



## Penerapan Metode *Learning Vector Quantization*, *Support Vector Machine* & *Support Vector Regression* Untuk Prediksi

Fauzan Abdurrahman, Candra Kirana, Renaldy Alamsyah

Jurnal Algoritma  
Sekolah Tinggi Teknologi Garut  
Jl. Mayor Syamsu No. 1 Jayaraga Garut 44151 Indonesia  
Email : [jurnal@sttgarut.ac.id](mailto:jurnal@sttgarut.ac.id)

1806065@sttgarut.ac.id

1806059@sttgarut.ac.id

1806053@sttgarut.ac.id

**Abstrak** – Seiring berkembangnya teknologi berbagai metode telah dibuat untuk memprediksikan suatu hal baik dalam segala bidang beberapa hal umum yang sering digunakan dalam memprediksikan sesuatu menggunakan metode vektor yaitu *Learning Vector Quantization*, *Support Vector Machine* & *Support Vector Regression*.

LVQ (*Learning Vector Quantization*) adalah suatu metode klasifikasi pola yang masing-masing unit output mewakili kategori atau kelompok tertentu. Algoritma *Support Vector Regression* (SVR) merupakan pengembangan dari algoritme *Support Vector Machine* (SVM) untuk regresi. Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari hyperplane terbaik yang berfungsi sebagai pemisah dua buah class pada input space.

Hasil yang dicapai dari ketiga metode ini ialah sama-sama berhasil memprediksi suatu hal dengan keakuratan tinggi namun dengan cara yang berbeda.

**Kata Kunci** – Aljabar Linear, Algoritma, Vektor, *Learning Vector Quantization*, *Support Vector Regression*, *Support Vector Machine*.

### I. PENDAHULUAN

Vektor merupakan besaran dalam fisika disamping memiliki nilai, juga memiliki arah [1]. Algoritma merupakan kumpulan perintah untuk menyelesaikan suatu masalah. Perintah-perintahnya dapat diterjemahkan secara bertahap dari awal hingga akhir. Masalah tersebut dapat berupa apapun dengan catatan untuk setiap masalah memiliki kriteria kondisi awal yang harus dipenuhi sebelum menjalankan algoritma. [2]

Jaringan Syaraf Tiruan terawasi (*supervised*) seperti LVQ (*Learning Vector Quantization*) adalah suatu metode klasifikasi pola yang masing-masing unit output mewakili kategori atau kelompok tertentu. Pemrosesan yang terjadi pada setiap neuron adalah mencari jarak terdekat antara suatu vector masukan ke bobot yang bersangkutan. Kelebihan metode ini adalah selain mencari jarak terdekat, selama pembelajaran unit output diposisikan dengan mengatur dan memperbaharui bobot melalui pembelajaran yang terawasi untuk memperkirakan keputusan klasifikasi [3].

Algoritma *Support Vector Regression* (SVR) merupakan pengembangan dari algoritme *Support Vector Machine* (SVM) untuk regresi [4]. Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari hyperplane terbaik yang berfungsi sebagai pemisah dua buah class pada input space. [5]

### II. URAIAN PENELITIAN

## A. Metode *Learning Vector Quantization (LVQ)*

### 1. Algoritma *Learning Vector Quantization*

Algoritma LVQ bertujuan akhir mencari nilai bobot yang sesuai untuk mengelompokkan vektor – vektor kedalam kelas tujuan yang telah di inisialisasi pada saat pembentukan jaringan LVQ. Sedangkan algoritma pengujiannya adalah menghitung nilai output (kelas vektor) yang terdekat dengan vektor input, atau dapat disamakan dengan proses pengklasifikasian (pengelompokan).

Keterangan yang kita gunakan adalah sebagai berikut:

$x$  : vektor pelatihan (input) ( $x_1, \dots, x_i, \dots, x_n$ )

$T$  : kategori yang tepat atau kelas untuk vektor pelatihan

$W_j$  : bobot vektor untuk unit output ke- $j$  ( $w_{1j}, \dots, w_{ij}, \dots, w_{nj}$ )

$C_j$  : kategori atau kelas yang ditampilkan oleh unit output ke- $j$

$\|x - w_j\|$  : jarak Euclidean antara vektor input dan bobot vektor untuk layer output ke- $j$

Berikut ini adalah algoritma pembelajaran LVQ:

langkah 0 : inisialisasi vektor referensi ; inisialisasi rating pembelajaran  $\alpha$  (0)

langkah 1 : ketika kondisi berhenti adalah false, lakukan langkah 2 sampai 6

langkah 2 : untuk setiap input pelatihan vektor  $x$  lakukan langkah 3 – 4

langkah 3 : temukan  $j$  hingga  $\|x - w_j\|$  minimum

langkah 4 : perbaharui  $w_j$  sebagai berikut :

jika  $T = C_j$ , maka

$W_j(\text{baru}) = W_j(\text{lama}) + \alpha[x - w_j(\text{lama})]$ ;

jika  $T \neq C_j$ , maka

$W_j(\text{baru}) = W_j(\text{lama}) - \alpha[x - w_j(\text{lama})]$ ;

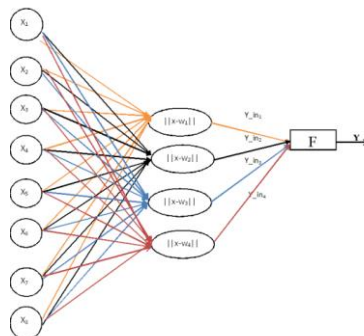
langkah 5 : kurang rating pelatihan

langkah 6 : tes kondisi berhenti: yaitu kondisi yang mungkin menetapkan sebuah jumlah tetap dari iterasi atau rating pembelajaran mencapai nilai kecil yang cukup.

Setelah proses pembelajaran *Learning Vector Quantization (LVQ)* maka langkah selanjutnya adalah proses pengambilan hasil output *Learning Vector Quantization (LVQ)*. Pengambilan hasil dilakukan hanya pada proses *Testing* (pengujian). Pada dasarnya, tahapan ini hanya memasukkan input bobot akhir kemudian mencari jarak terdekat dengan perhitungan Euclidian (jarak terdekat).

### 2. Arsitektur *Learning Vector Quantization*

Arsitektur LVQ terdiri dari lapisan input (*input layer*), lapisan kompetitif (terjadi kompetisi pada input untuk masuk ke dalam suatu kelas berdasarkan kedekatan jaraknya) dan lapisan output (*output layer*). Lapisan input dihubungkan dengan lapisan kompetitif oleh bobot. Dalam lapisan kompetitif, proses pembelajaran dilakukan secara terawasi. Input akan bersaing untuk dapat masuk ke dalam suatu kelas.



Gambar 1. Arsitektur LVQ pada penentuan Jurusan di SMA PGRI 1 Banjarbaru  
Sumber : Aplikasi Jaringan Syaraf Tiruan *Learning Vector Quantization* Untuk Penentuan Jurusan di SMA PGRI 1 Banjarbaru. 2016

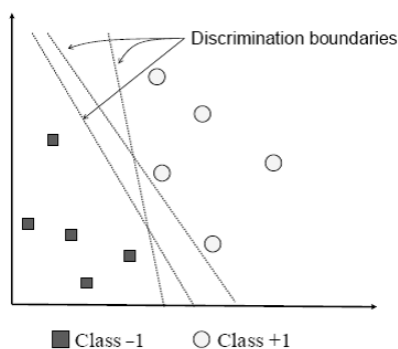
Pada tahap pengujian data selama 1 tahun dimana tujuannya untuk memprediksi hasil penjurusan disekolah dalam waktu 1 tahun. Langkahnya adalah data tersebut akan diinputkan terlebih dahulu yang nantinya akan dihitung jarak minimumnya antara bobot dan vector inputnya. Nilai jarak paling minimum dari ke-4 pola bobot yang nantinya akan dilihat Kelas bobot untuk menentukan jurusan prediksi jurusan. Nilai bobot yang didapat dari proses data training akan digunakan untuk menghitung prediksi penjurusan dengan menggunakan metode LVQ. Hasil Pengujian dapat dilihat pada tabel dibawah ini.

No	Pengujian Ke-	Iterasi	Alpha	Decalpha	minalpha	Akurasi
1	1	5	0.5	0.5	0.0001	Akurasi= 68,96%
2	2	10	0.5	0.5	0.0001	Akurasi= 75,86%
3	3	15	0.5	0.5	0.0001	Akurasi= 68,96%
4	4	20	0.5	0.5	0.0001	Akurasi= 37,93%
5	5	30	0.5	0.5	0.0001	Akurasi= 68,96%
6	6	40	0.5	0.5	0.0001	Akurasi= 68,96%
7	7	50	0.5	0.5	0.0001	Akurasi= 72,41%
8	8	60	0.5	0.5	0.0001	Akurasi= 79,31%
9	9	70	0.5	0.5	0.0001	Akurasi= 75,86%
10	10	80	0.5	0.5	0.0001	Akurasi= 55,17%
11	11	90	0.5	0.5	0.0001	Akurasi= 79,31%
12	12	100	0.5	0.5	0.0001	Akurasi= 75,86%
Rata -rata akurasi						Akurasi= 68,96%

Tabel 1. Hasil Pengujian dengan jumlah iterasi berbeda

### B. Metode Support Vector Machine (SVM)

Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari hyperplane terbaik yang berfungsi sebagai pemisah dua buah class pada input space.



Gambar 2 SVM berusaha menemukan hyperlane terbaik

Gambar 2 memperlihatkan beberapa pattern yang merupakan anggota dari dua buah class : +1 dan -1. Pattern yang tergabung pada class -1 disimbolkan dengan warna merah (kotak), sedangkan pattern pada class +1, disimbolkan dengan warna kuning (lingkaran). Problem klasifikasi dapat diterjemahkan dengan usaha menemukan garis (hyperplane) yang memisahkan antara kedua kelompok tersebut. Berbagai alternatif garis pemisah (*discrimination boundaries*) ditunjukkan pada gambar 2. Data yang tersedia dinotasikan sebagai  $x_i \in \mathbb{R}^d$  sedangkan label masing-masing dinotasikan  $y_i \in \{-1, +1\}$  untuk  $i = 1, 2, \dots, l$ , yang mana  $l$  adalah banyaknya data. Diasumsikan kedua class -1 dan +1 dapat terpisah secara sempurna oleh hyperplane berdimensi  $d$ , yang didefinisikan  $w \cdot x + b = 0$  (1)

$w$  = Bidang normal

$b$  = Posisi bidang relatif terhadap pusat koordinat Pattern  $x_i$  yang termasuk class -1

(sampel negatif) dapat dirumuskan sebagai pattern yang memenuhi pertidaksamaan  $w \cdot x + b \leq 1$  (2)

Sedang  $x_i$  yang termasuk class +1 (sampel positif)  $w \cdot x + b \geq 1$  (3)

Margin terbesar dapat ditemukan dengan memaksimalkan nilai jarak antara hyperplane dan titik terdekatnya, yaitu  $1/||w||$ . Hal ini dapat dirumuskan sebagai *Quadratic Programming (QP) problem*, yaitu mencari titik minimal persamaan (4), dengan memperhatikan constraint persamaan (5).

$$\frac{\min}{w} t(w) = \frac{1}{2} ||\vec{w}||^2 \quad (4)$$

$$y_i x_i \cdot w + b - 1 \geq 0, \forall i$$

Masalah ini dapat dipecahkan dengan berbagai teknik komputasi, di antaranya *Lagrange Multiplier*.

$$L(w, b, \alpha) = \frac{1}{2} ||\vec{w}||^2 - \sum_{i=1}^l \alpha_i \left( \left( \frac{\vec{w} \cdot \vec{x}_i}{||\vec{w}||} + b \right) - 1 \right) \quad (i = 1, 2, \dots, l)$$

$\alpha_i$  adalah Lagrange multipliers, yang bernilai nol atau positif ( $\alpha_i \geq 0$ ). Nilai optimal dari persamaan (6) dapat dihitung dengan meminimalkan  $L$  terhadap  $w$  dan  $b$ , dan memaksimalkan  $L$  terhadap  $\alpha_i$ . Dengan memperhatikan sifat bahwa pada titik optimal gradient  $L = 0$ , persamaan (6) dapat dimodifikasi sebagai maksimalisasi problem yang hanya mengandung saja  $\alpha_i$ , sebagaimana persamaan (7) di bawah.

Maximize :

$$\sum_{j=1}^l \alpha_j - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad (7)$$

Subject to :

$$\alpha_i \geq 0 \quad (i = 1, 2, \dots, l) \quad \sum_{i,j=1}^l \alpha_i y_i = 0 \quad (8)$$

Dari hasil dari perhitungan ini diperoleh  $\alpha_i$  yang kebanyakan bernilai positif. Data yang berkorelasi dengan  $\alpha_i$  yang positif inilah yang disebut sebagai support vector.

### C. Metode Support Vector Regression (SVR)

Algoritma *Support Vector Regression (SVR)* merupakan pengembangan dari algoritme *Support Vector Machine (SVM)* untuk regresi. Pada kasus regresi, *output* yang dihasilkan dalam bentuk bilangan nyata (*riil*) atau data sekuensial (*kontinue*) (Mustakim, et al., 2016). Terdapat penyempurnaan persamaan fungsi regresi nonlinier berdasarkan penelitian sebelumnya yaitu terdapat pada nilai bias. Nilai  $b$  dapat digantikan dengan vektor skalar ( $\lambda$ ) sehingga menghasilkan nilai regresi yang lebih baik ditunjukkan pada Persamaan (1). Berikut langkah-langkah proses *sequential learning Support Vector Regression* (Vijayakumar & Wu, 1999):

1. Inisialisasi parameter SVR. Parameter SVR diantaranya yaitu parameter  $\lambda$  (variabel skalar atau *lambda*),  $cLR$  (*constant learning rate*),  $\epsilon$  (nilai *epsilon*),  $C$  (*complexity*), inisialisasi  $\alpha_i$  dan  $\alpha_i = 0$  dan iterasi maksimum.
2. Menghitung Matriks *hessian* dengan persamaan sebagai berikut:  
 $R_{ij} = (K(x_i, x_j) + \lambda 2)$  (1)  
 Keterangan:  
 $R_{ij}$  = Matriks *hessian* baris ke- $i$  kolom ke- $j$   
 $K(x_i, x_j)$  = Fungsi Kernel  
 $\lambda 2$  = Variabel skalar atau *lambda*  
 $i, j$  = indeks

$l$  = Banyaknya data

Keluaran dari matriks *hessian* adalah nilai parameter *gamma* ( $\gamma$ ) yang digunakan pada tahap selanjutnya yaitu proses *sequential learning*. Nilai parameter *gamma* ( $\gamma$ ) dapat dihitung dengan Persamaan (2):

$$\gamma = \frac{cLR}{\max(\text{matriks Hessian})} \quad (2)$$

3. Pada data latih lakukan perhitungan Tahap a, b, dan c untuk tiap data *training point* yaitu:

a. Hitung nilai *error* dengan persamaan:

$$E_i = y_i - \sum_{j=1}^n (\alpha_i^* - \alpha_i) R_{ij} \quad (3)$$

Keterangan:

$E_i$  = Nilai *error* ke- $i$

$y_i$  = Nilai aktual

$\alpha_i^*, \alpha_i$  = Nilai *Lagrange multiplier*

$R_{ij}$  = Matriks *hessian* baris ke- $i$  kolom ke- $j$

$i, j$  = indeks

b. Hitung perubahan nilai *Lagrange multiplier* yang merupakan batas-batas untuk menentukan nilai *Lagrange multiplier* yang terbaru.

$$\delta \alpha_i^* = \min\{\max(\gamma(E_i - \epsilon), -\alpha_i^*), C - \alpha_i^*\} \quad (4)$$

$$\delta \alpha_i = \min\{\max(\gamma(-E_i - \epsilon), -\alpha_i), C - \alpha_i\} \quad (5)$$

Keterangan:

$\delta \alpha_i^*, \delta \alpha_i$  = perubahan nilai  $\alpha_i^*$  dan  $\alpha_i$

$\gamma$  = *Learning rate (gamma)*

$\epsilon$  = Epsilon

$C$  = Nilai kompleksitas

c. Hitung nilai *Lagrange multiplier* yang baru yaitu update nilai  $\alpha_i$  dan  $\alpha_i^*$ .

$$\alpha_i^*(\text{baru}) = \delta \alpha_i^* + \alpha_i^* \quad (6)$$

$$\alpha_i(\text{baru}) = \delta \alpha_i + \alpha_i \quad (7)$$

4. Langkah ke-3 diulangi sampai iterasi maksimum yang telah diinisialisasi di awal, atau telah mencapai konvergensi dengan syarat  $\max(|\delta \alpha_i^*|) < \epsilon$  dan  $\max(|\delta \alpha_i|) < \epsilon$ .

5. Hitung fungsi regresi dengan persamaan berikut :

$$f(x) = \sum_{i=1}^l (\alpha_i^* - \alpha_i)(K(x_i, x_j) + \lambda_2) \quad (8)$$

Keterangan :

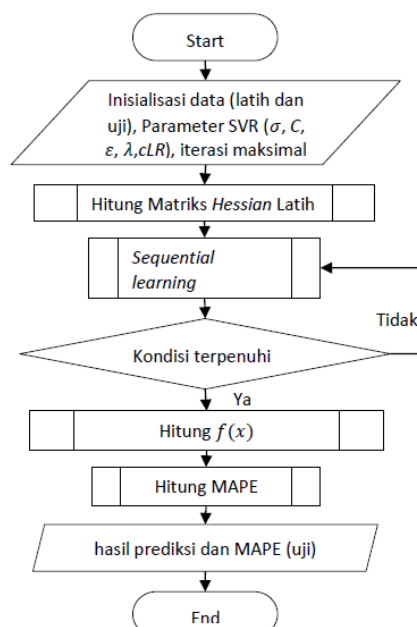
$f(x)$  = Nilai hasil regresi

$l$  = Banyaknya Data

$K(x_i, x_j)$  = Fungsi kernel

$\lambda_2$  = Variabel scalar

Diagram alir proses algoritme SVR yang diimplementasikan penelitian ini ditunjukkan pada Gambar 3.



Gambar 3 Diagram Alir SVR

### III. KESIMPULAN

1. Metode *Learning Vector Quantization* (LVQ) dapat diterapkan contoh dalam penentuan jurusan di SMA PGRI 1 Banjarbaru namun dalam penerapannya membutuhkan waktu 1 tahun.
2. Metode *Support Vector Machine* (SVM) dapat diterapkan contohnya dengan jumlah input data harga emas menunjukkan bahwa penggunaan algoritma support vector machine dengan pemaksimalan nilai parameter untuk prediksi penutupan harga emas dapat menghasilkan nilai yang cukup baik.
3. Algoritma *Support vector regression* (SVR) dapat diterapkan dalam kasus prediksi jumlah pengunjung. Proses pertama pada algoritme SVR ini adalah pemilihan data latih dan data uji dan juga pemilihan fitur. Fitur yang digunakan pada proses prediksi ini adalah bulan sebelumnya dan bersifat sekuensial.

### DAFTAR PUSTAKA

- [1] S. I. I. Susiharti, "STUDI KESALAHAN SISWA DALAM MENYELESAIKAN SOAL-SOAL VEKTOR DI SMA NEGERI 1 INDERALAYA," *Jurnal Inovasi dan Pembelajaran Fisika*, Vols. 4, no. 1, 2017.
- [2] I. F. S. N. H. R. W. Mutakhiroh, "Pemanfaatan Metode Heuristik dalam Pencarian Jalur Terpendek dengan Algoritma Semut dan Algoritma Genetika," *Jurnal Fakultas Hukum UII*, 2007.
- [3] R. O. S. & D. K. Meliawati, "Penerapan Metode Learning Vector Quantization (LVQ) Pada Prediksi Jurusan Di SMA PGRI 1 Banjarbaru," *Kumpulan jurnaL Ilmu Komputer (KLIK)*, pp. 11-20, 2016.
- [4] M. P. R. R. M. P. a. B. D. S. Raharyani, "Implementasi Algoritme Support Vector Regression Pada Prediksi Jumlah Pengunjung Pariwisata," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Vols. Vol. 2, No. 4, pp. 1501-1509, 2018.
- [5] N. D. Saputro, "Penerapan Algoritma Support Vector Machine untuk Prediksi Harga Emas," *Jurnal Informatika Upgris*, Vols. 1, no. 1 Juni, 2015.