

PENERAPAN BASIS NON-ORTOGONAL PADA CMA-ES UNTUK KLASIFIKASI SINYAL EEG

Munjiah Nur Sa'adah¹⁾, Rully Soelaiman¹⁾, Chastine Faticah¹⁾

¹⁾Teknik Informatika, Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
munjiah.nur11@mhs.if.its.ac.id

Abstrak. *Electroencephalograph (EEG) adalah suatu alat untuk merekam aktivitas listrik pada otak selama kurun waktu tertentu. Tujuan EEG berbasis Brain Computer Interface (BCI) adalah untuk mendeteksi pola pada sinyal EEG pengguna untuk mengontrol komputer atau perangkat eksternal. Teknik klasifikasi yang tepat dapat digunakan untuk mendeteksi pola pada data EEG sehingga dibutuhkan informasi masukan yang relevan. Data sinyal EEG bersifat time-series, sedangkan diketahui bahwa pola terbaik berada pada domain frekuensi, bukan pada domain waktu. Beberapa teknik klasifikasi sinyal EEG pada penelitian sebelumnya membutuhkan waktu komputasi cukup lama dan rata-rata akurasi 50%. Penggunaan algoritma CMA-ES dalam mencari sudut optimal untuk transformasi rotasi pada dataset yang didukung PCA untuk mereduksi dimensi dan Transformasi Whitening mampu meningkatkan akurasi hingga 69%. Namun, masih belum mampu mengatasi kebutuhan waktu komputasi yang cukup tinggi dalam melakukan klasifikasi sinyal EEG.*

Kata Kunci: *electroencephalograph, Brain Computer Interface, Covariance Matrix Adaptation Evolution Strategy, klasifikasi.*

Electroencephalograph (EEG) adalah suatu alat untuk merekam aktivitas listrik pada otak selama kurun waktu tertentu. EEG mengukur fluktuasi tegangan yang dihasilkan oleh aliran ion dalam otak. EEG banyak digunakan untuk diagnosa epilepsi, koma, tumor, stroke, dan gangguan otak lainnya. Penggunaan EEG mengalami penurunan seiring adanya teknik pencitraan anatomi dengan resolusi spasial tinggi, seperti *Computed Tomography (CT Scan)* dan *Magnetic Resonance Imaging (MRI)*. Walaupun EEG memiliki resolusi spasial terbatas, EEG tetap menjadi alat yang berharga untuk penelitian dan diagnosa, terutama ketika resolusi temporal dalam rentang milidetik dibutuhkan.

Tujuan EEG berbasis *Brain Computer Interface (BCI)* adalah untuk mendeteksi pola sinyal EEG pengguna untuk mengontrol komputer atau perangkat eksternal. Variabilitas yang tinggi dari pola EEG antara subyek yang berbeda menyebabkan pemilihan teknik klasifikasi yang tepat sangat dibutuhkan untuk mendeteksi pola pada data EEG.

Berbagai macam algoritma klasifikasi telah diuji pada penelitian BCI. Penelitian awal mengenai algoritma klasifikasi pada BCI, yaitu algoritma klasifikasi linear dan nonlinear [1].

Penelitian selanjutnya tentang survei yang lengkap dan terbaru mengenai teknik mesin belajar untuk BCI [2]. Berbagai algoritma klasifikasi yang disurvei adalah *Linear Discriminant Analysis (LDA)*, *Support Vector Machines (SVM)*, *Neural Networks (NN)*, klasifikasi bayes, dan *k-nearest neighbor (KNN)*.

Dalam kasus klasifikasi sinyal EEG, penentuan informasi masukan yang relevan menjadi hal yang sangat menentukan keberhasilan proses klasifikasi. Sinyal EEG pada dasarnya bersifat *time-series* yang diambil dari elektroda. Penggunaan data sinyal EEG yang bersifat *time-series* menghambat pembelajaran proses klasifikasi karena jumlah atribut yang terlalu besar. Diketahui bahwa jika hubungan antara jumlah data latih dan jumlah atribut pada dataset terlalu rendah, hasil klasifikasi akan *overfitting* terhadap data latih [3]. Dengan demikian, pembelajaran untuk mendapatkan pola yang tepat dan menghindari *overfitting* menjadi hal yang cukup sulit. Selain itu, diketahui bahwa pola terbaik terdeteksi pada domain frekuensi, bukan pada domain waktu. Oleh karena itu, preproses yang tepat dari sinyal mentah diakui sangat penting untuk mendapatkan akurasi klasifikasi yang tinggi.

Tiga jenis transformasi yang umum digunakan [4], yaitu filter spasial, *Fourier Transform*, dan filter band-pass. Filter spasial digunakan karena sinyal yang terdeteksi pada satu elektroda dapat berasal dari berbagai bagian otak. Filter tersebut menghasilkan sinyal lebih tepat untuk setiap elektroda. Sinyal juga harus ditransformasikan ke domain frekuensi menggunakan, *Fast Fourier Transform* (FFT) dan *Power Spectral Density* (PSD) karena diketahui bahwa pola terbaik terletak pada rentang frekuensi tertentu di domain frekuensi.

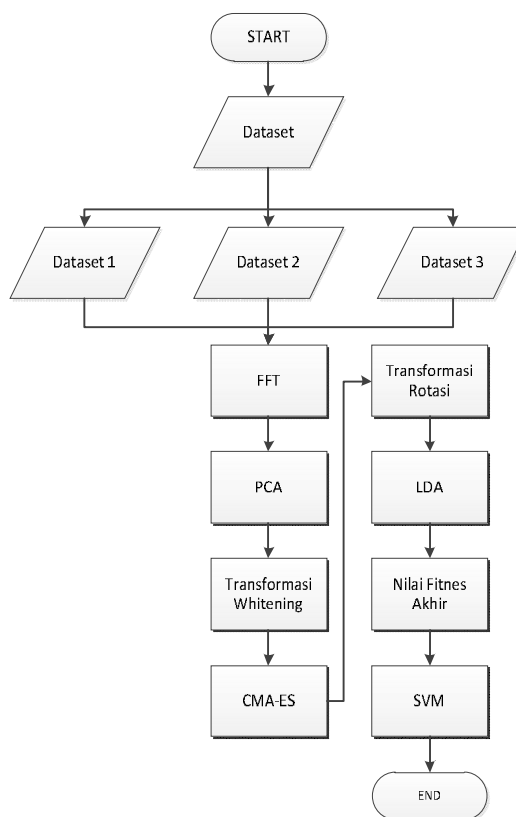
Pada penelitian sebelumnya, dilakukan optimisasi filter spasial dan filter seleksi frekuensi menggunakan *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES). Pada proses klasifikasi, CMA-ES mampu mengoptimalkan filter spasial dan filter seleksi frekuensi sehingga mampu memaksimalkan akurasi [5]. Namun, penggunaan algoritma CMA-ES pada preproses data EEG masih menghasilkan akurasi rata-rata 50%. Oleh karena itu, dibutuhkan optimasi komputasional berdasarkan algoritma *Principal Component Analysis* (PCA), transformasi whitening, dan transformasi rotasi pada algoritma CMA-ES untuk meningkatkan akurasi yang dihasilkan.

METODOLOGI

Terdapat dua tahap dalam metode yang diusulkan, yaitu tahap latih dan tahap uji. Pada tahap latih, dataset yang telah didapatkan dilakukan konversi dataset dari domain waktu ke domain frekuensi menggunakan *Fast Fourier Transform* (FFT). Kemudian dilakukan reduksi dimensi menggunakan *Principal Component Analysis* (PCA) dan transformasi whitening. Selanjutnya proses optimasi algoritma CMA-ES untuk mendapatkan sudut optimal untuk transformasi rotasi. Setiap nilai fitness terbaik dari algoritma CMA-ES digunakan untuk merotasi dataset yang berada pada domain frekuensi. Kemudian dilakukan proses LDA pada dataset yang telah disaring tersebut lalu dihitung *mean square error* (MSE) dari proses LDA. Nilai fitness akhir kemudian dihitung hingga didapatkan nilai fitness yang konvergen.

Pada tahap uji, data uji yang telah dimodelkan berdasarkan proses pada tahap latih diuji menggunakan algoritma klasifikasi SVM. Data uji diklasifikasi menggunakan SVM untuk

mendapatkan kelas sinyal EEG yang belum diketahui. Diagram alir dari metode yang diusulkan dapat dilihat pada Gambar 1.



Gambar 1 Diagram alir proses klasifikasi sinyal EEG berdasarkan CMA-ES

Konversi Data

Proses pertama pada tahap latih adalah konversi dataset dari domain waktu ke domain frekuensi menggunakan FFT. Konversi dataset ini tidak dilakukan secara langsung pada dataset, namun konversi dilakukan per *window* [5].

Sebuah *window* berjalan berukuran t bergerak sebanyak M langkah dan setiap menuju ke langkah selanjutnya *window* melompat sebesar δt . Proses ini menghasilkan submatriks berurutan dengan dimensi $t \times c$, yaitu $M_1, M_2, M_3, \dots, M_n, \dots$ yang diekstraksi dari M .

Masing-masing submatriks merepresentasikan sebuah potongan sinyal dan akan digunakan untuk membentuk data latih pada domain frekuensi. Sehingga M_1 diekstraksi dari M dari 1 sampai t , M_2 dari $1 + \delta t$ sampai $t + \delta t$, M_3 dari $1 + 2\delta t$ sampai $t + 2\delta t$, dan seterusnya.

Konversi dataset dari domain waktu ke domain frekuensi dilakukan pada submatriks M_n menggunakan FFT seperti pada persamaan (1)

$$M_n' = FFT(M_n) \quad (1).$$

Setelah FFT diterapkan, baris pada matriks M_n' dari 1 sampai $t/2$ merepresentasikan frekuensi $[0 - f/2]$ Hz, dimana f adalah frekuensi sampel dan t adalah ukuran *window*, dengan resolusi $\delta f = (f/t)$ Hz. Beberapa frekuensi dapat langsung dibuang karena diketahui di luar frekuensi $[8 - 32]$ Hz. Hal ini dilakukan karena tidak ada informasi psikologi yang penting di luar frekuensi tersebut. Oleh karena itu hanya baris $(8/\delta f)+1$ hingga baris $(32/\delta f)+1$ yang akan dipilih.

$$t' = \text{ceil}(8/\delta f) - \text{ceil}(32/\delta f) + 1 \quad (2).$$

Pada penelitian ini frekuensi sampel f yang digunakan adalah 512 Hz. Setiap satu detik data digunakan untuk membangun data latih sehingga ukuran *window* t yang digunakan adalah 512. Data latih disampel 16 kali per detik sehingga $\delta t = \frac{512}{16} = 32$. Jumlah elektroda c yang digunakan sejumlah 32 elektroda.

Principal Component Analysis (PCA)

PCA akan membangkitkan suatu himpunan basis orthonormal yang disebut *principal component*. Terdapat $X = [X_1, X_2, X_3, \dots, X_N]$ himpunan data sample dari dataset [6]. Setelah dilakukan normalisasi sehingga dihasilkan vektor yang mempunyai *unity norm* dan pengurangan *grand mean*, maka akan dihasilkan himpunan dataset baru yang dinotasikan dengan $Y = [Y_1, Y_2, Y_3, \dots, Y_N]$. Tiap Y_i merepresentasikan sebuah dataset yang telah dinormalisasi dengan dimensi N .

Selanjutnya matriks vektor eigen Φ dan matriks nilai eigen λ , Λ dapat dihitung dengan rumusan

$$(Y'Y)\Phi = \Phi\Lambda \quad (3).$$

Jika ukuran data sample n lebih kecil dari pada dimensi N . Maka metode berikut dapat diaplikasikan untuk menghemat biaya komputasi.

$$(Y'Y)\Psi = \Psi\Lambda_1 \quad (4),$$

$$\mathfrak{S} = Y\Psi \quad (5),$$

dengan $\Lambda_1 = \text{diag}(\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n)$ dan $\mathfrak{S} = [\Phi_1, \Phi_2, \dots, \Phi_n]$. Jika diasumsikan bahwa nilai eigen telah terurut secara *decreasing order*, $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$, maka m vektor eigen terbesar pertama digunakan untuk mendefinisikan matriks P

$$P = [\Phi_1, \Phi_2, \Phi_3, \dots, \Phi_m] \quad (6).$$

Himpunan ciri baru dengan dimensi m ($m \perp N$) dapat dihitung dengan rumusan

$$Z = P'Y \quad (7).$$

Transformasi Whitening

Setelah dilakukan transformasi dataset menggunakan PCA, maka himpunan ciri berdimensi rendah $Z \in \mathfrak{R}^{m \times n}$ (m adalah ukuran dimensi yang lebih rendah dan n adalah jumlah data sample pelatihan) menjadi subyek untuk Transformasi Whitening yang akan menghasilkan himpunan ciri baru, yaitu V seperti pada persamaan (8)

$$V = \Gamma Z \quad (8),$$

dengan

$$\Gamma = \text{diag}\{\lambda_1^{-1/2}, \lambda_2^{-1/2}, \lambda_3^{-1/2}, \dots, \lambda_m^{-1/2}\} \quad \text{dan} \\ \Gamma \in \mathfrak{R}^{m \times n}.$$

Tujuan Transformasi Whitening terhadap suatu vektor x adalah untuk mendapatkan vektor baru yang dinotasikan dengan \underline{d} secara linier, dimana komponen-komponennya tidak saling berkorelasi dan memiliki variansi yang sama dengan *unity*. Dapat dikatakan juga bahwa matriks kovarians dari \underline{d} sama dengan matriks identitas [6].

$$E(\underline{d}\underline{d}^T) = I \quad (9).$$

Tujuan lain dari Transformasi Whitening adalah mereduksi jumlah parameter yang harus diestimasi.

Transformasi Rotasi

Transformasi rotasi yang digunakan transformasi rotasi yang diwakili oleh matriks $c \times c'$, dinotasikan S . Hasil penerapan matriks rotasi S pada data yang telah dikonversi ke domain frekuensi adalah produk matriks yang ditunjukkan pada persamaan (10) [8].

$$M_n'' = M_n' * S \quad (10)$$

$$S = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (11).$$

Matriks M_n'' yang telah dirotasi adalah matriks yang memiliki dimensi sama dengan matriks asal M_n . Hanya saja tiap elemen matriks M_n'' berisi tiap elemen matriks M_n telah berubah sesuai dengan operator rotasi S . Pada penelitian ini besar sudut rotasi dibatasi pada $\left(0, \frac{\pi}{2}\right)$.

Persamaan (12) merangkum bagaimana I_n dihasilkan dari potongan sinyal M_n melalui S .

$$I_n = \text{whitening} (PCA(M_n)) * S \quad (12).$$

Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

Pengertian Umum

Algoritma CMA-ES diusulkan oleh Hansen dan Osterman [6,7]. Algoritma tersebut merupakan strategi evolusi untuk problem optimasi yang sulit pada domain kontinyu. Ciri khas dari algoritma ini adalah penggunaan matriks kovarians untuk menuntun proses pencarian. Pada bagian ini akan dijelaskan secara singkat mengenai (μ, λ) CMA-ES dimana μ adalah jumlah orang tua dan λ adalah jumlah keturunan.

CMA-ES memperkirakan distribusi probabilitas dari sampel dengan performa terbaik untuk menuntun pencarian menuju daerah yang menjanjikan dalam ruang pencarian. Distribusi probabilitas yang diperkirakan adalah distribusi normal multivariat $N(m, \delta^2.C)$ dimana m adalah rata-rata dan $\delta^2.C$ adalah matriks kovarians yang disusun dari matriks C dan step-size skalar δ^2 . Rata-rata mewakili lokasi saat pencarian dan bergerak menuju lokasi yang lebih baik sebagai kemajuan pencarian. Matriks kovarians mengontrol mutasi dan digunakan untuk memandu pencarian. Matriks kovarians dapat diartikan sebagai poin menuju solusi yang lebih baik dan diperkirakan dari sampel sebelumnya x_i yang memiliki performa baik terhadap fungsi fitness f . Tujuan yang ingin dicapai CMA-ES adalah untuk meminimalisir fungsi fitness $f(x): \mathbf{R}^p \rightarrow \mathbf{R}$, dimana p menunjukkan dimensi dari problem. Secara umum langkah-langkah dasar CMA-ES adalah sebagai berikut :

1. Inisialisasi parameter distribusi m, δ , dan C
2. Untuk tiap generasi (iterasi) $t = 0, 1, 2, 3, \dots$:

- a. Sampel λ titik dari $N(m, \delta^2.C)$: $x_i \leftarrow N(m, \delta^2.C)$
- b. Evaluasi fungsi fitness f dari x_1, \dots, x_λ
- c. Perbarui parameter distribusi m, δ , dan C berdasarkan performa terbaik dari x_1, \dots, x_μ .

Parameter distribusi m, δ , dan C selalu diperbarui setiap generasi atau setiap iterasi pada algoritma CMA-ES.

Diasumsikan bahwa terdapat i indeks telah diurutkan berdasarkan fungsi fitness f dan dipilih μ anggota dengan performa terbaik berdasarkan fungsi fitness f . Nilai rata-rata m selalu diperbarui berdasarkan persamaan (10)

$$m \leftarrow \sum_{i=1}^{\mu} w_i x_i, \text{ dimana } \sum w_i = 1 \quad (13).$$

CMA-ES memperbarui nilai *step size* δ dengan memperhatikan korelasi antara x_i individu dengan performa terbaik selama beberapa generasi. Urutan x_i berturut digunakan untuk menghitung jalur evolusi yang direpresentasikan oleh vektor s_δ . Ketika x_i dalam jalur berkorelasi, maka seluruh x_i akan ke arah yang sama dan jarak yang sama dapat tercakup dalam langkah yang lebih sedikit, tetapi lebih panjang. Oleh karena itu, nilai δ harus ditingkatkan. Di sisi lain, ketika jalur evolusi tidak berkorelasi, maka x_i pergi ke arah sebaliknya dan saling membatalkan. Nilai *step size* δ sebaiknya dibuat lebih kecil sehingga ruang pencarian dapat dieksplorasi dengan butiran halus. Secara singkat, diharapkan tidak adanya korelasi antara sampel di sepanjang jalur evolusi. Untuk alasan inilah, metode untuk memperbarui berdasarkan pada prinsip adaptasi nilai *step size* δ harus mengurangi perbedaan antara distribusi evolusi saat ini dan jalur evolusi di bawah seleksi acak. Persamaan (14) untuk memperbarui nilai *step size* δ

$$\delta \leftarrow \delta \exp \left\{ \beta \cdot \left(\frac{\|s_\delta\|}{\|E[N(0, I)]\|} - 1 \right) \right\} \quad (14),$$

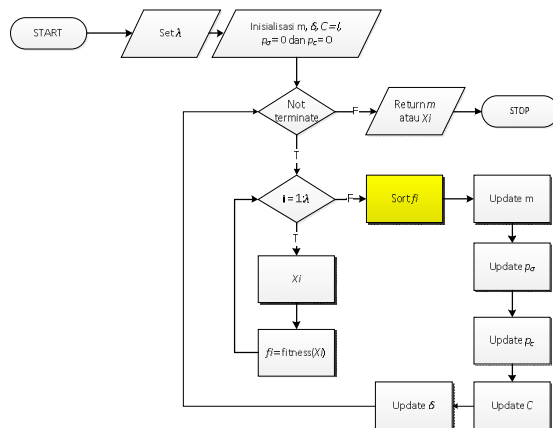
dimana β adalah parameter yang menentukan variasi *step size* δ antara generasi yang berurutan, $\|E[N(0, I)]\|$ adalah ekspektasi dari panjang sebuah distribusi random vektor $N(0, I)$ dalam \mathbf{R}^p , dan $\|s_\delta\|$ adalah panjang jalur evolusi saat ini.

Pada tahap akhir, adaptasi matriks kovarians C dideskripsikan. Jalur evolusi s_c dihitung dengan cara yang sama seperti s_δ . Matriks kovarians C diperbarui berdasarkan persamaan (15)

$$C \leftarrow (1 - c_{cov})C + \alpha_1.Cov(s_c) + \alpha_2.Cov(x_{i=1...μ}) \quad (15)$$

Berdasarkan persamaan tersebut dapat dilihat bahwa matriks kovarians C diperbarui dengan menambahkan dua komponen, yaitu $Cov(s_c)$ dan $Cov(x_{i=1...μ})$. Yang pertama berhubungan dengan kovarians jalur evolusi s_c , yaitu x_i langkah terbaik yang ditemukan selama riwayat pencarian. Yang kedua berhubungan dengan kovarians $μ$ sampel x_i terbaik pada generasi saat ini. Penggunaan kedua komponen tersebut adalah untuk mengambil keuntungan dari riwayat evolusi global (s_c) dan informasi yang diberikan oleh individu yang terbaik dalam generasi baru. $(1 - c_{cov})$ adalah nilai pembelajaran dan mengukur seberapa penting diberikan kepada C generasi sebelumnya, mengingat $\alpha_1 + \alpha_2 = c_{cov}$ meringkaskan kombinasi parameter bobot $Cov(s_c)$ dan $Cov(x_{i=1...μ})$.

Gambaran proses CMA-ES dapat dilihat lebih jelas pada digram alir Gambar 2.



Gambar 2 Digram alir CMA-ES

Optimasi Sudut Transformasi Rotasi

Covariance Matrix Adaptasi Evolution Strategy (CMA-ES) digunakan untuk mengoptimalkan secara simultan matriks rotasi S untuk classifier C . Operator rotasi direpresentasikan sebagai kromosom dan fungsi fitness yang akan dioptimalkan akan dijelaskan.

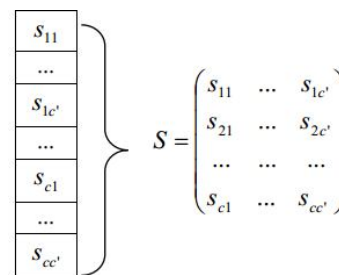
Kromosom mengandung elemen yg perlu dioptimalkan, yaitu operator rotasi S , seperti pada Gambar 3. Hal ini dapat diartikan sebagai matriks dimana setiap nilai menentukan seberapa besar setiap data dirotasikan.

CMA-ES berevolusi sebagai bilangan real sehingga matriks S juga harus dikodekan sebagai bilangan real. Oleh karena itu, prosedur

untuk mengubah bilangan real menjadi bilangan yang menyatakan sudut harus diterapkan. Sebelum matriks S dilakukan optimasi menggunakan CMA-ES, setiap elemen dari matriks S dikalikan dengan π (phi) agar tetap dapat dinyatakan sebagai sudut.

Setelah didapatkan matriks S terbaik dari proses evolusi CMA-ES, tiap elemen matriks S dijadikan operator rotasi berdasarkan persamaan (16). Sehingga operator rotasi dinyatakan sebagai

$$S = S_{11} * S_{12} * \dots * S_{1c} * \dots * S_{cc'} \quad (16)$$



Gambar 3 Struktur kromosom matriks rotasi S

Linear Discriminant Analysis (LDA)

Untuk mengevaluasi kualitas operator rotasi S yang dikodekan dalam sebuah kromosom, sebuah classifier C dibangun pada data latih yang telah dirotasi oleh S . Catatan penting disini adalah proses FFT hingga transformasi rotasi harus diterapkan untuk mengevaluasi setiap individu CMA-ES dalam setiap generasi. Sehingga evaluasi nilai fitness juga dilakukan pada setiap individu CMA-ES dalam setiap generasi.

Classifier C dibangun dari data latih I^n . Dalam hal ini, classifier C yang dipilih adalah LDA, yang dikenal sebagai classifier linier paling cepat [7]. Untuk menangani masalah multikelas, pendekatan *one-versus-all* telah diterapkan untuk LDA. Sebuah masalah klasifikasi N_c -kelas ditransformasikan ke masalah N_c -biner (2 kelas) yang bertujuan untuk memisahkan kelas i dari sisanya. Dengan demikian, LDA_i dipelajari untuk setiap masalah biner sehingga $LDA_i(x) > 0$ jika x adalah anggota dari kelas i dan $LDA_i(x) < 0$ jika x adalah anggota dari sisa kelas. Setiap LDA_i adalah sebuah *hyperplane* yang diwakili oleh persamaan (17)

$$LDA_i(x) = w_i * x + b_i \quad (17)$$

Pendekatan *one-versus-all* bekerja dengan mengelompokkan x berdasarkan persamaan (18)

$$LDA(x) = \max \arg_i LDA_i(x) = \max \arg_i (w_i * x + b_i) \quad (18)$$

Kualitas *classifier* dapat diukur berdasarkan tingkat akurasi (persentase dari anggota yang berhasil dilasifikasikan dengan benar, nilai antara 0% sampai 100%) atau kesalahan klasifikasi (1 dikurangi tingkat akurasi). Dalam hal memberikan umpan balik yang lebih tepat dan kontinyu, diputuskan untuk mengoptimalkan *mean squared error* (MSE) seperti pada persamaan (19)

$$MSE_{LDA} = \sum_n \sum_{i=1}^{i=N_c} (\sigma(w_i * I_n + b_i) - y_{i,I_n})^2 / N \quad (19)$$

dimana n adalah jumlah data, σ adalah fungsi sigmoid antara 0 dan 1, $\sigma(x) = 1/(1 + e^{(-x)})$, dan y_{i,I_n} adalah keluaran yang diharapkan untuk data I_n dan masalah biner i ($y_{i,I_n} = 1$ jika kelas sebenarnya dari I_n adalah i dan 0 jika sebaliknya). MSE_{LDA} dihitung pada data latih yang sama digunakan untuk membangun LDA (kumpulan data I_n).

Jumlah kolom c yang dipilih setelah dilakukan proses PCA menjadi salah satu parameter dalam fungsi fitness CMA-ES. Sebuah parameter λ digunakan sebagai bobot kepentingan minimisasi atribut atau minimisasi kesalahan. Persamaan (20) menunjukkan minimisasi fungsi fitness oleh CMA-ES

$$fitness(S) = MSE_{LDA} + \lambda * c \quad (20)$$

dimana MSE_{LDA} adalah MSE dari classifier LDA yang digunakan pada data latih dan λ adalah parameter regulasi. Parameter regulasi yang digunakan adalah 0,1 dan 0,2.

Support Vector Machine (SVM)

Matriks rotasi S optimal telah didapatkan pada tahap latih. Matriks rotasi tersebut digunakan untuk merotasi data uji kemudian dilanjutkan dengan klasifikasi data uji yang telah disaring menggunakan Support Vector Machine (SVM). SVM dipilih sebagai *classifier* pada tahap latih karena umumnya SVM mampu mencapai akurasi yang lebih tinggi daripada *classifier* lain, walaupun membutuhkan waktu klasifikasi lebih lama. Apabila matriks rotasi S

mampu bekerja baik pada *classifier* LDA di tahap latih, maka seharusnya matriks rotasi S juga bekerja baik atau bahkan lebih baik pada *classifier* SVM di tahap uji ini.

HASIL DAN PEMBAHASAN

Dataset

Dataset sinyal EEG yang digunakan adalah tiga dataset klasifikasi sinyal EEG yang dikeluarkan oleh IDIAP Research Institute (José del R. Millán). Tiap dataset terdiri dari tiga data latih dan satu data uji. Masing-masing dataset mengandung rekaman data EEG dari 3 subjek (orang) normal selama 4 sesi *feedback* yang melibatkan tiga tugas, yaitu imajinasi dari gerakan tangan kiri, imajinasi dari gerakan tangan kanan, dan pengucapan kata-kata yang dimulai dengan huruf acak yang sama. Keempat sesi *feedback* terjadi pada hari yang sama dan lama setiap sesi adalah 4 menit dengan sela (istirahat) 5-10 menit diantara dua sesi. Subjek mengerjakan satu tugas selama ± 15 detik kemudian melakukan tugas selanjutnya secara random (tugas lain atau mengulang tugas yang sama) sesuai dengan permintaan operator. Dataset IDIAP tidak dipisah ke dalam percobaan-percobaan, karena subjek secara kontinu melakukan setiap tugas mental (tidak ada struktur percobaan).

Data IDIAP disediakan dalam dua bentuk :

1. *Raw EEG signals* (sinyal mentah EEG) (dengan frekuensi sampel : 512 Hz) direkam dari 32 elektroda. Tiap baris data mengandung 32 data EEG dengan urutan Fp1, AF3, F7, F3, FC1, FC5, T7, C3, CP1, CP5, P7, P3, Pz, PO3, O1, Oz, O2, PO4, P4, P8, CP6, CP2, C4, T8, FC6, FC2, F4, F8, AF4, Fp2, Fz, Cz
2. *Precomputed Feature*. Data *power spectral density* (PSD) dalam 8-30 Hz setiap 62.5 ms (16 kali dalam satu detik) dengan resolusi frekuensi 2 Hz untuk 8 *parietal-channels* (C3, Cz, C4, CP1, CP2, P3, Pz, dan P4). Data mentah EEG potensial difilter spasial berdasarkan rata-rata sebuah *surface Laplacian* (8 channel, 12 komponen frekuensi).

Pada penelitian ini dataset yang digunakan hanya sinyal mentah EEG. Dataset tersebut dapat diunduh di <http://www.bbc.de/competition/iii/>.

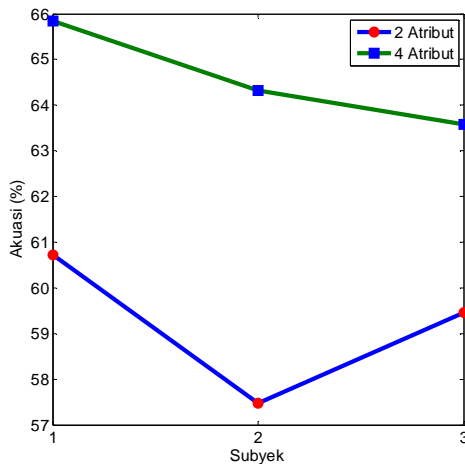
Hasil

Uji coba telah dilakukan menggunakan MATLAB untuk proses pada tahap latih dan WEKA untuk proses pada tahap uji. Uji coba dilakukan pada dataset yang terdiri dari 3 kelas dengan 3 data latih dan 1 data uji. Uji coba dilakukan dengan pemilihan jumlah atribut berdasarkan PCA sebanyak 2 atribut dan 4 atribut dan variasi nilai λ sebesar 0,1 dan 0,2 untuk kemudian dilakukan klasifikasi SVM dan mendapatkan akurasinya.

Hasil uji coba untuk $\lambda = 0,1$ dan atribut yang dipilih sebanyak 2 menghasilkan akurasi dengan rata-rata 59,22%. Sedangkan untuk atribut yang dipilih sebanyak 4 menghasilkan akurasi rata-rata 64,62%. Hasil uji coba lebih lengkap dapat dilihat pada Tabel 1. Gambar 4 menunjukkan perbandingan akurasi antara 2 atribut dan 4 atribut untuk $\lambda = 0,1$ dimana penggunaan 4 atribut menghasilkan akurasi lebih tinggi daripada 2 atribut.

Tabel 1 Hasil uji coba untuk $\lambda = 0,1$

Subyek	2 Atribut (persen)	4 Atribut (persen)
1	60,7251	65,8336
2	57,469	64,4157
3	59,4676	63,584



Gambar 4 Perbandingan akurasi untuk $\lambda = 0,1$

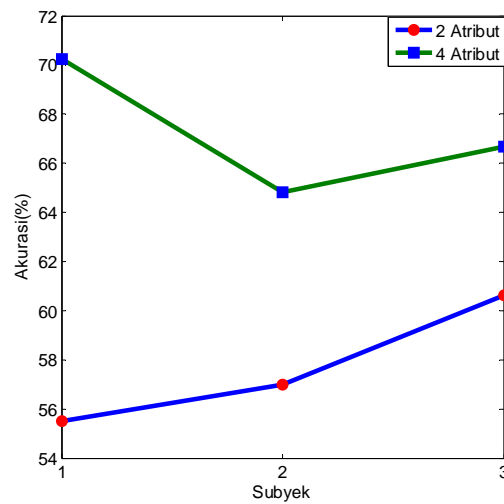
Hasil uji coba untuk $\lambda = 0,2$ dan atribut yang dipilih sebanyak 2 menghasilkan akurasi dengan rata-rata 57,71%. Sedangkan untuk atribut yang dipilih sebanyak 4 menghasilkan akurasi rata-rata 67,24%. Hasil uji coba lebih lengkap dapat dilihat pada Tabel 2. Gambar 5 menunjukkan perbandingan akurasi antara 2

atribut dan 4 atribut untuk $\lambda = 0,2$ dimana penggunaan 4 atribut menghasilkan akurasi lebih tinggi daripada 2 atribut.

Tabel 2 Hasil uji coba untuk $\lambda = 0,2$

Subyek	2 Atribut (persen)	4 Atribut (persen)
1	55,5067	70,2554
2	57,0011	64,8015
3	60,6202	66,6575

Namun, apabila akurasi antara $\lambda = 0,1$ dan $\lambda = 0,2$ dibandingkan maka akurasi untuk $\lambda = 0,2$ lebih tinggi daripada akurasi $\lambda = 0,1$. Akurasi maksimal $\lambda = 0,2$ adalah sebesar 70,26%, sedangkan akurasi maksimal $\lambda = 0,1$ hanya 65,83%.



Gambar 5 Perbandingan akurasi untuk $\lambda = 0,2$

SIMPULAN

Berdasarkan uji coba yang telah dilakukan dapat disimpulkan bahwa sistem klasifikasi sinyal EEG berdasarkan optimasi CMA-ES mampu meningkatkan akurasi sebesar 20% daripada akurasi penelitian sebelumnya hanya 50%. Nilai akurasi tertinggi didapatkan pada $\lambda = 0,2$ dan jumlah atribut berdasarkan PCA sebanyak 4 atribut.

DAFTAR PUSTAKA

- [1] K.-R. Muller, C.W. Anderson, G.E. Birch, Linear and nonlinear methods for brain-computer interfaces, *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 11 (2) (2003) 165–169.
- [2] Fabien Lotte, Marco Congedo, Anatole Lécuyer, Fabrice Lamarche, Bruno Arnaldi, A review of classification algorithms for EEG-based brain-computer interfaces, *Journal of Neural Engineering* 4 (2007).
- [3] C.M. Bishop, *Pattern Recognition and Machine Learning, The Curse of Dimensionality*, Springer, 2006. Chapter: Introduction.
- [4] G. Dornhege, M. Krauledat, K-R. Muller, B. Blankertz, *Toward Brain-Computer Interfacing*, MIT Press, 2007. pp. 207–234. Chapter: General Signal Processing and Machine Learning Tools for BCI Analysis.
- [5] Ricardo Aler, Inés M. Galván, José M. Valls, Applying evolution strategies to preprocessing EEG signals for brain-computer interfaces, *Information Sciences* 215 (2012) 53–66.
- [6] Gunn, Steve R. *Support Vector Machine for Classification and Regression*. University of South Hampton, 1998.
- [7] Bendat, J.S dan A.G. Piersol. (1986). *Random Data, Analysis And measurement Procedures*. John Wiley & Sons, 566 pp.
- [8] Soelaiman Rully, *Sistem Pengenalan wajah dengan Penerapan Algoritma Genetik pada Optimasi Basis Eigenface dan Proyeksi Fisherface*, 2006.