Implementasi Deteksi Kesalahan Pada Pabrik Kimia Secara *Real-Time* Dengan *On-The-Fly Semi-Supervised Learning* Berbasis *Support Vector Machines*

Reza Andriady [a,1,*], Dr.-Ing. Awang Noor Indra Wardana, S.T., M.T., M.Sc.[a,2], Nopriadi, S.T., M.Sc., Ph.D.[a,3]

[a] Program Studi Teknik Fisika, Departemen Teknik Nuklir Teknik Fisika Fakultas Teknik Universitas Gadjah Mada, Jalan Grafika 02, Sleman 55281, Indonesia
[1] reza.andriady@mail.ugm.ac.id*; [2] awang.wardana@ugm.ac.id; [3] nopriadi@ugm.ac.id

ARTICLE INFO

Article history
Received
Revised
Accepted

Keywords
deteksi kesalahan
support vector machines
semi-supervised learning
on-the-fly learning

ABSTRACT

Deteksi kesalahan berbasis data pada pabrik kimia umumnya membutuhkan kumpulan data berjumlah besar berisikan sampel-sampel ketika pabrik beroperasi secara normal dan ketika terdapat kesalahan pada proses pabrik, yang kemudian tiap sampel diberi label yang sesuai oleh ahli. Penelitian ini akan merancang dan menguji performa dari metode deteksi kesalahan dengan *on-the-fly semi-supervised learning* menggunakan metode *cluster-then-label*, di mana pelatihan model deteksi kesalahan dapat dilakukan sembari pabrik beroperasi, sehingga cocok untuk diterapkan apabila pabrik belum memiliki kumpulan data. Metode yang dirancang akan menggunakan data referensi dan penyangga, yang mana data referensi adalah sampel-sampel pertama dalam jumlah yang telah ditentukan sebelumnya sebagai acuan data normal bagi program, dan data penyangga berisi sampel yang masuk setelahnya. Digunakan metode KFDA untuk pra-pemrosesan, dan k-means untuk pengelompokan yang dilanjutkan oleh SVM untuk klasifikasi. Berdasarkan data latih, program dibangun dengan parameter yakni ukuran data referensi sebesar 150 sampel, ukuran penyangga sebesar 50, dan γ sebesar $5x10^{-5}$. Didapatkan hasil bahwa perlu dilakukan penyesuaian parameter terlebih dahulu agar didapatkan hasil yang optimum. Pada pengujian menggunakan data latih, parameter disesuaikan sehingga γ sebesar $5x10^{-4}$, sedangkan pada data validasi, parameter disesuaikan sehingga ukuran data referensi sebesar 350 sampel dan γ sebesar $5x10^{-3}$. Dengan penyesuaian parameter, deteksi kesalahan dapat dilakukan pada kedua data.

1. Pendahuluan

Dengan berkembangnya industri modern, kualitas produk dan keselamatan merupakan faktor krusial pada proses manufaktur, namun terjadinya kesalahan pada proses dapat mengancam kedua hal tersebut. Kesalahan didefinisikan sebagai perilaku abnormal pada proses yang berhubungan dengan kegagalan mesin, kelelahan mesin, atau gangguan ekstrim pada proses [1]. Dengan deteksi kesalahan, kesalahan dalam proses – proses pabrik yang kompleks dapat dideteksi sedini mungkin.

Umumnya, pembangunan model mesin belajar untuk deteksi kesalahan menggunakan teknik supervised learning yang membutuhkan kumpulan data dalam jumlah yang besar, berisi dengan sampel ketika pabrik beroperasi secara normal dan sampel ketika terdapat kesalahan dalam proses pabrik, yang tiap sampelnya sudah diberi label oleh para ahli untuk memberikan keterangan apakah sebuah sampel merupakan sampel normal atau sampel salah [2]. Kumpulan data tersebut digunakan sebagai data latih oleh strategi deteksi kesalahan, namun, proses pengumpulan kumpulan data merupakan tugas yang membutuhkan waktu yang lama.

Pada berbagai penelitian, SVM dilaporkan memiliki akurasi yang tinggi dalam melakukan deteksi kesalahan. Penelitian yang dilakukan oleh J. Liu dkk [3] meneliti performa SVM untuk mendeteksi kesalahan pada sistem pengereman Kereta Berkecepatan Tinggi. Meskipun dilatih menggunakan data yang tidak berimbang dengan mayoritas data latih merupakan data normal, SVM menghasilkan akurasi yang memuaskan. Pada 15 dataset yang digunakan, rangka kerja SVM yang ditawarkan menghasilkan 12 nilai F-*measurement* dan 9 nilai G-*mean* yang lebih tinggi dibandingkan 3 metode pembanding. K.-Y. Chen dkk [4] menguji SVM untuk mendeteksi kesalahan pada pembangkit listrik termal dengan SVM dengan kernel fungsi basis radial. SVM mengalahkan performa metode *Linear Discriminant Analysis* (79.95%) dan *Back Propagation Neural Network* (85.57%), dengan SVM dapat mendeteksi kesalahan mengidentifikasi jenis kesalahan pada pembangkit listrik termal dengan akurasi diatas 91.93%.

Pada penelitian yang telah disebutkan, data latih dan data uji diolah terlebih dahulu dengan metode reduksi dimensi untuk menyingkat waktu komputasi. Selain itu reduksi dimensi juga dapat mentransformasikan data yang tidak dapat dipisahkan secera linier menjadi dapat dipisahkan. Penelitian Zhu dan Song [5] menguji implementasi *kernel fisher discriminant analysis* (KFDA) pada deteksi kesalahan menggunakan data proses Tennessee Eastman. Pada penelitian tersebut, data proses Tennessee Eastman diolah menggunakan *principal component analysis*, *kernel principal component analysis*, *fisher discriminant analysis*, dan KFDA sebagai metode ekstraksi fitur sehingga data diproyeksikan kedalam ruang fitur 2 dimensi dan kemudian diklasifikasikan menggunakan metode klasifikasi. Pada penelitian tersebut, KFDA terbukti dapat memproyeksikan data yang tidak dapat dipisahkan secara linier menjadi 4 kluster yang terpisah pada ruang fitur 2 dimensi sehingga data tersebut menjadi terpisah secara linier. Secara keseluruhan, ekstraksi fitur dengan KFDA memiliki akurasi klasifikasi sebesar 92.75% menggunakan *gaussian mixture model* dan 90,81% menggunakan *k-nearest neighbor*.

Selain metode *supervised*, terdapat pula metode *unsupervised* yang dapat melakukan karakterisasi data tanpa menggunakan label. Metode *unsupervised* merupakan metode yang tidak membutuhkan pengetahuan sebelumnya mengenai keluaran atau label dari sebuah data. Fungsi dari metode *unsupervised* adalah memberikan perkiraan keluaran berdasarkan struktur dari kumpulan data yang disediakan [2]. Penelitian pada [6] mengajukan metode deteksi kesalahan pada proses produksi wafer semikonduktor. Metode yang diajukan dapat melakukan deteksi kesalahan secara *unsupervised* dengan *on-the-fly learning* yang kemudian dipasang pada sistem proses produksi sehingga dapat mendeteksi kesalahan proses secara *real-time*. Metode tersebut menggunakan data-data histori dari proses untuk mendapatkan acuan data normal yang kemudian digunakan untuk membangun model seleksi fitur dan *classifier* berupa metode *unsupervised learning* yakni 1-*Class* SVM yang hanya membutuhkan data dari 1 kelas untuk proses pembangunan model. Data yang telah diolah dimasukkan ke dalam sebuah *sliding window* sebagai data referensi. Pada penelitian ini mampu mendeteksi kesalahan pada 2 data industri dengan akurasi yang memuaskan, pada dataset 1 dihasilkan akurasi sebesar 95.65% pada kedua skenario *sliding window*, sedangkan pada dataset 2 dihasilkan akurasi sebesar 83.3 dan 91.67 pada masing-masing skenario.

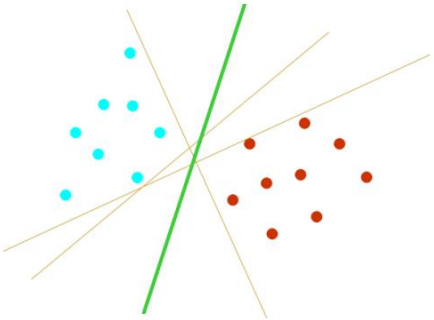
Semi-supervised learning mengombinasikan metode *supervised* dan *unsupervised* untuk saling melengkapi kekurangannya. Penelitian [2] menyebutkan bahwa pembangunan model secara semi-*supervised* menggunakan data berlabel dan tidak berlabel menghasilkan akurasi yang lebih tinggi dibandingkan metode *supervised* maupun *unsupervised*. Salah satu metode *semi-supervised* adalah metode *cluster-then-label*, yaitu metode yang menggunakan metode pengelompokan untuk memberikan *pseudo-label* dari sebuah dataset, *pseudo-label* tersebut kemudian digunakan oleh metode *classifier* untuk menentukan label akhir dari tiap data dan untuk membentuk batas pemisah. Penelitian [7] menguji metode pengelompokan yang diikuti oleh metode klasifikasi untuk mendeteksi anomali, yang mendapatkan hasil bahwa gabungan metode k-*medoid* dan metode SVM mendapatkan rata-rata akurasi sebesar 99.43% pada dataset kecil (10^4 sampel) dan 99.21% pada dataset besar (10^5 sampel).

Pada penelitian ini, teknik *semi-supervised learning* akan dilakukan secara *on-the-fly* menggunakan metode *cluster-then-label* yang terpisah seperti pada penelitian [7], sehingga pembangunan model deteksi kesalahan dapat dilakukan sembari program beroperasi. Dengan begitu, program deteksi kesalahan yang dirancang pada penelitian ini dapat digunakan untuk melakukan deteksi kesalahan apabila kumpulan data berlabel belum tersedia.

2. Metode Yang Diusulkan

2.1. *Support Vector Machines*

Support vector machines (SVM) merupakan pendekatan statistik multivarian yang relatif baru dan menjadi populer karena memiliki hasil yang banyak diminati pada persoalan klasifikasi dan regresi [8]. Prinsip fundamental dari SVM adalah membentuk hyperplane (batas keputusan) antar kelas yang harus memiliki jarak maksimum antara support vector tiap kelas. *Support vector* sendiri merupakan data pada masing-masing kelas yang bertindak sebagai batas dari kelas tersebut.



Gambar 1 Klasifikasi Menggunakan SVM [9]

Gambar 1 menunjukkan beberapa titik data yang merupakan anggota dari 2 kelas, anggota dari kelas pertama disimbolkan dengan lingkaran berwarna biru, dan anggota dari kelas kedua disimbolkan dengan lingkaran berwarna merah. Meggunakan Gambar 1 sebagai contoh, persoalan klasifikasi dapat diartikan sebagai persoalan untuk menemukan batas yang dapat memisahkan kedua kelas tersebut sehingga apabila sebuah data baru dimasukkan kedalam figur, kelas dari data baru dapat diprediksi berdasarkan letak data tersebut. Terdapat banyak kemungkinan garis batas yang dapat digunakan, pada Gambar 1 terdapat beberapa contoh alternatif garis batas.

Diasumsikan bahwa himpunan data yang memiliki 2 kelas berada didalam matriks X dengan ukuran matriks m x n, m merupakan jumlah sampel yang diamati dan n merupakan jumlah variabel yang diamati. Tiap sampel pada X dinotasikan sebagai x_i yang merupakan vektor baris ke-i pada X dan berukuran ukuran n. Tiap sampel pada X diasumsikan merupakan anggota dari salah satu diantara 2 kelas, yaitu kelas positif dan kelas negatif. Sehingga sebuah vektor kolom Y berukuran m bertindak sebagai label kelas untuk tiap sampel, dengan baris ke-i pada Y atau dinotasikan sebagai (y_i) dapat memiliki 2 nilai, yaitu +1 dan -1. Apabila y_i bernilai +1, maka sampel x_i merupakan anggota dari kelas positif, begitu pula sebaliknya. Diasumsikan bahwa data tersebut dapat dipisahkan menggunakan *hyperplane* dengan n dimensi, yang didefinisikan sebagai:

$$f(x) = w.x + b = 0 \quad (1)$$

w merupakan vektor berdimensi m dan b merupakan skalar. Parameter w dan b akan menentukan posisi dan orientasi dari hyperplane batas. Persoalan untuk menentukan hyperplane batas yang optimum merupakan sebuah persoalan optimasi, di mana hyperplane batas yang optimum harus memenuhi 2 persamaan berikut:

$$y_i f(x_i) = y_i(\langle w, x_i \rangle + b) \geq 1, i = 1, \ldots, m, \text{ dan} \quad (2)$$

$$\min_{w,b} = \frac{1}{2}\|w\|^2 \quad (3)$$

⟨w,x⟩ menotasikan inner product dari vektor w dan vektor x Untuk memecahkan persoalan optimasi pada persamaan (3) dengan batasan persamaan (2), dapat digunakan teknik *Langrange Multiplie*r, menjadi:

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i(y_i(\langle w, x_i \rangle + b) - 1) \quad (4)$$

Dengan α merupakan *Langrangian Multiplier.*

Pada keadaan riil, sering kali data tidak dapat diklasifikasikan secara sempurna, sehingga persamaan optimasi menjadi tidak dapat dipenuhi. Untuk mengatasi masalah ini, digunakan teknik *soft margin* dan persamaan ditulis kembali dengan menyisipkan variabel *slack* (ξ_i), sehingga persamaan optimasi menjadi:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \tag{5}$$

$$L(w,b,\xi,\alpha,\beta) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i(y_i(\langle w,x_i\rangle + b) - 1 + \xi_i) - \sum_{i=1}^{m}\beta_i\xi_i \tag{6}$$

2.2. Pengelompokan K-*means*

Proses k-*means* merupakan proses yang membagi sebuah populasi data kedalam k kelompok berbasis sebuah sampel. Aplikasi dari proses k-*means* meliputi pengelompokan berdasarkan kemiripan, prediksi non-linier, memperkirakan distribusi multivariabel dan uji ketergantungan dari beberapa variabel secara non-parametrik [10].

Aplikasi proses k-*means* pada pengelompokan berdasarkan kemiripan disebut sebagai pengelompokan k-*means*, yang merupakan metode pengelompokan yang banyak digunakan. Pengelompokan k-*means* bertujuan untuk mencari nilai minimal dari rerata jarak kuadrat antara pusat kelompok dengan sebuah data pada kelompok tersebut atau disebut sebagai *within cluster sum of squares* (WCSS). Pada pengelompokan k-*means*, ditinjau konstanta *k* sebagai jumlah kelompok yang telah ditentukan, matriks *X* sebagai kumpulan data berukuran matriks *m* x *n* dengan fitur sebanyak *m* dan sampel sejumlah *n*, maka pengelompokan k-*means* akan memiliki *objective function* (ϕ) sebagai berikut:

$$\phi = \sum_{j=1}^{k}\sum_{i=1}^{m}\left\|x_i^{(j)} - c_j\right\|^2 \tag{7}$$

Dengan c_j merupakan pusat kelompok ke-*j* dan $x_i^{(j)}$ merupakan sebuah sampel dari *X* yang termasuk kedalam c_j. Dari persamaan tersebut, diartikan bahwa J merupakan jumlah dari jarak kuadrat antara tiap pusat kelompok ke tiap sampel yang termasuk kedalam kelompok tersebut, dan pengelompokan k-*means* bertujuan untuk mencari nilai minimum dari ϕ. Algoritma dari pengelompokan k-*means* adalah sebagai berikut:

1. Dipilih pusat kelompok (*c*) sebanyak *k* secara acak dari sampel-sampel pada kumpulan data yang disediakan.
2. Untuk tiap $j \in \{1, 2, .. k\}$ dan $i \in \{1, 2, .. m\}$, golongkan tiap sampel (x_i) dari *X* sebagai anggota dari kelompok ke-*j* (C_j) apabila x_i lebih dekat ke c_j daripada pusat kelompok lain.
3. Untuk tiap $j \in \{1, 2, .. k\}$, tetapkan c_j sebagai mean dari C_j.
4. Ulangi tahap 2 dan 3 hingga *C* tidak berubah.

Dengan algoritma yang iteratif seperti yang telah disebutkan, pengelompokan k-*means* akan mencari posisi untuk tiap c_j yang menghasilkan ϕ yang minimum. Pada penelitian ini, digunakan metode k-*means*++ yang diperkenalkan pada penelitian [7] sehingga pusat kelompok pertama dipilih tidak secara acak, namun dengan iterasi tersendiri. Tahapan k-*means*++ dilakukan sebagai berikut:

1a. Tentukan pusat kelompok pertama c_1, dipilih secara acak dari sampel pada *X*.
1b. Tentukan pusat kelompok berikutnya c_i, yang dipilih secara acak dari sampel pada *X* dengan probabilitas $\frac{D(x)}{\sum_{x\epsilon X} D(x)^2}$.
1c. Ulangi tahap 1b, hingga c_k telah ditentukan
2-4. Dijalankan sama seperti algortima k-*means*.

2.3. *Kernel Fisher Discriminant Analysis*

Reduksi dimensi digunakan untuk mengurangi jumlah fitur (dimensi) pada sebuah himpunan data. Metode ini penting karena dapat memfasilitasi visualisasi dalam sebuah figur (2 dimensi atau 3 dimensi), mengurangi waktu yang dibutuhkan untuk komputasi classifier, serta meningkatkan akurasi klasifikasi [11]. Salah satu metode ekstraksi fitur yang sering digunakan adalah *fisher discriminant analysis (*FDA), metode tersebut banyak digunakan dan terbukti memiliki performa baik dalam berbagai aplilkasi. Namun, karena batasannya pada persoalan-persoalan linier, FDA memiliki performa yang buruk pada data yang tidak dapat dipisahkan secara linier [12]. Beberapa peneliti kemudian mengembangkan FDA yang dapat memiliki performa baik pada data yang tidak dapat dipisahkan secara linier, salah satunya adalah dengan menggunakan metode kernel dan disebut dengan *Kernel Fisher Discriminant Analysis* (KFDA).

Yang [13] pada penelitiannya merumuskan KFDA sebagai metode 2 tahap, yaitu *kernel principal component analysis* (KPCA) dan kemudian dilanjutkan dengan FDA. Diasumsikan sebuah himpunan data bernotasi *X* dengan *n* jumlah variabel, maka pada tahap KPCA, dicari ruang fitur kernel dari himpunan data menggunakan persamaan dari metode kernel yang digunakan.

Matriks transformasi KPCA dapat dihitung dengan persamaan:

$$z_{KPCA} = \left(\frac{\nu_1}{\sqrt{\lambda_1}}, \dots, \frac{\nu_n}{\sqrt{\lambda_n}}\right)^T \quad (8)$$

Dengan ν dan λ merupakan *eigenvalue* dan *eigenvector* dari K, dan K merupakan matriks $\widetilde{K}$ yang terpusat. Menggunakan perkalian *dot product* dengan z_{KPCA}, *X* dapat diproyeksikan kedalam ruang fitur baru dan dinotasikan sebagai $\hat{X}$, yang mana $\hat{X}$ kemudian digunakan untuk membentuk matriks transformasi KFDA (z_{KFDA}) pada tahap kedua, yaitu tahap FDA. z_{KFDA} didapatkan menerapkan FDA pada z_{KPCA} sehingga didapatkan *eigenvector* (φ) dari $\langle S^{-1}, Cov\rangle$, dengan *Cov* merupakan matriks kovarians dari mean antar kelas dan *S* merupakan variansi dari $\hat{X}$ pada tiap vektor fitur pada ruang fitur z_{KPCA}.

$$Cov = \frac{1}{m}\sum_{i=1}^{c} l_i (M_{c_i} - M_{\hat{X}})(M_{c_i} - M_{\hat{X}})^T \quad (9)$$

$$S = diag(\lambda_1, \lambda_2, \dots, \lambda_m) \quad (10)$$

Dengan *m* menotasikan jumlah sampel pada *X*, *c* menotasikan jumlah kelas, l_i menotasikan jumlah sampel pada kelas ke-*i*, M_{c_i} menotasikan *mean* dari kelas ke-*i*,dan M_0 menotasikan mean dari $\hat{X}$. Sebuah data baru bernotasi *x* dapat diproyeksikan pada ruang fitur KFDA dengan persamaan:

$$\check{x} = {z_{KFDA}}^T . ({z_{KPCA}}^T . x) \quad (11)$$

2.4. *Robust Scaler*

Penskalaan merupakan metode yang biasanya dilakukan pada tahap pra-proses data . Dengan menggunakan penskalaan, maka nilai tiap variabel pada sebuah data diubah menjadi memiliki rentang yang sama sehingga semua variabel memiliki proporsi yang sama [14]. *Robust scaler* didapatkan menggunakan persamaan:

$$x_i\ terskala = \frac{x_i - Q_1(f_i)}{Q_3(f_i) - Q_1(f_i)} \quad (12)$$

2.5 Metode Deteksi Kesalahan

Program memiliki 4 tahapan utama, yakni pengolahan referensi, ekstraksi fitur, pengelompokan data, dan klasifikasi data. Data dikirim melalui protokol komunikasi MQTT satu per satu untuk meniru aktifitas sensor pada pabrik dengan waktu cuplikan tertentu.

Data referensi adalah kelompok sampel yang diterima pada awal program digunakan, kelompok sampel ini akan dianggap sebagai acuan data normal bagi program dan disimpan oleh program. Data referensi kemudian juga digunakan untuk membentuk matriks KFDA dan *robust scaler*. Data referensi diperlukan karena dengan adanya data referensi normal, apabila terjadi sebuah sampel diproyeksikan terlalu jauh dari proyeksi data referensi, maka program deteksi kesalahan akan mengklasifikasikan sampel tersebut sebagai data salah. Sedangkan penyangga merupakan kelompok sampel yang diterima setelah referensi terisi, dengan penyangga akan secara terus menerus diisi oleh sampel baru yang masuk, hal tersebut sesuai dengan strategi *sliding windows*. Akurasi dari deteksi kesalahan akan bervariasi tergantung pada ukuran referensi dan penyangga, dikarenakan matriks KFDA dibentuk oleh data referensi sehingga penggunaan ukuran referensi akan berbeda akan menghasilkan proyeksi sampel yang berbeda pula.

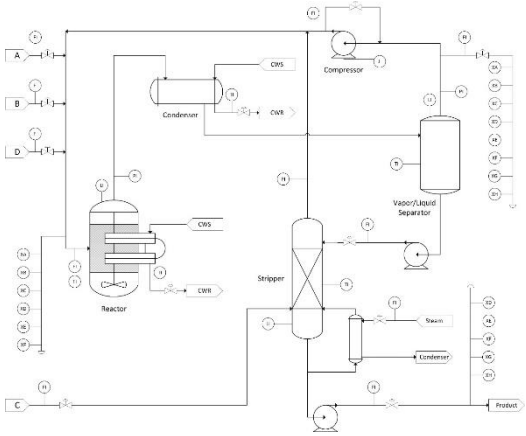
Data dari MQTT akan dimasukkan ke dalam data referensi hingga ukuran data referensi terpenuhi untuk kemudian digunakan untuk membangun matriks KFDA dan *robust scaler* sehingga didapatkan proyeksi data referensi pada ruang KFDA yang telah diskala. Data dari MQTT kemudian dimasukkan kedalam data penyangga, apabila data penyangga sudah terpenuhi ukurannya, maka data penyangga akan diolah menggunakan matriks KFDA dan *robust scaler* yang telah dibangun. Proyeksi data referensi dan data penyangga kemudian diolah menggunakan metode *semi-supervised* yaitu *cluster-then-label*, dimana tahap pengelompokan dan klasifikasi dilakukan secara terpisah, pengelompokan dilakukan menggunakan pengelompokan k-*means* dan klasifikasi dilakukan menggunakan SVM.

3. Metodologi

3.1. Bahan Penelitian

Digunakan 2 kumpulan data pada penelitian ini, yakni kumpulan data dari Tennessee Eastman Chemical Company dan data dari penelitian Brooks. Kumpulan data dari Tennessee Eastman Chemical Company memiliki 52 variabel dan digunakan untuk membangun program deteksi kesalahan pada penelitian ini sebagai data latih, sedangkan kumpulan data dari penelitian Brooks memiliki 20 variabel digunakan sebagai validasi dari program yang sudah dibangun. Kedua data didapatkan melalui gudang data South African Council of Control Automation.

1. Data Latih



Gambar 2 Diagram P&ID Data Latih [14]

Gambar 2 merupakan diagram perpipaan dan instrumentasi dari data latih yang digunakan pada penelitian ini, yaitu data Tennessee Eastman Process. Data tersebut memiliki 52 variabel yang tediri dari 22 variabel proses, 11 variabel yang dimanipulasi, serta 19 variabel dari pengukuran komposisi. Proses Tennessee Eastman memiliki 5 unit operasi utama, reactor, kondensor, separator, kompresor daur ulang, dan stripper. Reaktor akan menerima reaktan (A, D, E) dalam bentuk gas, di mana reaktan akan bereaksi. Produk keluar dari reactor dalam bentuk uap bersamaan dengan umpan yang tidak bereaksi. Produk kemudian masuk kedalam kondensor sehingga produk terkondensasi menjadi

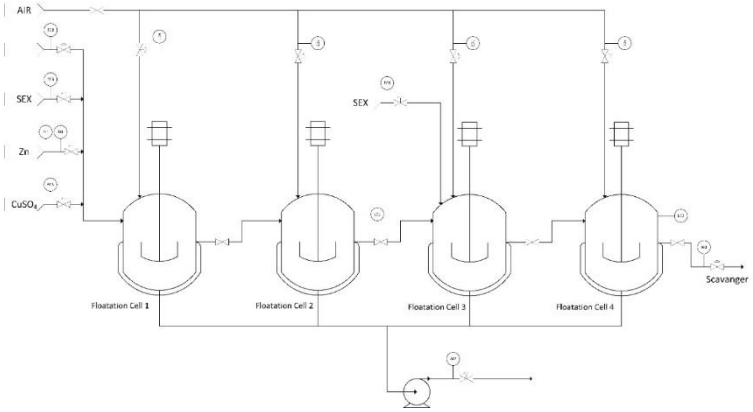
campuran uap dan cairan. Campuran tersebut kemudian akan masuk ke separator untuk dipisahkan ke proses yang berbeda. Komponen uap dari separator akan diumpankan Kembali ke reaktor oleh kompresor, sedangkan komponen uap akan masuk ke stripper untuk dipisahkan dari sisa reaktan untuk menghasilkan produk akhir G dan H [15].

Tabel 1 Deskripsi Jenis Data pada Data Latih

Jenis Data	**Deskripsi**
Data normal	Pabrik beroperasi dalam keadaan standar
Data salah tipe 2	Perubahan komposisi B, rasio A/C konstan
Data salah tipe 5	Perubahan suhu pada masukan pendingin kondensor
Data salah tipe 6	Penurunan aliran umpan A
Data salah tipe 7	Penurunan tekanan umpan C

Tabel 2.1 merupakan deskripsi dari tipe data yang digunakan sebagai data latih, yaitu data normal, data salah tipe 2, 5, 6 dan 7. Data salah tipe 2, 5, 6, dan 7 didapatkan dengan memberikan perubahan berupa masukan step pada salah satu variabel proses tertentu sesuai deskripsi yang tertera. Digunakan kumpulan data sebanyak 2420 sampel dengan rincian data normal sebanyak 500 sampel, data salah tipe 2 sebanyak 480 normal, data salah tipe 5 sebanyak 480 sampel, data salah tipe 7 sebanyak 480 sampel, dan data salah tipe 14 sebanyak 480 sampel.

2. Data Validasi



Gambar 3 Diagram P&ID Data Validasi [15]

Gambar 3 merupakan diagram perpipaan dan instrumentasi dari penelitian dilakukan oleh Brooks pada tahun 2018 [16] bertujuan untuk menguji performa metode data-driven dari sebuah paket perangkat lunak komersil untuk melakukan deteksi kesalahan dan rekonstruksi data pada sebuah kumpulan data sensor pabrik. Diagram menjelaskan alur proses dan instrumentasi dari pabrik tersebut. Proses terdiri dari 4 flotation cell yang berfungsi untuk memisahkan bijih seng sulfida dari mineral lain. 4 sel tersebut membentuk apa yang disebut sebagai rougher bank. Pada tiap sel, dengan mengalirkan air bersamaan dengan bijih yang telah dihancurkan kedalam feed, rougher bank akan melakukan potongan kasar dan memisahkan komponen yang mengapung sehingga terbentuk slurry. Cairan surfaktan berupa sodium ethyl xanthate (SEX) dicampurkan kedalam slurry untuk menurunkan tegangan permukaan, bersamaan dengan CuSO4 dan naphthalene sulphonate (NS). Udara dialirkan kedalam slurry untuk membentuk gelembung udara, partikel-partikel yang terikat dengan gelembung udara akan mengapung ke permukaan dan membentuk buih. Buih akan dipisahkan sebagai konsentrat dan diproses secara lebih lanjut melalui pump box. Sisa slurry akan masuk ke sel selanjutnya di mana slurry akan diproses seperti pada sel sebelumnya.

Pada data tersebut terdapat beberapa kejadian di mana satu atau lebih sensor mengalami kegagalan, sehingga tidak menghasilkan hasil pengukuran, ditandai dengan pengukuran bernilai 0. Pada data validasi, sebuah sampel diberi label normal apabila tidak terdapat pengukuran yang bernilai 0, dan diberi label salah apabila sebaliknya.

3.2. Tata Laksana Penelitian

Penelitian ini dilakukan dengan beberapa tahapan, yaitu pengujian KFDA dan SVM sebagai metode deteksi kesalahan, perancangan program, pembangunan program, dan pengujian program menggunakan data validasi.

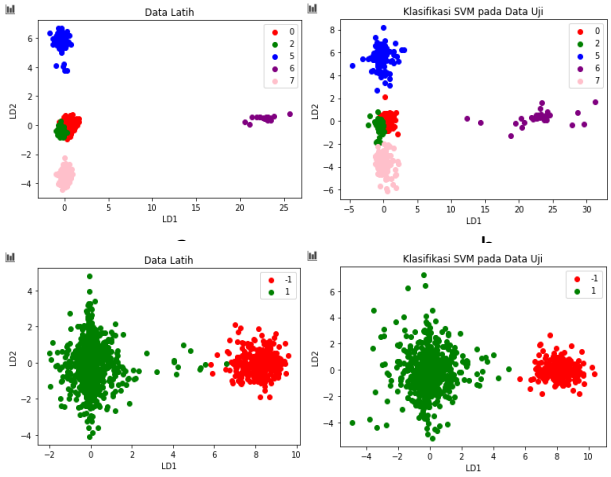
Tahap pengujian KFDA dan SVM dilakukan dengan menggunakan rangka kerja yang dibangun melalui studi literatur untuk menguji performa KFDA-SVM dalam mendeteksi kesalahan pada data pabrik. Pada tahap ini dilakukan pelatihan model dan pengujian model yang diiterasi untuk mencari nilai optimal dari parameter KFDA sehingga didapatkan akurasi klasifikasi SVM yang optimal pula.

Tahap perancangan program dilakukan setelah ditemukan parameter pada rangka kerja yang menghasilkan akurasi yang optimal, tahap ini dilakukan untuk membuat alur program yang dapat mengimplementasikan dengan baik rangka kerja yang telah diuji. Tahap perancangan program menjadi penting mengingat perbedaan antara tahap pengujian sebelumnya dengan kondisi nyata, di mana pada tahap Pengujian KFDA dan SVM, data latih masih memiliki label yang digunakan untuk membentuk matriks transformasi KFDA dan *classifier*, sedangkan pada kondisi riil, data yang masuk tidak memiliki label. Pada tahap ini, akan dilakukan perancangan program untuk mengimplementasikan KFDA-SVM yang telah diuji sebelumnya sehingga dapat melakukan deteksi kesalahan tanpa membutuhkan kumpulan data yang telah tersedia sebagai data latih. Perancangan program dilakukan dengan cara menguji metode pengelompokan yang sesuai untuk diimplementasikan bersama KFDA-SVM untuk membentuk *metode semi-supervised learning*. Apabila metode pengelompokan yang sesuai telah ditentukan, pengujian selanjutnya dilakukan untuk mencari parameter yang paling optimum untuk rangka kerja yang dibangun. Diakhir tahap ini didapatkan akurasi rangka kerja yang ditawarkan untuk mendeteksi kesalahan pada data latih dengan parameter yang telah ditentukan.

Setelah pengujian dilakukan dan alur tahapan program dirancang, program dibangun dengan mengikuti rancangan program yang telah dibuat, sehingga data pabrik dapat dikirimkan satu per satu menggunakan protokol komunikasi MQTT untuk meniru kegiatan kumpulan sensor dari pabrik yang memiliki waktu cuplikan tertentu. Program dibangun menggunakan parameter-parameter yang didapatkan pada tahap sebelumnya. Hasil akhir dari tahap ini adalah program deteksi kesalahan yang mampu mendeteksi data salah dari sebuah pabrik secara real-time tanpa memerlukan tahapan pelatihan model yang eksplisit. Setelah program dibangun, performa dari program diuji menggunakan data validasi, dengan parameter-parameter yang didapatkan melalui tahap perancangan program.

4. Hasil dan Pembahasan

4.1. Pengujian KFDA-SVM untuk Deteksi Kesalahan



Gambar 4 Hasil Pengujian Menggunakan KFDA-SVM

Performa rangka kerja KFDA-SVM diuji menggunakan data Tennessee Eastman Process untuk mendapatkan akurasi rangka kerja sebagai metode *supervised learning*. Hasil proyeksi dari KFDA-SVM dapat dilihat pada Gambar 4. Pada rangka kerja yang diuji, pada awalnya data dipisah menjadi

2, yaitu data latih dan data uji, dimana data latih akan digunakan untuk membangun matriks proyeksi KFDA, model penskalaan, dan model SVM. Data uji kemudian diproyeksikan pada KFDA-*space*, diolah menggunakan model penskalaan, dan diprediksi kelasnya menggunakan SVM yang telah dibangun menggunakan data latih. Pada pengujian tersebut, digunakan nilai γ sebesar 10^{-5} sehingga didapatkan nilai akurasi klasifikasi yang optimum. Gambar 4(a) merupakan proyeksi data latih apabila digunakan label yang sesuai dengan jenis datanya, sedangkan Gambar 4(c) merupakan proyeksi data latih apabila hanya terdapat 2 label, data normal dan data salah. Gambar 4(b) dan 4(d) merupakan hasil klasifikasi SVM terhadap data uji. Didapatkan akurasi klasifikasi sebesar 96,67% pada Gambar 4(b) dan akurasi klasifikasi sebesar 99,69% pada Gambar 4(d).

4.2. Perancangan Program

Pada pengujian parameter deteksi kesalahan, dilakukan beberapa pengujian untuk menentukan parameter-parameter program yang optimal. Parameter yang akan diuji antara lain ukuran data referensi dan penyangga dan metode pengelompokan yang optimal. Hasil pengujian ini dapat dilihat pada Gambar 5, 6, 7, 8, dan 9. Dari pengujian tersebut, didapatkan tren bahwa akurasi menurun seiring meningkatnya ukuran penyangga. Nilai γ dan referensi berpengaruh dalam membangun matriks proyeksi KFDA. Apabila nilai γ yang digunakan terlalu kecil, data akan diproyeksikan secara terlalu menyebar hingga data normal pada penyangga diproyeksikan terpisah dari data referensi, sehingga memperkecil nilai akurasi, sedangkan apabila ukuran referensi terlalu besar, data salah akan diproyeksikan didalam persebaran data referensi dan menghasilkan nilai akurasi yang lebih rendah. Akurasi tertinggi sebesar 97,04% diraih dengan menggunakan nilai γ sebesar 10^{-4}, ukuran sampel referensi sebesar 150, dan ukuran sampel penyangga sebesar 50. Sedangkan akurasi tersendah sebesar 86,92% diraih dengan menggunakan nilai γ sebesar $5x10^{-6}$, ukuran sampel referensi sebesar 150, dan ukuran sampel penyangga sebesar 200.

Saat program deteksi kesalahan dijalankan, program deteksi kesalahan harus mampu melakukan pelatihan SVM menggunakan data pada data referensi dan penyangga. Karena SVM merupakan sebuah metode klasifikasi supervised, SVM membutuhkan label dari data yang digunakan untuk dapat membuat garis klasifikasi. Pada bagian ini, akan dilakukan pengujian yang membandingkan performa metode pengelompokan, yaitu metode pengelompokan k-means dan metode DBSCAN. Pengujian dilakukan dengan cara mencari F-*score* dari metode pengelompokan k-*means* dan DBSCAN pada berbagai variasi jumlah sampel salah, nilai γ, dan ukuran referensi. Pada 80 pengujian yang telah dilakukan, didapatkan hasil bahwa penggunaan metode pengelompokan k-*means* menghasilkan F-*score* yang lebih tinggi dibandingkan metode pengelompokan DBSCAN pada 59 pengujian. Sebagai hasil dari pengujian ini, metode pengelompokan k-*means* akan digunakan sebagai metode pengelompokan pada program yang dirancang.

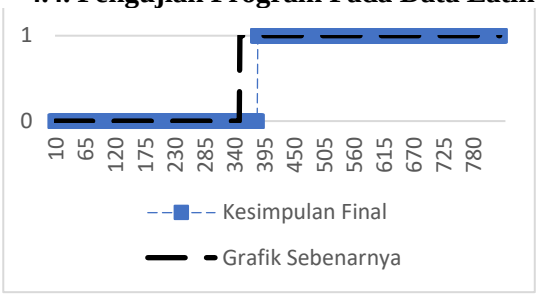
Dilakukan pula pengujian untuk menentukan batas ambang untuk memulai pengelompokan k-*means*. Batas ambang diperlukan karena pengelompokan k-*means* akan selalu mengelompokan penyangga ke dalam 2 kelompok, sehingga apabila di dalam penyangga hanya terdapat data normal, hasil dari pengelompokan k-*means* akan memberikan label yang keliru. Digunakan teknik *elbow method* untuk mendeteksi apabila sebuah data perlu dibagi menjadi 2 kelompok. *Elbow method* merupakan sebuah strategi sehingga penentuan jumlah kelompok optimal berdasarkan pembentukan siku pada grafik *within cluster sum of squares* (WCSS) [17]. Mempertimbangkan metode tersebut, penentuan ambang dilakukan dengan cara mencari rasio beda nilai WCSS pada tiap pengujian, dengan rasio didapatkan dengan rasio = $(WCSS_1 - WCSS_2)/(WCSS_2 - WCSS_3)$. Rasio yang paling kecil untuk tiap ukuran referensi ditetapkan sebagai ambang pada ukuran referensi tersebut agar pengelompokan k-means dapat segera dijalankan dengan jumlah sampel salah yang sedikit mungkin. Ambang yang didapatkan yakni: $ambang_{50} = 1.62$, $ambang_{100} = 3.18$, $ambang_{150} = 1.2$, dan $ambang_{200} = 3.46$.

Berdasarkan pengujian yang telah dilakukan, dipilih parameter yang akan ditetapkan sebagai parameter yang ditentukan, yakni nilai γ sebesar $5x10^{-5}$, ukuran referensi sebesar 150, dan ukuran penyangga sebesar 50.

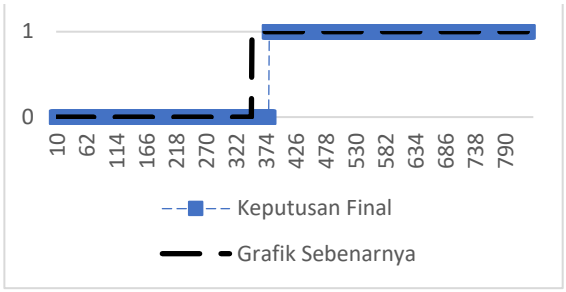
4.3. Pembangunan Program

Program dibangun menggunakan metode-metode yang telah diuji pada tahap sebelumnya, dan menggunakan parameter yang telah diuji sebagai parameter yang ditentukan. Program memiliki 4 tahapan utama, yakni pengolahan referensi, ekstraksi fitur, pengelompokan data, dan klasifikasi data. Program lalu memberikan kesimpulan apakah terdapat kesalahan proses pada data berdasarkan klasifikasi dari SVM. Data dikirim melalui protokol komunikasi MQTT satu per satu untuk meniru aktifitas sensor pada pabrik dengan waktu cuplikan tertentu.

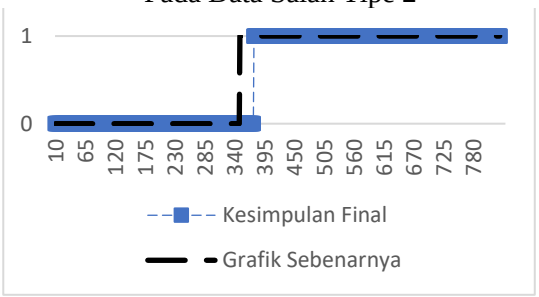
4.4. Pengujian Program Pada Data Latih



Gambar 5 Hasil Kesimpulan Final Program Pada Data Salah Tipe 2



Gambar 6 Hasil Kesimpulan Final Program Pada Data Salah Tipe 5

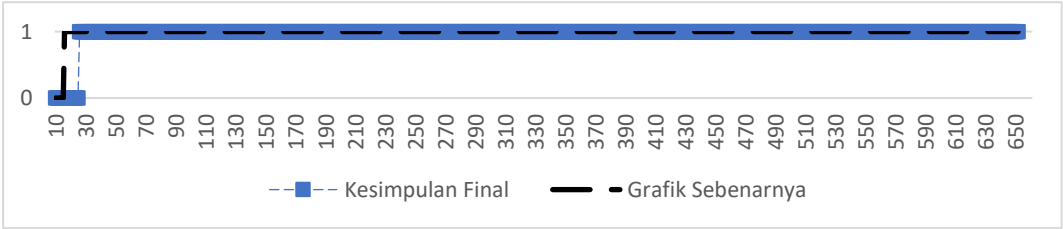


Gambar 7 Hasil Kesimpulan Final Program Pada Data Salah Tipe 6

Gambar 8 Hasil Kesimpulan Final Program Pada Data Salah Tipe 7

Gambar 5 hingga Gambar 8 merupakan hasil keputusan final pada program deteksi kesalahan. Berdasarkan data latih, digunakan nilai ambang sehingga keputusan final bahwa terdapat kesalahan diambil apabila terdapat minimal 15% data salah pada penyangga minimal selama 10 masukan data baru. Dengan nilai ambang tersebut, program dapat memberikan toleransi terhadap misklasifikasi sehingga deteksi kesalahan dapat dilakukan secara tepat.

4.5. Pengujian Program Pada Data Validasi



Gambar 9 Pengujian Program Pada Data Validasi Dengan Ukuran Referensi Sebesar 350 Sampel

Program mengalami kesulitan dalam mendeteksi kesalahan pada data validasi karena terdapat perubahan karakter data normal pada sensor FC1 dan FC2 secara signifikan yang tidak masuk ke dalam data referensi karena berada diluar ukuran data referensi. Hal tersebut menyebabkan program mendeteksi perubahan hasil bacaan FC1 dan FC2 tersebut sebagai kesalahan proses. Pengujian lanjutan dilakukan dengan cara memperbesar ukuran data referensi, sehingga perubahan hasil bacaan FC1 dan FC2 masuk ke dalam data referensi dan program dapat mengenali perubahan tersebut sebagai data normal. Gambar 9 merupakan hasil pengujian program pada data validasi dengan menggunakan ukuran referensi sebesar 350 sampel. Dengan menggunakan ukuran sampel yang lebih besar, program dapat memasukkan perubahan data FC1 dan FC2 sehingga program dapat mengenali karakter data tersebut sebagai data normal. Pengujian yang dilakukan mendapatkan hasil bahwa program dapat melakukan deteksi kesalahan dengan tepat apabila data referensi yang masuk mewakili karakter data normal pada proses, sehingga ukuran referensi harus disesuaikan dengan jumlah data yang dibutuhkan untuk mendapatkan data referensi yang dapat mewakili karakter data normal pada proses. Pada pengujian tersebut, juga digunakan nilai γ sebesar $5x10^{-3}$, sehingga dapat disimpulkan bahwa untuk mendapatkan hasil yang optimal, parameter program perlu disesuaikan dengan data yang digunakan.

5. Kesimpulan

Pada program yang dirancang, sampel-sampel yang pertama masuk ke dalam program digunakan sebagai data referensi, yakni acuan data normal bagi program. Data referensi kemudian digunakan untuk membangun matriks KFDA yang digunakan untuk memperkecil jumlah fitur dari data program menjadi 2, sehingga data dapat diproyeksikan kedalam grafik 2 dimensi. Setelah itu, sampel-sampel baru akan dimasukkan kedalam penyangga untuk diolah. Terdapat 3 tahap utama ketika sampel baru masuk kedalam program, pra-pemrosesan, pengelompokan, klasifikasi. Setelah data diolah dengan *robust scaler*, sampel baru akan diproyeksikan kedalam ruang fitur KFDA menggunakan matriks KFDA. Sampel baru dan sampel lainnya didalam penyangga kemudian diolah dengan metode pengelompokan berupa pengelompokan k-*means* untuk mendapatkan label dari tiap sampel. Metode klasifikasi berupa SVM kemudian digunakan untuk membuat garis pemisah yang optimum antara data referensi dan data normal dengan data salah. Nilai ambang batas digunakan untuk mengambil kesimpulan apakah terdapat kesalahan pada proses berdasarkan hasil klasifikasi oleh SVM.

Program yang dirancang dapat mendeteksi kesalahan pada kedua kumpulan data yang digunakan. Walaupun demikian, harus dilakukan penyesuaian parameter terlebih dahulu agar didapatkan hasil yang optimum pada data latih dan data validasi, sehingga program belum dapat seutuhnya digunakan untuk deteksi kesalahan secara *on-the-fly*. Selain itu, pada pengujian menggunakan data validasi, didapatkan pula bahwa program harus memiliki data yang mewakili karakter data normal sebagai data referensi untuk dapat melakukan deteksi kesalahan secara optimal. Walaupun parameter yang telah ditentukan sebelumnya tidak sesuai dengan data yang digunakan, parameter tersebut dapat digunakan sebagai titik mulai dalam pencarian parameter optimum untuk kedua data.

Daftar Pustaka

[1] L. H. Chiang, E. L. Russell, and R. D. Braatz, "Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis," *Chemom. Intell. Lab. Syst.*, vol. 50, no. 2, pp. 243–252, 2000, doi: 10.1016/S0169-7439(99)00061-1.

[2] Y. C. A. P. Reddy, P. Viswanath, and B. E. Reddy, "Semi - supervised learning : a brief review," vol. 7, pp. 81–85, 2018.

[3] J. Liu, Y. F. Li, and E. Zio, "A SVM framework for fault detection of the braking system in a high speed train," *Mech. Syst. Signal Process.*, vol. 87, no. August 2016, pp. 401–409, 2017, doi: 10.1016/j.ymssp.2016.10.034.

[4] K. Y. Chen, L. S. Chen, M. C. Chen, and C. L. Lee, "Using SVM based method for equipment fault detection in a thermal power plant," *Comput. Ind.*, vol. 62, no. 1, pp. 42–50, 2011, doi: 10.1016/j.compind.2010.05.013.

[5] Z. B. Zhu and Z. H. Song, "A novel fault diagnosis system using pattern classification on kernel FDA subspace," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 6895–6905, 2011, doi: 10.1016/j.eswa.2010.12.034.

[6] A. Hajj Hassan, S. Lambert-Lacroix, and F. Pasqualini, "Real-Time Fault Detection in Semiconductor Using One-Class Support Vector Machines," *Int. J. Comput. Theory Eng.*, vol. 7, no. 3, pp. 191–196, 2015, doi: 10.7763/ijcte.2015.v7.955.

[7] R. Chitrakar and H. Chuanhe, "Anomaly detection using Support Vector Machine classification with k-Medoids clustering," *Asian Himalayas Int. Conf. Internet*, pp. 1–5, 2012, doi: 10.1109/AHICI.2012.6408446.

[8] S. Yin, X. Gao, H. R. Karimi, and X. Zhu, "Study on support vector machine-based fault detection in Tennessee Eastman process," *Abstr. Appl. Anal.*, vol. 2014, 2014, doi: 10.1155/2014/836895.

[9] R. G. Brereton and G. R. Lloyd, "Support Vector Machines for classification and regression," 1998. doi: 10.1039/b918972f.

[10] J. MacQueen, "SOME METHODS FOR CLASSIFICATION AND ANALYSIS OF MULTIVARIATE OBSERVATIONS," 1967, doi: 10.1.1.308.8619.

[11] L. J. P. Van Der Maaten, E. O. Postma, and H. J. Van Den Herik, "Dimensionality Reduction: A Comparative Review," *J. Mach. Learn. Res.*, vol. 10, pp. 1–41, 2009, doi: 10.1080/13506280444000102.

[12] Q. Yang, "MODEL-BASED AND DATA DRIVEN FAULT DIAGNOSIS METHODS WITH APPLICATIONS TO PROCESS MONITORING," CASE WESTERN RESERVE UNIVERSITY, 2004.

[13] J. Yang, Z. Jin, J. Y. Yang, D. Zhang, and A. F. Frangi, "Essence of kernel Fisher discriminant: KPCA plus LDA," *Pattern Recognit.*, vol. 37, no. 10, pp. 2097–2100, 2004, doi: 10.1016/j.patcog.2003.10.015.

[14] X. Wan, "Influence of feature scaling on convergence of gradient iterative algorithm," *J. Phys. Conf. Ser.*, vol. 1213, no. 3, 2019, doi: 10.1088/1742-6596/1213/3/032021.

[15] J. J. Downs and E. F. Vogel, "A Plant-wide Industrial Problem Process," *Comput. Chem. Eng.*, vol. 17, no. 3, pp. 245–255, 1993.

[16] K. S. Brooks and M. Bauer, "Sensor validation and reconstruction: Experiences with commercial technology," *Control Eng. Pract.*, vol. 77, no. March 2017, pp. 28–40, 2018, doi: 10.1016/j.conengprac.2018.04.003.

[17] M. A. Syakur, B. K. Khotimah, E. M. S. Rochman, and B. D. Satoto, "Integration K-Means Clustering Method and Elbow Method for Identification of the Best Customer Profile Cluster," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 336, no. 1, 2018, doi: 10.1088/1757-899X/336/1/012017.