# Uji Konsep Paralel SVM dengan Dekomposisi SMO Pada Data Set Cancer Microarray

**Rahmat Ramadan Prasojoe[1], Setyorini, S.T, M..T[2]**

[1,2]Fakultas Informatika, Universitas Telkom, Bandung
[1]prasojoe@students.telkomuniversity.ac.id, [2]setyorini@telkomuniversity.ac.id

**Abstrak**

***Support Vector Machine* (SVM) adalah metode yang andal untuk melakukan klasifikasi dan regresi terutama dalam *supervised machine learning*. Akan tetapi SVM memiliki masalah skalabilitas dalam waktu komputasi dan penggunaan memori. Oleh karena itu banyak diusulkan *Parallel Support Vector Machine* (PSVM) untuk menambang data yang besekala besar. Pada penelitian ini penulis melakukan uji konsep PSVM dengan dekomposisi SMO yang dapat mendeteksi dan mengklasifikasika kanker dengan menggunakan data microarray. Penulis menerapkan teknik *Sequential Minimal Optimization* (SMO) yang menggunakan *lagrange multipliers* menyelesaikan masalah *quadratic programming* (QP) yang muncul selama pelatihan. Untuk menguji konsep dekomposisi SMO, data set akan dipecah ke dalam beberapa subset kemudia melakukan pelatihan SMO setiap subset secara independen dan menggabungkan setiap hasil pelatihan ke dalam satu model klasifikasi SMO. Evaluasi dilakukan dengan membandingkan akurasi dan performan dekomposisi SMO dan non-dekomposisi SMO. Evaluasi menghasilkan akurasi dekomposisi SMO 75% dan non-dekomposisi SMO 63% serta waktu pelatihan dekomposisi SMO 5.7 kali lebih cepat daripada non-dekomposisi SMO**

**Kata kunci : Microarray, Suport Vector Machine (SVM), Parallel Suport Vector Machine (PSVM), Sequential Minimal Optimization(SMO), Microarray**

**Abstract**

**Support Vector Machine (SVM) is a reliable method for performing classification and regression especially in supervised machine learning. However, SVM has scalability issues in compute time and memory usage. Therefore, there are many proposals for Parallel Support Vector Machine (PSVM) for mining large-scale data. In this study, the authors conducted a PSVM concept test with SMO decomposition that could be handled and classified cancer using microarray data. The author applies the Sequential Minimal Optimization (SMO) technique which uses lagrange multipliers to solve quadratic programming (QP) problems that arise during training. To test the concept of SMO decomposition, the data set will be broken down into several subsets and then independently conduct SMO training for each subset and combine each training result into one SMO classification model. Evaluation is done by comparing the accuracy and performance of SMO decomposition and non-decomposition SMO. Evaluation increased SMO decomposition 75% and non-SMO decomposition 63% as well as SMO decomposition training time 5.7 times faster according to non-SMO decomposition**

**Keywords: Microarray, Suport Vector Machine (SVM), Parallel Suport Vector Machine (PSVM), Sequential Minimal Optimization(SMO)**

## 1. Pendahuluan

### Latar Belakang

Seiring berkembangnya teknologi informasi jumlah data juga meningkat secara eksponensial dan ini menyebabkan para ilmuan kewalahan dengan meningkatnya jumlah kebutuhan pemrosesan data yang timbul dari banjir data yang mengalir melalui hampir setiap bidang sains, seperti bioinformatika [1-2], biomedis [3-4], Cheminformatika [5], web [6] dan seterusnya. Bukanlah hal yang mudah untuk mengukur total volume data terstruktu dan tidak terstruktur yang menggunakan teknologi dan sistem berbasis mesin agar data-data tersebut dapat di analisis secara penuh [7]. Teknik implementasi yang efisien adalah kunci untuk memenuhi skalabilitas dan persyaratan kinerja yang diperlukan dalam analisis data ilmiah tersebut.

Pembelajaran mesin telah diteliti oleh banyak ilmuan selama beretahun-tahun dan banyak metode penambangan data yang telah dikembangkan dan diterapkan. Support Vector Machine (SVM) adalah metode yang andal untuk melakukan klasifikasi dan regresi terutama dalam supervised machine learning [8]. Akan tetapi SVM memiliki masalah skalabilitas dalam waktu komputasi dan penggunaan memori seiring banyaknya training vector yang ada. Oleh karena itu banyak diusulkan Parallel Support Vector Machine (PSVM) untuk penambangan data yang besekala besar. PSVM membagi perkerjaan training dan klasifikasi data ke berbagai node yang berbeda, Hal ini dapat mempersingkat waktu komputasi dan mengurangi penggunaan memori.

Penulis menerapkan teknik *Sequential Minimal Optimization* (SMO) yang menggunakan *lagrange multipliers* menyelesaikan masalah quadratic programming (QP) yang muncul selama pelatihan SVM. Dalam melakukan dekomposisi SMO penulis memecah dataset kedalam beberapa subset dan menjalakan pelatihan SMO secara independent dan bergantian, proses ini dinamakan training lokal. setiap training SMO lokal akan memberikan nilai alpha (*lagrange multipliers*) lokal dan nilai bias lokal yang nantinya setiap nilai tersebut akan digabungkan menjadi nilai alpha global dan nilai bias global yang nantinya digunakan untuk membuat model klasifikasi SMO

**Topik dan Batasannya**

Pada penelitian tugas akhir kali ini akan dibangun sebuah sistem atau program SVM dengan menggunakan teknik SMO untuk menentukan label kelas kanker pada training data yang berupa data *microarray*. Penelitian ini berfokus dalam menguji konsep PSVM dengan dekomposisi SMO dan membandingkan hasil akurasi dan performan dekomposisi SMO dengan non-dekomposisi SMO.

**Tujuan**

Dari penjelasan latar belakang dan perumusan masalah dapat dihasilkan tujuan sebagai berikut:

- Membuat PSVM dan SVM dengan menggunakan teknik SMO untuk mendeteksi dan mengklasifikasi kanker menggunakan data microarray
- Menguji konsep PSVM dengan membuat program pelatihan SVM/SMO dengan dekomposisi dataset
- Membandingkan akurasi dan performan dari hasil pelatihan dekomposisi SMO dengan non-dekomposisi SMO menggunakan 4 dataset dan 2 kernel sebagai parameter pengujian.

**Organisasi Tulisan**

Penulisan tugas akhir ini mempunyai organisasi tulisan sebagai berikut. Bagian 2 adalah penelitian yang digunakan sebagai acuan dan penjabaran dari metode yan digunakan. Bagian 3 adalah detail dari sistem yang dibangun. Pada bagian 4 diberikan hasil eksperimen dan analisis dari sistem. Dan pada bagian 5 dijelaskan kesimpulan dari penelitian tugas akhir ini

## 2. Studi Terkait

Dalam penelitian ini penulis mengambil referensi dari penelitian sebelumnya terkait dengan latar belakang penelitian ini. Berikut beberapa studi yang terkait dengan pengujian PSVM :

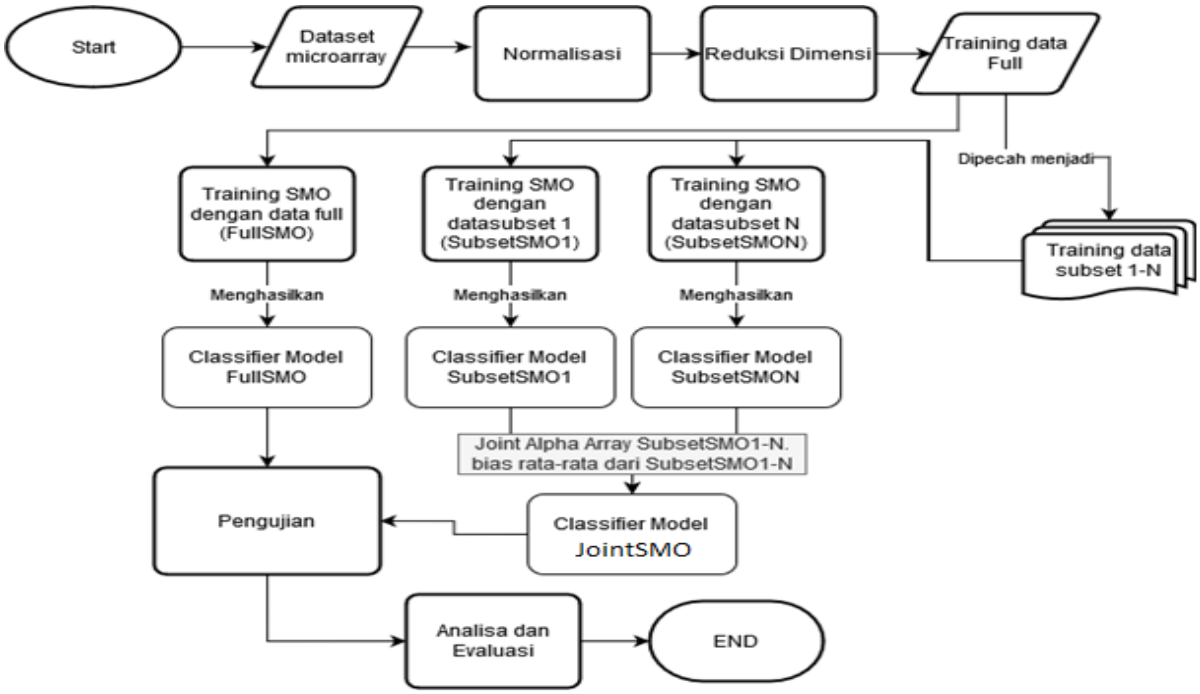
Pertama penelitian yang dilakukan oleh Sun, Zhanquan & Fox, Geoffrey. (2012). Yang berjudul "Study on Parallel SVM Based on MapReduce"[9]. Beberapa perangkat lunak MapReduce dikembangkan, seperti Hadoop, Twister dan sebagainya. Dalam tulisan ini, paralel SVM berdasarkan model iterative MapReduce Twister dipelajari. Alur program dikembangkan. Efisiensi metode ini diilustrasikan melalui analisis masalah praktis.

Selanjutnya penelitian yang dilakukan oleh Nasullah Khalid Alham, Maozhen Li, Yang Liu, Suhel Hammoud. (2011). Yang berjudul "A MapReduce-based distributed SVM algorithm for automatic image annotation"[10]. Dalam tulisan ini pararel SVM dilakukan denggan menggunakan metode MapReduce SMO dengan cara memecah training data kedalam beberapa subset dan melakukan Map Task yaitu training lokal untuk setiap subset, kemudian melakukan shuffle dan sort yang diikuti dengan Reduce Task berupa sum partial W dan join alpha Array. Hasilnya adalah Dengan mempartisi kumpulan data pelatihan menjadi subset kecil dan mengoptimalkan subset yang dipartisi di beberapa cluster node komputasi, algoritme MRSMO mengurangi waktu pelatihan secara signifikan dan mempertahankan tingkat akurasi yang tinggi dalam klasifikasi biner dan multikelas terutama untuk set data pelatihan yang jumblah besar.

Penelitian yang dilakukan oleh Kiran M, Amresh Kumar, Saikat Mukherjee & Ravi Prakash pada tahun 2013, yang berjudul "Verification and Validation of MapReduce Program Model for Parallel Support Vector Machine Algorithm on Hadoop Cluster"[7]. Dalam penelitian ini, Sequential Support Vector Machine di WEKA dan berbagai program MapReduce termasuk Parallel Support Vector Machine pada Hadoop cluster dianalisa dengan cara ini Algoritma diverifikasi dan divalidasi pada Cluster Hadoop menggunakan Konsep MapReduce. Hasil Eksperimental menunjukkan bahwa ketika jumlah node meningkat, waktu eksekusi menurun.

## 3. Sistem yang Dibangun

Pada penelitian tugas akhir kali ini akan dibangun sebuah sistem atau program PSVM dengan dekomposisi SMO untuk menentukan label kelas kanker pada training data yang berupa data *microarray*. Adapun skema sistem yg akan dibangun sebagai berikut :




Gambar 1. Skema Sistem yang dibuat

**Dataset Microarray**

Dataset yang akan digunakan adalah data colon tumor, breast cancer, lung cancer, dan leukemia yang didapatkan dari Kent Ridge Biomedical Data Repository[11].

Tabel 1.Data set Kanker Kent Ridge Biomedical Data Repository

| Data | Sampel | Fitur | Jumlah Kelas |
|---|---|---|---|
| Colon Tumor | 62(22 Positif, 40 Negatif) | 2000 | 2 |
| Breast Cancer | 97(46 Relapse, 51 non-relapse) | 24482 | 2 |
| Lung Cancer | 181(31 Mesothelioma, 150 ADCA) | 12533 | 2 |
| Leukemia | 72(47 ALL, 25 AML) | 7129 | 2 |

Pada kolom pertama terdapat data untuk digunakan sebagai nama klasifikasi kanker. Kolom kedua adalah data sampel pada setiap data klasifikasi. Kolom ketiga adalah jumlah fitur atau atribut. Dan pada kolom keempat terdapat jumlah kelas pada setiap datanya. Setiap data akan mempunyai kelas yang mengindentifikasikan apakah terkena kanker(relapse atau 1) atau tidak terkena kanker(non-relapse atau 0).

**Preprocessing Training Data**

Data microarray mempunyai permasalahan dimensi yang besar dan tingginya perbedaan nilai range dalam setiap fiturnya. Untuk mengurangi besarnya dimensi dilakukan proses reduksi dimensi. Sedangkan untuk mengatasi tingginya perbedaan nilai range dalam setiap fiturnya, akan dilakukan normalisasi data.

Pada proses normalisasi data, nilai atribut / fitur di setiap sampel data akan diubah range nilainya. Metode min - max akan digunakan untuk memperkecil range nilai pada setiap fiturnya. Rumus umum untuk normalisasi adalah:

$$Normalisasi = \frac{data - \min(data)}{\max(data) - \min(Data)} \quad (1)$$

proses ini akan menghasilkan nilai 0 sampai 1 pada setiap fitur dalam dataset yang diproses, sehingga rentang nilai yang dihasilkan tidak jauh.

Proses reduksi menggunakan metode Partial Least Square(PLS). Tahap pertama yang dilakukan adalah dengan melakukan mean-centered matrix, yaitu mencari rata-rata pada setiap fitur kemudian data akan dikurangi oleh rata-rata fitur tersebut. Langkah selanjutnya adalah dengan menjadikan atribut menjadi X dan label menjadi Y. Partial Least Square(PLS)akan memproses data dan akan dihasilkan data dengan jumlah atribut yang lebih

kecil. Proses reduksi dimensi ini akan menghasilkan fitur yang lebih sedikit dibandingkan dengan fiutr pada dataset awal. Hasil data inilah yang akan dijadikan data trainset.

**Training SMO**

Sequential minimal optimization (SMO) adalah algoritma untuk menyelesaikan masalah quadratic programming (QP) yang muncul selama pelatihan support-vector machines (SVM). diciptakan oleh John Platt pada tahun 1998 di Microsoft Research. SMO banyak digunakan untuk pelatihan mesin vektor dukungan dan diimplementasikan oleh alat LIBSVM yang popular.

Algoritma SMO dikembangkan oleh Platt [12] dan ditingkatkan oleh Keerthietal [13]. Platt mengambil dekomposisi yang ekstrem dengan memilih satu set hanya dua titik sebagai set kerja yang merupakan minimum karena kondisi beriku :

$$\sum_{i=1}^{n} \alpha_i y_i \tag{2}$$

Di mana $a_i$adalah multiplier Langrange dan $y$ adalah nama kelas. Ini memungkinkan sub masalah untuk mempunyai solusi analitis. Meskipun membutuhkan iterasi / perulangan yang lebih banyak, setiap iterasi hanya memiliki sedikit operasi; oleh karena itu algoritma menunjukkan percepatan keseluruhan dari beberapa urutan magnitude[16]. Gagasan Platt memberikan peningkatan efisiensi, dan SMO sekarang menjadi salah satu algoritma SVM tercepat yang tersedia[15]. mendefinisikan set indeks *I* berikut yang menunjukkan pola data pelatihan:

- $I_0 = \{i : y_i = 1, 0 < a_1 < c\} \cup \{i : y_i = -1, 0 < a_1 < c\}$
- $I_1 = \{i : y_i = 1, a_1 = 0\} (Positive\ Non - Support\ Vectors)$
- $I_2 = \{i : y_i = -1, a_1 = c\} (Bound\ Negative\ Support\ Vectors)$
- $I_3 = \{i : y_i = 1, a_1 = c\} (Bound\ Positive\ Support\ Vectors)$
- $I_4 = \{i : y_i = -1, a_1 = c\} (Negative\ Non - Support\ Vectors)$

Di mana *C* adalah parameter koreksi, kami juga mendefinisikan bias $b_{up}$ dan $b_{low}$ dengan index terkait mereka:

$$b_{up} = \min \{f_i : i \in I_0 \cup I_1 \cup I_2\}$$
$$I_{up} = arg \min_i f_i$$
$$b_{low} = \max \{f_i : i \in I_0 \cup I_3 \cup I_4\}$$
$$I_{low} = \arg \max_i f_i$$

Kondisi Optimalitas Dilacak Melalui Vektor :

$$f_i = \sum_{j=1}^{i} \alpha_j y_j K(X_j, X_i) - y_i \tag{3}$$

Di mana *K* adalah fungsi kernel dan $X_i$ adalah poin data training. SMO mengoptimalkan dua $a_i$ yang terkait dengan $b_{up}$ dan $b_{low}$ menurut ini :

$$a_2^{new} = a_2^{old} - y_2(f_1^{old} - f_2^{old})/n \tag{4}$$
$$a_1^{new} = a_1^{old} - s(a_2^{old} - a_2^{new})/n \tag{5}$$

Di mana $n = 2k(X_1, X_2) - k(X_1, X1) - k(X_2, X_2)$. Setelah mengoptimalkan $a_1$ dan $a_2$, $f_i$ yang menunjukkan error data pelatihan diperbarui sesuai dengan berikut ini :

$$f_i^{new} = f_i^{old} + (a_1^{new} - a_i^{old}) y_1 k(X_1, X_i) + (a_2^{new} - a_2^{old}) y_2 (X_2, X_i) \tag{6}$$

**Parallel SMO Dengan MapReduce**

Konsep Dekomposisi SMO pada penelitian ini berdasarkan Map Reduce Sequential Minimal Optimization (MRSMO) dengan menggunakan framework MapReduce yang digunakan oleh Nasullah Khalid [10]. Algoritma ini mempartisi semua training dataset menjadi subset yang lebih kecil m dan mengalokasikan setiap subset yang dipartisi ke satu tugas peta(MapTask). Jumlah MapTask sama dengan jumlah partisi. Masing-masing fungsi peta mengoptimalkan partisi secara parallel. Dalam kasus Linier SVM output setiap fungsi map adalah vector partial weight(Beban) untuk partisi lokal :

$$\underset{Parttial}{\vec{w}} = \sum_{i=1}^{I} y_i a_i \vec{x_i} \tag{7}$$

Dan value b(bias) untuk partisi tersebut. Kemudian reducer menjumlahkan total vector weight untuk menghitung vector weight global :

$$\vec{w}_{Global} = \sum_{i=1}^{n} \vec{w}_{partial} \tag{8}$$

Selanjutnya, reducer harus memecahkan masalah dengan nilai b karena nilai ini berbeda untuk setiap partisi. Oleh karena itu reducer juga harus menghitung value rata-rata b untuk setiap partisi. Kita hanya memerlukan vector weight global dan value b untuk menghitung output SVM sesuai dengan persamaan :

$$u = \vec{w}.\vec{x} - b \tag{9}$$

Dalam kasus non-linier SVM, output setiap fungsi map adalah alpha array untuk partisi local dan value b. reducer hanya menggabungkan partisi alpha array untuk menghasilkan global alpha array. Mirip dengan kasus linier reducer juga menghitung value rata-rata b untuk setiap partisi. Kita memerlukan alpha array, b dan training data yang sesuai dengan a > 0 untuk menghitung output SVM sesuai dengan:

$$u = \sum_{i=1}^{i} y_i^{a_i} K(X_i, X) + b \tag{10}$$

Dari MRSMO ini digunakan oleh Nasullah Khalid didapatkan kesimpulan bahwa nilai alpha array, bias, dan juga weight bisa didapatkan dalam training lokal serta proses mendapatkan nilai-nilai tersebut bisa dijalankan secara pararel untuk subset yang berbeda-beda dikarenakan nilai global bisa didapatkan dari gabungan nilai lokal untuk nilai alpha array dan weight dan nilai bias adalah nilai rata-rata setiap partisi/subset.

**Dekomposisi SMO**

Dalam Penelitian ini terdapat 3 jenis Training/Model SMO yaitu :

- FullSMO adalah Training SMO dengan menggunakan training data secara utuh tampa memecah data. FullSMO adalah non-dekomposisi SMO.
- SubsetSMO(1-N) adalah Training SMO dengan menggunakan subset yaitu training data yang telah dipecah menjadi beberapa bagian yang setara besarnya. Jumlah SubsetSMO adalah sama dengan jumlah data subset yang telah disediakan dan berjumlah genap. Jika ada 4 data subset maka akan ada 4 SubsetSMO. SubsetSMO1, SubsetSMO2, SubsetSMO3, SubsetSMO4. Training SubsetSMO akan di lakukan secara bergiliran dari 1-N, di mana N adalah banyaknya subset. Hasil data setiap SubsetSMO juga akan disimpan dikarenakan akan digabungkan menjadi satu model. SubsetSMO adalah proses dari dekomposisi SMO.
- JointSMO adalah Model klasifikasi SMO dengan menggunakan nilai – nilai yang didapatkan dan gabungan dari setiap SubsetSMO. Nilai utama yang dibutuhkan oleh JointSMO adalah alpha array yang didapatkan dari join alpha array setiap subset dan nilai bias didapatkan dari rata-rata nilai bias di setiap SubsetSMO. JointSMO adalah hasil akhir dari dekomposisi SMO.

**Evaluasi**

Setelah berhasil mendapatkan FullSMO dan JointSMO maka akan evaluasi akurasi dan performan. Evaluasi akurasi dilakukan untuk menentukan seberapa akurat hasil klasifikasi sistem yang dibuat. Evaluasi akurasi dilakukan dengan cara membandingkan jumlah klasifikasi data yang benar dengan jumlah data keseluruhan dan dapat dihitung dengan rumus :

$$Akurasi = \frac{Jumlah\ data\ benar}{jumlah\ training\ data} \tag{11}$$

Untuk evaluasi performa dilakukan khususnya untuk JointSMO dikarenakan JointSMO adalah percobaan simulasi PSVM. Terdapat 2 kriteria evaluasi perfoman yaitu speedup dan efficiency dengan rumus :

$$SpeedUp = \frac{Waktu\ Training\ FullSMO}{Max(Waktu\ Training\ SubsetSMO)} \tag{12}$$

$$Efficiency = \frac{SpeedUp}{Jumlah\ SubsetSMO} \tag{13}$$

Speedup didapatkan dengan membandingkan waktu training FullSMO dengan waktu training terlama dari setiap SubSetSMO dan efficiency didapatkan dari membandingkan speedup dan jumlah subsetSMO yang ada. Speedup mengukur percepatan waktu yang dibutuhkan untuk training dekomposisi SMO sedangkan efficiency mengukur seberapa efisien setiap training subset.

## 4. Evaluasi dan Analisa

Pada penelitian ini menggunakan empat dataset yang terdapat pada tabel 1. Pengujian dilakukan dengan FullSMO dan JointSMO yang sudah didapatkan. Setiap dataset dari tabel 1 telah direduksi dimensi dengan PLS menyadi 30 komponen dan dijadikan sebagai train dan test data. Dalam pengujian kali ini traindata akan dipecah menjadi 2 subset yang sama besar, sebab itu terdapat 2 SubsetSMO yang nantinya akan digabungkan menjadi JointSMO.

Dalam Pengujian ini terdapat 2 skenario pengujian yang sama-sama menggunakan C=100 dan bias=0 sebagai default value. Dalam kedua skenario pengujian dilakukan perbandingan akurasi dan performan dari FullSMO dan JointSMO.

Skenario pertama dilakukan dengan membandingkan akurasi dan performan dari FullSMO dan JointSMO menggunakan fungsi kernel liner yang didefisinikan:

$$K(x, z) = x^T z + b \quad (14)$$

Di mana *X* dan *Z* adalah array input vector, dan *b* adalah nilai bias opsional. Kernel ini menghitung kombinasi linier berpasangan dari titik-titik yang terdaftar di x dan z. Menggunakan kernel ini akan menghasilkan batasan keputusan linier.

Skenario kedua dilakukan dengan membandingkan akurasi dan performan dari FullSMO dan JointSMO menggunakan fungsi kernel Gaussian(lebih dikenal dengan radial basis function atau RBF) yang didefisinikan:

$$K(x, z) = \exp(\frac{-|x - z|^2}{2\sigma^2}) \quad (15)$$

Di mana *X* dan *Z* sama seperti kernel linier dan σ adalah parameter lebar yang menggambarkan seberapa lebar kernel dengan nilai default 1. Kernel ini menghitung kesamaan Gaussian antara contoh pelatihan yang terdaftar di x dan z, dengan nilai 1 yang menunjukkan bahwa titik-titik memiliki vektor fitur yang sama persis dan 0 menunjukkan vektor yang berbeda. Menggunakan kernel ini memungkinkan untuk konstruksi batas keputusan non-linier yang lebih kompleks.

**Hasil Evaluasi Data Colon Tumor**

Hasil evaluasi yang dihasilkan dari pengujian Data Colon Tumor sebagai berikut.

Tabel 2 Perbandingan Data Colon Tumor Kernel Linier

| SMO | RunTime | Akurasi |
|---|---|---|
| FullSMO | 6.5718s | 80.64% |
| SubsetSMO1 | 0.3007s | 80.64% |
| SubsetSMO2 | 1.0643s | 74.19% |
| JointSMO | - | 83.87% |
| SpeedUp=6.1747 | Efficiency=3.0873 | Akurasi Up |

Tabel 3 Perbandingan Data Colon Tumor Kernel Gaussian

| SMO | RunTime | Akurasi |
|---|---|---|
| FullSMO | 1.4887s | 32.25% |
| SubsetSMO1 | 0.3854s | 58.06% |
| SubsetSMO2 | 0.3477s | 45.16% |
| JointSMO | - | 82.25% |
| SpeedUp=3.8627 | Efficiency=1.9313 | Akurasi Up |

Pada data Colon Tumor, JointSMO berhasil memiliki akurasi yang tinggi dan stabil di kedua kernel yaitu sekitar 82.60% dan melebihi akurasi FullSMO di kedua kernel. Hal ini dikarenakan SubsetSMO menggunakan pecahan data train yang sudah di acak dibandingkan FullSMO menggunakan full data train yang memiliki pola data sampel. Kernel linier memberikan akurasi yang stabil di setiap SMO sedangkan kernel Gaussian cenderung memiliki akurasi yang rendah terkecuali akurasi JointSMO. Performan JointSMO terbaik didapatkan oleh kernel linier dengan SpeedUp=6.1747 dan Efficiency=3.0873.

**Hasil Evaluasi Data Breast Cancer**

Hasil evaluasi yang dihasilkan dari pengujian Data Breast Cancer sebagai berikut.

Tabel 4 Perbandingan Data Breast Cancer Kernel Linier

| SMO | RunTime | Akurasi |
|---|---|---|
| FullSMO | 523.40s | 60.41% |
| SubsetSMO1 | 39.85s | 52.08% |

| | | |
|---|---|---|
| SubsetSMO2 | 36.29s | 45.83% |
| JointSMO | - | 67.70% |
| SpeedUp=13.13425 | Efficiency=6.5671 | Akurasi Up |

Tabel 5 Perbandingan Data Breast Cancer Kernel Gaussian

| SMO | RunTime | Akurasi |
|---|---|---|
| FullSMO | 5.8650s | 56.25% |
| SubsetSMO1 | 0.5796s | 50% |
| SubsetSMO2 | 0.8869s | 41.66% |
| JointSMO | - | 50% |
| SpeedUp=6,6129 | Efficiency=3,3064 | Akurasi Down |

Pada data Breast Cancer akurasi cendurung stabil namun rendah pada setiap SMO dikedua kernel. Akurasi tertinggi didapatkan pada JointSMO kernel linier sebesar 67.70%. Runtime training SMO juga cenderung lama dengan FullSMO linier kerner memiliki waktu paling lama yaitu 524 detik. Akurasi rendah dan waktu traning yang lama diyakini karena dataset Breast Cancer memiliki fitur yang kompleks dan besar yaitu 24482. Oleh karena itu proses reduksi dimensi PLS tidak mampu untuk menghilangkan noise data Breast Cancer. Data Breast Cancer juga memakan waktu 2 sampai 2,5 jam untuk melakukan reduksi dimensi PLS menjadi 30 f itur. Waktu ini sangat lama jika dibandingkan dengan proses reduksi data yang lain yang hanya memakan waktu sekitar 10menit. Performan JointSMO terbaik didapatkan oleh kernel linier dengan SpeedUp=13.13425 dan Efficiency=6.5671.

**Hasil Evaluasi Data Lung Cancer**

Hasil evaluasi yang dihasilkan dari pengujian Data Lung Cancer sebagai berikut.

Tabel 6 Perbandingan Data Lung Cancer Kernel Linier

| SMO | RunTime | Akurasi |
|---|---|---|
| FullSMO | 1.7682s | 83.33% |
| SubsetSMO1 | 0.6682s | 94.44% |
| SubsetSMO2 | 0.7019s | 92.22% |
| JointSMO | - | 92.22% |
| SpeedUp=2,5191 | Efficiency=1,2595 | Akurasi Up |

Tabel 7 Perbandingan Data Lung Cancer Kernel Gaussian

| SMO | RunTime | Akurasi |
|---|---|---|
| FullSMO | 11.0316s | 49% |
| SubsetSMO1 | 3.9451s | 80% |
| SubsetSMO2 | 1.5521s | 21.22% |
| JointSMO | - | 56.11% |
| SpeedUp=2,7962 | Efficiency=1,3981 | Akurasi Up |

Pada data Lung Cancer, kernel Linier memiliki akurasi yang stabil dan tinggi disetiap SMO dengan rata – rata akurasi 90%. Disisi lain kernel Gaussian memiliki akurasi yang stabil namun rendah disetiap SMO dengan rata – rata akurasi 53%. JointSMO berhasil memiliki akurasi yang lebih tinggi dengan FullSMO di kedua kernel hal ini dikarenakan SubsetSMO menggunakan pecahan data train yang sudah di acak dibandingkan FullSMO menggunakan full data train yang memiliki pola data sampel. Performan JointSMO cenderung stabil dan sama di kedua kernel dengan performa terbaik didapatkan oleh kernel Gaussian dengan SpeedUp=2,7962 dan Efficiency=1,3981.

**Hasil Evaluasi Data Leukemia**

Hasil evaluasi yang dihasilkan dari pengujian Data Leukemia sebagai berikut.

Tabel 8 Perbandingan Data Leukemia Kernel Linier

| SMO | RunTime | Akurasi |
|---|---|---|
| FullSMO | 0.0846s | 88.46% |
| SubsetSMO1 | 0.0690s | 88.46% |
| SubsetSMO2 | 0.0624s | 73.07% |
| JointSMO | - | 96.15% |
| SpeedUp=0,1226 | Efficiency=0,0613 | Akurasi Up |

Tabel 9 Perbandingan Data Leukemia Kernel Gaussian

| SMO | RunTime | Akurasi |
|---|---|---|
| FullSMO | 1.1858s | 57.69% |
| SubsetSMO1 | 0.1159s | 42.30% |
| SubsetSMO2 | 0.0156s | 50% |
| JointSMO | - | 78.84% |
| SpeedUp=10,2312 | Efficiency=5,1156 | Akurasi Up |

Pada data Leukemia, JointSMO berhasil memiliki akurasi yang tinggi dan stabil dan melebihi akurasi FullSMO di kedua kernel. Hal ini dikarenakan SubsetSMO menggunakan pecahan data train yang sudah di acak dibandingkan FullSMO menggunakan full data train yang memiliki pola data sampel. Kernel linier memberikan akurasi yang stabil di setiap SMO sedangkan kernel Gaussian cenderung memiliki akurasi yang rendah terkecuali akurasi JointSMO. Performan JointSMO terbaik didapatkan oleh kernel Gaussian dengan SpeedUp=10,2312 dan Efficiency=5,1156. Untuk perbandingan nilai sum alpha array dan bias antara FullSMO dan JointSMO, kedua kernel tidak memiliki perbedaan nilai yang signifikan dan nilai hamper mendekati sama.

**Analisis Hasil Pengujian**

Berdasarkan skenario pengujian yang telah dilakukan, dekomposisi SMO dengan menggunakan subset dan joint SMO dapat meningkatkan performan SpeedUp dengan nilai rata-rata 5.7 untuk setiap kasus pelatihan SMO.

Halis akurasi klasifikasi JointSMO cenderung mengalami peningkatan daripada akurasi FullSMO disetiap skenario dengan rata-rata akurasi 75% untuk JointSMO dan 63% untuk FullSMO. Terdapat pengecualian dengan breast cancer kernel gaussian di mana akurasi JointSMO mengalami penurunan daripada akurasi FullSMO dikarenakan noise yang dihasilkan dari kurang baiknya data yang dihasilkan setelah reduksi data.

Untuik setiap dataset linier kernel memiliki akurasi yang lebih tinggi gaussian kernel di kedua SMO. Ini dikarenakan kedua kernel menggunakan parameter C=100(hard margin) tetapi gaussian kernel memiliki parameter lain yaitu gamma=1. Pemilihan parameter sangat sensitif dalam SVM oleh karena itu diperlukan penelitian dan ujicoba lebih lanjut untuk menentukan parameter yang optimal

Untuk setiap dataset gaussian memiliki waktu training yang lebih cepat dibandingkan dengan linier kernel. Dalam penelitian ini, klasifikasi dan deteksi kanker tentunya sangat dipengaruhi dengan dataset microarray yang ada. Pada dasarnya microarray adalah data bersifat non-linier dan memiliki dimensi yang sangat besar, oleh karena itu hasil reduksi dataset juga sangat memengaruhi performa dan akurasi.

## 5. Kesimpulan

Pada penelitian kali ini penulis berhasil membuat sistem SVM dengan menggunakan teknik SMO untuk mendeteksi dan mengklasifikasikan kanker dengan menggunakan data microarray. Tetapi microarray memiliki demensi yang sangat besar oleh karena itu proses reduksi dimensi dengan metode PLS dibutuhkan dan sangat berpengaruh pada hasil akurasi. Dari setiap skenario yang ada FullSMO dapat meningkatkan akurasi dan performan dengan nilai rata-rata akurasi 75% dan speedup 5.7. Dari 4 dataset dan 2 SMO linier kernel memiliki akurasi lebih tinggi daripada gaussian kernel dengan akurasi tertinggi 96% pada data leukimia JoinSMO. Pada SVM kombinasi yang tepat antara penggunaan kernel dan parameter yang digunakan seperti C, dan gammaj juga dapat meningkatkan akurasi dari sistem yang telah dibuat. Konsep dekomposisi SMO yang dilakukan dalam penelitian ini dapat di implementasikan kedalam PSVM yang menggunakan sistem parelisasi seperti MapReduce Apache Hadoop atau OpenMPI.

**Referensi**

[1] G C Fox, X H Qiu et al. Case Studies in Data Intensive Computing: Large Scale DNA Sequence Analysis. 2009. The Million Sequence Challenge and Biomedical Computing Technical Report,.
[2] X H Qiu, J Ekanayake, G C Fox et al. Computational Methods for Large Scale DNA Data Analysis. 2009. Microsoft eScience workshop.
[3] J A Blake, C J Bult. Beyond the data deluge: Data integration and bio-ontologies. 2006. Journal of Biomedical Informatics, 39(3), 314-320.
[4] J Qiu. Scalable Programming and Algorithms for Data Intensive Life Science. 2010. Applications Data-Intensive Sciences Workshop.
[5] R Guha, K Gilbert, G C Fox, et al. Advances in Cheminformatics Methodologies and Infrastructure to Support the Data Mining of Large, Heterogeneous Chemical Datasets. 2010. Current Computer-Aided Drug Design, 6: 50-67.
[6] C C Chang, B He, Z Zhang. Mining semantics for large scale integration on the web: evidences, insights, and challenges. 2004. SIGKDD Explorations, 6(2):67-76.
[7] Kiran M, Amresh Kumar, Saikat Mukherjee & Ravi Prakash. 2013. Verification and Validation of MapReduce Program Model for Parallel Support Vector Machine Algorithm on Hadoop Cluster.
[8] C. Cortes, V. Vapnik. Support Vector Networks. 1995. Machine Learning, 20: 273-297
[9] Sun, Zhanquan & Fox, Geoffrey. 2012. Study on Parallel SVM Based on MapReduce.
[10] Nasullah Khalid Alham, Maozhen Li, Yang Liu, Suhel Hammoud. A MapReduce-based distributed SVM algorithm for automatic imagean notation. 2011. Computers and Mathematics with Applications 62 ,2801–2811.
[11] Elvira Biomedical Dataset Repository. [Online]. Available: http://leo.ugr.es/elvira/DBCRepository/. [Accessed: 06-Apr-2019].
[12] J.C.Platt, Sequential minimal optimization: a fast algorithm for training support vector machines. 1998, available: http://research.microsoft.com/enus/um/people/jplatt/smoTR.pdf
[13] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy, Improvements to platt's SMO algorithm for svm classifier design, Neural Computing 13(2001)637–649.

**Lampiran**