gambar 3, sebagai contoh, 3 titik berwarna putih menyatakan *support vector*. Untuk mengatasi masalah ketidaklinieran (nonlinearity) yang sering terjadi dalam kasus nyata, kita bisa menerapkan metoda kernel. Metoda kernel memberikan pendekatan alternatif dengan cara melakukan mapping data  $x$

dari input space ke *feature space*  $F$  melalui suatu fungsi  $\varphi$  sehingga  $\varphi : x \rightarrow \varphi(x)$ . Karena itu suatu titik  $x$  dalam input space menjadi  $\varphi(x)$  dalam feature space.

### **Metode Kernel SVM**

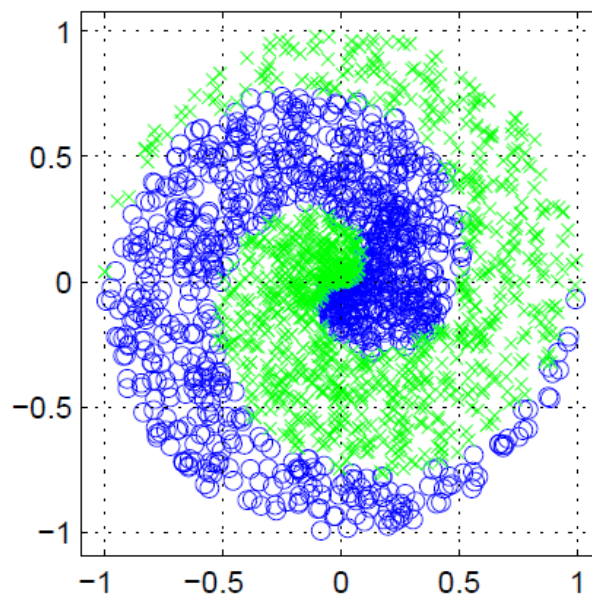
Banyak teknik data mining atau machine learning yang dikembangkan dengan asumsi kelinieran. Sehingga algoritma yang dihasilkan terbatas untuk kasus-kasus yang linier. Karena itu, bila suatu kasus klasifikasi memperlihatkan ketidaklinieran, algoritma seperti perceptron tidak bisa mengatasinya. Secara umum, kasus-kasus di dunia nyata adalah kasus yang tidak linier. Sebagai contoh, perhatikan Gambar 5. Data ini sulit dipisahkan secara linier. Metoda kernel adalah salah satu untuk mengatasinya.

Dengan metoda kernel suatu data  $x$  di ruang input petakan ke feature space  $F$  dengan dimensi yang lebih tinggi melalui pemetaan  $\varphi$  sebagai berikut  $\varphi : x \rightarrow \varphi(x)$ . Karena itu data  $x$  di input space menjadi  $\varphi(x)$  di feature space.

Sering kali fungsi  $\varphi(x)$  tidak tersedia atau tidak bisa dihitung. tetapi *dotproduct* dari dua vektor dapat dihitung baik di dalam *input space* maupun di *feature space*. Dengan kata lain, sementara  $\varphi(x)$  mungkin tidak diketahui, dot product  $\langle \varphi(x_1), \varphi(x_2) \rangle$  masih bisa dihitung di feature space. Untuk bisa memakai metoda kernel, pembatas (constraint) perlu diekspresikan dalam bentuk dot product dari vektor data  $x_i$ . Sebagai konsekuensi, pembatas yang menjelaskan permasalahan dalam klasifikasi harus

diformulasikan kembali sehingga menjadi bentuk dot product. Dalam feature space ini dot product  $\langle \cdot \rangle$  menjadi  $\langle \varphi(x), \varphi(x) \rangle$ . Suatu fungsi kernel,  $k(x, x')$ , bisa untuk menggantikan dot product  $\langle \varphi(x), \varphi(x) \rangle$ .

Kemudian di feature space, kita bisa membuat suatu fungsi pemisah yang linier yang mewakili fungsi nonlinear di input space. Gambar 4 mendeskripsikan suatu contoh feature mapping dari ruang dua dimensi ke feature space dua dimensi.



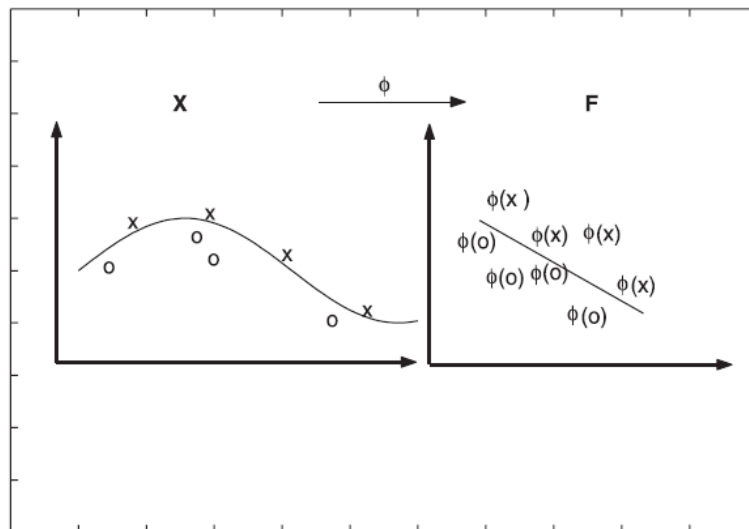
Gambar 2-5. Data Spiral yang Menggambarkan Ketidaklinieran [11].

Dalam input space, data tidak bisa dipisahkan secara linier, tetapi kita bisa memisahkan di feature space. Karena itu dengan memetakan data ke

feature Karena itu dengan memetakan data ke feature space menjadikan tugas klasifikasi menjadi lebih mudah.

Fungsi kernel yang biasanya dipakai dalam literatur SVM [11]:

- linier :  $x^T x$ ,
- Polynomial :  $(x^T x_i + 1)^p$ ,
- Radial basis function (RBF) :  $\exp\left[-\frac{1}{2\sigma^2} \|x - x_i\|^2\right]$
- Tangent hyperbolic (sigmoid) :  $\tanh(\beta x^T x_i + \beta_1)$ , dimana  $\beta, \beta_1 \in R$



Gambar 2-6. Suatu Kernel Map Mengubah Problem yang Tidak Linier Menjadi Linier dalam Space Baru [11].

Fungsi kernel mana yang harus digunakan untuk substitusi dot product di feature space sangat bergantung pada data. Biasanya metoda cross-validation digunakan untuk pemilihan fungsi kernel ini. Pemilihan fungsi

kernel yang tepat adalah hal yang sangat penting. Karena fungsi kernel ini akan menentukan feature space di mana fungsi klasifier akan dicari. Sepanjang fungsi kernelnya legitimate, SVM akan beroperasi secara benar meskipun kita tidak tahu seperti apa map yang digunakan. Fungsi kernel yang legitimate diberikan oleh Teori Mercer dimana fungsi itu harus memenuhi syarat: kontinu dan positive definite. Lebih mudah menemukan fungsi kernel daripada mencari map  $\varphi$  seperti apa yang tepat untuk melakukan mapping dari *input space* ke *feature space*. Pada penerapan metoda kernel, kita tidak perlu tahu map apa yang digunakan untuk satu per satu data, tetapi lebih penting mengetahui bahwa dot produk dua titik di feaure space bisa digantikan oleh fungsi kernel.

## II.5. Road Map Penelitian

Beberapa peneliti terdahulu telah melakukan penelitian sebelumnya tentang metode pembuangan limbah, antara lain :

1. **“Simulation Of Waste Processing, Transportation, And Disposal Operations”**, oleh Janis Trone, Sandia National Laboratory USA, 2000. Penelitian dilakukan dalam rangka pembuatan simulasi pembuangan limbah radio aktif dari 25 tempat di USA selama 35 tahun ke depan yang dilakukan oleh Sandia National Laboratory. Model yang dikembangkan dalam penelitian ini memungkinkan pengguna untuk mengamati proses

pembuangan limbah, mengamati proses pengangkutan jalur transportasi pembuangan limbah dan dapat mengevaluasi proses pembuangan limbah sampai ke penampungan.

Hubungan dengan penelitian yang akan dilakukan adalah adanya data tersimpan tentang konsentrasi unsur dalam limbah dan dapat dievaluasi untuk kepentingan pengelolaan limbah pada pertambangan nikel [15].

2. **“Modeling Instructible Robots For Waste Disposal Applications”**, oleh Lefteri H. Tsoukalas and Dimitrios T. Bargiotas. Purdue University and The University of Tennessee. Penelitian yang dilakukan adalah membuat aplikasi pembuangan limbah radio aktif menggunakan lengan robot dengan pendekatan *neurofuzzy*. Pada penelitian ini lengan robot mendeteksi adanya kebocoran aliran gamma, alpha dll, di *centralized interim storage facilities* (CIS), kemudian dengan hati-hati mengangkat atau membuangnya.

Hubungan dengan penelitian yang akan dilakukan yaitu proses pembelajaran untuk mengetahui kesesuaian data dengan parameter yang ditetapkan menggunakan metode *artificial intelligence* [16].

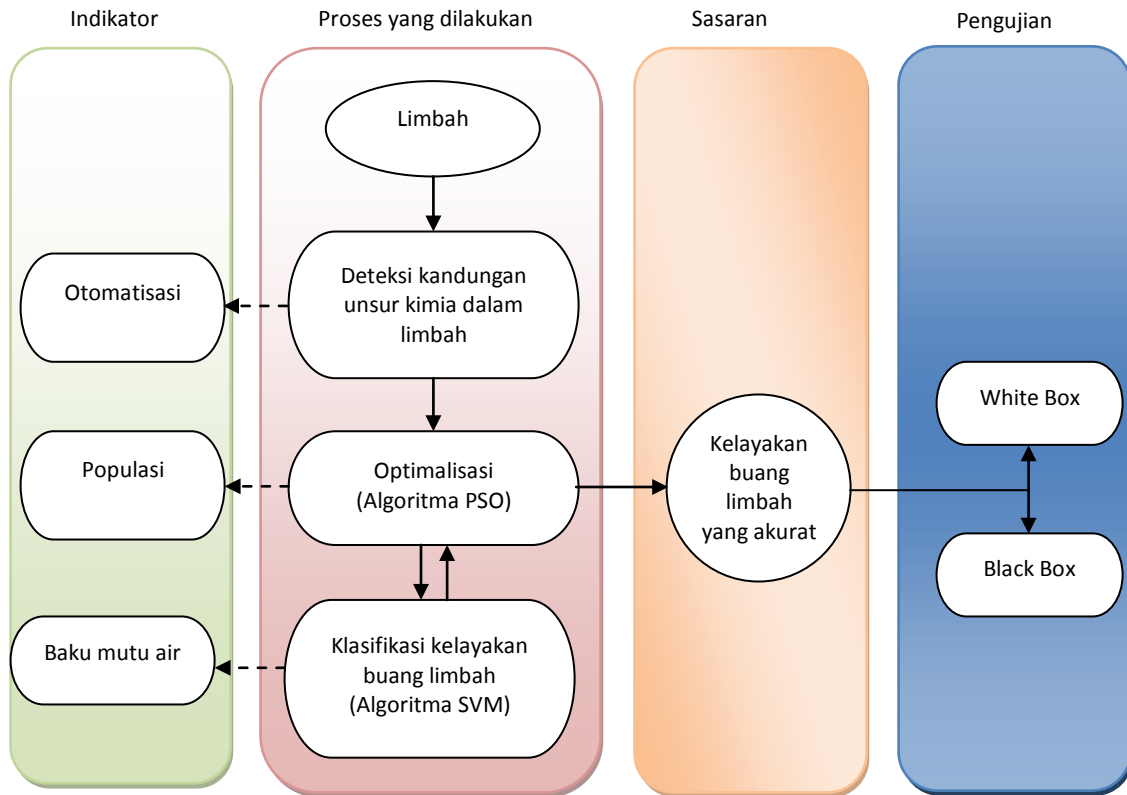
3. **“A PSO-SVM Lips Recognition Method Based on Active Basis Model”**, oleh Chih-Yu Hsu, Yung-Chih Chen, Min-chian Tsai, Department of Information and Communication Engineering, Chaoyang University of Technology, Department of Radiation Oology, China Medical University Hospital . Penelitian ini mengusulkan sebuah metode



Active Basis Model (ABM) untuk pengenalan bentuk bibir. Dengan menggunakan algoritma PSO dan SVM dapat menggolongkan bentuk-bentuk bibir dengan 4 tahapan, yaitu mengenal bentuk gambar bibir yang terbuka dan tertutup, memperoleh bentuk yang catat dari pengenalan gambar, membedakan bentuk-bentuk bibir dan yang terakhir mengklasifikasikan bentuk bibir.

Hubungan dengan penelitian yang akan dilakukan adalah menggunakan algoritma PSO-SVM untuk melakukan optimasi dan klasifikasi [17].

## II.6. Kerangka Pikir



## **BAB III**

### **METODE PENELITIAN**

#### **III.1. Waktu Penelitian**

Penelitian ini dilakukan selama empat bulan mulai Februari 2013 sampai dengan Mei 2013.

#### **III.2. Tahapan Penelitian**

Tahapan penelitian dilakukan sesuai dengan metode pendekatan yang telah dikemukakan sebelumnya dengan tahap-tahapan sebagai berikut :

##### **1. Tahap Identifikasi Masalah.**

Pada tahap ini dilakukan identifikasi masalah untuk memahami secara menyeluruh permasalahan yang ada sehingga dapat diperoleh pemecahan masalah ketika sistem baru diterapkan. Adapun masalah yang teridentifikasi :

- a. Bagaimana mengetahui kadar unsur yang terkandung dalam air limbah secara real time dan kontinu sehingga buka tutup pintu pembuangan air limbah dapat dilakukan secara otomatis.
- b. Bagaimana merancang sistem yang dapat memonitoring kadar unsur air limbah untuk pengaturan kadar unsur tersebut sesuai standar kelayakan

buang sehingga pembuangan air limbah dapat dilakukan sesuai waktu yang diinginkan.

- c. Bagaimana memiliki database terhadap seluruh aktifitas proses pembuangan air limbah di pertambangan nikel.

## **2. Tahapan Perencanaan**

Pada tahap ini dilakukan penyusunan konsep sistem yang diawali dengan analisis secara teoritis terhadap parameter unsur-unsur yang terkandung dalam air limbah pertambangan nikel, standarisasi pengukuran/penghitungan kadar unsur dalam air limbah dan metode penghitungan (proses) yang akan digunakan untuk otomatisasi kelayakan buang limbah, maka rencana penelitian disusun sebagai berikut :

- a. Membuat program aplikasi sebagai simulator yang mendeteksi kadar kandungan unsur dalam air limbah yang bekerja secara real time memberikan masukan data pada program utama. Program aplikasi ini juga mensimulasikan buka tutup pintu pembuangan air limbah sesuai dengan masukan yang diterima dari program utama.
- b. Membuat program aplikasi (program utama) untuk menganalisis kelayakan buang limbah berdasarkan masukan yang diterima dari simulator. Hasil analisis kemudian dikirim kembali ke program simulator sebagai masukan untuk membuka atau menutup pintu pembuangan air limbah.

### **3. Tahap Pelaksanaan**

Pada tahap ini dilakukan pembangunan sistem sesuai dengan rencana pada tahap sebelumnya. Pembangunan sistem ini didasari pada desain kebutuhan sistem. Desain sistem tersebut adalah :

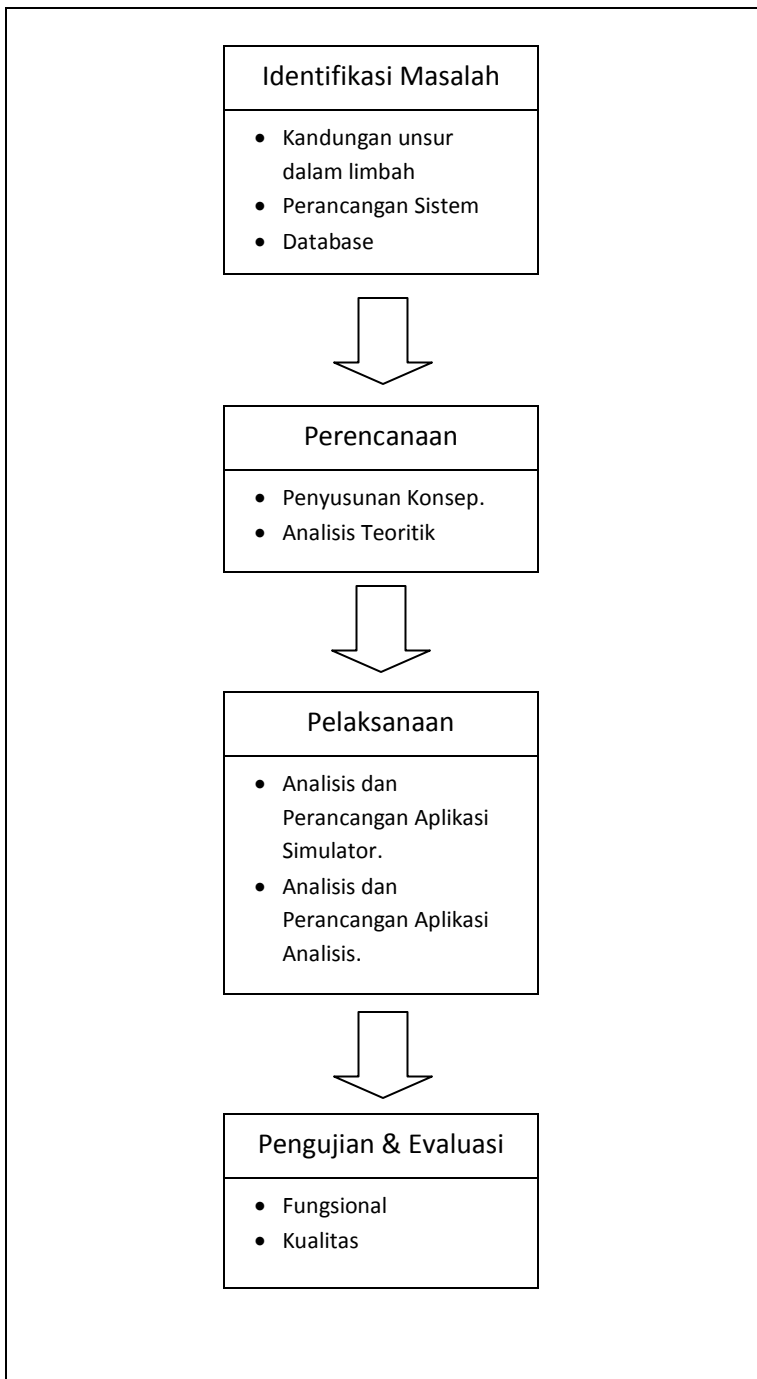
- a. Use Case Diagram. Menjelaskan setiap fungsi yang ada dalam aplikasi simulator dan aplikasi analisis (program utama).
- b. Activity Diagram. Menggambarkan urutan/tahapan aktifitas proses pada aplikasi simulator dan aplikasi analisis.
- c. Sequence Diagram. Menggambarkan interaksi objek yang disusun berdasarkan urutan waktu dari aplikasi simulator dan aplikasi analisis.
- d. Perancangan sistem meliputi bagan rancangan sistem dan flowchart metode yang diterapkan pada aplikasi analisis.

### **4. Tahap Pengujian dan Evaluasi Sistem**

Pada tahap ini dilakukan dua model pegujian yaitu :

- a. Pengujian fungsional mencakup, aplikasi simulator saat membangkitkan nilai kadar kandungan unsur dalam air limbah dan aplikasi analisis saat melakukan proses penentuan kelayakan buang limbah berdasarkan metode yang digunakan.
- b. Pengujian kualitas sistem mencakup ;

- Pengukuran komunikasi data antara aplikasi simulator dan aplikasi analisis mencakup integritas data dan ketepatan waktu pembangkitan nilai dan pemrosesan nilai.
- Evaluasi interface.



Gambar 3-1. Bagan Tahapan Penelitian

### **III.3. Metode Pendekatan**

Berdasarkan sifat permasalahan dari objek yang diteliti, maka digunakan metode pendekatan studi literature (library research) dan studi lapangan (field research) terhadap proses diagnosis awal dan perencanaan. Studi literatur meliputi pemahaman terhadap standar baku mutu air limbah pertambangan nikel sebagaimana yang telah diatur pada peraturan Menteri Negara Lingkungan Hidup No. 9 Tahun 2006, serta Standar Nasional Indonesia (SNI) tentang tata cara pengukuran unsur-unsur yang terkandung dalam air limbah pertambangan nikel. Studi lapangan dilakukan untuk mengamati metode penentuan kelayakan buang limbah yang dilakukan di PT. Vale Indonesia Tbk.

Dari dua metode pendekatan ini kemudian dilakukan analisis dan perancangan sistem menggunakan pendekatan metode proses desain arsitektur yaitu mendeskripsikan sistem, lingkungan sistem, serta hubungan antara sistem dengan lingkungannya sesuai dengan metode penelitian yang diimplementasikan, yaitu Partikel Swarm Optimization (PSO) dan Support Vector Machine (SVM).

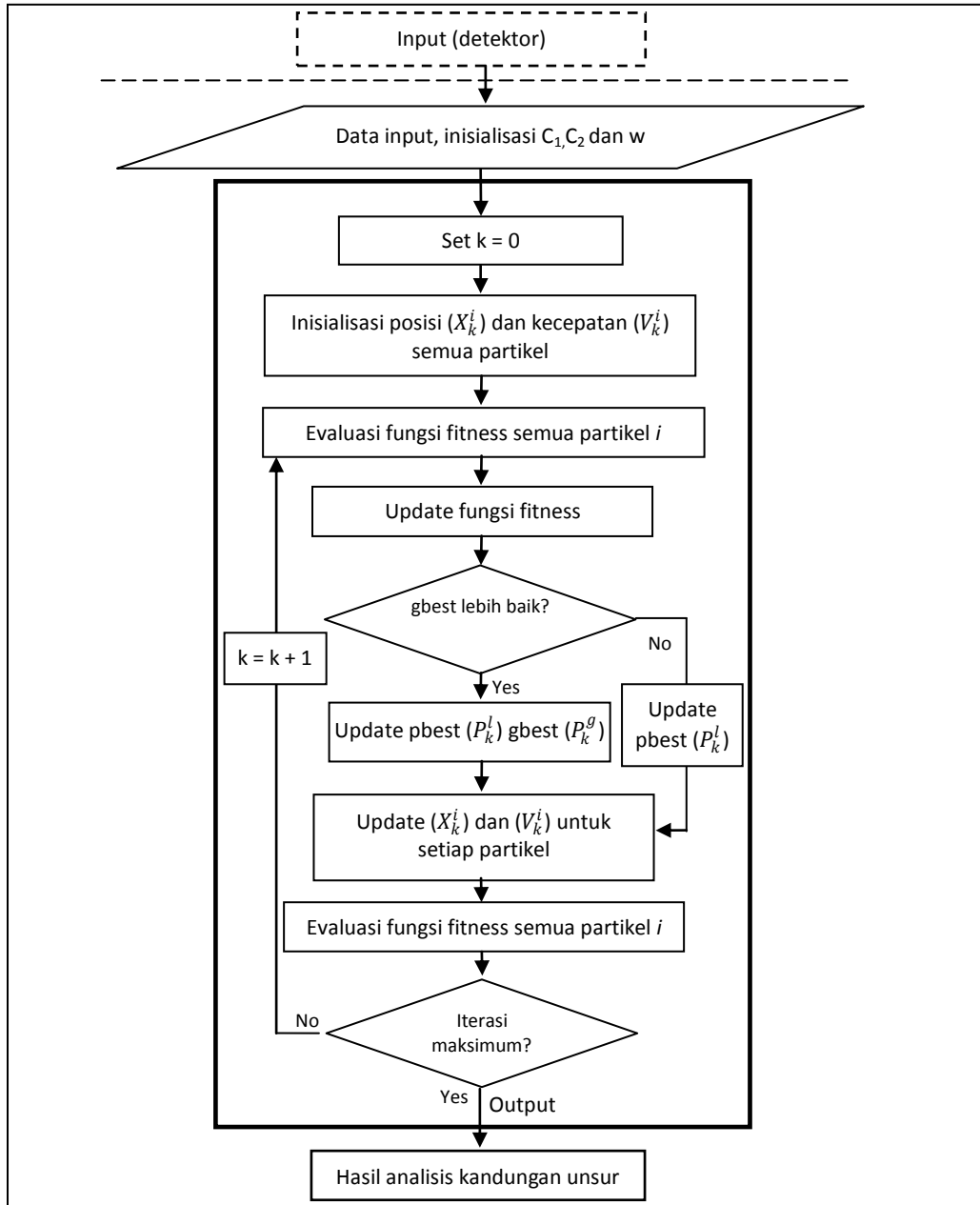
Untuk pengujian dan evaluasi sistem digunakan beberapa tahapan pengujian, yaitu pengujian fungsional sistem dilakukan dengan metode simulasi melalui skenario pengujian, dan pengujian kualitas sistem mencakup pengujian komunikasi dan pemrosesan data serta evaluasi



interface melalui metode kualitatif dengan memilih 10 orang yang diwawancarai sesuai dengan kebutuhan penggunaan sistem (operator komputer).

### III.4. Implementasi Algoritma

#### a. Algoritma PSO



Gambar 3-2. Implementasi Algoritma PSO

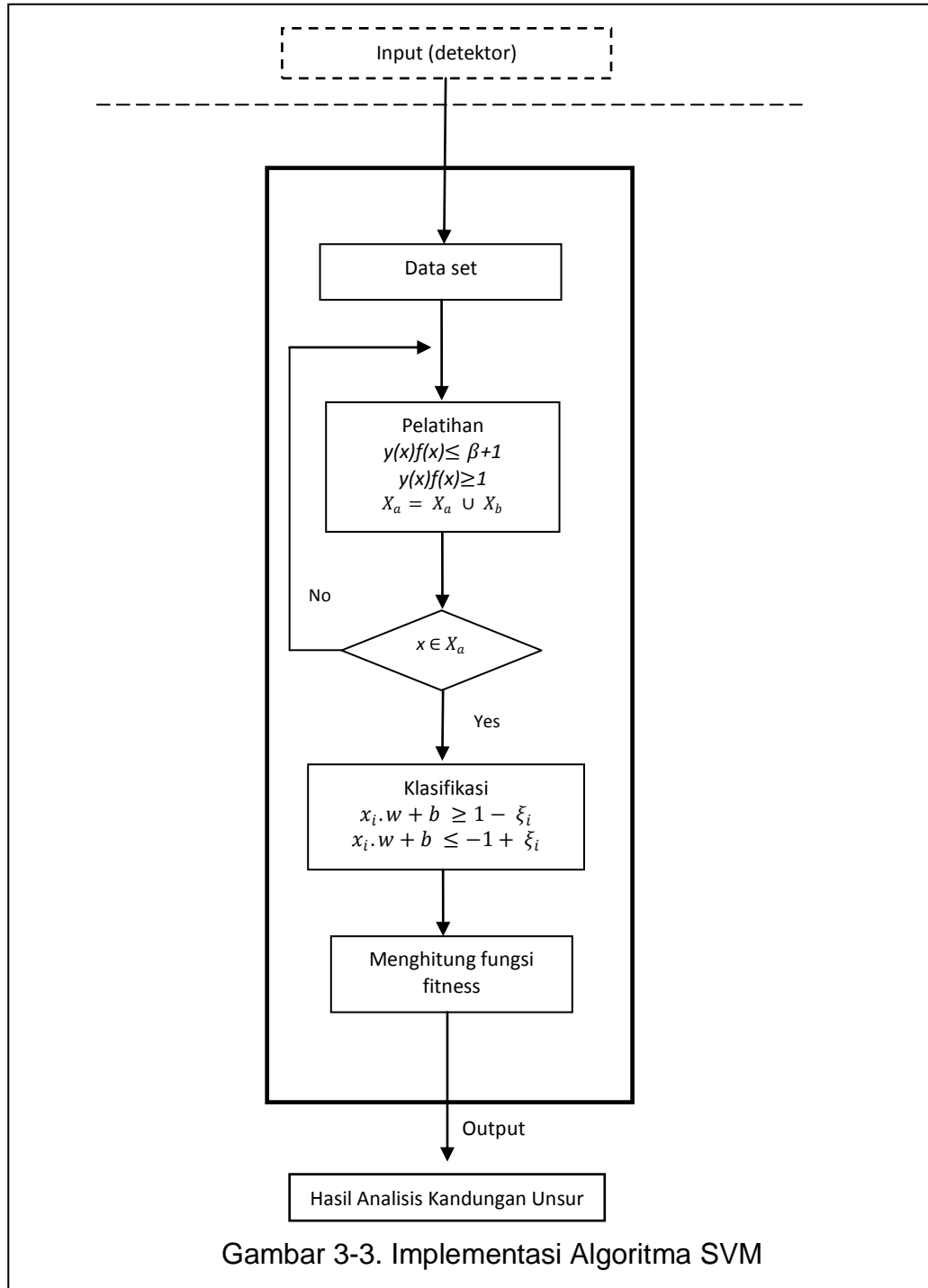
Tahapan implementasi algoritma PSO pada Gambar 3-2 adalah sebagai berikut :

1. Input data berasal dari detektor yang mendeteksi kadar kandungan limbah terukur, yaitu : pH, TSS, Cu, Cd, Zn, Pb, Ni, Cr<sup>(6+)</sup>, Cr total, Fe, dan Co. Proses deteksi menggunakan sensor diluar lingkup penelitian dan disimulasikan melalui aplikasi simulator.
2. Inisialisasi kadar kandungan limbah yang terinput secara realtime dari peralatan sensor, jumlah iterasi, inisialisasi nilai  $C_1$ ,  $C_2$ , dan inertia weight ( $w$ ).
3. Set waktu ( $k$ ) = 0. Berjalan pada algoritma PSO.
4. Inisialisasi posisi ( $X_k^i$ ) dan kecepatan/velocity ( $V_k^i$ ) dari kumpulan partikel yang dibangkitkan secara acak menggunakan batas atas ( $X_{max}$ ) dan batas bawah ( $X_{min}$ ).
5. Evaluasi nilai fitness untuk semua partikel  $i$  terhadap posisi ( $X_k^i$ ) dan kecepatan/velocity ( $V_k^i$ ).
6. Update fungsi fitness untuk  $k = k + 1$ .
7. Jika  $gbest$  lebih baik dari iterasi sebelumnya maka Update  $pbest$  ( $P_k^l$ ) dan  $gbest$  ( $P_k^g$ ), jika tidak maka Update  $pbest$  ( $P_k^l$ ).
8. Update posisi ( $X_k^i$ ) dan kecepatan/velocity ( $V_k^i$ ) berdasarkan  $pbest$  dan  $gbest$  yang baru.
9. Evaluasi nilai fitness untuk semua partikel  $i$ .

10. Jika iterasi maksimum maka diperoleh hasil analisis, jika tidak ke point

(5).

## b. Algoritma SVM



Gambar 3-3. Implementasi Algoritma SVM

Tahapan implementasi algoritma SVM pada Gambar 3-3 adalah sebagai berikut :

1. Input data berasal dari detektor yang mendeteksi kadar kandungan limbah terukur, yaitu : pH, TSS, Cu, Cd, Zn, Pb, Ni, Cr<sup>(6+)</sup>, Cr total, Fe, dan Co. Proses deteksi menggunakan sensor diluar lingkup penelitian dan disimulasikan melalui aplikasi simulator.
2. Inisialisasi data set untuk  $y_i = + 1$  dan  $y_i = - 1$ .
3. Lakukan pelatihan untuk semua data set sehingga diperoleh nilai  $w$  dan  $b$  dari nilai margin maksimal antara *support vector* dan *hyperplane*.
4. Lakukan klasifikasi terhadap data uji berdasarkan hasil point (3).
5. Hitung nilai fitness untuk memperoleh hasil analisis.

### **c. Algoritma PSO dan SVM**

Tahapan implementasi algoritma PSO-SVM pada Gambar 3-4 adalah sebagai berikut :

1. Input data berasal dari detektor yang mendeteksi kadar kandungan limbah terukur, yaitu : pH, TSS, Cu, Cd, Zn, Pb, Ni, Cr<sup>(6+)</sup>, Cr total, Fe, dan Co. Proses deteksi menggunakan sensor diluar lingkup penelitian dan disimulasikan melalui aplikasi simulator.

2. Inisialisasi kadar kandungan limbah yang terinput secara realtime dari peralatan sensor, jumlah iterasi, inisialisasi nilai  $C_1$ ,  $C_2$ , dan inertia weight ( $w$ ).
3. Set waktu ( $k$ ) = 0. Berjalan pada algoritma PSO.
4. Inisialisasi posisi ( $X_k^i$ ) dan kecepatan/velocity ( $V_k^i$ ) dari kumpulan partikel yang dibangkitkan secara acak menggunakan batas atas ( $X_{max}$ ) dan batas bawah ( $X_{min}$ ) melalui persamaan :

$$X_0^i = X_{min} + rand(X_{max} - X_{min}) \quad (3.1)$$

$$V_0^i = X_{min} + rand(X_{max} - X_{min}) \quad (3.2)$$

Dengan proses inisialisasi ini kumpulan partikel menjadi terdistribusi secara acak dalam bentuk vektor pada *design space*, dimana dimensi vektor disimbolkan sebagai  $n$ ,

$$X_k^i = (X_k^{i1}, X_k^{i2}, \dots, X_k^{in})^T, \quad (3.3)$$

$$V_k^i = (V_k^{i1}, V_k^{i2}, \dots, V_k^{in})^T. \quad (3.4)$$

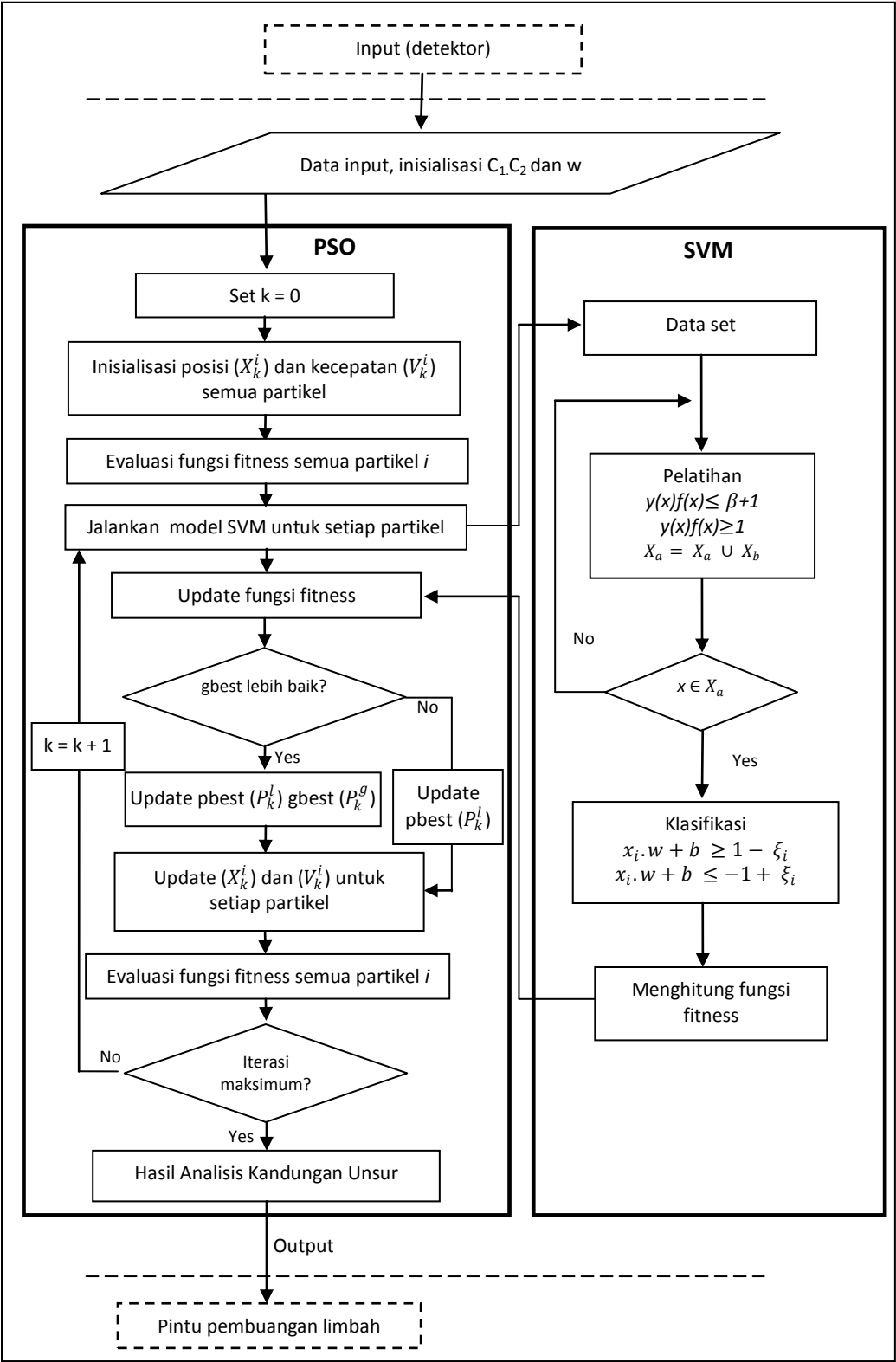
5. Evaluasi nilai fitness untuk semua partikel  $i$ .
6. Menjalankan model SVM untuk semua partikel yang telah terinisialisasi, partikel tersebut merupakan dataset (input) pada SVM.
7. Menjalankan proses pelatihan untuk setiap data yang telah terseleksi. Parameter pada proses pelatihan ini merupakan masukan yang berasal dari baku mutu air limbah yang telah ditentukan dan kemudian akan menjadi garis pemisah (*hyperplane*) untuk memisahkan 2 buah kelas

sebagai  $y(x)f(x)=1$ . Kandidat *support vector* diperoleh dari  $y(x)f(x)\leq \beta+1$  dan  $y(x)f(x)\geq 1$  ( $\beta$  adalah parameter yang ditentukan pengguna).

8. Jika terdapat data pelatihan baru dan model hasil pelatihan sebelumnya akan diperbaharui, maka dari data pelatihan lama cukup diambil sebagian data sebagai kandidat *support vector*.
9. Data hasil pelatihan diklasifikasikan dengan  $x_i \cdot w + b \geq 1$  untuk kelas 1, dan  $x_i \cdot w + b \leq -1$  untuk kelas 2.
10. PSO melakukan update fungsi fitness. Nilai fitness diperoleh dari hasil klasifikasi SVM.
11. Jika nilai fitness lebih baik dari gbest sebelumnya, maka dilakukan update pbest dan gbest, tetapi jika nilai fitness tidak lebih baik dari gbest maka hanya update pbest yang dilakukan.
12. Berdasarkan pbest (posisi terbaik partikel) dan gbest (posisi terbaik dalam populasi) yang baru maka dilakukan update posisi ( $X_{k+1}^i$ ) dan kecepatan ( $V_{k+1}^i$ ) untuk semua partikel dengan rumus :
$$V_{k+1}^i = w \cdot V_k^i + c_1 \text{rnd}(P_k^i - X_k^i) + c_2 \text{rnd}(P_k^g - X_k^i) \quad (3.6)$$
$$X_{k+1}^i = X_k^i + V_{k+1}^i \quad (3.7)$$
13. Melakukan evaluasi fungsi fitness terbaru semua partikel untuk iterasi berikutnya.
14. Kembali ke point 6 untuk proses pada modul SVM untuk iterasi berikutnya, jika iterasi sudah maksimum, maka algoritma PSO-SVM



selesai dan menghasilkan output apakah pembuangan limbah dilakukan atau tidak.



Gambar 3-4. Implementasi Algoritma PSO dan SVM

#### d. Ilustrasi Algoritma Particle Swarm Optimization

Misalkan kita mempunyai persoalan optimasi dengan satu variabel sebagai berikut :

$$f(x) = (100 - x)^2$$

$$\text{dimana } 60 \leq x \leq 120$$

1. Tentukan jumlah partikel  $N = 4$  Tentukan populasi awal secara random, misalkan di dapat :  
 $x_1(0) = 80, x_2(0) = 90, x_3(0) = 110, x_4(0) = 75.$
2. Evaluasi nilai fungsi tujuan untuk setiap partikel  $x_j(0)$  untuk  $j = 1, 2, 3, 4$  dan nyatakan dengan :  
 $f_1 = f(80) = 400, f_2 = f(90) = 100, f_3 = f(110) = 100, f_4 = f(75) = 625.$
3. Tentukan kecepatan awal  $v_1(0) = v_2(0) = v_3(0) = v_4(0) = 0$ . Tentukan iterasi  $i = 1$ ; Lalu ke langkah nomer 4.
4. Evaluasi nilai fungsi tujuan untuk setiap partikel  $x_j(0)$  untuk  $j = 1, 2, 3, 4$  dan nyatakan dengan :  
 $f_1 = f(80) = 400, f_2 = f(90) = 100, f_3 = f(110) = 100, f_4 = f(75) = 625.$
5. Tentukan kecepatan awal  $v_1(0) = v_2(0) = v_3(0) = v_4(0) = 0$ . Tentukan iterasi  $i = 1$ ; Lalu ke langkah nomer 4.
6. Temukan  $Pbest_1 = 80, Pbest_2 = 90, Pbest_3 = 110, Pbest_4 = 75, Gbest = 90.$

Hitung  $v(j)$  dengan  $c_1 = c_2 = 1$ . Misalkan nilai random yang didapat,  $r_1 = 0.4$ ,  $r_2 = 0.5$ , dengan rumus  $V_j(i) = V_j(i - 1) + c_1 r_1 [P_{best,j} - x_j(i - 1)] + c_2 r_2 [G_{best} - x_j(i - 1)]$  diperoleh

$$v_1(1) = 0 + 0.4(80 - 80) + 0.5(90 - 80) = 5$$

$$v_2(1) = 0 + 0.4(90 - 90) + 0.5(90 - 90) = 0$$

$$v_3(1) = 0 + 0.4(110 - 110) + 0.5(90 - 110) = -10$$

$$v_4(1) = 0 + 0.4(75 - 75) + 0.5(90 - 75) = 7.5$$

Sedangkan untuk nilai  $x$  adalah

$$x_1(1) = 80 + 5 = 85$$

$$x_2(1) = 90 + 0 = 90$$

$$x_3(1) = 110 - 10 = 100$$

$$x_4(1) = 75 + 7.5 = 82.5$$

7. Evaluasi nilai fungsi tujuan sekarang pada partikel  $x_j(1)$ ,

$$f_1(1) = f(85) = 225,$$

$$f_2(1) = f(90) = 100,$$

$$f_3(1) = f(100) = 0,$$

$$f_4(1) = f(82.5) = 306.25.$$

Sedangkan pada iterasi sebelumnya kita dapatkan

$$f_1(0) = f(80) = 400,$$

$$f_2(0) = f(90) = 100,$$

$$f_3(0) = f(110) = 100,$$

$$f_4(0) = f(75) = 625.$$

Nilai dari  $f$  dari iterasi sebelumnya tidak ada yang lebih baik sehingga  $Pbest$  untuk masing-masing partikel sama dengan nilai  $x$ -nya.  $Gbest = 100$

Cek apakah solusi  $x$  sudah konvergen, dimana nilai  $x$  saling dekat. Jika tidak, tingkatkan ke iterasi berikutnya  $i = 2$ .

Lanjutkan ke langkah 4.

1.  $Pbest1 = 85, Pbest2 = 90, Pbest3 = 100, Pbest4 = 82.5, Gbest = 100$

Hitung kecepatan baru dengan  $r1 = 0.3$  dan  $r = 0.6$  ( ini hanya sekedar contoh untuk menjelaskan penghitungan, dalam implementasi angka ini dibangkitkan secara acak).

$$v1(2) = 5 + 0.3(85 - 85) + 0.6(100 - 85) = 14$$

$$v2(2) = 0 + 0.3(90 - 90) + 0.6(100 - 90) = 6.$$

$$v3(2) = -10 + 0.3(100 - 100) + 0.6(100 - 100) = -10$$

$$v4(2) = 7.5 + 0.3(82.5 - 82.5) + 0.6(100 - 82.5) = 18$$

Sedangkan untuk nilai  $x$  adalah

$$x1(2) = 85 + 14 = 99$$

$$x2(2) = 90 + 6 = 96$$

$$x3(2) = 100 - 10 = 90$$

$$x4(2) = 82.5 + 18 = 100.5$$

2. Evaluasi nilai fungsi tujuan sekarang pada partikel  $xj(2)$ ,

$$f1(2) = f(99) = 1,$$

$$f2(2) = f(96) = 16,$$

$$f_3(2) = f(90) = 100$$

$$f_4(2) = f(100.5) = 0.25.$$

## **BAB IV**

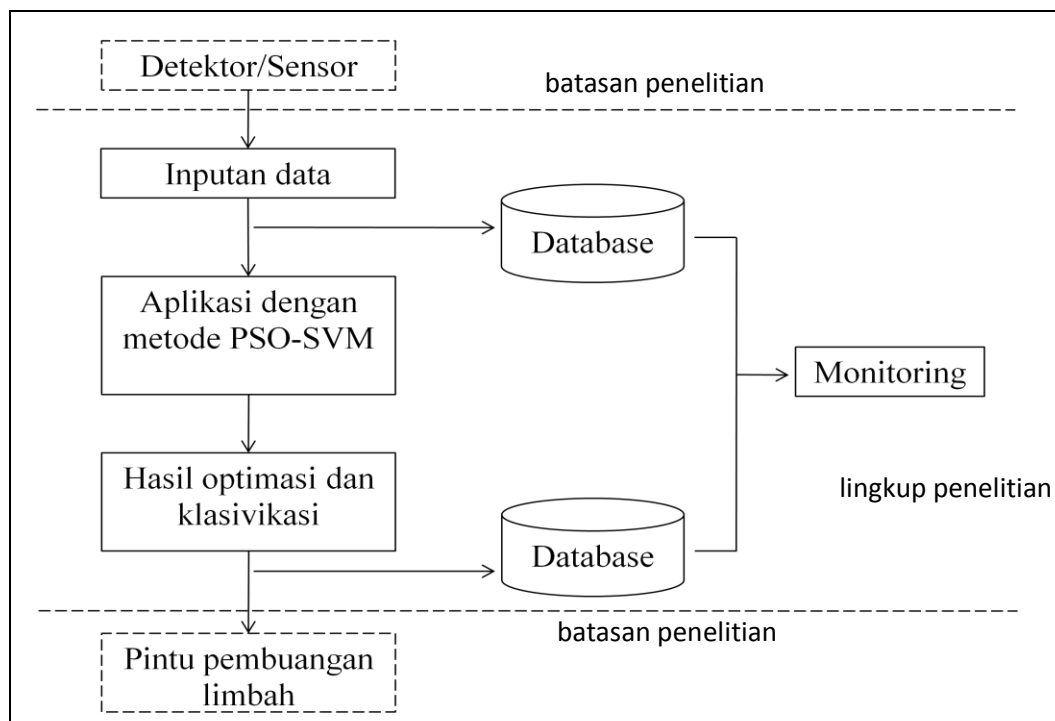
### **HASIL DAN PEMBAHASAN**

Otomatisasi pembuangan limbah pertambangan nikel merupakan ide dasar dilakukannya penelitian ini, sebab sampai saat ini proses pembuangan limbah di pertambangan nikel belum dapat dilakukan secara otomatis (*direct reading*). Proses pembuangan limbah masih melalui analisis laboratorium sebagai acuan kelayakan buang limbah. Kondisi lapangan tempat penelitian ini dilakukan masih menggunakan metode yang sama, dimana peralatan otomatisasi dilakukan hanya sebagai alat bantu awal untuk akurasi atau kecepatan dalam melakukan analisis laboratorium, dimana proses analisis laboratorium dimulai dari pengambilan sampel air limbah setiap 2 jam, kemudian dilakukan analisis untuk mengetahui apakah limbah dalam penampungan sudah dapat dibuang ke badan air sesuai standar baku mutu air limbah yang ditetapkan pemerintah. Kesulitan mendasar adalah belum adanya peralatan yang dapat membaca kandungan unsur logam dalam air limbah secara langsung.

Berdasarkan kondisi di atas penelitian ini dilakukan sebagai langkah awal untuk menemukan metode baru pembuangan limbah pertambangan nikel secara langsung (otomatis). Pada penelitian ini menitik beratkan pada analisis kandungan unsur yang telah dideteksi oleh alat sensor menggunakan

algoritma Particle Swarm Optimization (PSO) dan Support Vector Machine (SVM).

Pada penelitian ini implementasi algoritma PSO dan SVM untuk menghasilkan otomatisasi pembuangan limbah pertambangan nikel dijelaskan sebagai berikut :



Gambar 4-1. Bagan Lingkup Kerja Sistem Otomatisasi Kelayakan Buang Limbah Pertambangan Nikel Menggunakan PSO dan SVM.

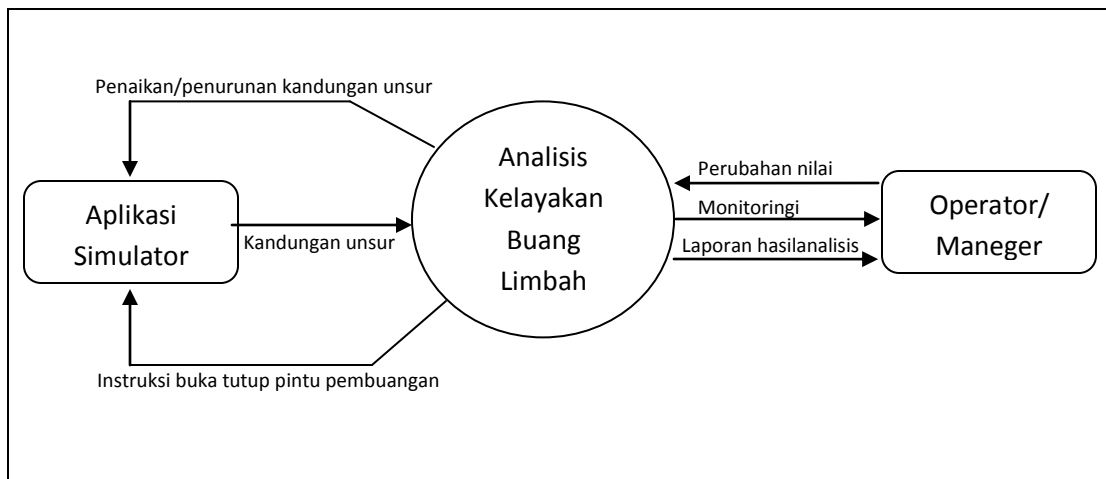
Dari gambar di atas menjelaskan batasan penelitian dan ruang lingkup penelitian yang dilakukan. Detektor/sensor disimulasikan dalam bentuk



aplikasi untuk memperoleh data kandungan unsur dalam limbah. Data yang dibangkitkan oleh aplikasi simulasi detektor (aplikasi simulator) kemudian dianalisis. Hasil analisis yang menggunakan PSO dan SVM kemudian dikirim kembali ke aplikasi simulator dalam bentuk instruksi membuka atau menutup pintu pembuangan limbah.

#### IV.1. Analisis Sistem

Pada Bab III sekilas telah dibahas analisis dan perancangan sistem yang digunakan pada penelitian ini. Dan untuk lebih jelasnya ditampilkan diagram konteks dan Diagram Aliran Data (DFD) level 0 berikut ini sebagai analisis dari sistem yang dibuat.

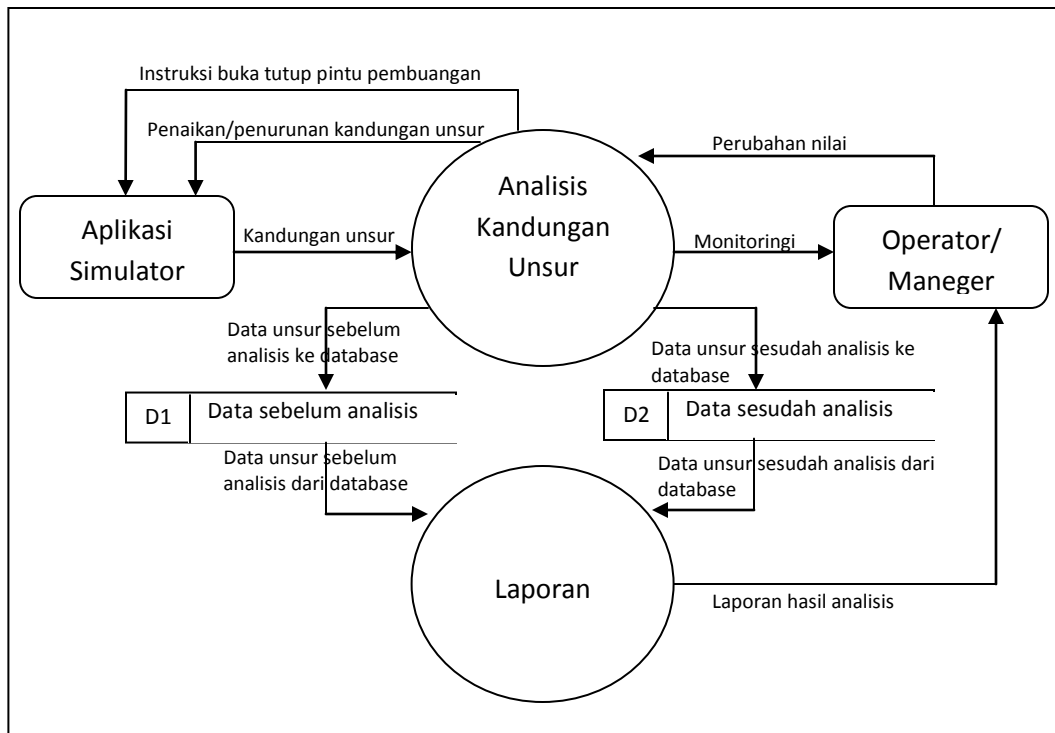


Gambar 4-2. Diagram Konteks Aplikasi Analisis Pembuangan Limbah

Pada diagram konteks di atas (Gambar 4-2.) terdapat dua entitas yang berhubungan dengan sistem melalui enam aliran data. Entitas dan aliran data tersebut adalah :

1. Entitas aplikasi simulator yang terdiri dari tiga aliran data, yaitu :
  - a. Aliran data kandungan unsur yang dibangkitkan oleh aplikasi simulator dan menjadi masukan pada sistem analisis.
  - b. Aliran data perubahan kandungan unsur (bernilai naik/turun) yang merupakan keluaran dari sistem.
  - c. Aliran data instruksi buka tutup pintu pembuangan limbah yang merupakan keluaran dari sistem.
2. Entitas operator/manager yang terdiri dari dua aliran data, yaitu :
  - a. Aliran data perubahan nilai oleh operator/manager untuk menurunkan atau menaikkan nilai kandungan unsur untuk memperoleh kondisi kelayakan buang limbah. Aliran data ini merupakan masukan sistem.
  - b. Aliran data monitoring untuk melihat hasil analisis secara realtime. Aliran data ini merupakan keluaran dari sistem.
  - c. Aliran data laporan untuk operator/manejer tentang hasil analisis pembuangan limbah dan merupakan keluaran dari sistem.

Dari diagram konteks di atas dapat diuraikan secara rinci melalui diagram aliran data (DFD) level 0 untuk melihat bagian-bagian detail dalam sistem analisis yang bekerja untuk menentukan kelayakan buang limbah. DFD level 0 disajikan sebagai berikut :



Gambar 4-3. DFD Level 0 Aplikasi Analisis Pembuangan Limbah

Pada DFD level 0 (Gambar 4-3) menjelaskan secara rinci bagian dari aplikasi analisis pembuangan limbah yang terdiri dari dua modul, yaitu modul analisis kandungan unsur dan modul laporan. Dimana tiga aliran data yang berhubungan dengan entitas aplikasi simulator masuk dan keluar pada modul analisis kandungan unsur. Demikian pula dua aliran data yang berhubungan dengan entitas operator/manager keluar dan masuk pada modul analisis kandungan unsur. Aliran data yang keluar dari sistem melalui modul laporan hanya satu aliran data, yaitu aliran data laporan hasil analisis menuju entitas operator/manager.

DFD level 0 juga memperlihatkan adanya database yang terdiri dari dua tabel, yaitu tabel data sebelum analisis dan data sesudah analisis. Hal ini menjelaskan bahwa data yang dibangkitkan dari aplikasi simulator akan tersimpan sebelum dilakukan analisis dan hasil analisis menggunakan algoritma PSO dan SVM juga tersimpan ke dalam database. Hal ini bertujuan agar supaya ketika operator atau manager melakukan analisis dari laporan, dapat terlihat kondisi menyeluruh kandungan unsur baik sebelum dianalisis maupun sesudah dianalisis.

## **IV.2. Perancangan Aplikasi**

Untuk mencapai tujuan penelitian maka dibuat dua aplikasi yaitu aplikasi analisis untuk melakukan analisis kandungan unsur dalam limbah menggunakan algoritma PSO dan SVM dan aplikasi kedua adalah aplikasi simulator yang menjalankan fungsi detektor/sensor limbah dan menjalankan proses buka tutup pintu pembuangan limbah.

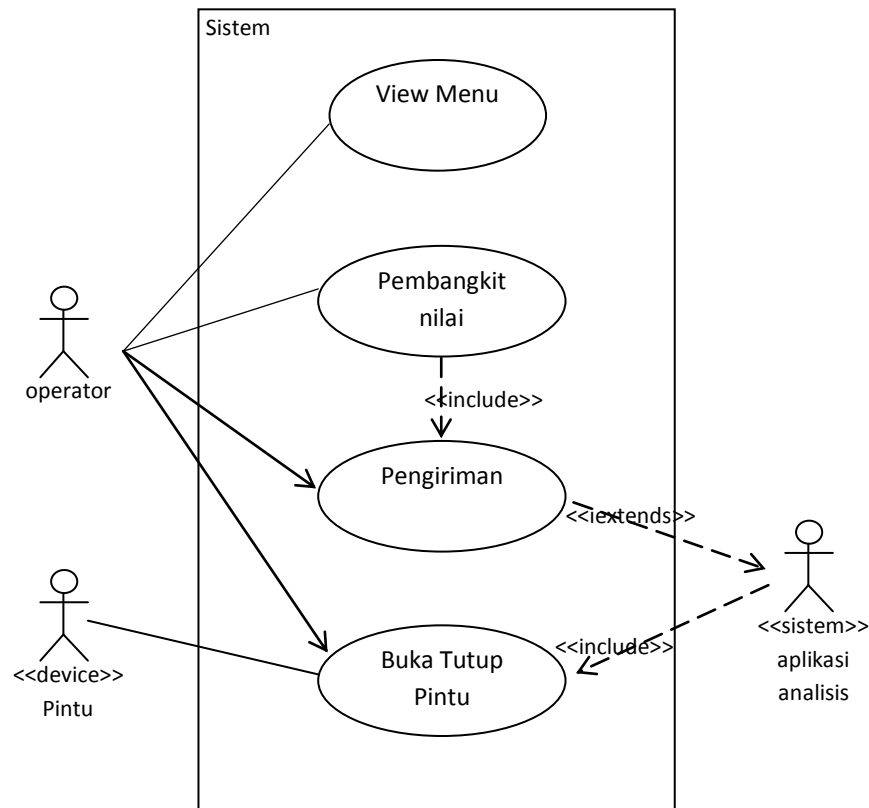
### **1. Perancangan Aplikasi Simulator**

#### **1. 1. Use Case Diagram Aplikasi Simulator**

Use Case aplikasi simulator adalah diagram yang digunakan menggambarkan fungsionalitas sistem dari aplikasi simulator dan

bagaimana aplikasi tersebut tersebut berinteraksi dengan dunia luar.

Diagram use case aplikasi simulator adalah sebagai berikut :



Gambar 4-4. Use Case Diagram Aplikasi Simulator

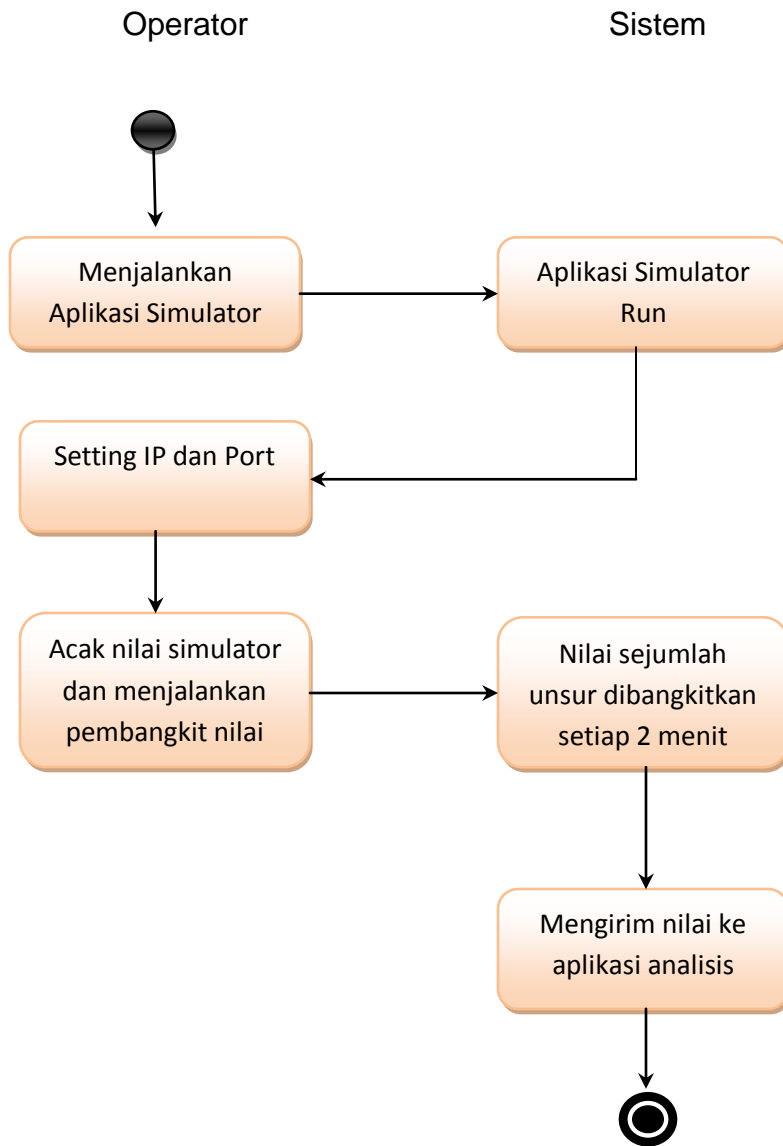
Use Case diagram aplikasi simulator menggambarkan interaksi antara aktor-aktor diluar sistem dengan case-case yang berada didalam sistem terdiri dari dua aktor yaitu aktor operator dan aktor sistem (aplikasi analisis) dan empat case, yaitu case view menu, case pembangkit nilai, case pengiriman data dan case buka tutup pintu pembuangan limbah.

- Aktor operator berinteraksi dengan sistem secara aktif pada dua case yaitu case view menu dan pembangkit nilai dan secara pasif pada case pengiriman data dan case buka tutup pintu pembuangan limbah. Aktor operator berinteraksi dengan case view menu saat akan menjalankan aplikasi dan membuka menu pembangkit nilai. Aktor operator berhubungan dengan case pembangkit nilai ketika akan menjalankan simulasi deteksi kadar unsur dalam air limbah.
- Aktor sistem (aplikasi analisis) berinteraksi secara aktif dengan dua case, yaitu case pengirim data (nilai yang telah dibangkitkan) yang include dengan case pembangkit nilai, dimana pengiriman data dilakukan setelah ada nilai yang dibangkitkan sebagai simulasi dari kadar unsur yang terdeteksi. Aktor sistem berinteraksi dengan case buka tutup pintu saat aktor sistem mengirimkan informasi hasil analisis pembuangan limbah untuk dilakukan oleh aplikasi simulator melalui case buka tutup pintu.

## **1.2. Activity Diagram Aplikasi Simulator**

Activity diagram aplikasi simulator menggambarkan aliran aktivitas sistem pada aplikasi simulator berdasarkan setiap case atau beberapa case yang ada pada diagram use case aplikasi simulator.

- a. Activity Diagram pada Case View Menu, Pembangkit Nilai dan Pengiriman data.



Gambar 4-5. Activity Diagram pada Case View Menu, Pembangkit Nilai dan Pengiriman Data.

Pada Gambar 4-5. dijelaskan bahwa aktivitas sistem pada aplikasi simulator dimulai dengan operator membuka/menjalankan aplikasi simulator, setelah aplikasi terbuka, operator melakukan pengaturan alamat IP dan Port dari komputer dimana aplikasi analisis berada. Tujuannya adalah nilai yang dibangkitkan terkirim dan dapat diterima oleh aplikasi analisis yang berada dimesin yang berbeda.

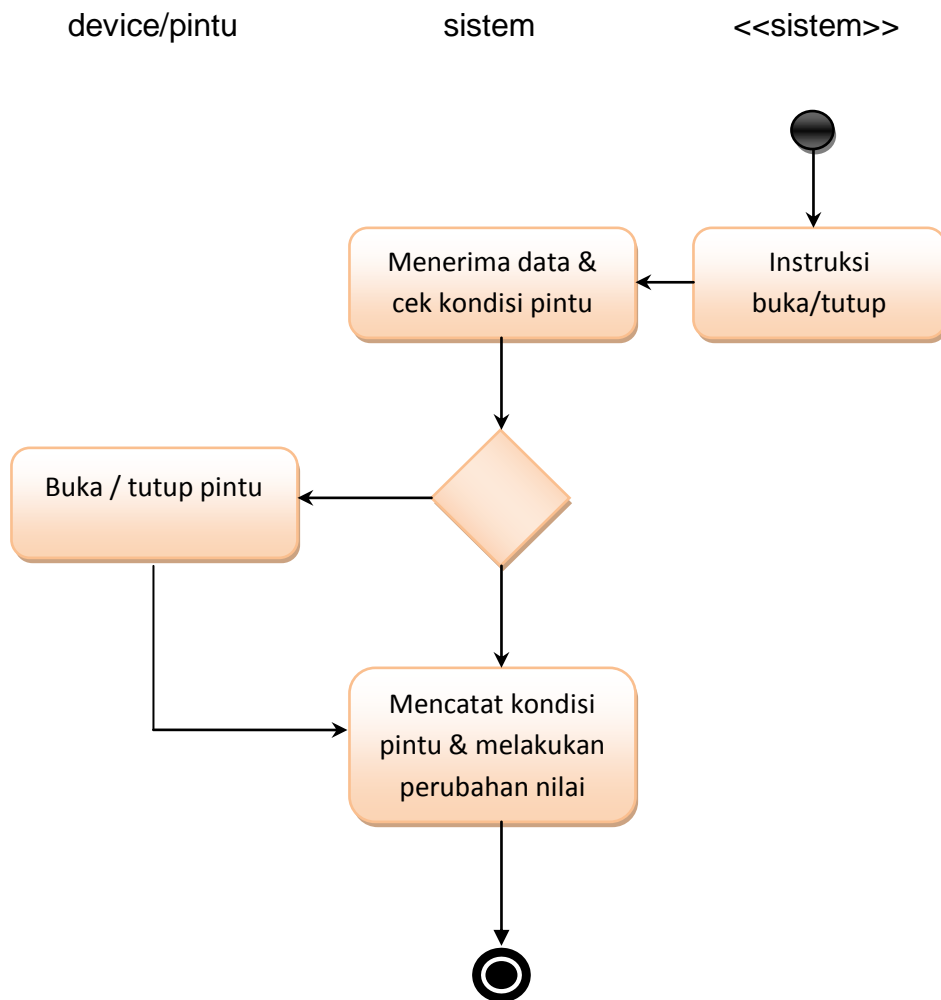
Selanjutnya operator melakukan pengacakan nilai awal dengan tujuan nilai yang akan dibangkitkan menjadi heterogen. Kondisi ini diciptakan untuk melihat sejauh mana aplikasi analisis bekerja untuk melakukan otomatisasi pembuangan dan memberikan informasi terhadap perlakuan yang harus dikerjakan jika kondisi kelayakan buang belum tercapai. Jika nilai telah dibangkitkan maka nilai tersebut akan dikirim ke aplikasi analisis. Proses pembangkitan nilai dan pengiriman data dilakukan setiap dua menit. Pengaturan waktu ini dilakukan untuk memenuhi prinsip real time pada proses otomatisasi pembuangan limbah, dimana nilai yang terkirim telah selesai diproses oleh aplikasi analisis barulah pembangkitan nilai dilakukan untuk nilai berikutnya sampai kondisi buka tutup pintu tercapai.

Pembangkitan nilai berikutnya untuk setiap unsur apakah dinaikkan atau diturunkan bergantung dari informasi yang diterima dari aplikasi analisis setelah melakukan analisis terhadap nilai pertama yang



dikirim. Hal ini dilakukan untuk untuk mencapai kondisi buka tutup pintu pembuangan limbah.

b. Activity Diagram pada Case Buka Tutup Pintu Pembuangan Limbah.



#### Gambar 4-6. Activity Diagram Case Buka Tutup Pintu Pembuangan

Pada Gambar 4-6 aktifitas dimulai dari pengiriman data hasil analisis aplikasi, selanjutnya sistem (case buka tutup pintu) menerima data berupa instruksi untuk membuka/menutup pintu pembuangan. Sistem akan melakukan pengecekan kondisi pintu, jika kondisi pintu sesuai dengan instruksi dari aplikasi analisis, maka sistem akan mencatat kondisi pintu dan melakukan perubahan nilai. Jika instruksi tidak sesuai dengan kondisi pintu, maka sistem akan memerintahkan pintu (device) sesuai dengan instruksi yang diterima dari aplikasi analisis. Selanjutnya pintu akan menginformasikan kondisinya kepada sistem dan sistem akan mencatat kondisi pintu dan melakukan perubahan nilai. Perubahan nilai yang dilakukan adalah :

- a. Jika instruksi membuka pintu sama dengan kondisi pintu maka nilai akan dinaikkan.
- b. Jika instruksi menutup pintu sama dengan kondisi pintu maka nilai akan diturunkan.
- c. Jika instruksi membuka tidak sama dengan kondisi pintu maka pintu akan dibuka dan nilai akan dinaikkan.
- d. Jika instruksi menutup tidak sama dengan kondisi pintu maka pintu akan ditutup dan nilai akan diturunkan.

Nilai yang dinaikan atau diturunkan setelah aktivitas buka tutup pintu pada case ini akan menjadi acuan pada case pembangkit nilai untuk pembangkitan nilai berikutnya. Sehingga kondisi simulator ini akan bekerja sesuai kondisi lapangan dimana aktivitas buka tutup pintu pembuangan limbah ditentukan oleh kadar unsur yang terkandung dalam air limbah. Pada case ini percepatan dan perlambatan menaikkan atau menurunkan nilai dapat dilakukan jika hal tersebut dikehendaki aplikasi analisis sebagai hasil dari analisis nilai yang terkirim. Hal ini bertujuan untuk mempercepat atau memperlambat aktivitas buka tutup pintu sesuai kebutuhan lapangan.

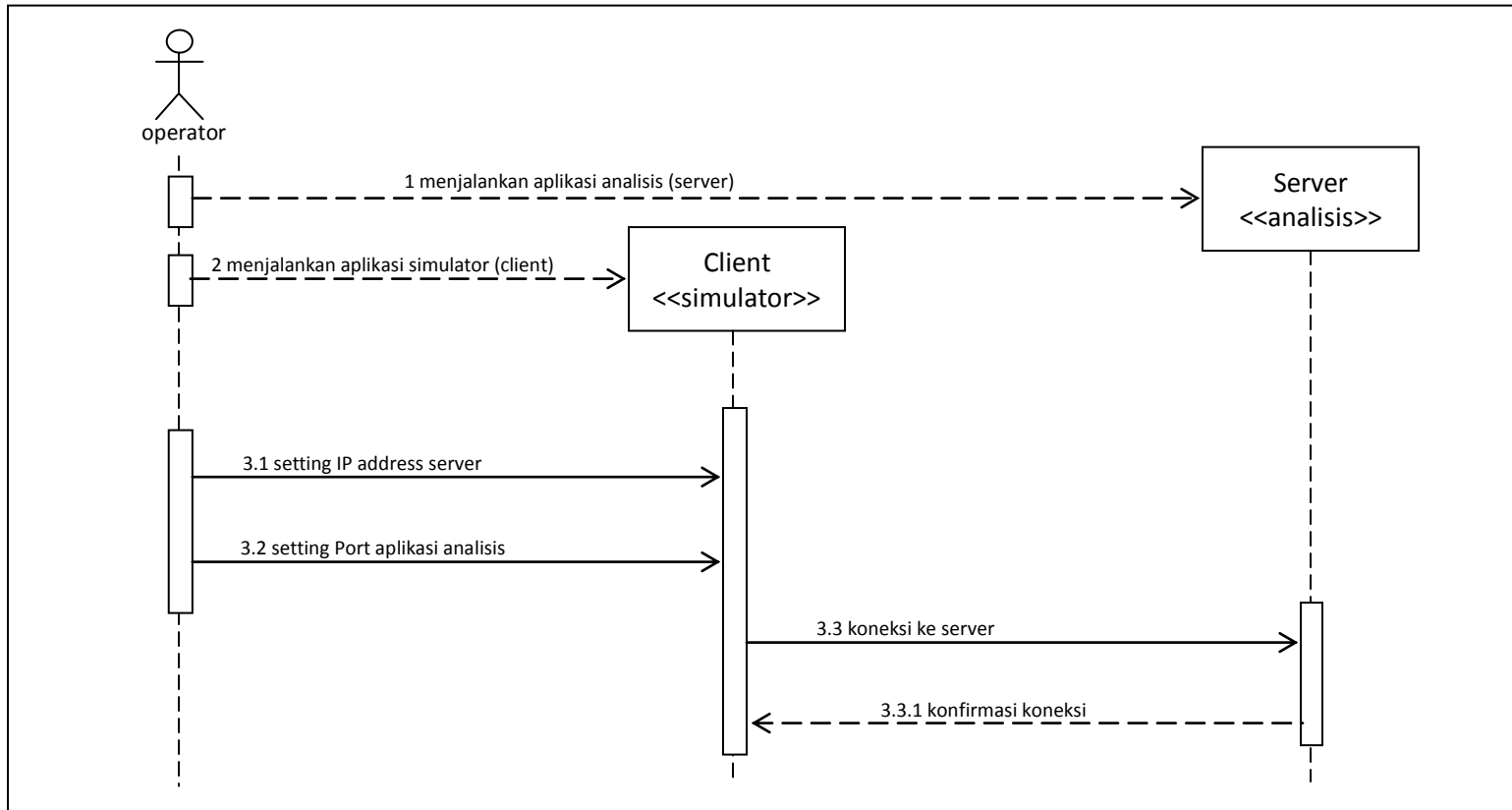
### **1.3. Sequensial Diagram Aplikasi Simulator**

Sequence diagram aplikasi simulator digunakan untuk mendesain aplikasi simulator yang menggambarkan perilaku objek pada proses dengan mendeskripsikan waktu hidup dan pesan yang dikirimkan dan diterima objek dalam aplikasi simulator. Perilaku dari setiap objek berdasarkan proses yang terjadi dalam aplikasi simulator digambarkan pada Gambar 4-7.

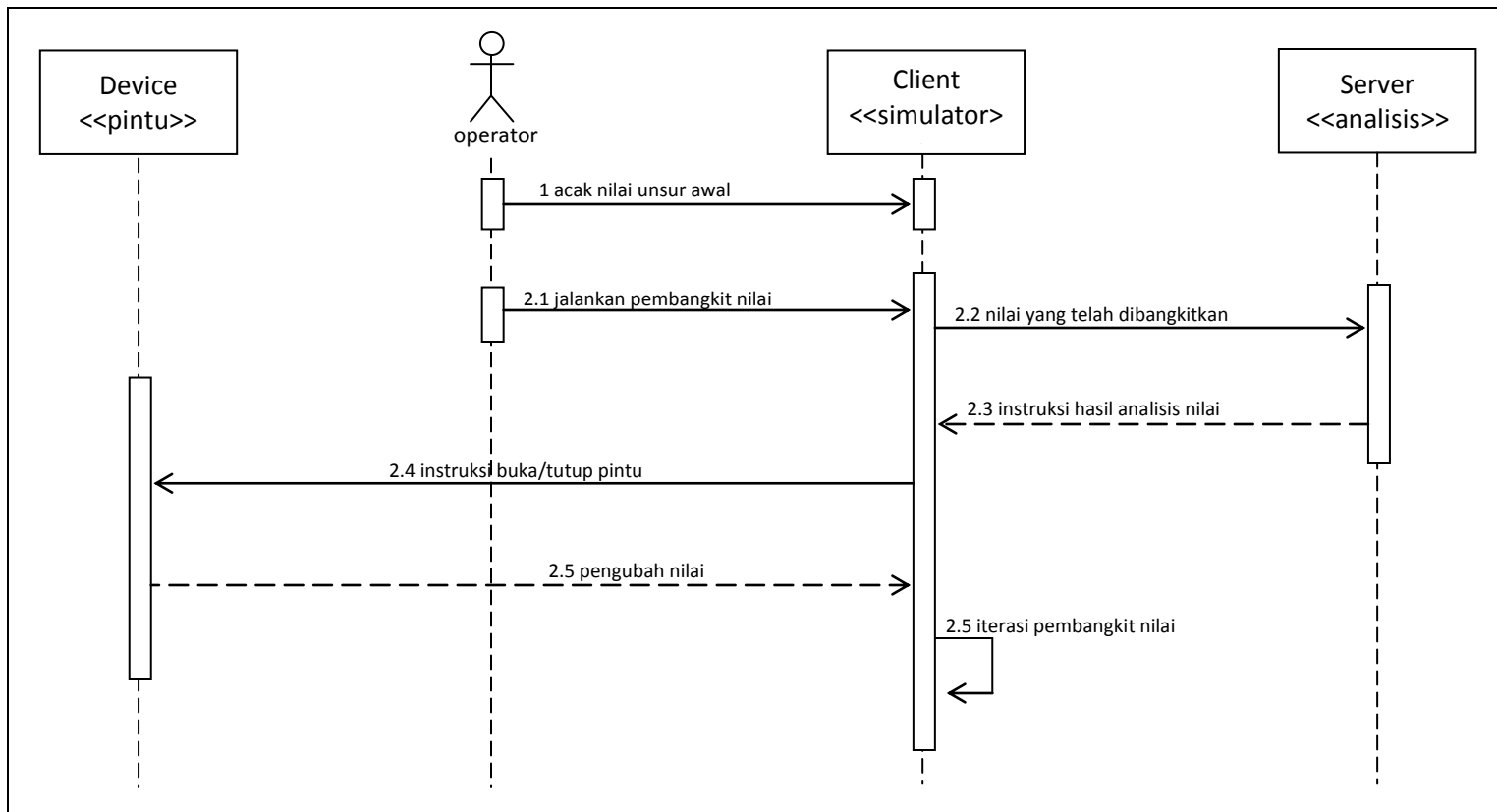
Pada Gambar 4-7 memperlihatkan perilaku objek saat terjadi koneksi antara aplikasi simulator sebagai class client dan aplikasi analisis sebagai class server. Koneksi ini dilakukan oleh class operator sebagai

aktor dengan menjalankan aplikasi analisis terlebih dahulu, dan kemudian aplikasi simulator.

Setelah kedua aplikasi ini berjalan, aktor melakukan setting alamat IP server dan nomor port untuk aplikasi analisis yang ada di server. Koneksi dilakukan dari client ke server. Setelah mendapatkan informasi alamat IP dan nomor port yang tepat maka aplikasi analisis akan membuka saluran komunikasi dengan aplikasi simulator, dan selanjutnya aplikasi analisis akan memberi informasi kepada aplikasi simulator.



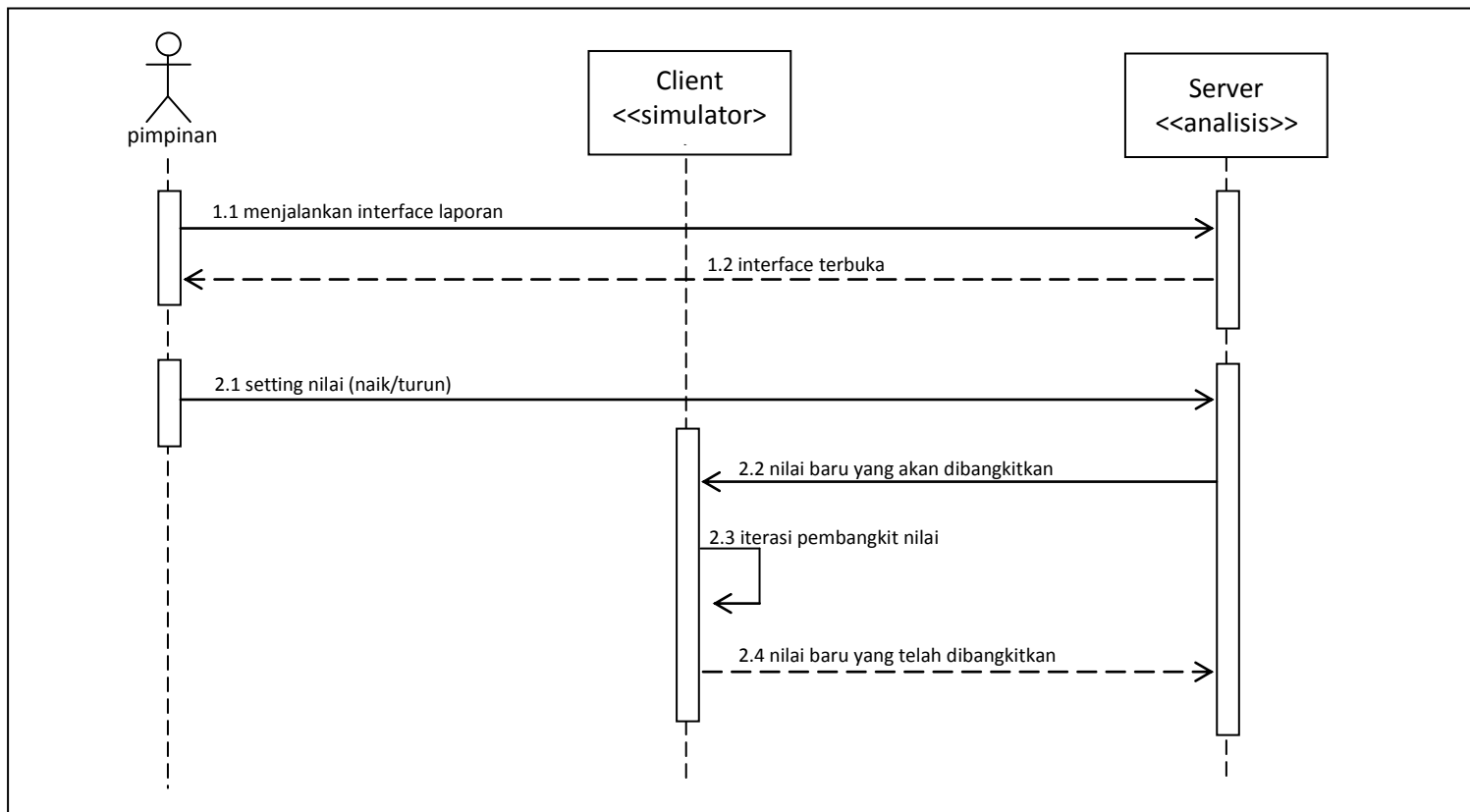
Gambar 4-7. Sequence Diagram Koneksi Client Server.



Gambar 4-8. Sequence Diagram Buka Tutup Pintu Pembuangan Limbah.

Pada Gambar 4-8 memperlihatkan perilaku objek yang terjadi pada saat proses membuka atau menutup pintu pembuangan limbah. Urutan kejadian/prilaku berdasarkan waktu dimulai oleh oprator melakukan pengacakan nilai awal yang akan dibangkitkan. Setelah nilai awal ditetapkan, operator menjalankan pembangkitan nilai pada aplikasi simulator, nilai yang dibangkitkan selanjutnya dikirim ke aplikasi analisis yang berada di server.

Aplikasi analisis akan melakukan analisis terhadap nilai unsur yang diterima dari aplikasi simulator. Hasil analisis mengasilkan satu keputusan untuk melakukan tindakan, tindakan tersebut kemudian dikirim kembali ke aplikasi simulator dan kemudian informasi tindakan tersebut diteruskan ke pintu pembuangan limbah. Setelah tindakan dilakukan maka aplikasi simulator akan mengubah nilai unsur yang akan dibangkitkan pada iterasi pembangkitan nilai selanjutnya. Perubahan nilai tersebut sesuai dengan kondisi pintu pembuangan limbah seperti yang telah dijelaskan pada activity diagram Gambar 4-6.



Gambar 4-9. Sequence Diagram Pengubah Nilai.



Pada Gambar 4-9 memperlihatkan perilaku objek melakukan perubahan nilai unsur untuk dibangkitkan pada iterasi pembangkitan nilai berikutnya. Pengubahan nilai unsur tidak saja dapat dilakukan setelah instruksi buka tutup pintu, tetapi dapat juga dilakukan dari aplikasi analisis jika dari hasil analisis ingin mempercepat proses buka tutup pintu pembuangan limbah.

Perilaku berdasarkan urutan waktu diawali oleh pimpinan yang membuka interface menu laporan pada aplikasi analisis, setelah interface terbuka pimpinan dapat melakukan setting nilai yang diinginkan untuk mempercepat pencapaian kondisi pembuangan limbah. Nilai baru yang diinginkan kemudian dikirim dari aplikasi analisis ke aplikasi simulator untuk dibangkitkan pada iterasi pembangkit nilai periode berikutnya. Selanjutnya nilai baru yang telah dibangkitkan dikirim kembali ke aplikasi analisis untuk dilakukan analisis kelayakan buang dari nilai unsur baru yang telah dibangkitkan.

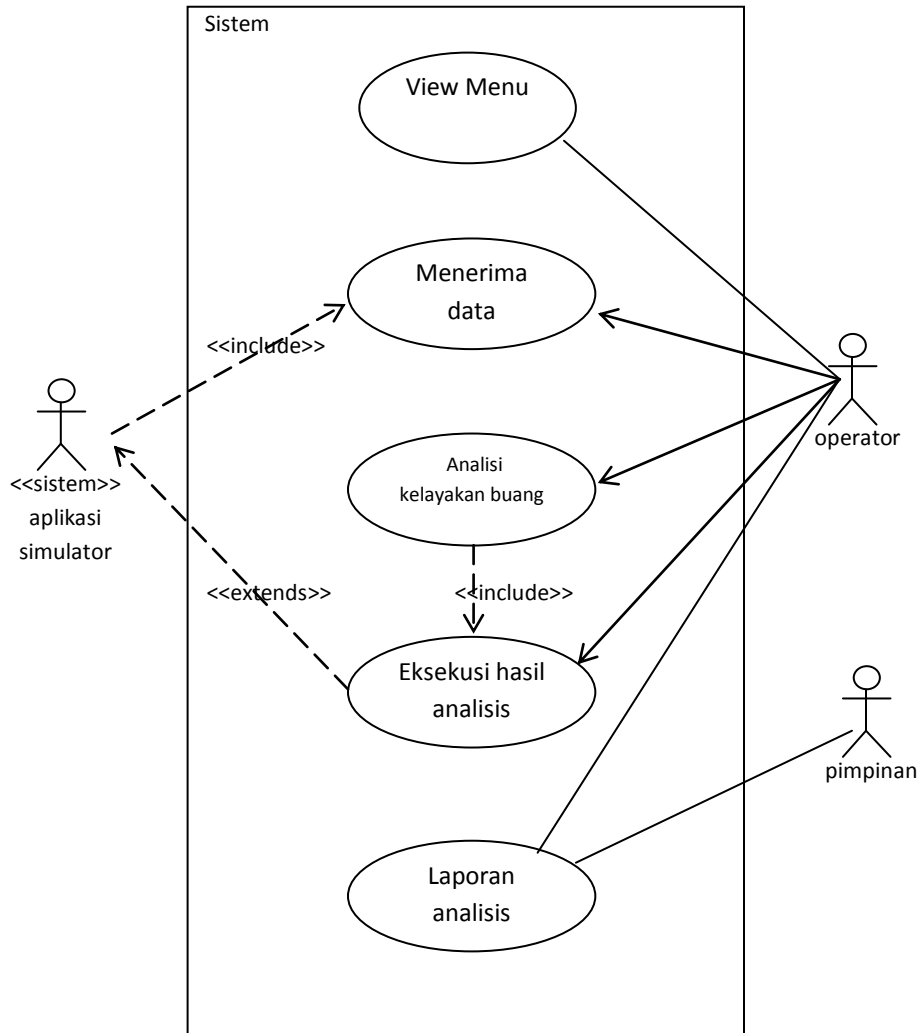
## **2. Perancangan Aplikasi Analisis**

### **2.1. Use Case Diagram Aplikasi Analisis**

Use Case diagram aplikasi analisis menggambarkan interaksi antara 3 aktor yang berada diluar sistem, yaitu aktor sistem (aplikasi simulator, aktor operator dan aktor pimpinan).

Ketiga aktor tersebut berinteraksi dengan lima case yang ada dalam sistem. Interaksi yang terjadi dijelaskan sebagai berikut :

- a. Aktor operator berinteraksi secara aktif kepada case view menu dan laporan hasil analisis. Interaksi tersebut terjadi saat operator menjalankan aplikasi analisis dan membuka menu-menu seperti menu analisis dan menu laporan hasil analisis. Sementara untuk case menerima data, analisis kelayakan buang limbah dan eksekusi hasil analisis, aktor operator berinteraksi secara pasif, dimana aktor berinteraksi untuk mengawasi proses-proses yang terjadi tanpa melakukan interaksi aktif terhadap proses yang sedang berjalan.
- b. Aktor pimpinan berinteraksi aktif dengan case laporan analisis, dimana interaksi ini pemimpin akan mengambil keputusan untuk melakukan tindakan-tindakan yang diperlukan sehubungan dengan pembuangan limbah.
- c. Aktor sistem (aplikasi simulator) berinteraksi dengan case menerima data, dimana penerimaan data terjadi jika pengiriman data dari aktor telah dilakukan. Selain itu aktor sistem ini akan terextends saat case eksekusi hasil analisis berproses. Saat case ini bekerja maka sistem juga akan bekerja sesuai tugas dari case eksekusi hasil analisis.



Gambar 4-10. Use Case Diagram Aplikasi Analisis

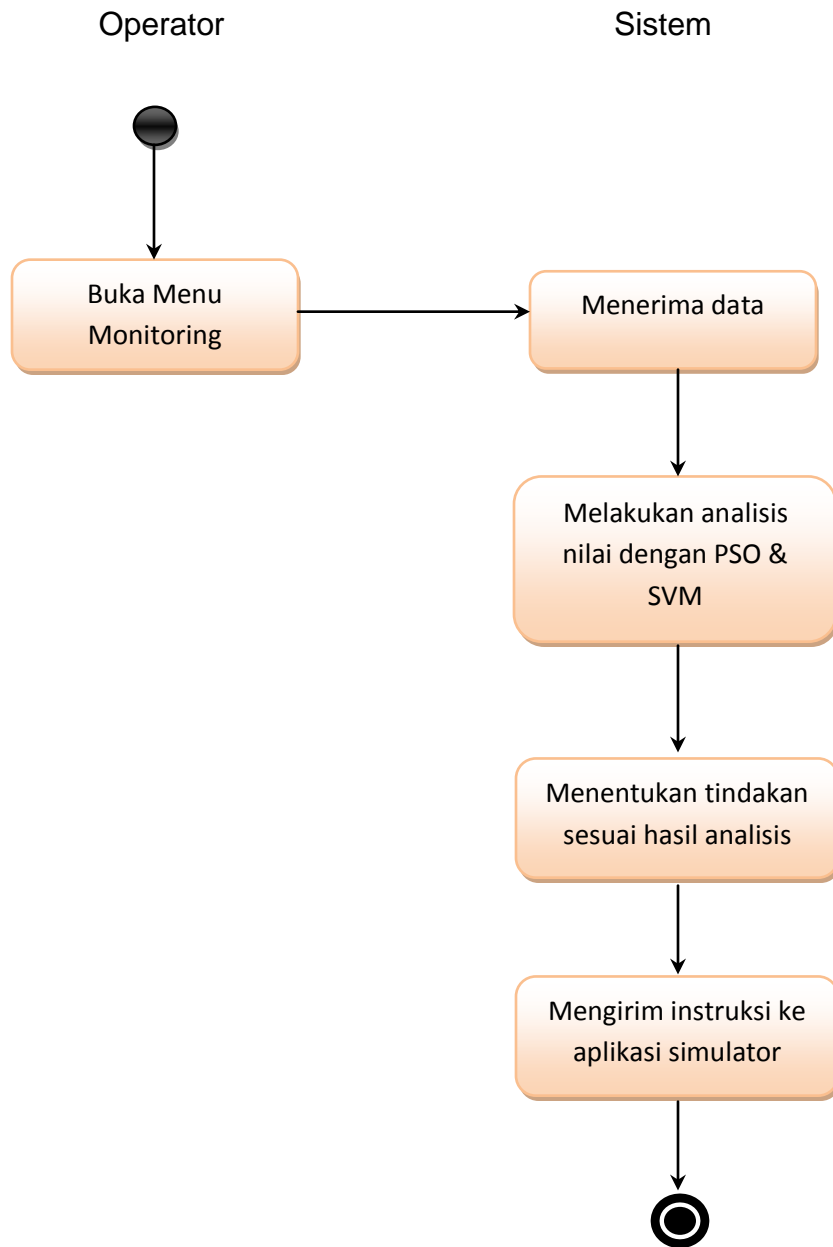
## 2.2. Activity Diagram Aplikasi Analisis

Pada Gambar 4-11 terlihat alur aktivitas mulai dari tampilan menu yang ditampilkan oleh sistem, kemudian operator membuka menu monitoring. Pada menu ini terjadi proses penerimaan data yang dikirim dari

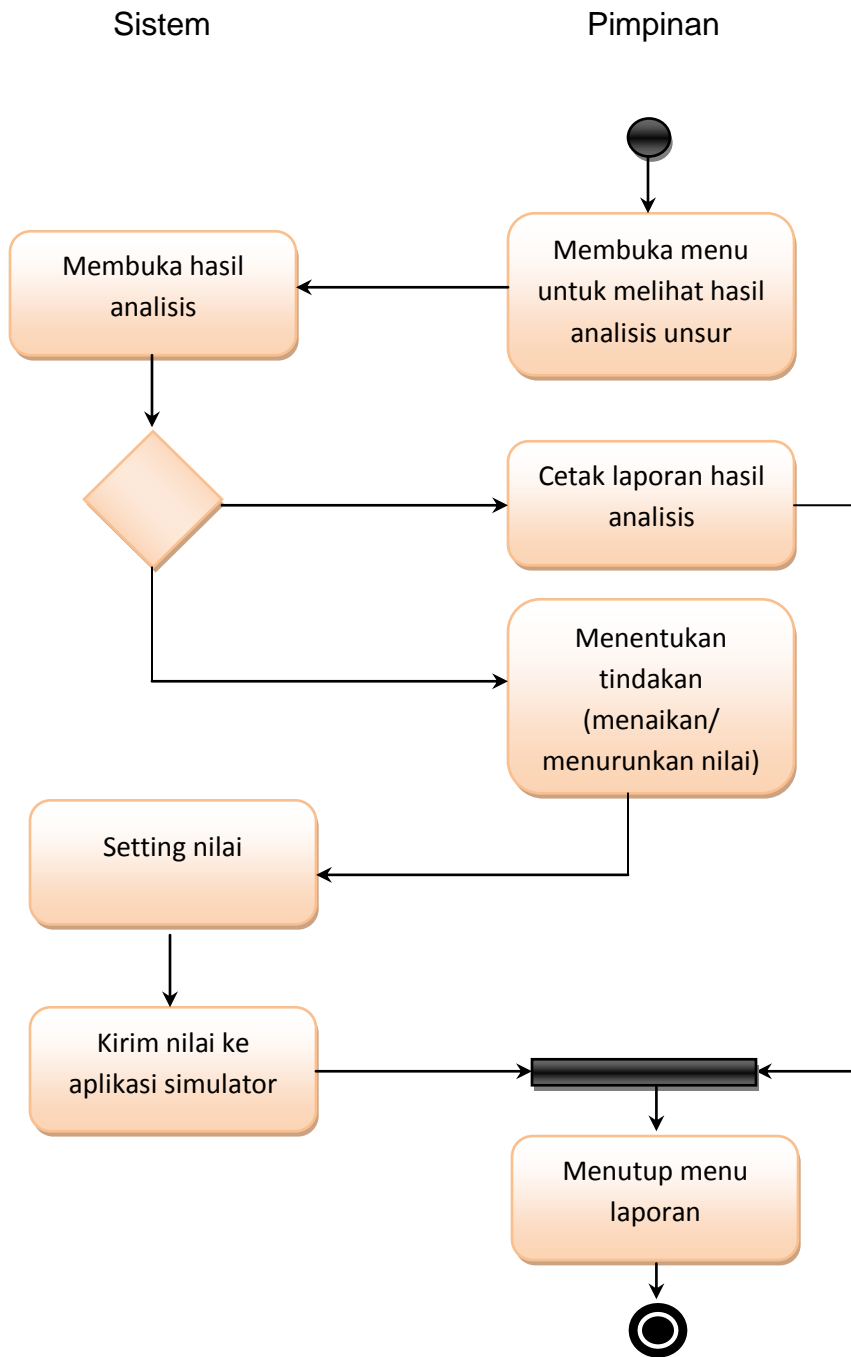
aplikasi simulator secara real time. Data/nilai yang diterima kemudian dianalisis menggunakan algoritma Partikel Swarm Optimization (PSO) dan algoritma Support Vector Machine (SVM). Data/nilai yang dianalisis sebanyak 100 data setiap unsur, dimana unsur yang dianalisis sebanyak 11 unsur. Dengan demikian terdapat 1100 data yang diterima untuk sekali analisis. Hasil analisis tersebut akan menghasilkan tindakan yang akan dilakukan, dan tindakan tersebut dikirim kembali ke aplikasi simulator untuk dijalankan sesuai dengan tahapan/aktivitas pada aplikasi simulator yang telah dijelaskan sebelumnya. Proses penerimaan data sampai mengirimkan hasil berupa sebuah tindakan ke aplikasi simulator harus telah selesai dikerjakan oleh sistem sebelum terjadi pengiriman data berikutnya oleh aplikasi simulator. Hal ini terjadi untuk memenuhi prinsip kerja real time pada sistem secara keseluruhan.

### **2.3. Activity Diagram pada Case Laporan**

Aktivitas pada Case Laporan seperti yang terlihat pada Gambar 4-12 dapat dijelaskan sebagai berikut ; Sistem menyimpan hasil analisis semua unsur dan dapat dilihat/evaluasi oleh pimpinan untuk mengambil tindakan/kebijakan untuk mempercepat/memperlambat proses pembuangan pembuangan limbah dengan cara menaikkan/menurunkan nilai dari beberapa unsur yang dianggap penting untuk mencapai kondisi yang diinginkan.



Gambar 4-11. Activity Diagram Case View Menu, Menerima Data, Analisis dan Eksekusi Hasil Analisis.



Gambar 4-12. Activity Diagram Case Laporan

Pada Gambar 4-12 aktivitas dimulai dari pimpinan yang membuka menu laporan pada sistem, setelah sistem membuka data hasil analisis terdapat pilihan apakah laporan akan dicetak atau akan dilakukan tindakan untuk menaikkan atau menurunkan nilai sehingga kondisi pembuangan limbah dapat tercapai sesuai yang diinginkan. Jika pimpinan hanya akan mencetak laporan, maka aktivitas berakhir setelah sistem mencetak laporan yang analisis yang diinginkan. Jika pimpinan akan melakukan tindakan untuk menaikkan atau menurunkan nilai maka sistem akan melakukan penyetingan nilai sesuai yang diinginkan dan kemudian nilai tersebut akan dikirimkan ke aplikasi simulator dan kemudian menjadi dasar pembangkitan nilai selanjutnya untuk mencapai kondisi yang diinginkan. Setelah sistem mengirimkan nilai ke aplikasi simulator maka aktivitas laporan selesai dan pimpinan mengakhiri aktivitas dengan menutup menu laporan.

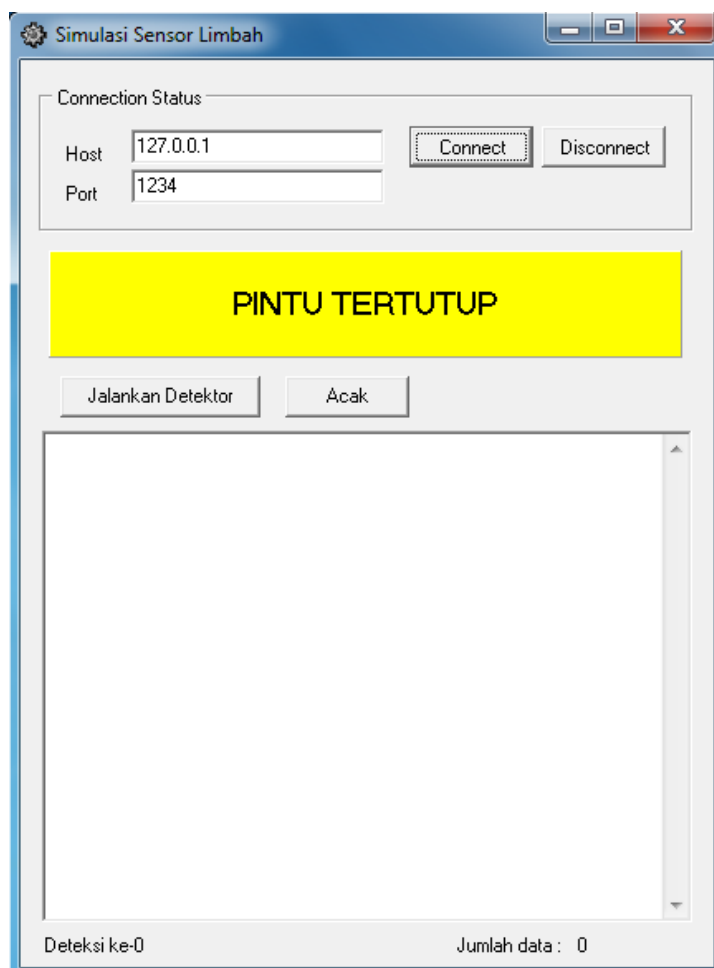
### **IV.3. Perancangan Antarmuka**

Perancangan antarmuka pada penelitian ini dimaksudkan untuk membuat interaksi antara aplikasi dengan pengguna menjadi sederhana dan efisien sehingga mencapai tujuan yang maksimal dalam penggunaan aplikasi. Sesuai dengan tujuan penelitian untuk membuat proses otomatisasi, maka antarmuka dirancang supaya pengguna lebih banyak berinteraksi

secara visual untuk melihat hasil analisis secara otomatis. Berikut ini adalah rancangan antarmuka pada aplikasi seimulator dan aplikasi analisis.

### 1. Rancangan Antarmuka Aplikasi Simulator

Pada aplikasi simulator hanya terdapat satu antarmuka, dimana semua proses otomatisasi bercalan dalam antarmuka tersebut. Berikut gambar rancangan antarmuka pada aplikasi simulator :



Gambar 4-13. Rancangan Antarmuka Aplikasi Simulator



Pada gambar rancangan antarmuka aplikasi simulator (Gambar 4-13) terdapat tombol untuk melakukan koneksi dan memutuskan koneksi ke aplikasi analisis, dimana syarat terjadinya koneksi menggunakan alamat IP komputer yang terpasang aplikasi analisis dan juga nomor port saluran komunikasi aplikasi analisis.

Saat koneksi telah terjadi pengguna dapat menjalankan proses detektor dengan menekan tombol jalankan detektor atau menunggu aplikasi berjalan secara otomatis. Ketika proses deteksi berjalan maka aplikasi akan membangkitkan nilai-nilai yang diasumsikan sebagai detektor/sensor yang mendeteksi kandungan unsur dalam limbah. Detektor akan membangkitkan nilai-nilai kandungan unsur sebanyak 100 nilai untuk setiap unsur, dimana unsur yang dideteksi sebanyak 11 unsur.

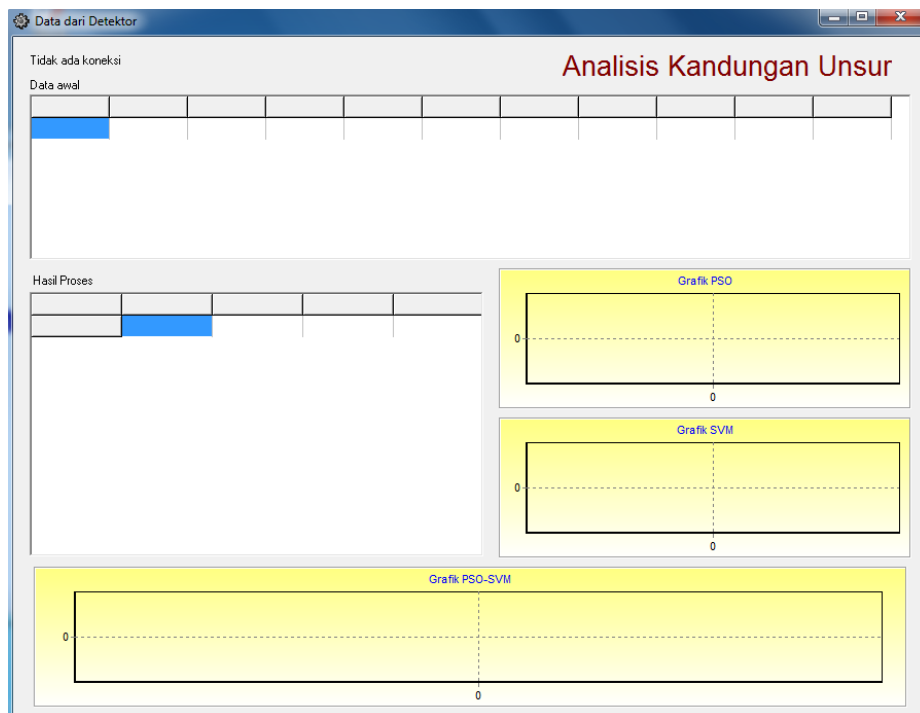
Untuk membuat kondisi beragam sehingga dapat melihat bagaimana aplikasi analisis bekerja dengan menggunakan dua metode (algoritma) maka disediakan tombol acak untuk mengacak nilai-nilai unsur yang terkandung. Nilai yang dibangkitkan kemudian akan ditampilkan terlebih dahulu sebelum dikirim ke aplikasi analisis untuk dianalisis. Untuk menjaga konsistensi jumlah data yang dibangkitkan, maka pada bagian bawah diberikan indikator untuk mengetahui jumlah data pada setiap pembangkitan data. Selain itu pengguna dapat mengetahui jumlah pembangkitan data yang telah dilakukan.

Panel berwarna kuning merupakan simulasi dari proses buka dan atau tutup pintu pembuangan limbah. Jika pada panel tersebut tertulis pintu

tertutup berarti hasil analisis kandungan unsur belum mencapai standar baku mutu sehingga pembuangan limbah tidak dapat dilakukan dan demikian pula sebaliknya. Aplikasi simulator akan bekerja secara real time dalam membangkitkan dan mengirimkan dilai yang telah dibangkitkat setiap satu menit.

## 2. Rancangan Antarmuka Aplikasi Analisis

Rancangan antarmuka pada aplikasi analisis terdiri dari dua modul yaitu antarmuka analisis kandungan unsur (monitoring) dan rancangan antarmuka pembuatan laporan. Berikut gambar rancangan antarmuka aplikasi analisis.



Gambar 4-14. Rancangan Antarmuka Analisis Kandungan Unsur

Pada Gambar 4-14 menampilkan rancangan antarmuka dari aplikasi analisis pada modul analisis kandungan unsur. Pada bagian awal terdapat indikator koneksi dari aplikasi simulator. Pada tabel data awal akan ditampilkan unsur-unsur yang terdeteksi hasil dari pembangkitan nilai aplikasi simulator. Data unsur disusun berdasarkan kolom-kolom ke-11 unsur yang dibangkitkan nilainya.

Pada tabel hasil proses menampilkan hasil analisis yang menggunakan algoritma PSO dan SVM. Terdapat empat kolom pada tabel hasil proses, yaitu kolom hasil perhitungan dengan PSO, kolom hasil perhitungan dengan SVM, kolom hasil perhitungan PSO-SVM dan kolom standar bakumutu limbah pertambangan nikel.

Pada rancangan antarmuka modul ini selain dhasi analisis ditampilkan dalam bentuk tabel, juga dirancang tampilan dalam bentuk grafik untuk hasil analisis menggunakan PSO, SVM dan PSO-SVM.

#### **IV.4. Perancangan Database**

Fungsi rancangan database pada penelitian ini adalah untuk memperoleh penampungan data yang baik terhadap data-data yang dihasilkan baik data yang dibangkitkan oleh aplikasi simulator maupun aplikasi analisis. Rancangan database yang baik akan menghasilkan tampungan data yang baik dan handal serta dapat berelasi baik dengan tabel

lain dalam database tersebut, sehingga pengelolaan data-data tersebut untuk memperoleh informasi menjadi efektif. Rancangan database pada penelitian ini terdiri dari dua tabel, yaitu tabel data awal, yaitu tabel yang menampung nilai-nilai dari unsur dalam limbah dan tabel yang menampung hasil proses dari aplikasi analisis. Berikut adalah tabel database yang dirancang :

1. Tabel Data Awal

Nama Field	Tipe	Panjang	Kunci	Keterangan
kode_field	Int	11	primary	-
tanggal_catat	date	0	-	-
nama_unsur	varchar	15	-	-
nilai	float	0	-	-

2. Tabel Hasil Proses

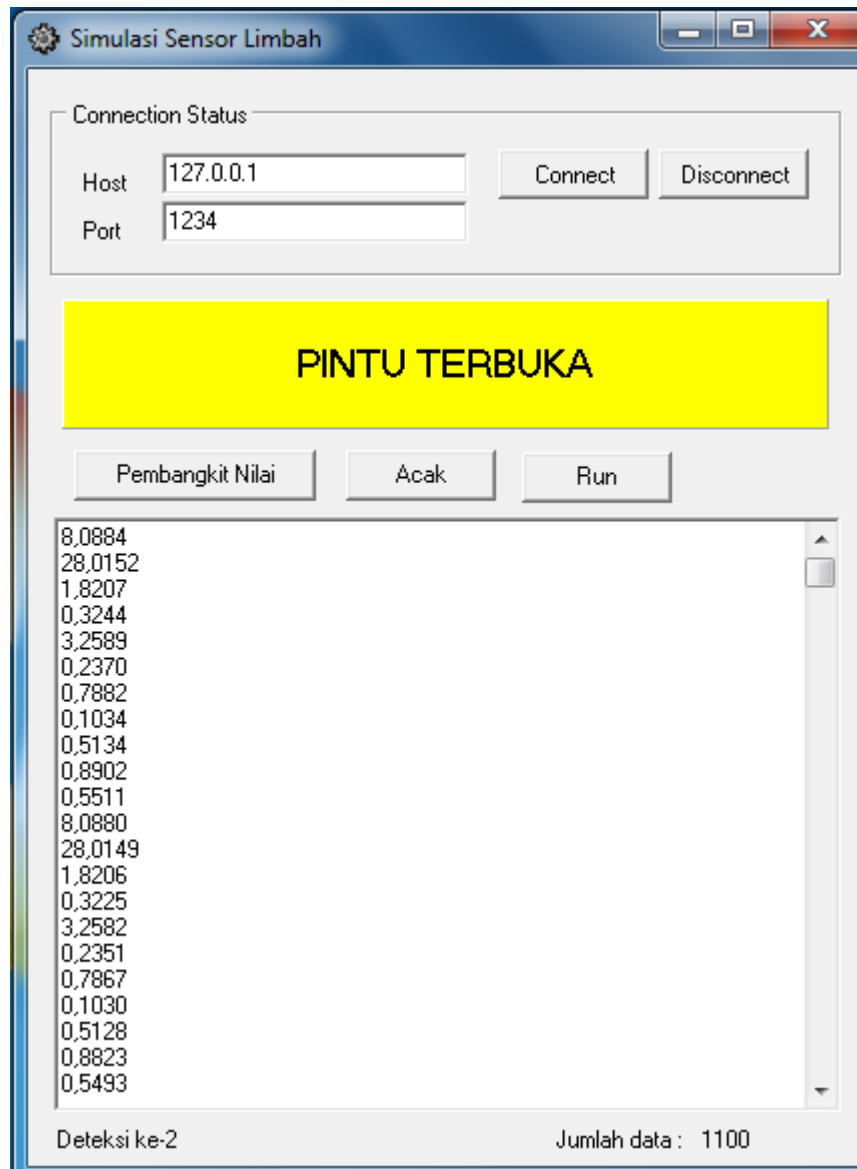
Nama Field	Tipe	Panjang	Kunci	Keterangan
kode_field	Int	11	primary	-
tanggal_catat	date	0	-	-
pso	float	0	-	-
svm	float	0	-	-
pso_svm	float	0	-	-

## **IV.5 Implementasi Sistem**

Seperti yang telah dijelaskan pada bagian rancangan antarmuka, berikut ini ditampilkan implementasi dari sistem yang dibangun meliputi implementasi aplikasi simulator dan aplikasi analisis.

### **1. Aplikasi Simulator**

Implementasi aplikasi simulator akan menghasilkan data yang dibangkitkan kemudian dikirim ke aplikasi analisis. Pada implementasi ini akan ditampilkan 1100 data dari 11 unsur yang nilainya dibangkitkan. Pada implementasi ini juga memperlihatkan respon dari pembukaan/penutupan pintu pembuangan limbah berdasarkan hasil analisis yang diterima dari aplikasi analisis.



Gambar 4-15 Implementasi Aplikasi Simulator

Pada Gambar 4-15 menampilkan data yang dibangkitkan dan pada indikator terlihat jumlah data yang dibangkitkan sebesar 1100 data. Dalam sekali pembangkitan data setiap unsur dibangkitkan sebanyak 100 data .

Pada gambar terlihat nilai pertama 8,0884 adalah nilai pH, selanjutnya TSS = 28,0152, Cu = 1,8207, Cd = 0,3244, Zn = 3,2589, Pb = 0,2370, Ni = 0,7882, Cr<sup>(6+)</sup> = 0,1034, Cr (Total) = 0,5134, Fe = 0,8902, Co = 0,5511, dan nilai pH berikutnya adalah 0,0880. Pola pembangkitan data dilakukan untuk tiap unsur dan dilakukan sebanyak 100 kali sehingga total data dalam sekali pembangkitan sebesar 1100 data. Data yang dibangkitkan sudah terkodefikasi sehingga data yang diterima aplikasi analisis nantinya dapat mengenal kepemilikan nilai-nilai tersebut.

Pada indikator deteksi menunjukkan angka dua yang berarti nilai-nilai di atas merupakan data yang dibangkitkan kedua kalinya. Dan pada indikator buka/tutup pintu adalah "PINTU TERBUKA", berarti data yang dibangkitkan pada pembangkitan sebelumnya setelah dianalisis berada di bawah baku mutu air limbah (Tabel 2-1) sehingga aplikasi detektor melakukan pembukaan pintu pembuangan limbah. Data di atas adalah pembangkitan kedua yang akan dikirim ke aplikasi analisis. Hasilnya akan dikembalikan ke aplikasi detektor. Jika hasilnya dibawah bakumutu air limbah maka aplikasi detektor tetap akan membuka pintu, sementara jika nilai di atas bakumutu maka aplikasi detektor akan menutup pintu pembuangan limbah.

### 1.a. Cara Pembangkitan Nilai Awal

Pembangkitan nilai dimulai saat aplikasi simulator telah terkoneksi dengan aplikasi Simulasi. Pembangkitan nilai pada awal aplikasi simulator dijalankan dapat dilakukan dengan dua cara, yaitu :

1. Pembangkitan nilai dengan cara menekan tombol Pembangkit Nilai.

Tombol ini akan dijalankan juga oleh komponen timer setiap satu menit, itu berarti nilai akan terus dibangkitkan setiap satu menit. Untuk memperoleh nilai pada awal pembangkitan digunakan persamaan ( $a = b + \text{rand}(b)$ ).  $a$  = nilai yang dibangkitkan,  $b$  adalah nilai yang ditetapkan antara 0 – 1,  $\text{rand}(b)$  adalah nilai yang diperoleh secara acak antara 0 –  $b$ .

Sebagai contoh untuk pH  $b = 0,002$  (ditetapkan),  $\text{rand}(b)$  diperoleh 0,0013 maka nilai yang dibangkitkan adalah 0,0033. Pada satu kali pembangkitan data dilakukan 100 iterasi sehingga diperoleh 100 nilai untuk setiap unsur.

Nilai pada iterasi berikutnya dari contoh unsur pH di atas, nilai  $b = 0,0033$  (nilai  $a$  dari iterasi sebelumnya),  $\text{rand}(b)$  diperoleh 0,0027 maka nilai yang diperoleh pada iterasi kedua pembangkitan pertama adalah 0,006. Demikian seterusnya sampai iterasi ke-100 untuk setiap unsurnya. Nilai  $b$  awal untuk TSS = 0,00034, Cu = 0,001, Cd = 0,005, Zn = 0,001, Pb = 0,0022, Ni = 0,0041,  $\text{Cr}^{(6+)} = 0,0013$ ,  $\text{Cr}^{(\text{total})} =$



0,0063, Fe = 0,0081 dan Co = 0,0029. Penetapan nilai b yang berbeda-beda diawal pembangkitan nilai dimaksudkan untuk memperoleh nilai yang beragam untuk setiap nilai yang dibangkitkan.

2. Pembangkitan nilai dengan nilai b awal diperoleh secara acak. Proses ini dilakukan dengan cara menekan tombol acak sebelum pembangkit nilai dijalankan. Nilai yang diperoleh melalui pengacakan ini dilakukan dengan persamaan  $b = \text{rand}(c) + \text{rand}(d)$  dimana nilai c dan d ditetapkan.

Sebagai contoh untuk pH ditetapkan nilai c = 2, dan nilai d = 12 maka nilai c diperoleh dari hasil nilai acak 1 – 2 , yaitu 1, dan nilai d diperoleh dari hasil nilai acak 1 – 12, yaitu 3. Jadi  $b = 1 + 3$ . Nilai 4 yang adalah nilai b menjadi nilai awal saat pembangkit nilai dijalankan dengan persamaan pada point (a.1). Proses yang sama terjadi untuk semua unsur dimana nilai c dan d ditentukan sebelumnya sebagai berikut : TSS (c = 5, d = 120), Cu (c = 0,018, d = 3,0), Cr (c = 0,00044, d = 0,3), Zn (c = 0,074, d = 4,0), Pb (c = 0,0029, d = 0,3), Ni (c = 0,0031, d = 1,0),  $\text{Cr}^{(6+)}$  (c = 0,0086, d = 0,3),  $\text{Cr}^{(\text{total})}$  (c = 0,065, d = 2,3), Fe (c = 0,12, d = 5,5) dan Co (c = 0,0062, d = 0,5).

Kedua cara pembangkitan nilai seperti yang telah dijelaskan di atas dilakukan untuk memperoleh kondisi yang mendekati kondisi sebenarnya dimana kandungan unsur dalam air limbah sangat bervariasi

sehingga aplikasi analisis dapat melakukan analisis untuk semua nilai yang dibangkitkan dan mewakili keadaan sebenarnya.

### **1.b. Cara Pembangkitan Nilai Berikut**

Pembangkitan nilai berikutnya dipengaruhi oleh hasil analisis terhadap nilai yang dibangkitkan sebelumnya. Tetapi nilai  $b$  yang akan digunakan pada pembangkitan nilai berikutnya adalah nilai  $b$  pada iterasi terakhir pada pembangkitan nilai sebelumnya.

Hasil analisis pada aplikasi analisis menghasilkan instruksi apakah nilai pada pembangkitan berikutnya akan di naikkan atau diturunkan. Jika nilai batas atas dari toleransi nilai baku mutu telah terlampaui maka aplikasi akan memberikan instruksi untuk menurunkan nilai unsur dengan persamaan  $a = (b - \text{rand}(b))$ .

Pada pembangkitan berikutnya nilai yang diturunkan akan terus diturunkan sampai mencapai batas bawah dari toleransi baku mutu yang ditetapkan. Jika nilai batas bawah dari toleransi baku mutu telah dilampaui maka aplikasi analisis akan mengirimkan instruksi untuk menaikkan nilai pada pembangkitan berikutnya dengan persamaan  $a = (b + \text{rand}(b))$ . Proses pembangkitan nilai seperti ini berlaku untuk semua unsur dengan toleransi batas atas adalah (baku mutu x 1,15) dan toleransi batas bawah ditentukan melalui (baku mutu x 0,2).

Pembangkitan nilai dan analisis terus berjalan sampai ke-11 unsur berada pada kondisi di bawah toleransi batas atas maka pintu pembuangan dibuka. Jika dari hasil analisis diperoleh hasil diluar range bakumutu air, maka pintu pembuangan ditutup.

Tabel 4-1. Asumsi Toleransi Batas Bawah dan Batas Atas Baku Mutu

Parameter	Satuan	Kadar Maksimum	Batas Bawah	Batas Atas
pH	-	6-9	6	9
TSS	mg/L	100	20	115
Cu	mg/L	2	0,4	2,3
Cd	mg/L	0,05	0,01	0,0575
Zn	mg/L	5	1	5,75
Pb	mg/L	0,1	0,02	0,115
Ni	mg/L	0,5	0,1	0,575
Cr <sup>(6+)</sup>	mg/L	0,1	0,02	0,115
Cr total	mg/L	0,5	0,1	0,575
Fe	mg/L	5	1	5,75
Co	mg/L	0,4	0,08	0,46

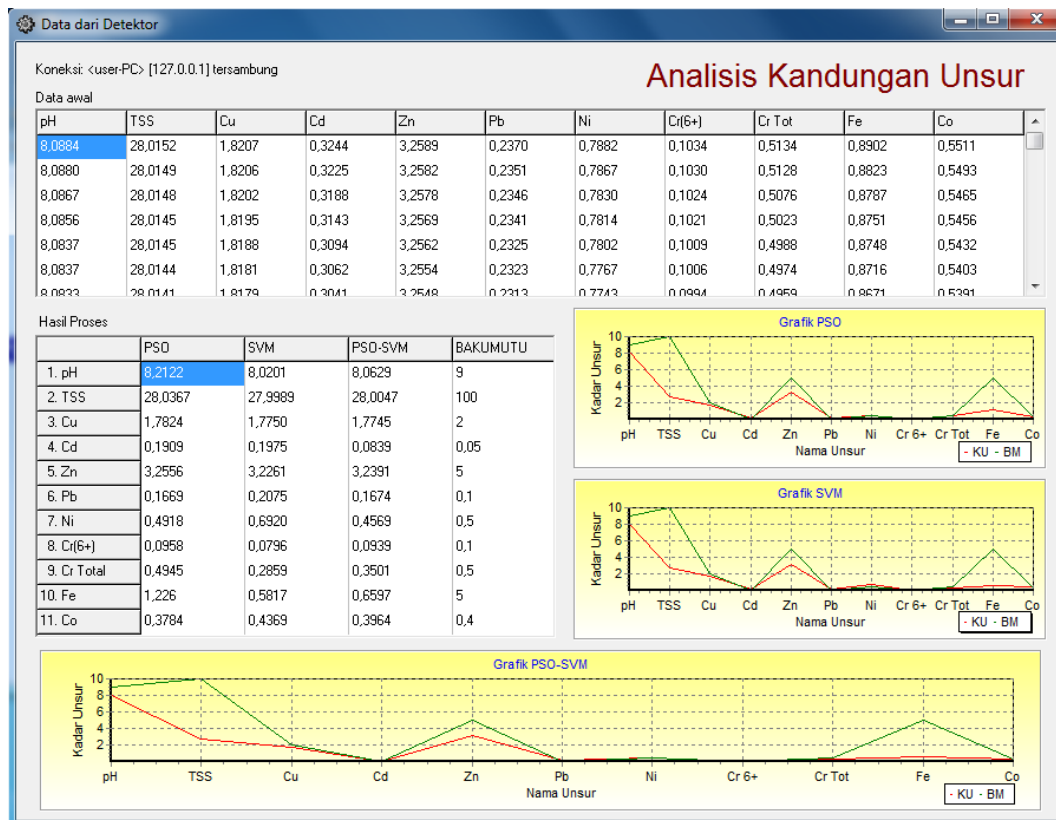
Pada Tabel 4-1 diperlihatkan asumsi toleransi batas bawah dan batas atas baku mutu air limbah. Asumsi toleransi batas bawah dan batas atas ini digunakan untuk menguji kepekaan aplikasi analisis saat merespon kandungan unsur yang berubah-ubah sesuai batas toleransi yang ditetapkan. Jika nilai mencapai batas atas maka secara otomatis aplikasi akan menurunkan nilai dan jika nilai mencapai batas bawah maka aplikasi akan menaikkan nilai. Menaikan atau menurunkan nilai dilakukan untuk semua unsur pada saat unsur tersebut mencapai batas atas atau batas bawah,

sehingga akan diperoleh kondisi yang berbeda tiap-tiap unsur disebabkan oleh perbedaan nilai batas atas dan batas bawah serta perbedaan besaran menaikkan atau menurunkan nilai tiap unsur.

## 2. Aplikasi Analisis

Implementasi aplikasi analisis menampilkan data-data dan informasi dalam bentuk tabel dan grafik. Berikut tampilan implementasi aplikasi analisis

:



Gambar 4-16. Implementasi Aplikasi Analisis

Pada Gambar 4-16 ditampilkan hasil implementasi aplikasi analisis yang terdiri dari tiga bagian utama, yaitu data awal dalam bentuk tabel, hasil proses/analisis dalam bentuk tabel dan grafik. Berikut penjelasan tentang hasil analisis dari aplikasi analisis.

**a. Data Awal**

Koneksi: <user-PC> [127.0.0.1] tersambung

**Analisis Kandungan Unsur**

Data awal

pH	TSS	Cu	Cd	Zn	Pb	Ni	Cr(6+)	Cr Tot	Fe	Co
8,0884	28,0152	1,8207	0,3244	3,2589	0,2370	0,7882	0,1034	0,5134	0,8902	0,5511
8,0880	28,0149	1,8206	0,3225	3,2582	0,2351	0,7867	0,1030	0,5128	0,8823	0,5493
8,0867	28,0148	1,8202	0,3188	3,2578	0,2346	0,7830	0,1024	0,5076	0,8787	0,5465
8,0856	28,0145	1,8195	0,3143	3,2569	0,2341	0,7814	0,1021	0,5023	0,8751	0,5456
8,0837	28,0145	1,8188	0,3094	3,2562	0,2325	0,7802	0,1009	0,4988	0,8748	0,5432
8,0837	28,0144	1,8181	0,3062	3,2554	0,2323	0,7767	0,1006	0,4974	0,8716	0,5403
8,0833	28,0141	1,8179	0,3041	3,2548	0,2313	0,7743	0,0994	0,4959	0,8671	0,5391

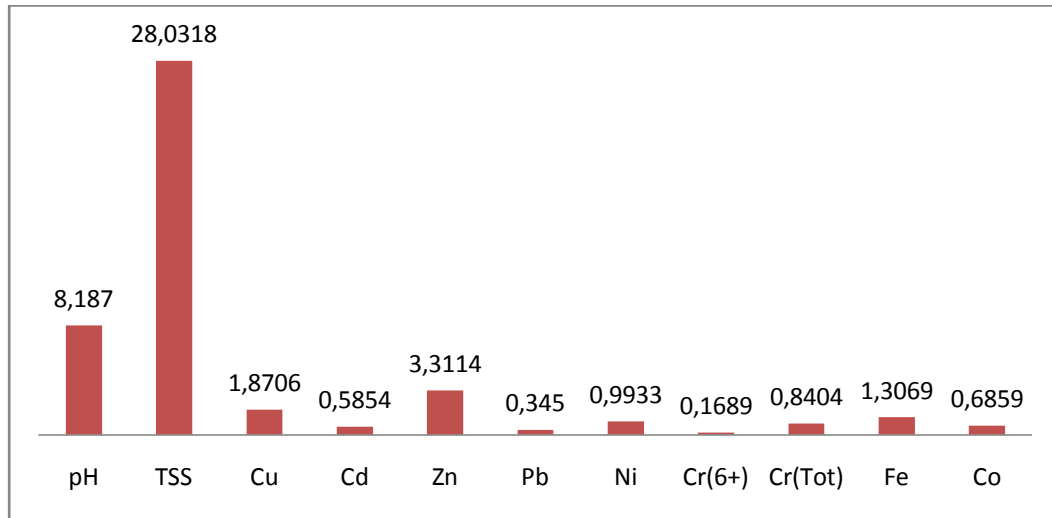
Gambar 4-17. Data Awal Aplikasi Analisis

Pada Gambar 4-17 di atas adalah data awal yang akan dianalisis oleh aplikasi analisis. Data tersebut diterima dari aplikasi detektor seperti yang terlihat pada Gambar 4-15. Data yang diterima dikelompokkan berdasarkan nama masing-masing unsur sehingga lebih mudah terbaca dan dipahami. pH dimulai dengan nilai 8,0884, TSS = 28,0152, Cu = 0,3244, Zn = 3,2589, Pb = 0,2370, Ni = 0,7882, Cr<sup>(6+)</sup> = 0,1034, Cr (Total) = 0,5134, Fe = 0,8902, Co = 0,5511. Nilai-nilai ini sama dengan nilai yang tersusun secara vertikal pada aplikasi detektor (Gambar 4-15).

Data dalam bentuk tabel pada Gambar 4-17 adalah data yang akan dianalisis menggunakan PSO dan SVM. Data tersebut sebanyak 100 data

untuk masing masing unsur dan tersusun dari bernilai besar ke nilai yang rendah dengan selisih masing-masing nilai berbeda. Hal tersebut dapat dijelaskan sebagai berikut :

1. Nilai data mulai dari nilai besar ke nilai kecil disebabkan pada saat data tersebut dibangkitkan, aplikasi detektor telah mendapat instruksi dari aplikasi analisis untuk menurunkan nilai unsur sebagai hasil analisis data yang dibangkitkan sebelumnya telah melampau batas maksimum (bakumutu air limbah) sehingga nilainya harus diturunkan agar supaya pintu pembuangan limbah dapat mencapai kondisi terbuka. Contoh nilai pertama pH = 8,0884, nilai kedua 8,0880, nilai ketiga 8,0867, nilai keempat 8,0856 sampai nilai ke 100 adalah 7,9978. Apabila hasil analisis telah mencapai batas minimum dari kandungan unsur, maka aplikasi analisis akan memerintahkan aplikasi detektor untuk menaikkan nilai pembangkitan nilai berikutnya.
2. Selisih penurunan angka berbeda-beda (tidak linier). Hal ini terjadi karena saat aplikasi detektor membangkitkan data jumlah penurunannya dilakukan secara acak. Keadaan ini merupakan asumsi bahwa sensor pada kondisi sebenarnya akan mendeteksi keadaan unsur dalam air limbah dengan nilai yang berbeda baik saat kadar unsur naik atau kadar unsur turun.



Grafik 4-1. Data Awal pada Kondisi Pertama

Pada Grafik 4-1 memperlihatkan data tertinggi kandungan tiap unsur dari hasil pembangkitan 100 nilai. Untuk pH nilai pertama 0.0884 (Gambar 4-17) dan nilai pembangkitan ke-100 adalah 8.1870 (Grafik 4-1). TSS 28.0152 – 28.0318, Cu 1.8207 – 1.8706, Cd 0.3244 – 0.5854, Zn 3.2589 – 3.3114, Pb 0.2370 – 0.3450, Ni 0.7882 – 0.9933, Cr(6+) 0.1034 – 0.1689, Cr(Tot) 0.5134 – 0.8404, Fe 0.8902 – 1.3069 dan Co 0.5511 – 0.6859.

Pembangkitan data pada kondisi pertama ini memperlihatkan terjadinya kenaikan nilai, dimana data yang dibangkitkan pada awalnya kecil dan data terakhir bernilai lebih besar dari data awal.

## b. Hasi Proses

Untuk memperoleh hasil analisis yang mewakili semua kondisi pembangkitan nilai maka hasil analisis akan dibahas dengan enam kondisi nilai yang dibangkitkan, yaitu :

### 1. Nilai dalam Range Baku Mutu dari Batas Atas ke Batas Bawah

Analisis dilakukan terhadap data yang sedang diturunkan nilainya dari batas atas ke batas bawah dari standar baku mutu seperti pada Gambar 4-17.

Hasil Proses				
	PSO	SVM	PSO-SVM	BAKUMUTU
1. pH	8,2122	8,0201	8,0629	9
2. TSS	28,0367	27,9989	28,0047	100
3. Cu	1,7824	1,7750	1,7745	2
4. Cd	0,1909	0,1975	0,0839	0,05
5. Zn	3,2556	3,2261	3,2391	5
6. Pb	0,1669	0,2075	0,1674	0,1
7. Ni	0,4918	0,6920	0,4569	0,5
8. Cr(6+)	0,0958	0,0796	0,0939	0,1
9. Cr Total	0,4945	0,2859	0,3501	0,5
10. Fe	1,226	0,5817	0,6597	5
11. Co	0,3784	0,4369	0,3964	0,4

Gambar 4-18 Hasil Proses Nilai Menurun dalam Range Baku Mutu



Pada Gambar 4-18 adalah hasil proses nilai-nilai pada Gambar 4-17 yang dianalisis menggunakan PSO, SVM dan PSO-SVM. Hasil tersebut ditampilkan dalam bentuk tabel dengan pengelompokan hasil untuk analisis menggunakan PSO, SVM, PSO-SVM untuk setiap unsur dan sebagai pembandingan pada kolom terakhir di tampilan bakumutu air limbah. Penjelasan hasil analisis sebagai berikut :

1. pH ; hasil analisis menggunakan PSO = 8.2122, SVM = 8,0201 dan PSO-SVM = 8,0629 dengan pembandingan bakumutu air limbah = 9 yang berarti hasil analisis dari data yang dibangkitkan berada dibawah baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO memiliki nilai lebih tinggi dari nilai SVM dan nilai PSO-SVM berada diantara PSO dan SVM.
2. TSS ; hasil analisis menggunakan PSO = 28.0367, SVM = 27,9989 dan PSO-SVM = 28,0047 dengan pembandingan bakumutu air limbah = 100 yang berarti hasil analisis dari data yang dibangkitkan berada dibawah baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO memiliki nilai lebih tinggi dari nilai SVM dan nilai PSO-SVM berada diantara PSO dan SVM.
3. Cu ; hasil analisis menggunakan PSO = 1,7824, SVM = 1,7750 dan PSO-SVM = 1,7745 dengan pembandingan bakumutu air limbah = 2 yang berarti hasil analisis dari data yang dibangkitkan berada dibawah baku mutu air limbah. Dari data tersebut terlihat hasil

analisis PSO memiliki nilai lebih tinggi dari nilai SVM dan nilai PSO-SVM berada diantara PSO dan SVM.

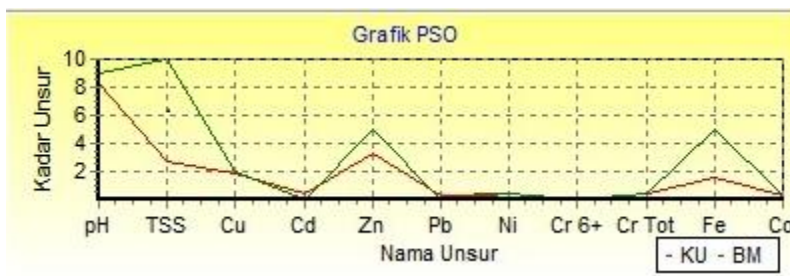
4. Cd ; hasil analisis menggunakan PSO = 0,1909, SVM = 0,1975 dan PSO-SVM = 0,0839 dengan pembanding bakumutu air limbah = 0,05 yang berarti hasil analisis dari data yang dibangkitkan berada di atas baku mutu air limbah. Dari data tersebut terlihat hasil analisis SVM memiliki nilai lebih tinggi dari nilai PSO dan nilai PSO-SVM.
5. Zn ; hasil analisis menggunakan PSO = 3,2526, SVM = 3,2261 dan PSO-SVM = 3,2391 dengan pembanding bakumutu air limbah = 5 yang berarti hasil analisis dari data yang dibangkitkan berada di bawah baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO memiliki nilai lebih tinggi dari nilai SVM dan nilai PSO-SVM berada diantara PSO dan SVM.
6. Pb ; hasil analisis menggunakan PSO = 0,1669, SVM = 0,2075 dan PSO-SVM = 0,1674 dengan pembanding bakumutu air limbah = 0,1 yang berarti hasil analisis dari data yang dibangkitkan berada di atas baku mutu air limbah. Dari data tersebut terlihat hasil analisis SVM memiliki nilai lebih tinggi dari nilai PSO-SVM dan nilai PSO paling rendah.
7. Ni ; hasil analisis menggunakan PSO = 0,4918, SVM = 0,6920 dan PSO-SVM = 0,4569 dengan pembanding bakumutu air limbah = 0,5 yang berarti hasil analisis dari data yang dibangkitkan berada di atas

baku mutu air limbah untuk SVM dan di bawah baku mutu air limbah untuk PSO dan SVM. Dari data tersebut terlihat hasil analisis SVM memiliki nilai lebih tinggi dari nilai PSO dan nilai PSO-SVM paling rendah.

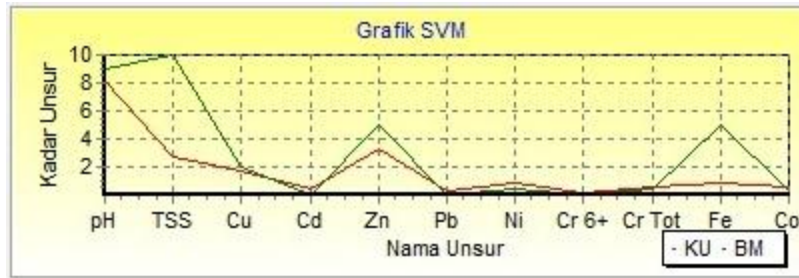
8.  $Cr^{(6+)}$  ; hasil analisis menggunakan PSO = 0,0958, SVM = 0,0796 dan PSO-SVM = 0,0939 dengan pembandingan bakumutu air limbah = 0,1 yang berarti hasil analisis dari data yang dibangkitkan berada di bawah baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO memiliki nilai lebih tinggi dari nilai SVM dan nilai PSO-SVM berada diantara PSO dan SVM.
9.  $Cr^{(Total)}$  ; hasil analisis menggunakan PSO = 0,4945, SVM = 0,2859 dan PSO-SVM = 0,3501 dengan pembandingan bakumutu air limbah = 0,5 yang berarti hasil analisis dari data yang dibangkitkan berada di bawah baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO memiliki nilai lebih tinggi dari nilai SVM dan nilai PSO-SVM berada diantara PSO dan SVM.
10. Fe ; hasil analisis menggunakan PSO = 1,2260, SVM = 0,5817 dan PSO-SVM = 0,6597 dengan pembandingan bakumutu air limbah = 5 yang berarti hasil analisis dari data yang dibangkitkan berada di bawah baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO memiliki nilai lebih tinggi dari nilai SVM dan nilai PSO-SVM berada diantara PSO dan SVM.

11. Co ; hasil analisis menggunakan PSO = 0,3784, SVM = 0,4369 dan PSO-SVM = 0,3964 dengan pembanding bakumutu air limbah = 0,4 yang berarti hasil analisis dari data yang dibangkitkan berada di bawah baku mutu air limbah untuk SVM dan di bawah baku mutu air untuk PSO dan PSO-SVM. Dari data tersebut terlihat hasil analisis SVM memiliki nilai lebih tinggi dari nilai PSO dan nilai PSO-SVM dimana PSO memiliki nilai paling rendah. Untuk unsur Co analisis menggunakan SVM lebih baik dibandingkan PSO maupun PSO-SVM.

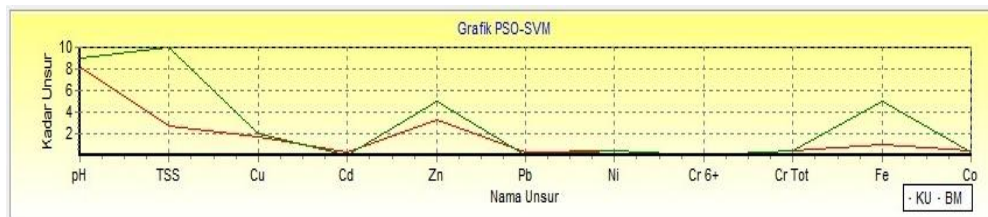
Hasil analisis ke-11 unsur tersebut di atas dapat dilihat pada grafik perbandingan nilai hasil analisis yang ditandai dengan garis merah dan baku mutu air limbah ditandai dengan garis hijau. Grafik 4-2 menunjukkan hasil analisis dengan PSO, Grafik 4-3 menunjukkan hasil analisis dengan SVM dan Grafik 4-4 menunjukkan hasil analisis dengan PSO-SVM.



Grafik 4-2. Hasil Analisis PSO Data Kondisi Pertama



Grafik 4-3. Hasil Analisis SVM Data Kondisi Pertama



Grafik 4-4. Hasil Analisis PSO-SVM Data Kondisi Pertama

Tabel 4-2. Bobot Hasil Analisis Kondisi Pertama

Unsur	PSO	SVM	PSO-SVM
pH	3	1	2
TSS	3	1	2
Cu	3	1	2
Cd	2	3	1
Zn	3	1	2
Pb	1	3	2

Ni	2	3	1
Cr (6+)	3	1	2
Cr Total	3	1	2
Fe	3	1	2
Co	1	3	2
Total	27	19	20

Hasil analisis kemudian diberikan bobot nilai tertinggi (3), sedang (2) dan (1) seperti yang terlihat pada Tabel 4-1. Dari tabel tersebut dapat disimpulkan bahwa untuk 11 unsur dalam limbah, ternyata PSO memiliki range nilai tertinggi (3) yang paling dominan, yaitu untuk 7 unsur pH, TSS, Cu, Zn, Cr<sup>(6+)</sup>, Cr Total, dan Fe, sedangkan untuk SVM hanya 4 nilai tertinggi masing-masing Cd, Pb, Ni dan Co, sementara untuk penggabungan keduanya, tidak terdapat bobot dengan nilai tertinggi. Untuk bobot nilai terendah (1) banyak terdapat (mayoritas) pada SVM masing-masing pada nilai unsur pH, TSS, Cu, Zn, Cr<sup>(6+)</sup>, Cr Total dan Fe, sedangkan pada PSO, hanya pada Pb dan Co. Sedangkan untuk penggabungan keduanya hanya terdapat pada nilai unsur Cd dan Ni. Sementara untuk penggabungan antara PSO dan SVM mayoritas bobot nilainya adalah (2), masing-masing pada nilai unsur pH, TSS, Cu, Zn, Pb,

Cr<sup>(6+)</sup>, Cr Total, Fe dan Co. Sedangkan bobot (1) hanya terdapat pada unsur Cd dan Ni. Untuk bobot (3) tidak ada sama sekali.

## 2. Nilai dalam Range Baku Mutu dari Batas Bawah ke Batas Atas

Pada Gambar 4-19 terlihat data yang diterima dari aplikasi simulator dibangkitkan dari nilai rendah ke tinggi menuju batas atas baku mutu. Untuk pH nilai pertama yang dibangkitkan sebesar 3,0010 dan nilai data yang ke-100 sebesar 3,0986. Demikian pula unsur lainnya nilai dibangkitkan dimana nilai berikutnya lebih tinggi dari nilai sebelumnya.

Koneksi: <user-PC> [127.0.0.1] tersambung

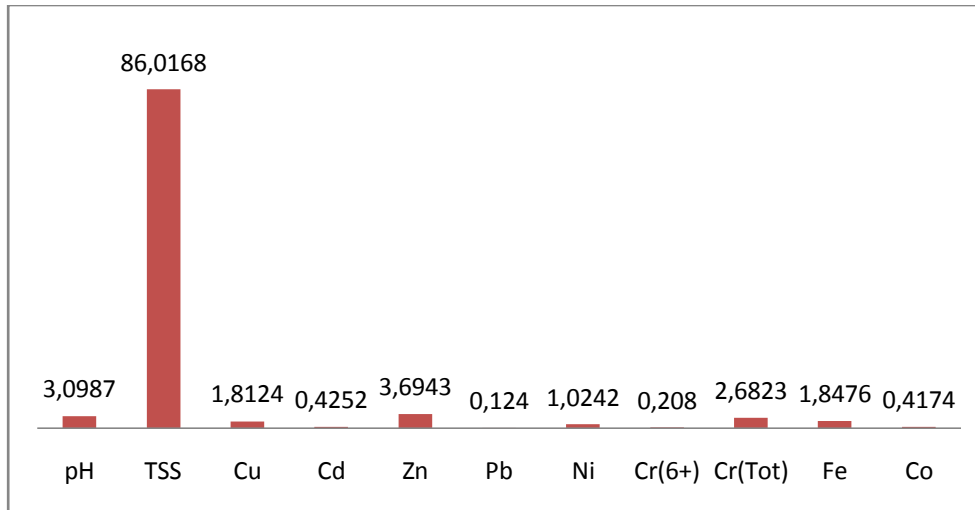
**Analisis Kandungan Unsur**

Data awal

pH	TSS	Cu	Cd	Zn	Pb	Ni	Cr(6+)	Cr Tot	Fe	Co
3,0010	86,0001	1,7605	0,1499	3,6518	0,0099	0,8175	0,1439	2,3434	1,4314	0,2827
3,0019	86,0001	1,7608	0,1454	3,6522	0,0108	0,8199	0,1445	2,3491	1,4372	0,2848
3,0023	86,0004	1,7610	0,1444	3,6522	0,0130	0,8201	0,1446	2,3506	1,4410	0,2856
3,0032	86,0007	1,7611	0,1420	3,6530	0,0147	0,8204	0,1447	2,3569	1,4435	0,2882
3,0042	86,0008	1,7616	0,1413	3,6530	0,0149	0,8236	0,1459	2,3609	1,4463	0,2910
3,0058	86,0012	1,7620	0,1394	3,6535	0,0155	0,8242	0,1469	2,3635	1,4497	0,2914
3,0059	86,0012	1,7620	0,1395	3,6545	0,0175	0,8265	0,1477	2,3655	1,4509	0,2924

Gambar 4-19 Data Awal Kondisi Menaikan Nilai

Gambar 4-19 memperlihatkan data pertama yang dibangkitkan pada kondisi ke dua dengan nilai pH 3.0010 dan nilai ke-100 dari pembangkitan nilai pada kondisi kedua ini sebesar 3.0987 (Grafik 4-5). Selanjutnya untuk mengetahui nilai awal pembangkitan tiap unsur ditampilkan pada Gambar 4-19 dan nilai akhir dari pembangkitan nilai tiap unsur dapat dilihat pada Grafik 4-5.



Grafik 4-5. Data Awal pada Kondisi Kedua

Selanjutnya data pada Gambar 4-19 dianalisis dan memperoleh hasil seperti pada Gambar 4-20 dengan penjelasan begai berikut :

1. pH ; hasil analisis menggunakan PSO = 3.0469, SVM = 3,0328 dan PSO-SVM = 3,0593 dengan pembanding bakumutu air limbah = 9 yang berarti hasil analisis dari data yang dibangkitkan berada dibawah baku mutu air limbah. Dari data tersebut PSO-SVM memiliki nilai tertinggi, kemudian PSO dan paling rendah adalah SVM. Untuk pH bobot nilai tertinggi pada PSO-SVM.



Hasil Proses				
	PSO	SVM	PSO-SVM	BAKUMUTU
1. pH	3,0469	3,0328	3,0593	9
2. TSS	86,0547	86,0104	86,0395	100
3. Cu	1,8348	1,7770	1,831	2
4. Cd	0,1222	-0,0350	0,4046	0,05
5. Zn	3,6934	3,6761	3,6965	5
6. Pb	0,1038	0,0554	0,0975	0,1
7. Ni	0,9152	0,9712	0,9861	0,5
8. Cr(6+)	0,1586	0,1696	0,1649	0,1
9. Cr Total	2,4152	2,5239	2,5155	0,5
10. Fe	2,2177	1,7679	1,9307	5
11. Co	0,4098	0,4323	0,3883	0,4

Gambar 4-20. Hasil Proses Nilai Naik dalam Range Baku Mutu

2. TSS ; hasil analisis menggunakan PSO = 86,0547, SVM = 86,0104 dan PSO-SVM = 86,0395 dengan pembandingan baku mutu air limbah = 100 yang berarti hasil analisis dari data yang dibangkitkan berada dibawah baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO memiliki nilai lebih tinggi dari nilai SVM dan nilai PSO-SVM berada diantara PSO dan SVM. Untuk unsur TSS analisis menggunakan PSO lebih baik dibandingkan SVM maupun PSO-SVM.
3. Cu ; hasil analisis menggunakan PSO = 1,8348, SVM = 1,7770 dan PSO-SVM = 1,8310 dengan pembandingan baku mutu air limbah = 2

yang berarti hasil analisis dari data yang dibangkitkan berada dibawah baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO memiliki nilai lebih tinggi dari nilai SVM dan nilai PSO-SVM berada diantara PSO dan SVM. Untuk unsur Cu analisis menggunakan PSO lebih baik dibandingkan SVM maupun PSO-SVM.

4. Cd ; hasil analisis menggunakan PSO = 0,1222, SVM = -0,0350 dan PSO-SVM = 0,4046 dengan pembanding bakumutu air limbah = 0,05 yang berarti hasil analisis dari data yang dibangkitkan berada di atas baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO-SVM memiliki nilai lebih tinggi dari nilai PSO dan nilai PSO-SVM. Untuk unsur Cd analisis menggunakan PSO-SVM lebih baik dibandingkan PSO maupun SVM.
5. Zn ; hasil analisis menggunakan PSO = 3,6394, SVM = 3,6761 dan PSO-SVM = 3,6965 dengan pembanding bakumutu air limbah = 5 yang berarti hasil analisis dari data yang dibangkitkan berada di bawah baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO-SVM memiliki nilai lebih tinggi dari nilai PSO dan nilai SVM. Untuk unsur Zn analisis menggunakan PSO-SVM lebih baik dibandingkan PSO maupun SVM.
6. Pb ; hasil analisis menggunakan PSO = 0,1038, SVM = 0,0554 dan PSO-SVM = 0,0975 dengan pembanding bakumutu air limbah = 0,1

yang berarti hasil analisis dari data yang dibangkitkan berada di atas baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO memiliki nilai lebih tinggi dari nilai PSO-SVM dan nilai SVM paling rendah. Untuk unsur Pb analisis menggunakan PSO lebih baik dibandingkan PSO-SVM maupun SVM.

7. Ni ; hasil analisis menggunakan PSO = 0,9152, SVM = 0,9712 dan PSO-SVM = 0,9861 dengan pembandingan bakumutu air limbah = 0,5 yang berarti hasil analisis dari data yang dibangkitkan berada di atas baku mutu air. Dari data tersebut terlihat hasil analisis PSO-SVM memiliki nilai lebih tinggi dari nilai SVM dan nilai PSO paling rendah. Untuk unsur Ni analisis menggunakan PSO-SVM lebih baik dibandingkan SVM maupun PSO.
8. Cr<sup>(6+)</sup> ; hasil analisis menggunakan PSO = 0,1586, SVM = 0,1696 dan PSO-SVM = 0,1649 dengan pembandingan bakumutu air limbah = 0,1 yang berarti hasil analisis dari data yang dibangkitkan berada di atas baku mutu air limbah. Dari data tersebut terlihat hasil analisis SVM memiliki nilai lebih tinggi dari nilai PSO dan nilai PSO-SVM. Untuk unsur Cr<sup>(6+)</sup> analisis menggunakan SVM lebih baik dibandingkan PSO maupun PSO-SVM.
9. Cr<sup>(Total)</sup> ; hasil analisis menggunakan PSO = 2,4152, SVM = 2,5239 dan PSO-SVM = 2,5155 dengan pembandingan bakumutu air limbah = 0,5 yang berarti hasil analisis dari data yang dibangkitkan berada di

atas baku mutu air limbah. Dari data tersebut terlihat hasil analisis SVM memiliki nilai lebih tinggi dari nilai PSO dan nilai PSO-SVM. Untuk unsur  $Cr^{(Total)}$  analisis menggunakan SVM lebih baik dibandingkan PSO maupun PSO-SVM.

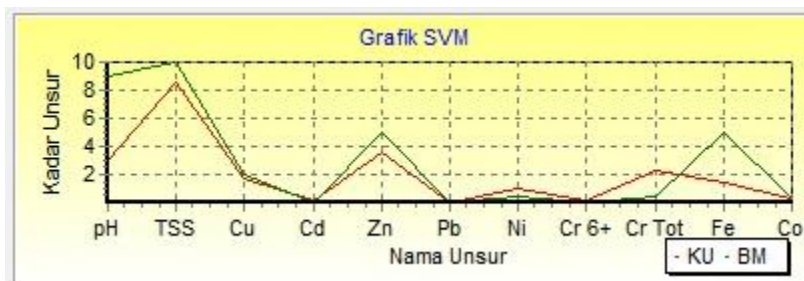
10. Fe ; hasil analisis menggunakan PSO = 2,2177, SVM = 1,7679 dan PSO-SVM = 1,9307 dengan pembanding bakumutu air limbah = 5 yang berarti hasil analisis dari data yang dibangkitkan berada di bawah baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO memiliki nilai lebih tinggi dari nilai SVM dan nilai PSO-SVM berada diantara PSO dan SVM. Untuk unsur Fe analisis menggunakan PSO lebih baik dibandingkan SVM maupun PSO-SVM.
11. Co ; hasil analisis menggunakan PSO = 0,4098, SVM = 0,4323 dan PSO-SVM = 0,3883 dengan pembanding bakumutu air limbah = 0,4 yang berarti hasil analisis dari data yang dibangkitkan berada di bawah baku mutu air limbah untuk SVM dan PSO di atas baku mutu air sementara PSO-SVM berada di bawah baku mutu. Dari data tersebut terlihat hasil analisis SVM memiliki nilai lebih tinggi dari nilai PSO dan nilai PSO-SVM.

Hasil analisis ke-11 unsur tersebut di atas dapat dilihat pada grafik perbandingan nilai hasil analisis yang ditandai dengan garis merah dan baku mutu air limbah ditandai dengan garis hijau. Grafik 4-6

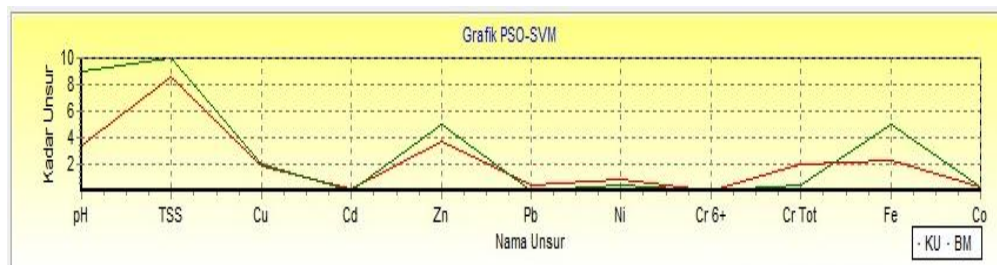
menunjukkan hasil analisis dengan PSO, Grafik 4-7 menunjukkan hasil analisis dengan SVM dan Grafik 4-8 menunjukkan hasil analisis dengan PSO-SVM.



. Grafik 4-6. Hasil Analisis PSO Data Kondisi Kedua



Grafik 4-7. Hasil Analisis SVM Data Kondisi Kedua



Grafik 4-8. Hasil Analisis PSO- SVM Data Kondisi Kedua

Tabel 4-3. Bobot Hasil Analisis Kondisi Kedua

Unsur	PSO	SVM	PSO-SVM
pH	2	1	3
TSS	3	1	2
Cu	3	1	2
Cd	2	1	3
Zn	2	1	3
Pb	3	1	2
Ni	1	2	3
Cr (6+)	1	3	2
Cr Total	1	3	2
Fe	3	1	2
Co	2	3	1
Total	23	18	25

Bobot nilai pada Tabel 4-2 tidak terdapat metode yang dominan terhadap 11 unsur yang dianalisis. PSO memiliki bobot tertinggi pada unsur TSS,Cu,Pb dan Fe dan SVM memiliki bobot yang tinggi pada Cr<sup>(6+)</sup>, Cr<sup>(total)</sup> dan Co. Bobot tertinggi untuk PSO-SVM pada pH, Cd, Zn dan Ni.

### 3. Nilai di Luar Toleransi Batas Atas Baku Mutu

Nilai yang akan dibangkitkan dan dianalisis adalah nilai yang berada diluar range baku mutu dan berada di atas toleransi batas atas. Toleransi batas atas untuk tiap unsur sesuai Tabel 4-1.

Koneksi: <user-PC> [127.0.0.1] tersambung

**Analisis Kandungan Unsur**

Data awal

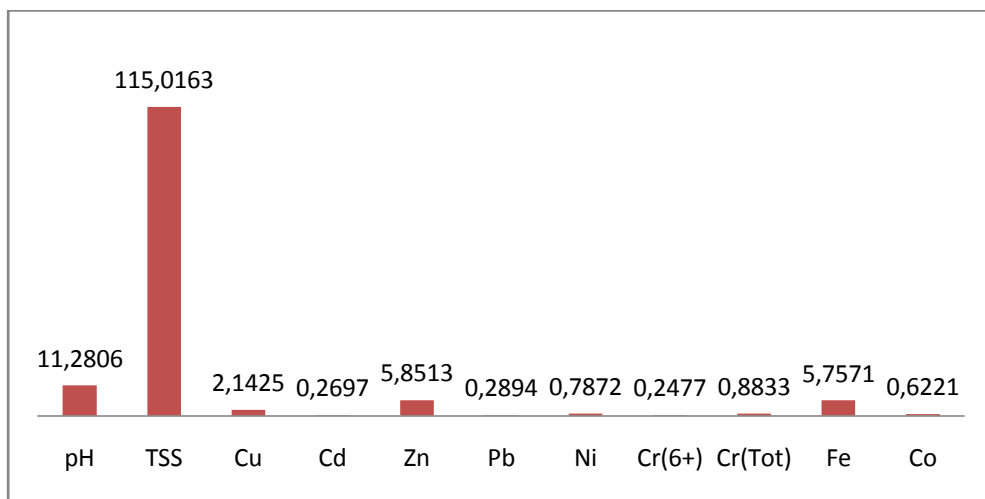
pH	TSS	Cu	Cd	Zn	Pb	Ni	Cr(6+)	Cr Tot	Fe	Co
11,2926	115,0179	2,1447	0,2688	5,8527	0,2916	0,7940	0,2344	0,9249	5,7800	0,6385
11,2907	115,0177	2,1446	0,2668	5,8525	0,2894	0,7929	0,2344	0,9214	5,7753	0,6376
11,2901	115,0176	2,1436	0,2629	5,8523	0,2893	0,7928	0,2341	0,9186	5,7725	0,6371
11,2885	115,0175	2,1428	0,2595	5,8516	0,2882	0,7895	0,2331	0,9176	5,7705	0,6364
11,2875	115,0173	2,1422	0,2568	5,8509	0,2870	0,7877	0,2323	0,9155	5,7681	0,6341
11,2868	115,0172	2,1421	0,2531	5,8504	0,2868	0,7857	0,2312	0,9146	5,7640	0,6312
11,2852	115,0169	2,1418	0,2495	5,8497	0,2855	0,7855	0,2299	0,9134	5,7599	0,6284

Gambar 4-21. Data Awal Aplikasi Analisis di Atas Toleransi Batas Atas

Pada Gambar 4-21 memperlihatkan nilai yang dibangkitkan sebagai data awal dengan nilai di atas toleransi batas atas untuk tiap unsur (Tabel 4-1). Nilai-nilai seperti pada gambar di atas dibangkitkan untuk melihat kinerja aplikasi dalam merespon kondisi nilai yang berada di atas toleransi batas atas.

Terlihat aplikasi merespon dengan baik kondisi nilai yang berada di atas toleransi batas atas dengan melakukan penurunan nilai untuk mencapai kondisi nilai dalam range baku mutu sehingga proses buka pintu pembuangan dapat terjadi. Nilai pH dimulai dengan nilai 11,2926 untuk data pertama, dan nilai data terakhir 11,2806 (Grafik 4-9).

Dengan demikian terjadi penurunan nilai pH, demikian halnya dengan unsur lainnya, Gambar 4-21 memperlihatkan nilai awal yang dibangkitkan dan Grafik 4-9 memperlihatkan nilai akhir dari pembangkitan nilai setiap unsur.



Grafik 4-9. Data Awal pada Kondisi Ketiga

Selanjutnya nilai awal tersebut akan dianalisis untuk melihat kinerja PSO, SVM dan PSO-SVM dalam melakukan analisis terhadap data yang bernilai di atas toleransi batas atas. Hasil analisis diperlihatkan pada Gambar 4-22.



Hasil Proses				
	PSO	SVM	PSO-SVM	BAKUMUTU
1. pH	11,0064	11,2150	11,1701	9
2. TSS	114,9674	115,0040	114,9984	100
3. Cu	2,0853	2,1020	2,0967	2
4. Cd	0,1158	0,1247	0,0498	0,05
5. Zn	5,7943	5,8206	5,797	5
6. Pb	0,2219	0,2542	0,2171	0,1
7. Ni	0,4856	0,7232	0,5425	0,5
8. Cr(6+)	0,11	0,2095	0,1002	0,1
9. Cr Total	0,5006	0,7063	0,4971	0,5
10. Fe	5,0049	5,4859	5,0421	5
11. Co	1,5323	0,5247	0,4457	0,4

Gambar 4-23. Hasil Proses Nilai di Atas Toleransi Batas Atas

Pada Gambar 4-23 terlihat hasil analisis terhadap data yang bernilai di atas toleransi batas atas dari baku mutu dengan penjelasan sebagai berikut :

1. pH ; hasil analisis menggunakan PSO = 11,0064, SVM = 11,2150 dan PSO-SVM = 11,1701 dengan pembanding bakumutu air limbah = 9 yang berarti hasil analisis dari data yang dibangkitkan berada di atas baku mutu air limbah. Dari data tersebut SVM memiliki nilai tertinggi, kemudian PSO-SVM dan paling rendah adalah PSO. Untuk pH bobot nilai tertinggi pada SVM.

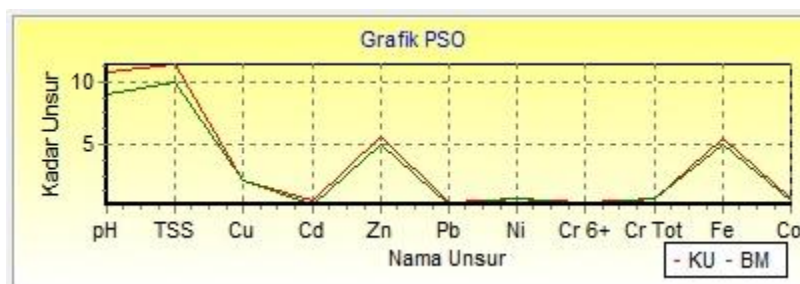
2. TSS ; hasil analisis menggunakan PSO = 114,9674, SVM = 115,0040 dan PSO-SVM = 114,9984 dengan pembanding bakumutu air limbah = 100 yang berarti hasil analisis dari data yang dibangkitkan berada di atas baku mutu air limbah. Dari data tersebut terlihat hasil analisis SVM memiliki nilai lebih tinggi dan kemudian PSO-SVM dan terakhir adalah PSO. Untuk unsur TSS analisis menggunakan SVM lebih baik dibandingkan PSO maupun PSO-SVM.
3. Cu ; hasil analisis menggunakan PSO = 2,0853, SVM = 2,1020 dan PSO-SVM = 2,0967 dengan pembanding bakumutu air limbah = 2 yang berarti hasil analisis dari data yang dibangkitkan berada di atas baku mutu air limbah. Dari data tersebut terlihat hasil analisis SVM memiliki nilai lebih tinggi dari nilai PSO-SVM dan nilai PSO. Untuk unsur Cu analisis menggunakan SVM lebih baik dibandingkan PSO maupun PSO-SVM.
4. Cd ; hasil analisis menggunakan PSO = 0,1158, SVM = 0,1247 dan PSO-SVM = 0,0498 dengan pembanding bakumutu air limbah = 0,05 yang berarti hasil analisis dari data yang dibangkitkan berada di atas toleransi batas atas baku mutu air limbah untuk PSO dan SVM, sementara PSO-SVM berada di bawah toleransi batas atas. Dari data tersebut terlihat hasil analisis SVM memiliki nilai lebih tinggi dari nilai PSO dan nilai PSO-SVM. Untuk unsur Cd analisis menggunakan SVM lebih baik dibandingkan PSO maupun PSO-SVM.

5. Zn ; hasil analisis menggunakan PSO = 5,7943, SVM = 5,8206 dan PSO-SVM = 5,7970 dengan pembandingan baku mutu air limbah = 5 yang berarti hasil analisis dari data yang dibangkitkan berada di atas baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO-SVM memiliki nilai lebih tinggi dari nilai PSO dan nilai SVM. Untuk unsur Zn analisis menggunakan PSO-SVM lebih baik dibandingkan PSO maupun SVM.
6. Pb ; hasil analisis menggunakan PSO = 0,2219, SVM = 0,2542 dan PSO-SVM = 0,2171 dengan pembandingan baku mutu air limbah = 0,1 yang berarti hasil analisis dari data yang dibangkitkan berada di atas baku mutu air limbah. Dari data tersebut terlihat hasil analisis SVM memiliki nilai lebih tinggi dari nilai PSO dan nilai PSO-SVM paling rendah. Untuk unsur Pb analisis menggunakan SVM lebih baik dibandingkan PSO-SVM maupun PSO.
7. Ni ; hasil analisis menggunakan PSO = 0,4856, SVM = 0,7232 dan PSO-SVM = 0,5425 dengan pembandingan baku mutu air limbah = 0,5 yang berarti hasil analisis dari data yang dibangkitkan berada di atas baku mutu air kecuali untuk PSO. Dari data tersebut terlihat hasil analisis SVM memiliki nilai lebih tinggi dari nilai PSO-SVM dan nilai PSO paling rendah. Untuk unsur Ni analisis menggunakan SVM lebih baik dibandingkan PSO maupun PSO-SVM.

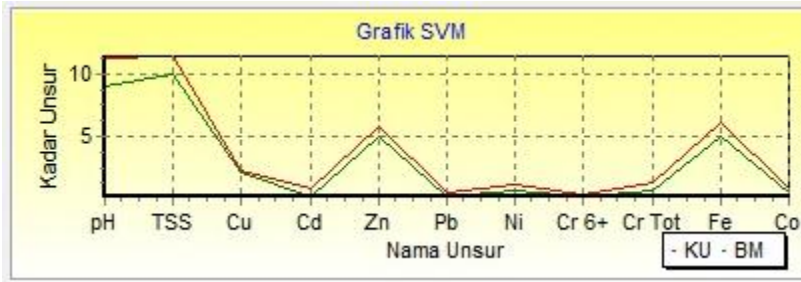
8.  $\text{Cr}^{(6+)}$  ; hasil analisis menggunakan PSO = 0,1100, SVM = 0,2095 dan PSO-SVM = 0,1002 dengan pembanding bakumutu air limbah = 0,1 yang berarti hasil analisis dari data yang dibangkitkan berada di atas baku mutu air limbah. Dari data tersebut terlihat hasil analisis SVM memiliki nilai lebih tinggi dari nilai PSO dan nilai PSO-SVM. Untuk unsur  $\text{Cr}^{(6+)}$  analisis menggunakan SVM lebih baik dibandingkan PSO maupun PSO-SVM.
9.  $\text{Cr}^{(\text{Total})}$  ; hasil analisis menggunakan PSO = 0,5006, SVM = 0,7063 dan PSO-SVM = 0,4971 dengan pembanding bakumutu air limbah = 0,5 yang berarti hasil analisis dari data yang dibangkitkan berada di atas baku mutu air limbah kecuali PSO-SVM. Dari data tersebut terlihat hasil analisis SVM memiliki nilai lebih tinggi dari nilai PSO dan nilai PSO-SVM. Untuk unsur  $\text{Cr}^{(\text{Total})}$  analisis menggunakan SVM lebih baik dibandingkan PSO maupun PSO-SVM.
10. Fe ; hasil analisis menggunakan PSO = 5,0049, SVM = 5,4859 dan PSO-SVM = 5,0421 dengan pembanding bakumutu air limbah = 5 yang berarti hasil analisis dari data yang dibangkitkan berada di atas baku mutu air limbah. Dari data tersebut terlihat hasil analisis SVM memiliki nilai lebih tinggi dari nilai PSO dan nilai PSO-SVM. Untuk unsur Fe analisis menggunakan SVM lebih baik dibandingkan PSO maupun PSO-SVM.

11. Co ; hasil analisis menggunakan PSO = 1,5323, SVM = 0,5247 dan PSO-SVM = 0,4457 dengan pembanding bakumutu air limbah = 0,4 yang berarti hasil analisis dari data yang dibangkitkan berada di atas baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO memiliki nilai lebih tinggi dari nilai SVM dan nilai PSO-SVM.

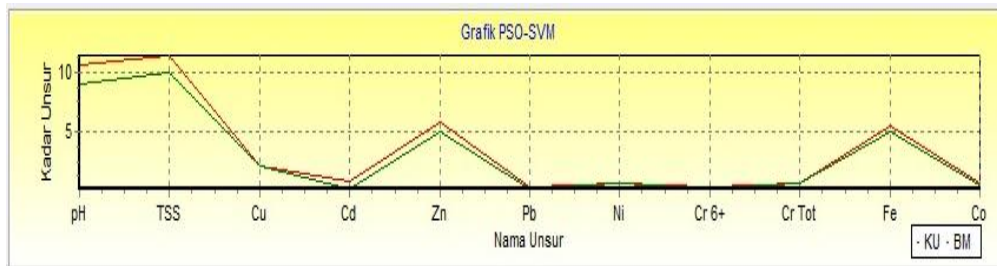
Hasil analisis ke-11 unsur tersebut di atas dapat dilihat pada grafik perbandingan nilai hasil analisis yang ditandai dengan garis merah dan baku mutu air limbah ditandai dengan garis hijau. Grafik 4-10 menunjukkan hasil analisis dengan PSO, Grafik 4-11 menunjukkan hasil analisis dengan SVM dan Grafik 4-12 menunjukkan hasil analisis dengan PSO-SVM.



Grafik 4-10. Hasil Analisis PSO Data Kondisi Ketiga



Grafik 4-11. Hasil Analisis SVM Data Kondisi Ketiga



Grafik 4-12. Hasil Analisis PSO-SVM Data Kondisi Ketiga

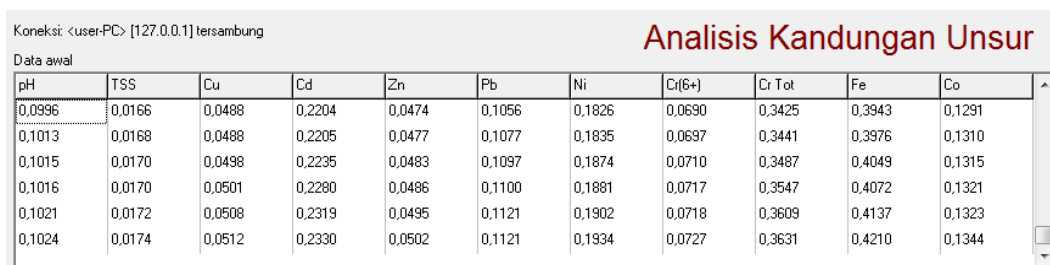
Tabel 4-4. Bobot Hasil Analisis Kondisi Ketiga

Unsur	PSO	SVM	PSO-SVM
pH	1	3	2
TSS	1	3	2
Cu	1	3	2
Cd	2	3	1
Zn	1	3	2
Pb	2	3	1
Ni	1	3	2
Cr (6+)	2	3	1
Cr Total	2	3	1
Fe	1	3	2
Co	3	2	1
Total	17	32	17

Pada Tabel 4-4 terlihat bobot nilai tertinggi pada SVM hampir terdapat pada semua unsur kecuali Co. PSO memiliki nilai tertinggi hanya pada Co, sementara PSO-SVM tidak pernah mendapatkan nilai tertinggi pada kondisi ini.

#### 4. Nilai di Luar Toleransi Batas Bawah Baku Mutu

Nilai yang akan dibangkitkan dan dianalisis adalah nilai yang berada diluar range baku mutu dan berada di bawah toleransi batas bawah. Toleransi batas bawah untuk tiap unsur sesuai Tabel 4-1.



Koneksi: <user-PC> [127.0.0.1] tersambung

Analisis Kandungan Unsur

Data awal

pH	TSS	Cu	Cd	Zn	Pb	Ni	Cr(6+)	Cr Tot	Fe	Co
0,0996	0,0166	0,0488	0,2204	0,0474	0,1056	0,1826	0,0690	0,3425	0,3943	0,1291
0,1013	0,0168	0,0488	0,2205	0,0477	0,1077	0,1835	0,0697	0,3441	0,3976	0,1310
0,1015	0,0170	0,0498	0,2235	0,0483	0,1097	0,1874	0,0710	0,3487	0,4049	0,1315
0,1016	0,0170	0,0501	0,2280	0,0486	0,1100	0,1881	0,0717	0,3547	0,4072	0,1321
0,1021	0,0172	0,0508	0,2319	0,0495	0,1121	0,1902	0,0718	0,3609	0,4137	0,1323
0,1024	0,0174	0,0512	0,2330	0,0502	0,1121	0,1934	0,0727	0,3631	0,4210	0,1344

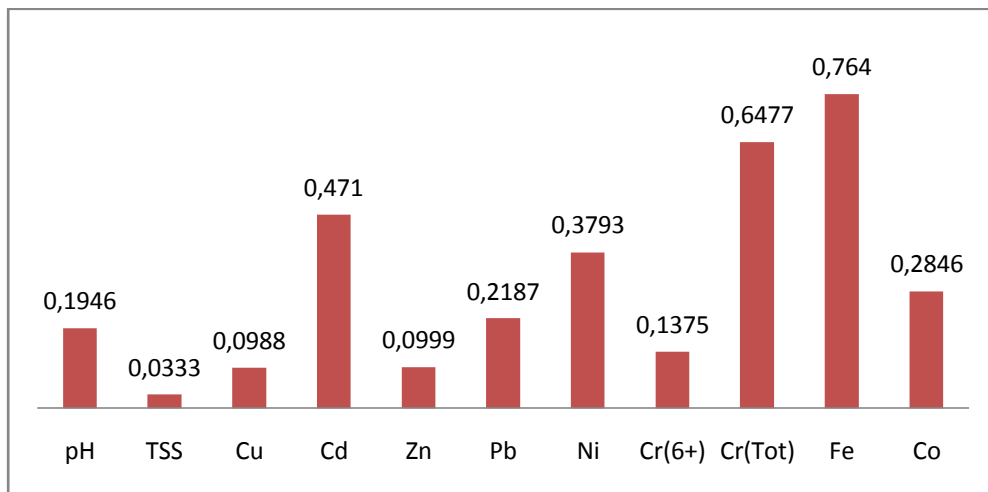
Gambar 4-24. Data Awal Aplikasi Analisis di Atas Toleransi Batas Atas

Pada Gambar 4-24 memperlihatkan nilai yang dibangkitkan sebagai data awal dengan nilai di bawah toleransi batas bawah untuk tiap unsur (Tabel 4-1). Nilai-nilai seperti pada gambar di atas dibangkitkan untuk melihat kinerja aplikasi dalam merespon kondisi nilai yang berada di bawah toleransi batas bawah.

Terlihat aplikasi merespon dengan baik kondisi nilai yang berada di bawah toleransi batas bawah dengan menaikkan nilai untuk mencapai kondisi nilai dalam range baku mutu sehingga proses tutup pintu pembuangan dapat terjadi. Nilai pH dimulai dengan nilai 0,0017 untuk data pertama, dan nilai data terakhir 0,1946 (Grafik 4-13). Dengan demikian terjadi kenaikan nilai pH, demikian halnya dengan unsur



lainnya. Gambar 4-24 memperlihatkan nilai awal yang dibangkitkan dan Grafik 4-13 memperlihatkan nilai akhir dari pembangkitan nilai setiap unsur.



Grafik 4-13. Data Awal pada Kondisi Keempat

Selanjutnya nilai awal tersebut akan dianalisis untuk melihat kinerja PSO, SVM dan PSO-SVM dalam melakukan analisis terhadap data yang bernilai di bawah toleransi batas bawah. Hasil analisis diperlihatkan pada Gambar 4-25.

Hasil Proses				
	PSO	SVM	PSO-SVM	BAKUMUTU
1. pH	0,0649	0,0319	0,0658	9
2. TSS	0,0505	0,0097	0,0379	100
3. Cu	0,0739	0,0213	0,0697	2
4. Cd	0,0414	0,1747	0,0388	0,05
5. Zn	0,0423	0,0226	0,0444	5
6. Pb	0,0999	0,0378	0,0963	0,1
7. Ni	0,3358	0,1506	0,2401	0,5
8. Cr(6+)	0,046	0,0297	0,0617	0,1
9. Cr Total	0,2175	0,2176	0,2879	0,5
10. Fe	0,7815	0,3489	0,4956	5
11. Co	0,0608	0,1344	0,1522	0,4

Gambar 4-25. Hasil Proses Nilai di Bawah Toleransi Batas Bawah

Pada Gambar 4-25 terlihat hasil analisis terhadap data yang bernilai di bawah toleransi batas bawah dari baku mutu dengan penjelasan sebagai berikut :

1. pH ; hasil analisis menggunakan PSO = 0,0649, SVM = 0,0319 dan PSO-SVM = 0,0658 dengan pembanding bakumutu air limbah = 9 yang berarti hasil analisis dari data yang dibangkitkan berada di bawah baku mutu air limbah. Dari data tersebut PSO-SVM memiliki nilai tertinggi, kemudian PSO dan paling rendah adalah SVM. Untuk pH bobot nilai tertinggi pada PSO-SVM.

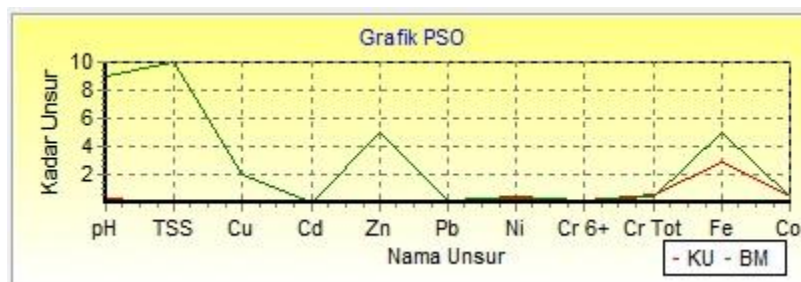
2. TSS ; hasil analisis menggunakan PSO = 0,0505, SVM = 0,0097 dan PSO-SVM = 0,0379 dengan pembanding bakumutu air limbah = 100 yang berarti hasil analisis dari data yang dibangkitkan berada di bawah baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO memiliki nilai lebih tinggi dan kemudian PSO-SVM dan terakhir adalah SVM. Untuk unsur TSS analisis menggunakan PSO lebih baik dibandingkan PSO-SVM maupun SVM.
3. Cu ; hasil analisis menggunakan PSO = 0,0739, SVM = 0,0213 dan PSO-SVM = 0,0697 dengan pembanding baku mutu air limbah = 2 yang berarti hasil analisis dari data yang dibangkitkan berada di bawah baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO memiliki nilai lebih tinggi dari nilai PSO-SVM dan nilai SVM. Untuk unsur Cu analisis menggunakan PSO lebih baik dibandingkan PSO-SVM maupun SVM.
4. Cd ; hasil analisis menggunakan PSO = 0,0414, SVM = 0,1747 dan PSO-SVM = 0,0388 dengan pembanding bakumutu air limbah = 0,05 yang berarti hasil analisis dari data yang dibangkitkan berada di atas toleransi batas bawah baku mutu air limbah untuk PSO dan PSO-SVM, sementara SVM berada di dalam range baku mutu. Dari data tersebut terlihat hasil analisis SVM memiliki nilai lebih tinggi dari nilai PSO dan nilai PSO-SVM. Untuk unsur Cd analisis menggunakan SVM lebih baik dibandingkan PSO maupun PSO-SVM.

5. Zn ; hasil analisis menggunakan PSO = 0,0423, SVM = 0,0226 dan PSO-SVM = 0,0444 dengan pembandingan bakumutu air limbah = 5 yang berarti hasil analisis dari data yang dibangkitkan berada di bawah baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO-SVM memiliki nilai lebih tinggi dari nilai PSO dan nilai SVM. Untuk unsur Zn analisis menggunakan PSO-SVM lebih baik dibandingkan PSO maupun SVM.
6. Pb ; hasil analisis menggunakan PSO = 0,0999, SVM = 0,0378 dan PSO-SVM = 0,0963 dengan pembandingan bakumutu air limbah = 0,1 yang berarti hasil analisis dari data yang dibangkitkan berada di bawah baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO memiliki nilai lebih tinggi dari nilai PSO-SVM dan nilai SVM paling rendah. Untuk unsur Pb analisis menggunakan PSO lebih baik dibandingkan PSO-SVM maupun SVM.
7. Ni ; hasil analisis menggunakan PSO = 0,3358, SVM = 0,1506 dan PSO-SVM = 0,2401 dengan pembandingan bakumutu air limbah = 0,5 yang berarti hasil analisis dari data yang dibangkitkan berada di bawah baku mutu. Dari data tersebut terlihat hasil analisis PSO memiliki nilai lebih tinggi dari nilai PSO-SVM dan nilai SVM paling rendah. Untuk unsur Ni analisis menggunakan PSO lebih baik dibandingkan PSO-SVM maupun SVM.

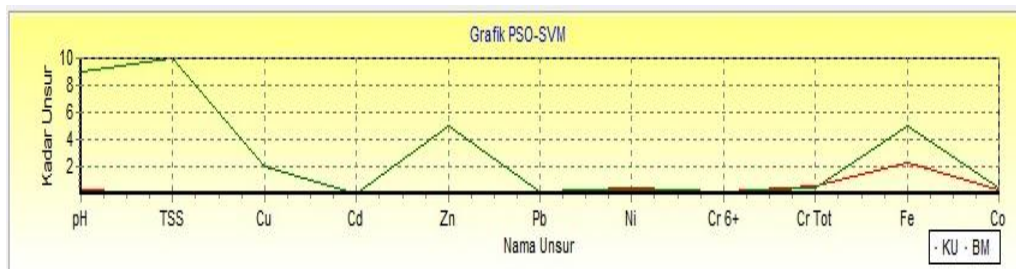
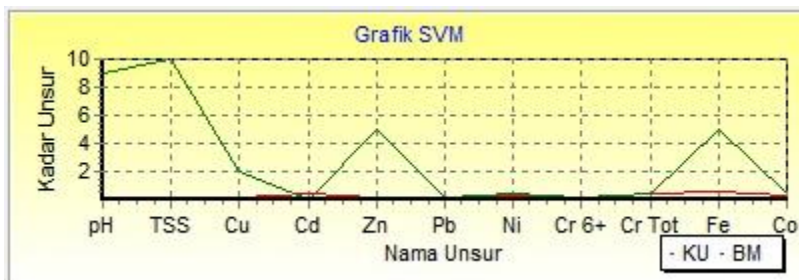
8.  $\text{Cr}^{(6+)}$  ; hasil analisis menggunakan PSO = 0,0460, SVM = 0,0297 dan PSO-SVM = 0,0617 dengan pembandingan baku mutu air limbah = 0,1 yang berarti hasil analisis dari data yang dibangkitkan berada di bawah baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO-SVM memiliki nilai lebih tinggi dari nilai PSO dan nilai SVM. Untuk unsur  $\text{Cr}^{(6+)}$  analisis menggunakan PSO-SVM lebih baik dibandingkan PSO maupun SVM.
9.  $\text{Cr}^{(\text{Total})}$  ; hasil analisis menggunakan PSO = 0,2175, SVM = 0,2176 dan PSO-SVM = 0,2879 dengan pembandingan baku mutu air limbah = 0,5 yang berarti hasil analisis dari data yang dibangkitkan berada di bawah baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO-SVM memiliki nilai lebih tinggi dari nilai PSO dan nilai SVM. Untuk unsur  $\text{Cr}^{(\text{Total})}$  analisis menggunakan PSO-SVM lebih baik dibandingkan PSO maupun SVM.
10. Fe ; hasil analisis menggunakan PSO = 0,7815, SVM = 0,3489 dan PSO-SVM = 0,4956 dengan pembandingan baku mutu air limbah = 5 yang berarti hasil analisis dari data yang dibangkitkan berada di bawah baku mutu air limbah kecuali PSO. Dari data tersebut terlihat hasil analisis PSO memiliki nilai lebih tinggi dari nilai PSO-SVM dan nilai SVM. Untuk unsur Fe analisis menggunakan PSO lebih baik dibandingkan PSO-SVM maupun SVM.

11. Co ; hasil analisis menggunakan PSO = 0,0608, SVM = 0,1344 dan PSO-SVM = 0,1522 dengan pembanding bakumutu air limbah = 0,4 yang berarti hasil analisis dari data yang dibangkitkan berada di bawah baku mutu air limbah. Dari data tersebut terlihat hasil analisis PSO-SVM memiliki nilai lebih tinggi dari nilai SVM dan nilai PSO-SVM.

Hasil analisis ke-11 unsur tersebut di atas dapat dilihat pada grafik perbandingan nilai hasil analisis yang ditandai dengan garis merah dan baku mutu air limbah ditandai dengan garis hijau. Grafik 4-14 menunjukkan hasil analisis dengan PSO, Grafik 4-15 menunjukkan hasil analisis dengan SVM dan Grafik 4-16 menunjukkan hasil analisis dengan PSO-SVM.



Grafik 4-14. Hasil Analisis PSO Data Kondisi Keempat



Grafik 4-16. Hasil Analisis PSO-SVM Data Kondisi Keempat

Tabel 4-5. Bobot Hasil Analisis Kondisi Keempat

Unsur	PSO	SVM	PSO-SVM
pH	2	1	3
TSS	3	1	2
Cu	3	2	1
Cd	2	3	1
Zn	2	1	3
Pb	3	1	2

Ni	3	1	2
Cr (6+)	2	1	3
Cr Total	1	2	3
Fe	3	1	2
Co	1	2	3
Total	25	16	25

Pada Tabel 4-5 terlihat bobot nilai PSO dan PSO-SVM memiliki nilai bobot sama dimana PSO-SVM memiliki nilai tertinggi untuk pH, Zn, Cr<sup>(6+)</sup>, Cr<sup>(total)</sup> dan pada PSO nilai tertinggi pada TSS, Cu, Pb Ni. SVM memiliki nilai tertinggi hanya pada Cd.

Untuk memperoleh kesimpulan terhadap implementasi aplikasi analisis ini maka Tabel 4-6 memperlihatkan jumlah pembobotan masing-masing algoritma.



Tabel 4-6. Rekapitulasi Nilai Bobot pada Empat Kondisi

	PSO	SVM	PSO-SVM
Kondisi I	27	19	20
Kondisi II	23	18	25
Kondisi III	17	32	17
Kondisi IV	25	16	25
Total	92	85	87

Pada Tabel 4-6 memperlihatkan rekapitulasi nilai bobot dari setiap kondisi pengamatan. Dari empat kondisi pengamatan terhadap kinerja aplikasi analisis diperoleh nilai bobot tertinggi pada algoritma PSO sebesar 92, PSO-SVM sebesar 87 dan SVM sebesar 85. Dapat disimpulkan bahwa algoritma paling baik dalam menentukan kelayakan buang limbah pertambangan nikel adalah PSO.

#### **IV.6. Pengujian Sistem**

Pengujian pada penelitian ini menggunakan metode *white box testing* dan *black box testing* untuk memperoleh hasil pengujian yang sesuai dengan indikator keberhasilan proses yang diharapkan.

##### **1. *White Box Testing***

Pengujian *White Box* adalah pengujian untuk mengetahui cara kerja perangkat lunak secara internal dan untuk menjamin operasi-operasi internal bekerja sesuai spesifikasi yang telah ditetapkan.

Metode ini bertujuan mengukur kompleksitas logika dari perancangan prosedur utama. Untuk menghitung tingkat kompleksitas logika program maka digunakan metode *Cyclometric Complexity (CC)*. CC merupakan suatu sistem pengukuran yang menyediakan ukuran kuantitatif dari kompleksitas logika suatu program. Pada *Basis Path Testing*, hasil dari CC digunakan untuk menentukan banyaknya *independent paths*. *Independent path* adalah sebuah kondisi pada program yang menghubungkan *node* awal dengan *node* akhir. Terdapat dua persamaan untuk mengukur CC, yaitu :

$$V(G) = E - N + 2 \text{ atau } V(G) = P + 1, \text{ dimana}$$

$$V(G) = \text{Cyclomatic Complexity untuk flowgraph } G$$

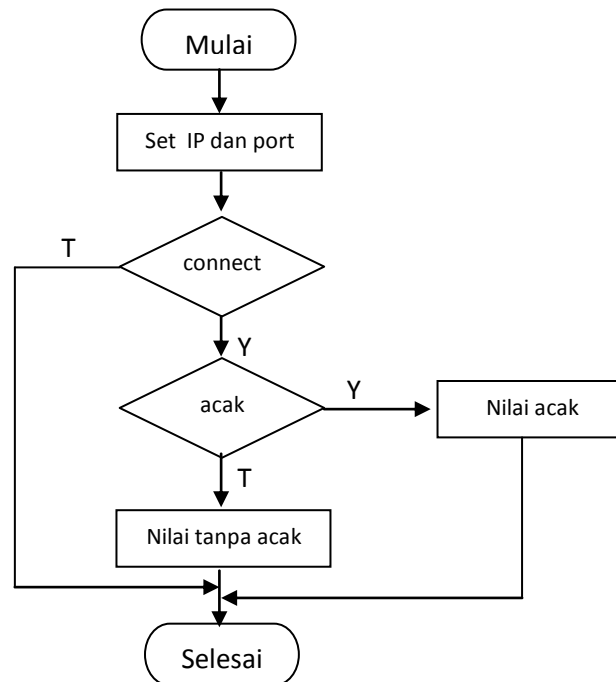
E = Jumlah *edge* (panah) yang menggambarkan aliran kontrol dari setiap *node* dan harus menuju sebuah *node*.

N = Jumlah *node* (lingkaran) yang menggambarkan satu atau lebih perintah prosedural.

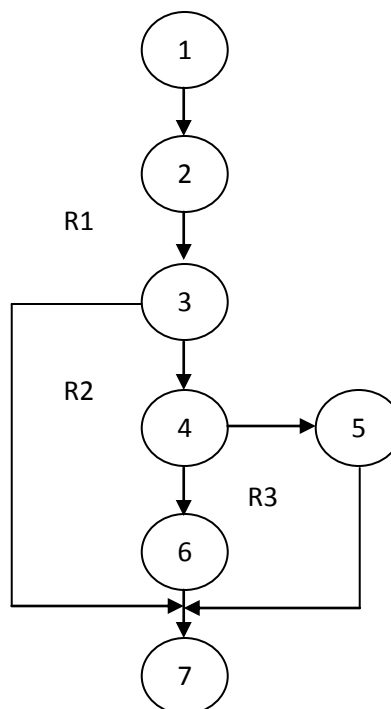
P = Jumlah *predicate node* adalah kondisi yang terdapat pada *node* dan mempunyai karakteristik dua atau lebih *edge* lainnya.

Region = Daerah yang dibatasi *node* dan *edge* dan daerah luar.

a. Pengujian white box pada aplikasi simulator.



Gambar 4-26. Flowchart Aplikasi Simulator



Gambar 4-27. *Flowgraph* Aplikasi Simulator

Perhitungan *Cyclometric Complexity* dari *Flowgraph* Aplikasi Simulator adalah sebagai berikut :

1. Region = 3

2. Menghitung *Cyclometric Complexity* dari *Edge* dan *Node*

dengan rumus :  $V(G) = E - N + 2$

dimana :  $E = 8, N = 7$

penyelesaian :  $V(G) = E - N + 2$

$$= 8 - 7 + 2$$

$$= 3$$

atau

Mengetahui *Cyclometric Complexity* dari P

P adalah titik yang menyatakan logika dalam diagram alir

$$V(G) = P + 1$$

$$= 2 + 1$$

$$= 3$$

3. Path 1 : 1 - 2 - 3 - 7

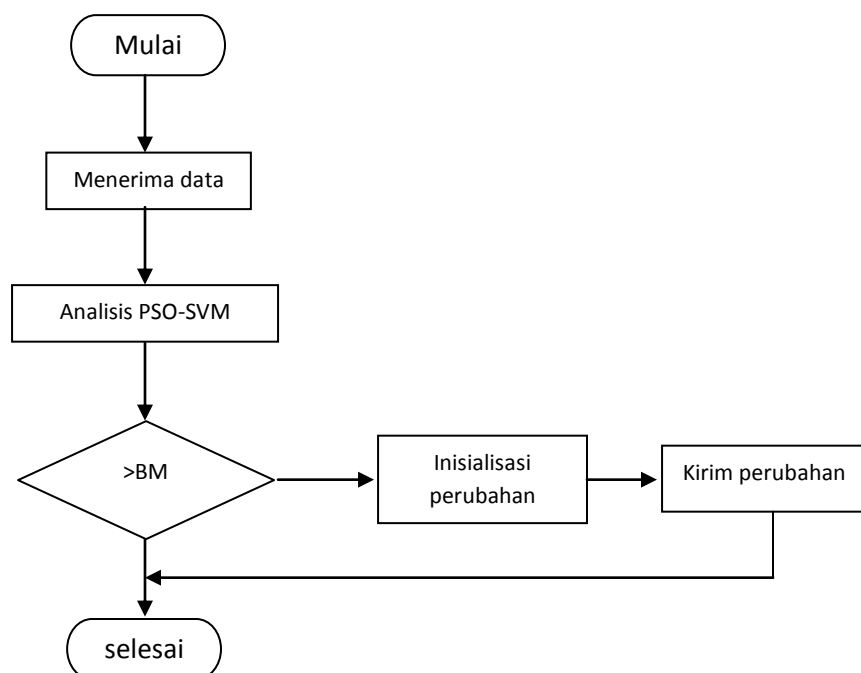
Path 2 : 1 - 2 - 3 - 4 - 5 - 7

Path 3 : 1 - 2 - 3 - 4 - 5 - 6 - 7

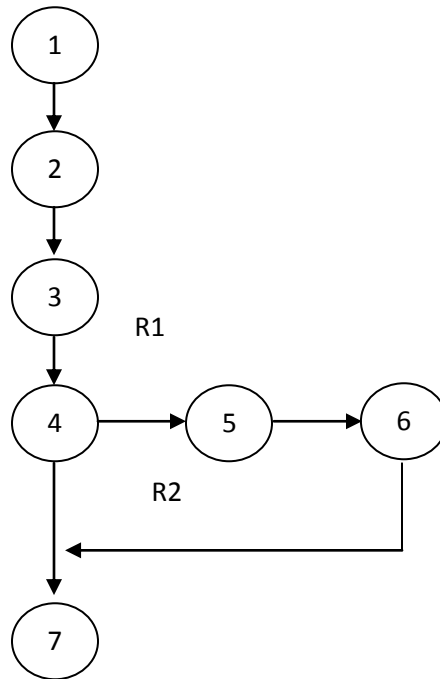
Kesimpulan  $CC = 3$ ,  $Region = 3$  dan  $Path = 3$ , maka aplikasi simulator dinyatakan benar.

b. Pengujian *white box* pada aplikasi analisis.

Pada aplikasi analisis proses berjalan secara otomatis mulai dari menerima data dari aplikasi simulator kemudian dianalisis menggunakan PSO dan SVM. Hasil analisis kemudian dibandingkan dengan baku mutu air limbah. Jika melewati baku mutu air limbah maka aplikasi analisis akan mengirimkan informasi pada aplikasi simulator untuk menurunkan nilai untuk setiap unsur yang melewati baku mutu air limbah. Jika hasil analisis tidak melewati baku mutu air limbah maka aplikasi analisis tidak mengirimkan instruksi perubahan nilai. Berikut pengujian *white box* pada aplikasi analisis.



Gambar 4-28. *Flowchart* Aplikasi Analisis



Gambar 4-29. *Flowgraph* Aplikasi Analisis

Perhitungan *Cyclometric Complexity* dari *Flowgraph* Aplikasi Simulator adalah sebagai berikut :

1. Region = 2
2. Menghitung *Cyclometric Complexity* dari *Edge* dan *Node*

dengan rumus :  $V(G) = E - N + 2$

dimana :  $E = 7, N = 7$

penyelesaian :  $V(G) = E - N + 2$

$$= 0 + 2$$

$$= 2$$

atau

Mengetahui *Cyclometric Complexity* dari P

P adalah titik yang menyatakan logika dalam diagram alir

$$\begin{aligned}V(G) &= P + 1 \\ &= 1 + 1 \\ &= 2\end{aligned}$$

3. Path 1 : 1 – 2 – 3 – 4 – 7

Path 2 : 1 – 2 – 3 – 4 – 5 – 6 – 7

Kesimpulan CC = 2, Region = 2 dan Path = 2, maka aplikasi analisis dinyatakan benar.

## 2. ***Black Box Testing***

Pengujian *Black Box* adalah metode pengujian fungsional dari sebuah aplikasi. Pengujian dilakukan dengan himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program dengan hasil yang sesuai yang diharapkan. Pengujian *black box* merupakan pengujian yang melengkapi pengujian *white box*.

a. Pengujian *Black Box* Aplikasi Simulator.

Tabel 4-7. Pengujian Aplikasi Simulator

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian
1	Melakukan koneksi ke aplikasi analisis (server)	Tombol Connect	Koneksi ke aplikasi analisis tersambung	<input checked="" type="checkbox"/> Terima
			<input type="checkbox"/> Tolak	
		Tombol Disconnect	Koneksi ke aplikasi analisis terputus	<input checked="" type="checkbox"/> Terima
			<input type="checkbox"/> Tolak	
2	Membangkitkan nilai unsur	Tombol Jalankan Detektor	Pembangkitan nilai terjadi sebanyak 1100 data (11x100)	<input checked="" type="checkbox"/> Terima
			<input type="checkbox"/> Tolak	
3	Mengacak nilai yang akan dibangkitkan	Tombol Acak	Nilai yang dibangkitkan teracak	<input checked="" type="checkbox"/> Terima
			<input type="checkbox"/> Tolak	



b. Pengujian *Black Box* Aplikasi Analisis

Tabel 4-8. Pengujian Aplikasi Analisis

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian
1	Menerima data dari aplikasi	Otomatis	Tabel nilai unsur yang dibangkitkan terisi	<input checked="" type="checkbox"/> Terima <input type="checkbox"/> Tolak
2	Menjalankan analisis dengan PSO-SVM	Otomatis	Tabel hasil analisis terisi dan grafik kondisi unsur terbentuk	<input checked="" type="checkbox"/> Terima <input type="checkbox"/> Tolak
3	Perubahan nilai	Otomatis	Nilai unsur yang diubah diterima dari pembangkitan berikutnya menurun.	<input checked="" type="checkbox"/> Terima <input type="checkbox"/> Tolak

Dari hasil pengujian menggunakan *black box* diperoleh hasil bahwa fungsi-fungsi aplikasi yang ada pada aplikasi simulator maupun aplikasi analisis berjalan dengan baik berdasarkan syarat-syarat fungsional input yang telah ditetapkan sebelumnya. Dan dari hasil pengujian menggunakan

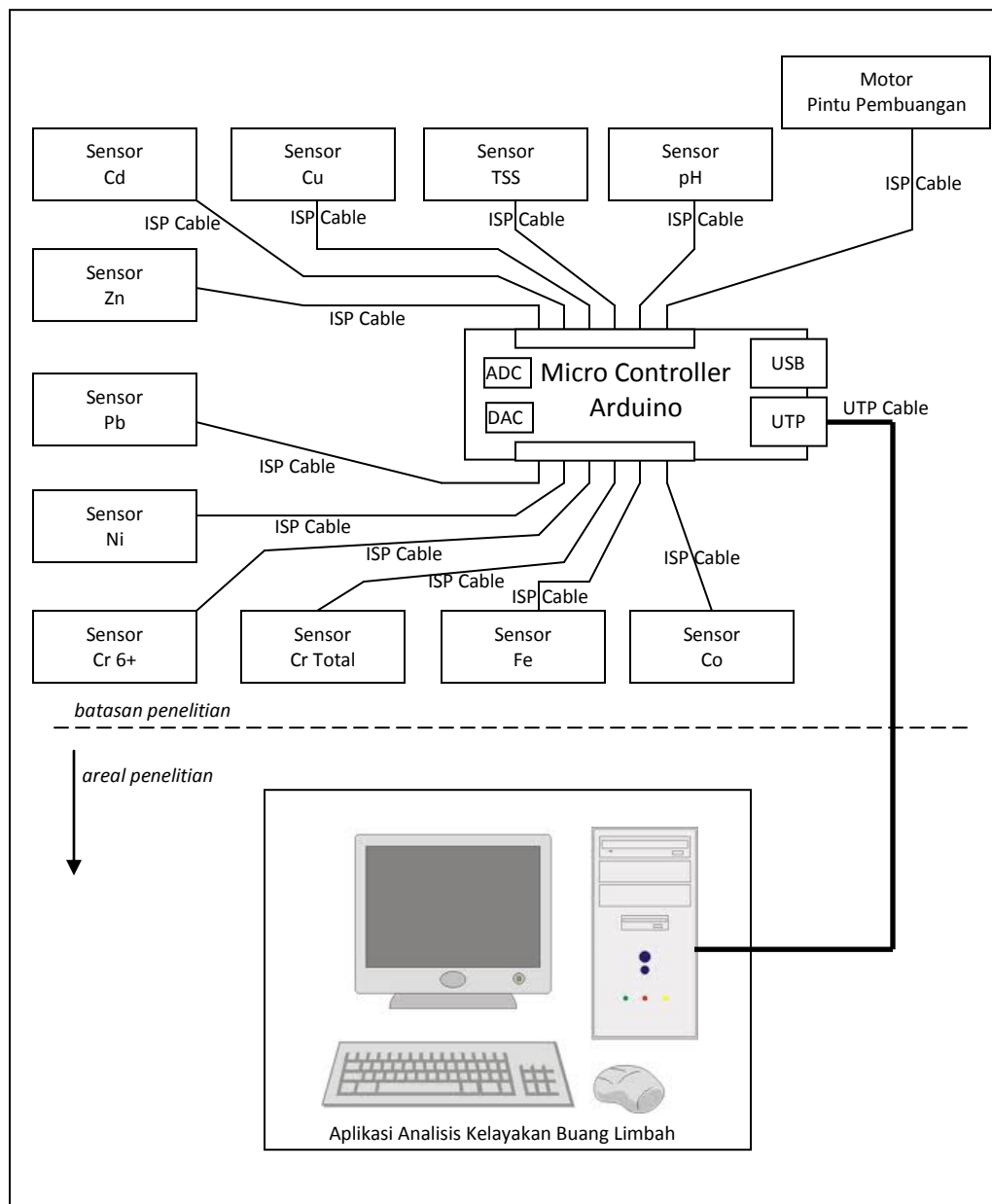
*white box* maupun *black box* diperoleh hasil bahwa secara logika dan fungsional kedua aplikasi berjalan dengan baik.

#### **IV.7. Diskusi *Real Time Measurement***

Faktor penting dalam implementasi otomatisasi adalah *real time* (waktu nyata). *Real time* dapat dipahami sebagai sebuah kondisi pengoperasian dari suatu sistem perangkat keras dan perangkat lunak yang dibatasi oleh rentang waktu dan memiliki batas waktu (*deadline*) yang jelas, terhadap suatu peristiwa atau operasi yang terjadi. Kondisi *real time* yang tidak tercapai akan mengakibatkan kegagalan implementasi otomatisasi.

Pada penelitian ini perangkat keras berupa peralatan detektor/sensor dan mekanik buka tutup pintu pembuangan limbah disimulasikan oleh sebuah aplikasi, yaitu aplikasi simulator. Kondisi ini mengakibatkan pengukuran waktu nyata (*real time measurement*) dalam proses otomatisasi belum dapat dijadikan acuan baku terhadap kinerja aplikasi analisis hubungannya dengan *real time*, karena komunikasi yang terjadi antara aplikasi simulator dan aplikasi analisis adalah komunikasi data digital sehingga komunikasi tersebut dapat terjadi dalam waktu yang relatif lebih cepat, sementara pada kondisi nyata komunikasi antara aplikasi dengan perangkat keras masih harus dikonversi dari data digital menjadi analog dan sebaliknya, sehingga waktu

nyata akan berbeda. Berikut digambarkan bagan arsitektur untuk mengimplementasi *real time measurement* pada kondisi nyata.



Gambar 4-30. Bagan *Real Time Measurement*

Pada Gambar 4-30 diperlihatkan bagan implementasi *real time* untuk kondisi sebenarnya, dimana penelitian yang dilakukan saat ini pada pembuatan aplikasi otomatisasi kelayakan buang limbah dan untuk memperoleh hasil yang maksimal dapat dilakukan penelitian lanjutan seperti gambar arsitektur di atas, sehingga implementasi *real time measurement* untuk otomatisasi kelayakan buang limbah pertambangan nikel dapat dilakukan. Diperlihatkan pada Gambar 4-30 terdapat 11 sensor yang meneteksi kandungan unsur sesuai dengan fungsi masing-masing sensor dan motor penggerak buka-tutup pintu pembuangan limbah. Peralatan-peralatan tersebut terhubung dengan micro controller dan kemudian micro controller terhubung dengan komputer dimana aplikasi analisis kelayakan buang limbah pertambangan nikel terinstall.

Tujuan diskusi ini adalah untuk membandingkan *real time measurment* pada kondisi nyata seperti gambar bagan di atas dengan kondisi dimana bagian sensor dan motor penggerak pintu disimulasikan melalui aplikasi simulator. Pada kondisi aplikasi simulator berkomunikasi dengan aplikasi analisis *real time measurment* diperoleh dalam waktu 15 detik. Hal tersebut tercapai karena komunikasi data yang terjadi adalah komunikasi data digital sehingga tidak terjadi masalah dan hambatan yang berarti. Jika implementasi *real time measurment* pada bagan di atas diperkirakan nilai *real time* akan lebih besar. Asumsi tersebut dikarenakan karakteristik peralatan dan data yang ditransmisikan berbeda. Data dari aplikasi analisis dalam

bentuk data digital untuk sampai pada peralatan harus melalui *micro controller* dimana data digital tersebut harus dikonversi ke dalam bentuk data analog melalui komponen *Digital to Analog Converter (DAC)* yang ada pada *micro controller* dan dan demikian pula sebaliknya. Kandungan unsur yang terdeteksi akan dikirimkan ke aplikasi analisis melalui *micro controller* yang mana data analog tersebut harus diubah terlebih dahulu menjadi data digital melalui *Analog to Digital Converter (ADC)* sehingga data yang terkirim dari peralatan dapat diproses pada aplikasi analisis.

Peneliti berasumsi bahwa akan terjadi perbedaan waktu nyata sebagai akibat dari proses seperti yang telah dipaparkan sebelumnya. Oleh karna perlu dilakukan penelitian lanjutan dan mengimplementasikan *real time measurment* pada bagan yang telah digambarkan pada Gambar 4-30 di atas.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **V.1. Kesimpulan**

Setelah melakukan seluruh langkah penelitian mulai dari analisis sampai dengan testing, maka dapat diambil kesimpulan pada penelitian ini adalah sebagai berikut :

1. Otomatisasi pembuangan limbah dapat dilakukan melalui analisis menggunakan algoritma Particle Swarm Optimization dan Support Vector Machine dengan lama waktu analisis setiap dua menit untuk 1100 data uji yang digunakan.
2. Algoritma yang paling baik dalam melakukan otomatisasi kelayakan buang limbah ini adalah Particel Swarm Optimizaton, dimana dari 11 unsur yang dianalisis tujuh unsur memiliki bobot nilai tertinggi dalam setiap pengukuran yaitu pH, TSS, Cu, Zn, Cr<sup>(6+)</sup>, Cr Total, dan Fe., sementara Support Vector Machine hanya empat unsur yang memiliki nilai bobot tertinggi dari setiap pengukuran, yaitu unsur Cd, Pb, Ni dan Co.
3. Penggabungan algoritma Particle Swarm Optimization dan Support Vector Machine tidak memberikan hasil yang lebih baik dalam

menganalisis kandungan unsur kimia dalam air limbah, karena dalam setiap pengujian tidak pernah mendapatkan bobot nilai tertinggi.

## **V.2. Saran-Saran**

Oleh karena penelitian yang telah dilakukan ini bersifat *off-line* dimana peralatan detektor/sensor disimulasikan dalam bentuk aplikasi maka sangat perlu penelitian ini dilanjutkan menggunakan objek penelitian secara *on-line*, yaitu dengan membuat alat detektor atau sensor kandungan unsur logam dalam air limbah pertambangan nikel sehingga pengukuran secara real-time (*real-time measurement*) dapat diimplementasikan ketika detektor/sensor logam dihubungkan dengan aplikasi analisis kelayakan buang limbah pertambangan nikel.

Penelitian yang mengimplementasi *real-time measurement* secara *on-line* antara alat sensor dan aplikasi sangat penting dilakukan untuk melihat perbedaan waktu antara penentuan kelayakan buang limbah berdasarkan hasil analisis laboratorium dengan penentuan kelayakan buang limbah secara otomatis menggunakan aplikasi analisis kelayakan buang limbah pertambangan nikel.

## **LAMPIRAN**



## Pseudocode

### 1. Pseudocode PSO

```
bangkitkan populasi
set  $w_1, w_2, r_1, r_2$ 
 $x \leftarrow 0$ ;
 $v \leftarrow 0$ ;
inisialisasi pBest, gBest;
for  $i = 1$  to 50 do
begin
    evaluasi fungsi fitness
     $v_{(i)} \leftarrow V_j(i-1) + c_1r_1[pbest, j - x_j(i-1)] + c_2r_2[gbest - x_j(i-1)];$ 
     $v_{(i)} = X(i-1) + V_j(i)$ 
    update fungsi fitness;
end;
```

### 2. Pseudocode SVM

```
Inisialisasi data set untuk  $y_i = +1$  dan  $y_i = -1$ 
for  $i = 1$  to 60 do
begin
    hitung nilai solusi  $w, b$  untuk semua data set
    hitung  $f(x) = w \cdot x + b$  untuk semua  $y = +1$ 
    for  $j = 1$  to  $(y_i + 1)$  do
    begin
        if  $(\sum_{i \in I} (1+y_i))/2 = 0$ 
            hitung  $i^* = \arg \max_{i \in I} f_i$ 
            set  $y_i^* = +1$ 
    end;
end;
output nilai  $w$  dan  $b$ 
for  $i = 61$  to 100 do
begin
    if  $w \cdot x_i + b > 1$  then
        hitung margin
        set  $y_i = +1$ 
```

```
end;
```

### Listing Program Aplikasi Simulator

#### 1. Listing pembangkitan nilai

```
procedure TClient.Button3Click(Sender: TObject);
var ii,jj,yy : integer;
begin
  // nilaiawal;
  nilai:=0;
  label5.Caption := '0';
  memo2.Lines.Clear;
  memo3.Lines.Clear;
  //nilai atur = 1 adalah naik, 0 adalah turun
  yy:=0;

  for ii := 1 to 11 do
  begin
    tampung[ii] := memo3.Lines[yy];
    yy:=yy+1;
  end;

  if cup < 1 then
  begin
    //bangkitkan nilai random
    nilairand[1] := 0.003;
    nilairand[2] := 0.02;
    nilairand[3] := 0.009;
    nilairand[4] := 0.005;
    nilairand[5] := 0.007;
    nilairand[6] := 0.0022;
    nilairand[7] := 0.0041;
    nilairand[8] := 0.0013;
    nilairand[9] := 0.0063;
    nilairand[10] := 0.0081;
    nilairand[11] := 0.0029;
    //aktivasi nilai unsur awal
    for ii := 1 to 100 do
    begin
      unsur[1]:=unsur[1]+random*nilairand[1];
      memo2.Lines.Add(format('%.*f',[4,unsur[1]]));
      nilai:=nilai+1;
    end;
  end;
end;
```

```

unsur[2]:=unsur[2]+random*nilairand[2];
memo2.Lines.Add(format('%.*f',[4,unsur[2]]));
nilai:=nilai+1;
unsur[3]:=unsur[3]+random*nilairand[3];
memo2.Lines.Add(format('%.*f',[4,unsur[3]]));
nilai:=nilai+1;
unsur[4]:=unsur[4]+random*nilairand[4];
memo2.Lines.Add(format('%.*f',[4,unsur[4]]));
nilai:=nilai+1;
unsur[5]:=unsur[5]+random*nilairand[5];
memo2.Lines.Add(format('%.*f',[4,unsur[5]]));
nilai:=nilai+1;
unsur[6]:=unsur[6]+random*nilairand[6];
memo2.Lines.Add(format('%.*f',[4,unsur[6]]));
nilai:=nilai+1;
unsur[7]:=unsur[7]+random*nilairand[7];
memo2.Lines.Add(format('%.*f',[4,unsur[7]]));
nilai:=nilai+1;
unsur[8]:=unsur[8]+random*nilairand[8];
memo2.Lines.Add(format('%.*f',[4,unsur[8]]));
nilai:=nilai+1;
unsur[9]:=unsur[9]+random*nilairand[9];
memo2.Lines.Add(format('%.*f',[4,unsur[9]]));
nilai:=nilai+1;
unsur[10]:=unsur[10]+random*nilairand[10];
memo2.Lines.Add(format('%.*f',[4,unsur[10]]));
nilai:=nilai+1;
unsur[11]:=unsur[11]+random*nilairand[11];
memo2.Lines.Add(format('%.*f',[4,unsur[11]]));
nilai:=nilai+1;
end;
end
else
begin
for ii := 1 to 100 do
begin
for jj := 1 to 11 do
begin
if tampung[jj] = '1' then
unsur[jj]:=unsur[jj]+random*nilairand[jj]
else
unsur[jj]:=unsur[jj]-random*nilairand[jj];

memo2.Lines.Add(format('%.*f',[4,unsur[jj]]));

```

```

        nilai:= nilai+1;
    end;
end;
end;

cup := cup + 1;
label6.Caption := 'Deteksi ke-'+inttostr(cup);
label5.Caption := inttostr(nilai);
Button4Click(Sender);
end;

```

2. Listing Buka Tutu Pintu Pembuangan Limbah

```

procedure TClient.Timer1Timer(Sender: TObject);
begin
    label7.Caption := inttostr(memo3.Lines.Capacity );
    if memo3.Lines[12] = '1' then
        panel1.Caption := 'PINTU TERTUTUP'
    else
        panel1.Caption := 'PINTU TERBUKA';
    Button3Click(Sender);
end;

```

3. Listing acak nilai yang akan dibangkitkan

```

procedure TClient.Button5Click(Sender: TObject);
var ik,jk : integer;
begin
    for ik := 1 to 10 do
    begin
        for jk := 1 to 15 do
        begin
            unsur[1] := 2+random(12);
            unsur[2] := 5 + random(120);
            unsur[3] := random*0.018 + random * 3.0;
            unsur[4] := random*0.00044 + random * 0.3;
            unsur[5] := random*0.074 + random * 4.0;
            unsur[6] := random*0.0029 + random * 0.3;
            unsur[7] := random*0.0031 + random * 1.0;
            unsur[8] := random*0.0086 + random * 0.3;
            unsur[9] := random*0.065 + random * 2.3;
            unsur[10] := random*0.12 + random * 5.5;
            unsur[11] := random*0.0062 + random * 0.5;

```

```
end;  
end;  
  
end;
```

### Listing Aplikasi Analisis

1. Listing penerimaan data dari aplikasi detektor.

```
procedure TCaptureData.Button1Click(Sender: TObject);  
var k : integer;  
    xmax,xmin,x,v : real;  
  
begin  
    //membersihkan array  
    fillchar(arr,SizeOf(arr),0);  
  
    //mengisi data tercapture di memo1 ke array (arr)  
    jj:=0;  
    for i := 1 to 100 do  
    begin  
        for j := 1 to 11 do  
        begin  
            arr[i,j] := memo1.Lines[jj];  
            jj := jj+1;  
        end;  
    end;  
  
    //menampilkan isi array ke dalam stringgrid1.  
    for i := 1 to 100 do  
    begin  
  
        for j := 1 to 11 do  
        begin  
  
            if i = 1 then  
                StringGrid1.Cells[i-1,0] := 'pH';  
            if i = 2 then  
                StringGrid1.Cells[i-1,0] := 'TSS';  
            if i = 3 then  
                StringGrid1.Cells[i-1,0] := 'Cu';  
            if i = 4 then
```

```

    StringGrid1.Cells[i-1,0] := 'Cd';
  if i = 5 then
    StringGrid1.Cells[i-1,0] := 'Zn';
  if i = 6 then
    StringGrid1.Cells[i-1,0] := 'Pb';
  if i = 7 then
    StringGrid1.Cells[i-1,0] := 'Ni';
  if i = 8 then
    StringGrid1.Cells[i-1,0] := 'Cr(6+)';
  if i = 9 then
    StringGrid1.Cells[i-1,0] := 'Cr Tot';
  if i = 10 then
    StringGrid1.Cells[i-1,0] := 'Fe';
  if i = 11 then
    StringGrid1.Cells[i-1,0] := 'Co';

  stringgrid1.cells[j-1,i] := arr[i,j];

end;
  StringGrid1.RowCount:=i+1;

end;
//membersihkan memo1 sebagai penampung data tercapture sementara
memo1.Lines.Clear;

//memulai proses algoritma PSO-SVM
//=====
//=====

//1. set generasi (k) = 0

k := 0;

//2. inisialisasi
// kecepatan (Vk-i) untuk semua partikel masing2 unsur....
fillchar(velo,SizeOf(velo),0);

xmax := 1.0;
xmin := 0.1;
for i := 1 to 100 do
begin
  for j := 1 to 11 do
  begin

```

```

    v := xmin + random*(xmax-xmin);
    velo[i,j]:= floattostr(v);
    // stringgrid2.cells[j-1,i] := velo[i,j];
end;
// StringGrid2.RowCount:=i+1;
end;

// posisi (Xk-i) untuk semua partikel masing2 unsur...
fillchar(posi,SizeOf(posi),0);

xmax := 1.0;
xmin := 0.1;
for i := 1 to 100 do
begin
    for j := 1 to 11 do
    begin
        x := xmin + random*(xmax-xmin);
        posi[i,j]:= floattostr(x);
        // stringgrid3.cells[j-1,i] := posi[i,j];
    end;
    // StringGrid3.RowCount:=i+1;

end;
BitBtn1Click(Sender);
Button2Click(Sender);

end;

```

## 2. Listing analisis PSO, SVM dan PSO-SVM

```

procedure TCaptureData.BitBtn1Click(Sender: TObject);
begin
    BitBtn2Click(Sender);

    //menjalankan algoritma PSO
    //evaluasi fungsi tujuan setiap partikel  $f(x) = (\text{bakumutu} - x)^2$ 
    //menghitung fungsi Fi

    for i := 1 to 100 do
    begin
        for j := 1 to 11 do

```

```

    fisbl[i,j] := format('%.*f',[4, (SQR(strtfloat(bakumutu[j])-strtfloat(arr[i,j])))) ]]);
end;
//set kecepatan awal v=0
for i := 1 to 100 do
begin
    for j := 1 to 11 do
        vsbl[i,j] := floattostr(0);
    end;
//set posisi awal x sbl
for i := 1 to 100 do
begin
    for j := 1 to 11 do
        xsbl[i,j] := arr[i,j];
    end;

//set pbest awal
for i := 1 to 100 do
begin
    for j := 1 to 11 do
        pbest[i,j] := arr[i,j];
    end;

//set gbest awal
for i := 1 to 100 do
begin
    for j := 1 to 11 do
        gbest[j] := arr[25,j];
    end;

//proses iterasi
for iter := 1 to 50 do
begin
//hitung velocity
for i := 1 to 100 do
begin
    for j := 1 to 11 do
begin
        vsdh[i,j] := format('%.*f',[4, ( strtfloat(vsbl[i,j]) + ( strtfloat(velo[25,j]) * (
strtfloat(pbest[i,j]) - strtfloat(xsbl[i,j]) ) ) +
( strtfloat(velo[20,j]) * ( strtfloat(gbest[j]) - strtfloat(xsbl[i,j]) ) ) ) ]]);
        end;
    end;
end;

//hitung posisi

```



```

for i := 1 to 100 do
begin
for j := 1 to 11 do
begin
xsdh[i,j] := floattostr( strtofloat(xsbl[i,j]) + strtofloat(vsdh[i,j]) );
end;
end;

//evaluasi fungsi tujuan (Fi) === fisdh
for i := 1 to 100 do
begin
for j := 1 to 11 do
begin
fisdh[i,j] := format('%.*f',[4,(SQR(strtoffloat(bakumutu[j])-strtfloat(xsdh[i,j])) )]);
end;
end;

//update pbest dan gbast
for i := 1 to 100 do
begin
for j := 1 to 11 do
begin
if strtfloat(fisbl[i,j]) < strtfloat(fisdh[i,j]) then
begin
pbest[i,j] := xsbl[i,j];
gbest[j] := xsbl[i,j];
end
else
begin
pbest[i,j] := pbest[i,j];
gbest[j] := gbest[j];
end;
end;

//update Fi, v dan x
fisbl[i,j] := fisdh[i,j];
vsbl[i,j] := vsdh[i,j];
xsbl[i,j] := xsdh[i,j];
end;
end;

//akhir dari iterasi.....
end;

```

```

//tampilkan grafik
for i := 3 to 11 do
begin
  Chart1.Series[0].Add(strtfloat(xsbl[100,i]));
  Chart1.Series[1].Add(strtfloat(bakumutu[i]));
end;

end;

procedure TCaptureData.BitBtn2Click(Sender: TObject);
var sv : real;
begin
  // 4. menjalankan SVM dengan dataset yang telah terinisialisasi pada algoritma PSO
  // Training SVM untuk mendapatkan fungsi fitness sebagai gbest pada PSO
  series1.Clear;
  series2.Clear;
  LineSeries1.Clear;
  LineSeries2.Clear;
  for i := 1 to 11 do
  begin
    gamma[i] := floattostr(abs( strtfloat(posi[1,i])-strtfloat(posi[2,i])));
    if strtfloat(posi[1,i]) > strtfloat(posi[2,i]) then
      svm[i] := floattostr(strtfloat(posi[2,i]) + (abs(strtfloat(posi[1,i])-
strtfloat(posi[2,i]))/2))
    else
      svm[i] := floattostr(strtfloat(posi[1,i]) + (abs(strtfloat(posi[1,i])-
strtfloat(posi[2,i]))/2));
    end;

    sv := 1;
    for i := 3 to 100 do
    begin
      for j := 1 to 11 do
      begin

        if strtfloat(posi[i,j]) > strtfloat(svm[j]) then
          begin
            if (strtfloat(posi[i,j])-strtfloat(svm[j])) < strtfloat(gamma[j]) then
              begin
                gamma[j] := floattostr(strtfloat(posi[i,j])-strtfloat(svm[j]));
                hassv[j] := posi[i,j];
                tamsv[j] := arr[i,j];
              end
          end
        end;
      end;
    end;
  end;
end;

```

```

        else

        end
        else
        end;
    end;

    for i := 3 to 11 do
    begin
        Chart2.Series[0].Add(strtfloat(tamsv[i]));
        Chart2.Series[1].Add(strtfloat(bakumutu[i]));
    end;

end;

procedure TCaptureData.Button2Click(Sender: TObject);
begin
    lineseries3.Clear;
    lineseries4.Clear;
    //menjalankan algoritma PSO
    //evaluasi fungsi tujuan setiap partikel  $f(x) = (bakumutu - x)^2$ 
    //menghitung fungsi Fi
    for i := 1 to 100 do
    begin
        for j := 1 to 11 do
            fisblps[i,j] := format('%.*f',[4, (SQR(strtfloat(bakumutu[j])-strtfloat(arr[i,j])))) ]);
        end;
        //set kecepatan awal v=0
        for i := 1 to 100 do
        begin
            for j := 1 to 11 do
                vsblps[i,j] := floattostr(0);
            end;
            //set posisi awal x sbl
            for i := 1 to 100 do
            begin
                for j := 1 to 11 do
                    xsblps[i,j] := arr[i,j];
                end;
            end;

            //set pbest awal
            for i := 1 to 100 do
            begin

```

```

    for j := 1 to 11 do
        pbestps[i,j] := arr[i,j];
    end;

//set gbest awal
for i := 1 to 100 do
begin
    for j := 1 to 11 do
        gbestps[j] := tamsv[j];
    end;

//proses iterasi
for iter := 1 to 50 do
begin
    //hitung velocity
    for i := 1 to 100 do
    begin
        for j := 1 to 11 do
        begin
            vsdhps[i,j] := format('%.*f',[4, ( strtofloat(vsbpls[i,j]) + ( strtofloat(velo[25,j]) * (
            strtofloat(pbestps[i,j]) - strtofloat(xsbpls[i,j]) ) ) +
            ( strtofloat(velo[20,j]) * ( strtofloat(gbestps[j]) - strtofloat(xsbpls[i,j]) ) ) ) ]));
        end;
    end;

//hitung posisi
for i := 1 to 100 do
begin
    for j := 1 to 11 do
    begin
        xsdhps[i,j] := floattostr( strtofloat(xsbpls[i,j]) + strtofloat(vsdhps[i,j]) );
    end;
end;

//evaluasi fungsi tujuan (Fi) === fisdh
for i := 1 to 100 do
begin
    for j := 1 to 11 do
    begin
        fisdhps[i,j] := format('%.*f',[4,(SQR(strtofload(bakumutu[j])-strtofloat(xsdhps[i,j]))))
]);
    end;
end;
end;

```

```

//update pbest dan gbast
for i := 1 to 100 do
begin
for j := 1 to 11 do
begin
if strtofloat(fisblps[i,j]) < strtofloat(fisdhps[i,j]) then
begin
pbestps[i,j] := xsblps[i,j];
gbestps[j] := xsblps[i,j];
end
else
begin
pbestps[i,j] := pbestps[i,j];
gbestps[j] := gbestps[j];
end;

//update Fi, v dan x
fisblps[i,j] := fisdhps[i,j];
vsblps[i,j] := vsdhps[i,j];
xsblps[i,j] := xsdhps[i,j];
end;
end;

//akhir dari iterasi.....
end;

```

```

//tampilkan grafik

```

```

for i := 3 to 11 do
begin
Chart3.Series[0].Add(strtfloat(xsblps[100,i]));
Chart3.Series[1].Add(strtfloat(bakumutu[i]));
end;
//mengisi tabel hasil proses
for i := 1 to 11 do
begin
for j := 1 to 4 do
begin
if j=1 then
hasilakhir[i,1] := xsbl[100,i];
if j=2 then
hasilakhir[i,2] := tamsv[i];

```

```

    if j=3 then
        hasilakhir[i,3] := xsblps[100,i];
    if j=4 then
        hasilakhir[i,4] := bakumutu[i];

        stringgrid2.cells[j,i] := hasilakhir[i,j];
    end;
    StringGrid2.RowCount:=i+1;
end;

for i := 1 to 11 do
begin

    for j := 1 to 4 do
    begin
        if i = 1 then
            StringGrid2.Cells[i,0] := 'PSO';
        if i = 2 then
            StringGrid2.Cells[i,0] := 'SVM';
        if i = 3 then
            StringGrid2.Cells[i,0] := 'PSO-SVM';
        if i = 4 then
            StringGrid2.Cells[i,0] := 'BAKUMUTU';

        stringgrid2.cells[j,i] := hasilakhir[i,j];
    end;
    StringGrid2.RowCount:=i+1;
end;

    StringGrid2.Cells[0,1] := ' 1. pH';
    StringGrid2.Cells[0,2] := ' 2. TSS';
    StringGrid2.Cells[0,3] := ' 3. Cu';
    StringGrid2.Cells[0,4] := ' 4. Cd';
    StringGrid2.Cells[0,5] := ' 5. Zn';
    StringGrid2.Cells[0,6] := ' 6. Pb';
    StringGrid2.Cells[0,7] := ' 7. Ni';
    StringGrid2.Cells[0,8] := ' 8. Cr(6+)';
    StringGrid2.Cells[0,9] := ' 9. Cr Total';
    StringGrid2.Cells[0,10] := '10. Fe';
    StringGrid2.Cells[0,11] := '11. Co';

//Analisis kandungan unsur terkandung
for i := 1 to 11 do

```

```

begin
  if tp = 0 then
    begin
      kondisi[i] := 1;
    end
  else
    begin
      if (strtofloat(hasilakhir[i,3]) > (strtofloat(bakumutu[i]) + (strtofloat(bakumutu[i])/4)))
then
      kondisi[i] := 0
      else
      kondisi[i] := kondisi[i];
      if strtofloat(hasilakhir[i,3]) < (strtofloat(bakumutu[i]) -
(strtofloat(bakumutu[i])/1.15)) then
      kondisi[i] := 1;
      end;
    end;
    tp := tp + 1;

    //mengisi kondisi ke memo2
    memo2.Lines.Clear;
    for i := 1 to 11 do
      begin
        memo2.Lines.Add(inttostr(kondisi[i]));
      end;

    //BUKA TUTUP PINTU

    label5.Caption := inttostr(memo2.Lines.Count);
    btnBroadcastClick(Sender);

  end;

```