

OPTIMASI VEKTOR BOBOT PADA LEARNING VECTOR QUANTIZATION DENGAN ALGORITMA GENETIKA UNTUK KLASIFIKASI TINGKAT RESIKO PENYAKIT STROKE

Muhammad Rifqi¹, Imam Cholissodin², Edy Santoso³

^{1,2,3}Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya

Jl. Veteran No.8 Malang, Informatika, Gedung A PTIHK – UB

Email : rifqi.brd@gmail.com¹, imamcs@ub.ac.id², edy144@ub.ac.id³

ABSTRAK

Otak manusia merupakan pusat pengaturan yang terdiri atas jutaan sel saraf atau neuron. Otak mengatur dan mengkoordinir sebagian besar, gerakan, perilaku dan fungsi tubuh seperti detak jantung, tekanan darah, keseimbangan cairan tubuh dan suhu tubuh, maka jika terjadi gangguan akan memberikan dampak yang mematikan. Stroke atau gangguan peredaran darah otak merupakan penyakit yang perlu ditangani secara cepat dan tepat. Penyakit stroke pada prinsipnya bisa dicegah, akan tetapi pada tahap deteksi dini masih sulit dan membutuhkan waktu yang lama. Oleh karena itu, dibutuhkan sistem cerdas yang mampu mempercepat dan memudahkan pendeteksian awal penyakit stroke. Learning Vector Quantization (LVQ) merupakan metode klasifikasi yang memiliki struktur yang simpel dan pembelajaran yang cepat serta dapat diandalkan untuk klasifikasi tingkat resiko penyakit stroke, namun jika vektor bobot tidak tepat maka hasil klasifikasi yang didapatkan bisa jadi tidak optimal karena terjebak pada optimum lokal. Algoritma Genetika (GA) dapat diterapkan secara luas dengan metode kecerdasan buatan lain. Pada penelitian ini membahas vektor bobot pada LVQ yang digunakan untuk klasifikasi tingkat resiko penyakit stroke yang akan dioptimasi dengan menggunakan GA agar mendapatkan vektor bobot yang paling optimal dalam melakukan klasifikasi. Percobaan dilakukan menggunakan data pasien sebanyak 200 dengan 5 fitur dan 3 kategori atau kelas yang menunjukkan metode yang diusulkan mampu melakukan klasifikasi dengan hasil pengujian akurasi yang baik. Hasil pengujian didapatkan akurasi terbaik sebesar 93%.

Kata Kunci : Stroke, Learning Vector Quantization, Algoritma Genetika, Klasifikasi.

ABSTRACT

The human brain is the control center of millions of nerve cells or neurons. Brain control and coordinate most movement, behavior and bodily functions such as heart rate, blood pressure, body fluid balance and body temperature, so if there is disruption will have an impact deadly. Stroke or circulatory disorders of the brain is a disease that needs to be handled quickly and appropriately. stroke can be prevented in principle, but at this stage of early detection is difficult and takes a long time. Therefore, it need an intelligent system that is able to accelerate and facilitate the early detection of stroke. Learning Vector Quantization (LVQ) is a classification method that has a simple structure and fast learning also reliable for the classification the risk level of stroke, but if the weight vector is not appropriate, the classification results may not be optimal because it stuck in a local optimum. Genetic Algorithm (GA) can be applied widely to other methods of artificial intelligence. This research on LVQ weight vector used for the classification of the risk level of stroke that will be optimized by GA to obtain the most optimal weight vector within the classification. The experiments were performed using 200 data patient with five features and three categories or classes that show the proposed method is able to classify with good accuracy result. Testing result obtained best accuracy of 93%.

Keyword : Stroke, Learning Vector Quantization, Genetic Algorithm, Classification.

1. PENDAHULUAN

Manusia memiliki sel saraf hingga 100 miliar dan memiliki triliunan sambungan saraf pada otaknya. Meskipun otak hanya memiliki 2 persen dari total berat tubuh yang dimiliki manusia, akan tetapi 70 persen kebutuhan oksigen dan juga kebutuhan gizi lainnya sangat diperlukan tubuh manusia agar otak dapat berkerja dengan baik. Tidak seperti otot, otak tidak mampu menyimpan cadangan zat gizi, sehingga diperlukan aliran darah agar otak bisa bekerja secara maksimal[1].

Stroke merupakan sebuah penyakit yang menyebabkan gangguan yang terjadi pada otak. Stroke dapat di tandai pada saat hilangnya fungsi bagian tubuh tertentu (kelompokan), hal tersebut disebabkan terjadinya gangguan pada aliran darah otak yang berguna untuk mengelola bagian tubuh yang kehilangan fungsinya. Menurut Statistik, penyakit stroke dalam satu tahun dapat mencapai 0.2% dari jumlah penduduk dan sekitar 1% lebih bisa dijumpai pada orang yang telah berusia lebih dari 65 tahun[2]. Penyakit stroke umumnya menjadi faktor kematian

peringkat ke-3 pada usia lanjut, setelah faktor tertinggi yaitu penyakit jantung dan kanker. Penyakit stroke banyak menyebabkan kelumpuhan pada usia lebih dari 45 tahun[3]. Menurut WHO, di Indonesia stroke menduduki peringkat ke-97 dunia dengan jumlah penderita terbanyak, dengan jumlah kematian yang mencapai sekitar 138.268 orang atau 9,70% dari total kematian tahun 2011.

Berdasarkan hasil laporan yang telah dilakukan oleh Menteri Kesehatan RI, dr. Nafsiah Mboi, Sp.A, MPH, menurutnya ada beberapa hal utama yang menjadi permasalahan pada kesehatan otak dan juga saraf, salah satunya yaitu: Penyakit otak dan saraf menimbulkan angka kecacatan dan angka kematian yang tinggi, yang kedua peningkatan pada usia harapan hidup (UHH) berdampak pada organ tubuh yang mengalami proses penuaan termasuk otak dan jaringan saraf. Beberapa upaya yang telah dilakukan oleh pemerintah adalah membangun rumah sakit pusat otak nasional. Namun, pada tahun 2014 baru diresmikan satu rumah sakit yang berfokus pada penyakit stroke. Oleh karena itu, dibutuhkan sebuah sistem cerdas dalam yang mampu digunakan dalam identifikasi secara dini untuk tingkat resiko seseorang menderita stroke khususnya untuk tim medis dan pasien.

Pada penelitian yang telah dilakukan oleh Maharani Dessy Wuryandari dan Irawan Afrianto (2012) dengan membandingkan metode *backpropagation* dan metode *Learning Vector Quantization* (LVQ) pada pengenalan wajah, tingkat kesamaan hasil pengenalan bergantung pada kondisi parameter yang ditentukan saat proses pembelajarannya. Metode LVQ memiliki hasil akurasi yang lebih baik jika dibandingkan dengan *backpropagation*, hasil akurasi yang dimiliki LVQ mencapai 37,63% dengan waktu pengenalan 32 milisecond, jika dibandingkan dengan dengan *backpropagation* akurasinya hanya mencapai 37,33% akan tetapi waktu pengenalan yang dibutuhkan 130 milisecond[4]. Penelitian yang dilakukan Ning Chen dkk (2010), menggabungkan algoritma genetika dan LVQ. LVQ digunakan karena metode tersebut sangat cocok dalam klasifikasi untuk menyelesaikan permasalahan dalam memprediksi kebangkrutan. Algoritma genetika digunakan untuk mengoptimalkan konfigurasi dari LVQ dan mendapatkan hasil yang terbaik[5]. Untuk meningkatkan akurasi yang dihasilkan LVQ maka diperlukan optimasi menggunakan algoritma genetika.

Berdasarkan fakta dan permasalahan yang ada maka pada skripsi ini digunakan judul "Optimasi Vektor Bobot pada *Learning Vector Quantization* dengan Algoritma Genetika untuk klasifikasi tingkat resiko penyakit stroke". Metode tersebut digunakan karena memiliki hasil yang terbaik dibandingkan dengan metode lainnya. Diharapkan dengan metode tersebut bisa mendapatkan hasil akurasi yang lebih baik.

2. PERMASALAHAN

Dari paparan pendahuluan, penelitian ini merumuskan permasalahan bagaimana menerapkan LVQ dan GA untuk deteksi tingkat resiko penyakit stroke dan bagaimana tingkat akurasi yang dihasilkan dari LVQ dan GA untuk deteksi tingkat resiko penyakit stroke.

3. TINJAUAN PUSTAKA

3.1 Studi terkait

Pada penelitian yang dilakukan oleh Ning Chen yang berjudul "*Hybrid Genetic Algorithm and Learning Vector Quantization Modeling for Cost-Sensitive Bankruptcy Prediction*" LVQ digunakan untuk menyelesaikan permasalahan dalam prediksi kebangkrutan sebagai sebuah klasifikasi. Hybridisasi algoritma genetika dengan algoritma klasifikasi akan baik untuk memprediksi kebangkrutan. Diharapkan dengan pendekatan tersebut bisa meminimalisir kesalahan klasifikasi yang terjadi[5].

3.2 Pengumpulan Data

Lokasi pengumpulan data penelitian ini di Laboratorium klinik sejahtera kabupaten Probolinggo. Penelitian ini menggunakan data hasil pemeriksaan pasien sebanyak 200 data dan berdasarkan batasan masalah yang sudah ditentukan.

3.3 Dasar Teori

Stroke

Stroke merupakan penyakit yang sering mengakibatkan kecacatan berupa kelumpuhan gangguan bicara, gangguan gerak, kemampuan mengingat, proses berpikir, dan juga kecacatan yang lain sebagai akibat gangguan fungsi otak. Penyebab utama gangguan pembuluh darah adalah penyumbatan kolesterol. Resiko stroke sesuai data dari Laboratorium klinik sejahtera diklasifikasikan menjadi 3 kelas. Beberapa faktor yang paling berpengaruh yang biasanya diperiksa dalam laboratorium untuk deteksi stroke adalah umur, total kolesterol, HDL, LDL, dan trigliserida.

Learning Vector Quantization

Algoritma pada jaringan LVQ memiliki tujuan untuk mendapatkan unit *output* yang paling mendekati dengan vektor *input*. Untuk mencapai tujuan itu, jika x dan w_c memiliki kelas yang sama, maka vektor bobot akan disesuaikan agar mendekati dengan vektor *input*. Jika x dan w_c memiliki kelas yang berbeda, maka vektor bobot akan disesuaikan agar menjauhi vektor masukan. Tata nama yang dipakai untuk algoritma ini adalah sebagai berikut[6]:

x → Vektor pelatihan ($x_1, \dots, x_i, \dots, x_n$).

T → Kelas atau kategori untuk vektor pelatihan.

w_j → Vektor bobot terhadap kelas ke- j .

C_j → Kategori atau kelas ke- j

$||x-w_j||$ → Jarak Euclidean antara vektor input

dan (vektor bobot untuk) output ke- j .

Langkah-langkah algoritma pembelajaran metode LVQ dapat dijelaskan sebagai berikut:

- Langkah 0** Menginisialisasi vektor referensi dan menginisialisasi laju pembelajaran (α).
- Langkah 1** Selama kondisi berhenti adalah salah, lakukan Langkah 2 sampai dengan Langkah 6.
- Langkah 2** Untuk setiap vektor pelatihan x , lakukan Langkah 3 sampai dengan Langkah 4.
- Langkah 3** Tentukan j (sebut dengan C_j), dimana $\|x - w_j\|$ adalah minimum.
- Langkah 4** Update w_j sebagai berikut:
Jika $T = C_j$, maka
 $w_j(\text{baru}) = w_j(\text{lama}) + \alpha(x(t) - w_j(\text{lama}))$
Jika $T \neq C_j$, maka
 $w_j(\text{baru}) = w_j(\text{lama}) - \alpha(x(t) - w_j(\text{lama}))$
- Langkah 5** Kurangi laju pembelajaran.
 $\alpha = \alpha * \text{pengurang}$
- Langkah 6** Uji kondisi berhenti:
Kondisi berhenti dapat ditentukan dengan menetapkan jumlah iterasi atau laju pembelajaran mencapai nilai yang cukup kecil yaitu tidak kurang dari laju pembelajaran minimum.

Algoritma Genetika

Algoritma genetika yang menggunakan pengkodean biner memiliki kelemahan jika digunakan untuk optimasi fungsi karena tidak mampu menjangkau beberapa titik solusi jika solusi tersebut berada pada daerah kontinyu. Sehingga optimasi fungsi yang lebih kompleks dan membutuhkan banyak generasi, operasi transformasi dari biner ke desimal ataupun sebaliknya memerlukan waktu yang lebih lama. Oleh karena itu, pengkodean real (real-coded genetic algorithm/RCGA) mampu menjadi solusi untuk permasalahan tersebut.

Siklus RCGA yang dilakukan adalah [7]:

1. Representasi *Chromosome*
Pada bagian ini variabel keputusan akan langsung menjadi gen string *chromosome* sehingga panjang *chromosome* akan sesuai dengan variabel yang ada.
2. Inisialisasi Populasi
Populasi dibangkitkan secara acak sesuai dengan jumlah popsize yang telah ditentukan terlebih dahulu.
3. Reproduksi
Metode *crossover* menggunakan *extended intermediate crossover* dimana *crossover* dilakukan dengan memilih dua induk (parent) secara acak dari populasi, kemudian akan menghasilkan offspring dari kombinasi parent tersebut. Jika P1 dan P2 adalah dua kromosom yang akan digunakan untuk *crossover* maka

offspring yang dihasilkan akan sesuai dengan rumusan berikut :

$$C_1 = P_1 + \alpha(P_2 - P_1)$$

$$C_2 = P_2 + \alpha(P_1 - P_2)$$

Nilai α dipilih secara acak antara [-0,25;1,25]

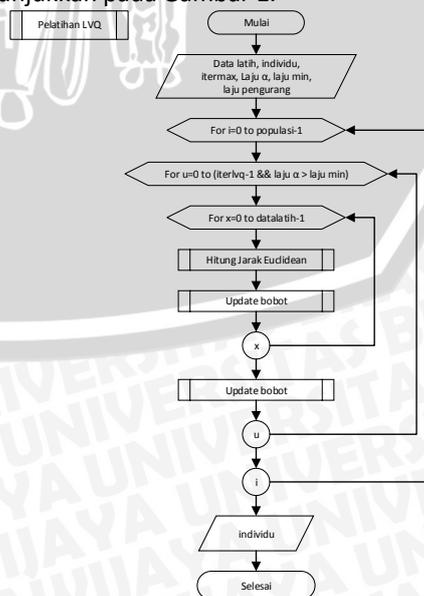
Metode Mutation menggunakan random mutation dimana mutation dilakukan dengan memilih satu induk, kemudian nilai gen yang terpilih dengan bilangan random yang kecil akan ditambah atau dikurangi. Jika domain variabel x_j adalah $[\text{min}_j, \text{max}_j]$ dan offspring yang dihasilkan adalah $C = [x'_1, \dots, x'_n]$, maka offspring yang dihasilkan akan sesuai dengan rumusan berikut:

$$x'_i = x_i + r(\text{max}_i - \text{min}_i)$$

Seleksi Elitism merupakan sebuah metode seleksi yang bekerja dengan mengumpulkan seluruh individu (*parent*) pada populasi dan *offspring* kedalam satu variabel (tempat) yang sama. Individu terbaik pada variabel tersebut akan dipilih dan masuk ke generasi selanjutnya. Metode ini akan menjamin individu yang terbaik yang akan terpilih.

4. METODOLOGI

Pada penelitian ini dibutuhkan data berupa daftar kondisi tubuh pasien yang diperlukan untuk data latih dan data uji. Penelitian ini menggunakan algoritma LVQ untuk klasifikasi dan algoritma genetika untuk menghasilkan bobot yang terbaik. Sistem diharapkan dapat memberikan keluaran berupa bobot yang optimal dengan akurasi yang paling mendekati. Siklus algoritma LVQ-GA merupakan urutan penyelesaian masalah dengan optimasi vektor bobot *Learning Vector Quantization* menggunakan Algoritma Genetika. Dari algoritma tersebut dihasilkan vektor bobot terbaik untuk melakukan klasifikasi. Alur kerja algoritma LVQ-GA ditunjukkan pada Gambar 1.



Gambar 1. Diagram Alur



Inisialisasi nilai parameter

Pada studi kasus ini inisialisasi parameter yang diperlukan seperti : data latih, jumlah iterasi, laju α , laju minimal, dan laju pengurang dan lain sebagainya. Individu merupakan vektor bobot yang nantinya akan digunakan untuk klasifikasi hingga didapatkan vektor bobot yang paling optimal.

Bangkitkan Populasi

Pada tahap ini proses yang dilakukan adalah membangkitkan populasi sejumlah 5 individu. Dimana setiap individu adalah sebuah vektor bobot yang akan digunakan pada Pelatihan LVQ. Populasi yang dibangkitkan dapat dilihat pada tabel 1.

Tabel 1. Data Vektor Bobot(Individu)

Individu 1					
W ₁	54	198	38.7	142.1	86
W ₂	54	186	36.6	109.8	198
W ₃	57	212	36.9	102.9	361
Individu 2					
W ₁	62	168	35.1	119.1	114
W ₂	50	67	42.1	112.1	164
W ₃	81	233	40.7	159.5	164
Individu 3					
W ₁	34	140	38.8	137.8	128
W ₂	41	209	37.1	138.3	168
W ₃	64	192	36.1	101.9	260
Individu 4					
W ₁	62	147	35.1	94.5	87
W ₂	54	186	36.6	109.8	198
W ₃	44	291	46.7	180.5	319
Individu 5					
W ₁	63	249	41.5	183.7	119
W ₂	28	185	42.3	111.5	156
W ₃	53	246	44.9	153.5	238

Pelatihan LVQ

Proses pelatihan dengan algoritma LVQ dalam perhitungan manual ini diperlukan data latih sebanyak 6 data. 6 data tersebut terdiri dari 2 data dengan tingkat resiko normal (N), 2 data dengan tingkat resiko rentan (R) serta 2 data dengan tingkat resiko mengkhawatirkan (M). Lihat Tabel 2.

Tabel 2. Tabel Data Latih

No	Umur	KT	HDL	LDL	Trigl	Kls	Status
1	48	197	30.5	120.5	79	1	N
2	61	205	45.2	140.6	96	1	N
3	52	319	57.2	228	169	2	R
4	32	237	46.4	159	158	2	R
5	44	391	46.7	180.5	319	3	M
6	67	276	51.4	188.2	182	3	M

Dari data latih diatas maka dapat dilakukan pelatihan sebagai berikut :

Langkah 1 : Inisialisasi Parameter

- Menginisialisasi vektor bobot sesuai dengan tabel 1.
- Menginisialisasi laju pembelajaran $\alpha = 0,1$
- Menginisialisasi laju pembelajaran minimum = 0,0001
- Menginisialisasi iterasi=0
- Menginisialisasi iterasi maksimum = 2

Langkah 2 : jika iterasi < iterasi maksimum dan laju pembelajaran $\alpha >$ laju pembelajaran minimum lakukan langkah berikutnya

Langkah 3 : Untuk vektor pelatihan x pertama dan bobot individu pertama, tentukan kelas C_j dimana $||x - w_j||$ dengan persamaan jarak Euclidean adalah minimum.

- Jarak ke w₁ dihitung dengan cara :

$$D_1 = \sqrt{\sum_{i=1}^5 (x_1 - w_{i1})^2}$$

- Jarak ke w₂ dihitung dengan cara :

$$D_2 = \sqrt{\sum_{i=1}^5 (x_1 - w_{i2})^2}$$

- Jarak ke w₃ dihitung dengan cara :

$$D_3 = \sqrt{\sum_{i=1}^5 (x_1 - w_{i3})^2}$$

Langkah 4 : Jika T=1 dan C=1, maka perbarui bobot sesuai dengan persamaan :

$$w_j(\text{baru}) = w_j(\text{lama}) + \alpha(x(t) - w_j(\text{lama}))$$

- Hingga diperoleh bobot baru individu pertama :

Tabel 3. Individu Baru Pertama

W ₁	54	198	38.7	142.1	86
W ₂	54	186	36.6	109.8	198
W ₃	57	212	36.9	102.9	361

Langkah 5 : Lakukan langkah ke 3-4 dengan melanjutkan vektor pelatihan berikutnya dan tetap pada vektor bobot pertama hingga mencapai batas maksimal iterasi atau laju pembelajaran minimal hingga akan dihasilkan vektor bobot individu pertama sebagai berikut :

Tabel 4. Bobot Akhir Individu Pertama

W ₁	54.389	187.427	36.824	131.176	78.669
W ₂	49.265	185.577	36.686	110.158	193.661
W ₃	55.349	222.033	38.144	112.755	355.666

Langkah 6 : Lanjutkan ke vektor bobot individu ke 2. Kemudian lakukan kembali langkah 3-5 untuk semua individu hingga akan dihasilkan vektor bobot untuk masing2 individu sebagai berikut.

Tabel 5. Bobot Akhir Individu

Individu 1					
W ₁	54.38918	187.4274	36.82464	131.1768	78.669
W ₂	49.2651	185.577	36.68667	110.158	193.661
W ₃	55.349	222.033	38.1446	112.7552	355.666
Individu 2					
W ₁	60.28298	175.891	35.82992	115.0681	107.784
W ₂	50	67	42.1	112.1	164
W ₃	83.54918	234.8952	40.23372	157.6942	184.836
Individu 3					
W ₁	38.94336	154.5514	38.64979	136.179	118.453
W ₂	37.84782	218.1978	39.09486	146.1081	164.974
W ₃	61.46	204.573	37.4462	111.8822	267.493
Individu 4					
W ₁	60.28298	159.8863	35.82992	103.1789	87.206
W ₂	49.21355	181.489	36.05718	106.5425	194.446
W ₃	43.04785	287.6675	45.4503	174.8466	336.852
Individu 5					
W ₁	67.36101	234.9828	39.42295	173.832	104.113
W ₂	26	183.8	43.48	110.6	163.7
W ₃	53.75811	252.9813	45.58712	158.9416	242.262



Algoritma Genetik Operator

Pada tahap ini individu yang telah diperbaharui bobotnya kemudian di *crossover*, *mutation* dan *selection*. Untuk menghasilkan individu generasi berikutnya. Sebelum di seleksi akan dievaluasi terlebih dahulu hingga menghasilkan akurasi, akurasi terbaik akan dipilih sejumlah popsize yaitu 5 individu tertinggi.

Langkah 1 : Inialisasi dan hitung nilai *crossover offspring* dan *mutation offspring*

- Nilai cr 0,4.
- Nilai mr 0,2.
- Nilai r = 0. 030687
- Nilai α

Tabel 5. Nilai α

0.323251	-0.12945	1.055757	0.491571	0.904976
----------	----------	----------	----------	----------

- Nilai Min Max

Tabel 6. Nilai Min dan Max

23	67	30.2	38.3	33.8
95	514	57.2	338.1	813

Langkah 2 : Menghitung *Child* dari *crossover*. Tentukan 2 *Parent* yang akan digunakan.

- Hitung *Child* pertama dengan persamaan :

$$C_1 = P_1 + \alpha(P_2 - P_1)$$

- Hingga dihasilkan *Child* ke-1 sebagai berikut :

Tabel 7. Offspring ke-1

w_1	58.17498	245.395	38.60668	155.3229	117.0912
w_2	29.82982	179.3471	38.85036	128.0548	164.853
w_3	56.24775	259.2479	36.99229	135.8086	265.0955

- Ulangi Langkah 2 dengan mengganti parent sebagai berikut :

$$C_2 = P_2 + \alpha(P_1 - P_2)$$

- Hingga dihasilkan *Child* ke-2 sebagai berikut :

Tabel 8. Offspring ke-2

w_1	48.1294	144.1392	39.46606	154.6881	105.4762
w_2	34.018	222.6507	43.7245	128.6534	163.8211
w_3	58.97036	198.3064	46.04103	135.0153	244.6601

Langkah 3 : Menghitung *Child* dari *mutation*. Tentukan 1 *Gen* yang akan di ubah. Sebagai contoh disini *gen* yang akan dipilih adalah *gen* ke 4.

- Hitung *Mutation* pada gen ke 4 dengan persamaan :

$$x_i = x_{ii} + r(\max_i - \min_i)$$

- Hingga dihasilkan offspring mutation sebagai berikut :

Tabel 9. Offspring Mutation ke-1

w_1	60.28298	175.891	35.82992	124.268	107.784
w_2	50	67	42.1	121.3	164
w_3	83.54918	234.8952	40.23372	166.8941	184.8366

Evaluasi Hasil dan Seleksi

Pada tahap ini proses yang dilakukan untuk mengukur akurasi yang akan digunakan untuk seleksi. Untuk itu langkah yang perlu dilakukan adalah sebagai berikut :

Langkah 1 : Lakukan langkah 1 hingga 3 pada tahap pelatihan LVQ. Hanya saja iterasi yang dilakukan hanya 1 kali iterasi. Hitung pada semua individu yang dihasilkan pada langkah 6 pelatihan LVQ.

- Hitung seluruh vektor pelatihan pada individu pertama akan menghasilkan kelas terdekat (C) pada masing-masing vektor pelatihan sebagai berikut :

Tabel 10. Evaluasi Hasil Individu

Vektor Pelatihan ke	T	C
1	1	1
2	1	1
3	2	2
4	2	2
5	3	3
6	3	2

- Hitung akurasi yang dihasilkan pada pelatihan LVQ untuk individu pertama dengan persamaan sebagai berikut :

$$Akurasi = \frac{\text{jumlah data relevan}}{\text{jumlah seluruh data}} \times 100\%$$

Maka hasilnya adalah :

$$Akurasi = \frac{5}{6} \times 100\%$$

$$Akurasi = 83,33\%$$

- Ulangi langkah ini hingga seluruh individu serta *child* dihitung dan hasilnya sebagai berikut :

Tabel 11. Evaluasi Hasil Individu dan Child

Individu ke-	Akurasi
Individu 1	83,33%
Individu 2	66,67%
Individu 3	83,33%
Individu 4	66,67%
Individu 5	66,67%
Child 1	50%
Child 2	83,33%
Child 3	66,67%

- Seleksi individu yang akan digunakan untuk iterasi berikutnya sejumlah popsize.

Maka Hasil Seleksi yang didapatkan sebagai berikut:

Tabel 12. Evaluasi Hasil Individu dan Child

Individu ke-	Individu lama yang dipilih
Individu 1	Individu 1
Individu 2	Individu 3
Individu 3	Child 2
Individu 4	Individu 4
Individu 5	Individu 5

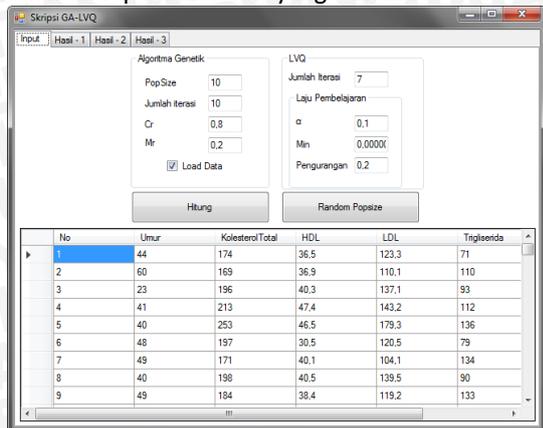
Hasilnya adalah untuk yang menjadi individu pertama pada iterasi berikutnya adalah individu 1 sesuai dengan hasil pelatihan LVQ pada langkah ke 6. begitu pula untuk individu yang lainnya.

5. IMPLEMENTASI

Antarmuka Optimasi Vektor Bobot pada *Learning Vector Quantization* dengan Algoritma Genetika untuk klasifikasi tingkat resiko penyakit stroke digunakan oleh pengguna sistem untuk dapat berinteraksi secara langsung dengan sistem. Antarmuka dari sistem ini terdiri dari empat bagian yaitu antarmuka parameter dan data, antarmuka hasil, halaman hasil akurasi dan sorting dan antarmuka grafik akurasi.

Antarmuka Parameter dan Data

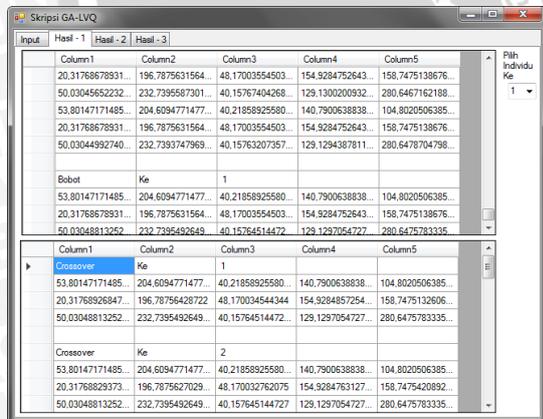
Pada gambar 2 ini merupakan tampilan dari halaman parameter dan data. Halaman berisi form untuk memberi masukkan parameter pada parameter LVQ-GA. Parameter tersebut antara lain yaitu jumlah popsize, jumlah iterasi, nilai *cross rate* dan *mutation rate*, jumlah iterasi LVQ, laju pembelajaran α , laju pembelajaran minimal dan juga laju pengurang pembelajaran. Setiap parameter akan menjadi batasan untuk mendapatkan bobot yang terbaik.



Gambar 2. Antarmuka Parameter dan Data

Antarmuka Hasil

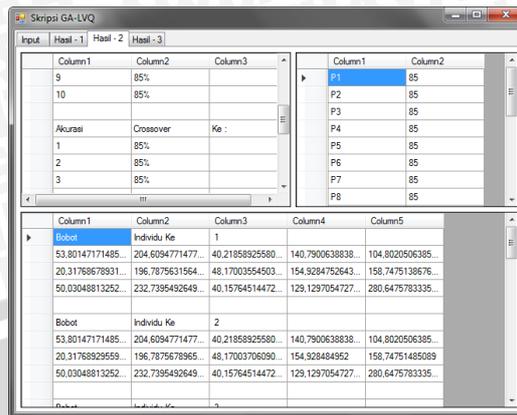
Pada gambar 3 merupakan halaman hasil, dimana *datagriedview* yang atas berisi individu hasil pelatihan LVQ sedangkan yang bawah berisi offspring dari *crossover* serta *mutation*.



Gambar 3. Antarmuka Hasil

Antarmuka Akurasi dan Sorting

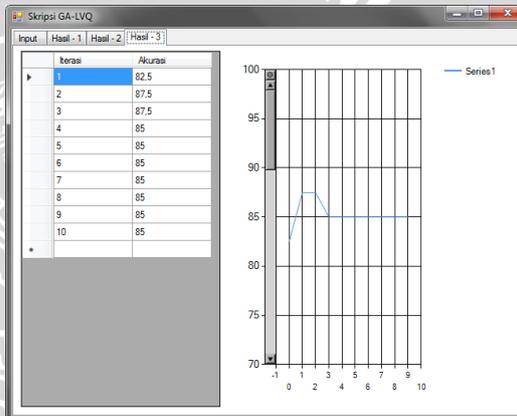
Pada gambar 4 merupakan halaman hasil akurasi dan hasil *sorting*. Dimana pada halama ini berisi tiga *datagridview*, pada *griedview* kiri atas akan menampilkan seluruh akurasi yang dihasilkan, pada *griedview* kanan atas akan menampilkan akurasi yang telah *disorting*, sedangkan pada *griedview* yang bawah akan menampilkan individu baru yang telah *disorting*.



Gambar 4. Antarmuka Akurasi dan Sorting

Antarmuka Grafik Akurasi

Pada gambar 5 merupakan halaman yang berisi grafik serta nilai akurasi tertinggi pada tiap iterasi. Hasil akurasi tertinggi tersebut ditampilkan kedalam grafik akurasi. Tampilan halaman grafik akurasi dapat dilihat seperti berikut.

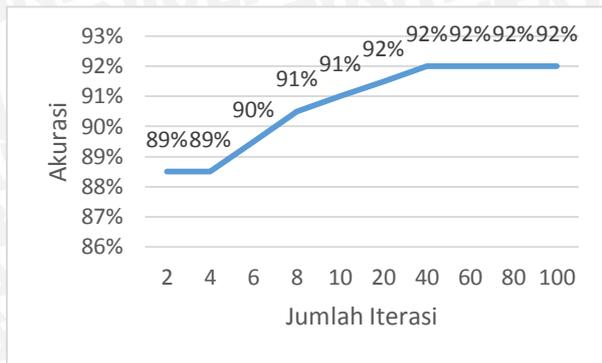


Gambar 5. Antarmuka Grafik Akurasi

6. HASIL DAN PEMBAHASAN

Pengujian tingkat akurasi dapat dilakukan dengan membandingkan data aktual dengan data prediksi dari algoritma LVQ dan GA. Pada penelitian ini dilakukan pencarian nilai akurasi rata-rata dari setiap pengujian parameter. Pada tiap percobaan digunakan data latih sebanyak 160 data dan data uji sebanyak 40 data.

- Pengujian Jumlah Iterasi Algoritma Genetika
 Pengujian jumlah iterasi algoritma genetika akan digunakan untuk mencari jumlah iterasi terbaik untuk menghasilkan individu yang terbaik untuk melakukan klasifikasi.



Gambar 6. Pengujian Jumlah Iterasi GA

Pada Gambar 6 menunjukkan bahwa pada iterasi 2 hingga 4 tidak terjadi perubahan akurasi, perubahan akurasi baru mulai mengalami peningkatan saat mencapai iterasi 6 dan terus mengalami kenaikan hingga akhirnya nilainya mulai konvergen saat mencapai iterasi 40. Iterasi algoritma geetik berpengaruh untuk menghasilkan individu baru saat algoritma genetika operator dilakukan. Pengujian ini menunjukkan semakin besar iterasi yang dilakukan maka akan dihasilkan akurasi yang semakin baik pula.

- Pengujian Iterasi Laju Pembelajaran LVQ

Pengujian jumlah iterasi laju pembelajaran akan digunakan untuk mencari jumlah iterasi terbaik untuk menghasilkan individu yang terbaik setelah mengalami proses pelatihan pada LVQ.

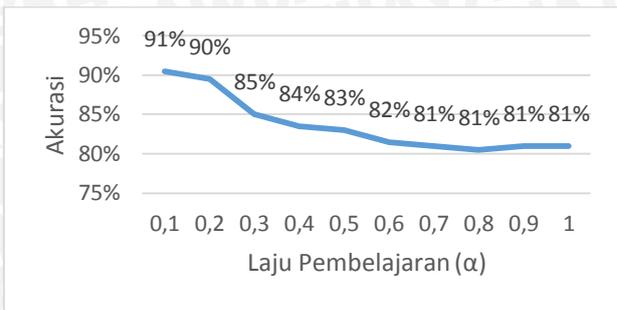


Gambar 7. Pengujian Iterasi Laju Pembelajaran LVQ

Pada Gambar 7 sejak iterasi 2 menuju 4 sudah mengalami kenaikan meskipun peningkatan yang terjadi tidak terlalu signifikan. Walau jarak jumlah iterasi tidak terlalu besar akan tetapi selalu terjadi peningkatan hingga mencapai iterasi 14. Dimana hal tersebut menunjukkan bahwa iterasi pelatihan pada LVQ, cukup berpengaruh pada saat proses pelatihan.

- Pengujian Laju Pembelajaran

Pengujian laju pembelajaran digunakan untuk mencari nilai laju pembelajaran yang terbaik agar mampu memperbaiki individu sehingga mampu menghasilkan hasil klasifikasi yang lebih baik.



Gambar 8. Pengujian Laju Pembelajaran

Pada Gambar 8 menunjukkan nilai akurasi yang mengalami penurunan. Dari hasil tersebut diketahui bahwa semakin besar nilai laju pembelajaran mengakibatkan semakin rendah tingkat akurasinya. Pada LVQ, laju pembelajaran akan berpengaruh pada saat perbaikan nilai bobot sesuai dengan persamaan yang digunakan untuk memperbarui nilai bobot dalam algoritma pelatihan LVQ.

- Pengujian Laju Pembelajaran Minimal

Pengujian laju pembelajaran minimal digunakan untuk mencari nilai laju pembelajaran minimal yang tedekat dimana laju pembelajaran minimal akan berfungsi seperti iterasi laju pembelajaran agar mampu memperbaiki individu sehingga mampu menghasilkan hasil klasifikasi yang lebih baik.



Gambar 9. Pengujian Laju Pembelajaran Minimal

Akurasi pada Gambar 9 mengalami kenaikan ketika jumlah laju pembelajaran minimal semakin kecil. Laju pembelajaran minimal juga merupakan salah satu kondisi berhenti ketika pelatihan pada LVQ. Semakin banyak iterasi maka semakin kecil pula nilai laju pembelajaran, laju pembelajaran minimum yang makin kecil menunjukkan bahwa iterasi semakin banyak dan akurasi juga semakin baik.

- Pengujian Laju Pengurang Pembelajaran

Pengujian laju pengurang pembelajaran digunakan untuk mencari nilai laju pengurang pembelajaran yang paling optimal sehingga dapat menghasilkan akurasi yang lebih baik serta akan menentukan batasan iterasi pula jika mencapai nilai tertentu.

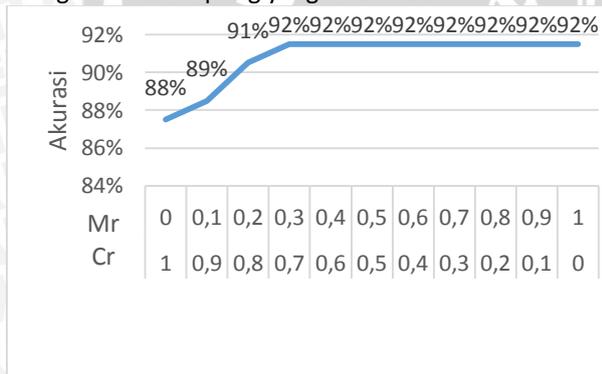


Gambar 10. Pengujian Laju Pengurang Pemb

Grafik pada Gambar 10 mengalami nilai yang stabil saat nilai pengurang pembelajaran dari 0,1-0,3 akan tetapi mulai tidak stabil ketika nilai laju pengurang semakin besar. Laju pengurang mempengaruhi besar kecilnya pergeseran nilai laju pembelajaran saat pelatihan LVQ. Grafik yang tidak stabil bisa dipengaruhi oleh perbedaan data uji yang digunakan pada tiap percobaan.

• Pengujian Nilai *Cr* dan *Mr*

Pengujian nilai *cr* dan *mr* digunakan untuk mencari kombinasi nilai *cr* dan *mr* yang terbaik agar mampu menghasilkan offspring yang lebih baik.

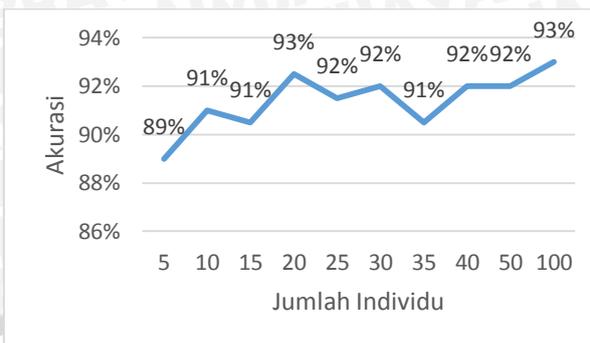


Gambar 11. Pengujian Nilai *Cr* dan *Mr*

Grafik pada Gambar 11 menunjukkan bahwa *mr* lebih berpengaruh ketika jumlah *mr* semakin besar dan *cr* semakin kecil. Hal tersebut ditunjukkan saat *mr* 0 hingga 0,3 kemudian nilainya mulai konvergen. Proses mutasi terjadi hanya pada salah satu sel, akan tetapi memberikan hasil yang lebih baik dibandingkan *crossover*.

• Pengujian Jumlah Individu

Pengujian jumlah individu digunakan untuk mencari jumlah serta kombinasi individu yang dapat menghasilkan akurasi yang terbaik. Setiap individu dirandom sesuai dengan *popsiz*e yang telah ditentukan.



Gambar 12. Pengujian Jumlah Individu

Pada Gambar 12 grafik menunjukkan akurasi mengalami naik turun yang tidak stabil yang disebabkan oleh *random* individu yang tidak menentu akurasinya. Hasil *random* setiap individu yang dihasilkan akan sangat berpengaruh dengan nilai akurasinya. Semakin banyak *popsiz*e maka semakin banyak kombinasi individu yang dihasilkan sehingga akurasi akan semakin baik. Akurasi terbaik didapatkan saat *popsiz*e berjumlah 100.

7. PENUTUP

1. Kesimpulan

Berdasarkan pada hasil perancangan, implementasi, dan pengujian yang telah dilakukan maka kesimpulan yang didapat adalah didapatkan kesimpulan sebagai berikut.

1. Untuk mengimplementasikan metode *Learning Vector Quantization* dan Algoritma Genetika (LVQ-GA) untuk klasifikasi tingkat resiko stroke adalah menentukan inisialisasi parameter yang digunakan, melakukan pelatihan dengan menggunakan LVQ, melakukan optimasi dengan algoritma genetika operator, melakukan evaluasi dan seleksi. Diawali dengan menentukan parameter yang dilakukan untuk menghasilkan bobot yang terbaik. Kemudian membangkitkan populasi yaitu vektor bobot yang akan dioptimasi serta yang akan menjadi pengukur untuk melakukan klasifikasi. Setiap vektor bobot akan dilatih dengan vektor pelatihan pada LVQ dimana setiap vektor pelatihan merupakan data latih, vektor bobot yang telah mengalami proses pelatihan akan dioptimasi dengan melakukan *crossover* serta *mutation* hingga menghasilkan *offspring*. Setiap individu populasi dan juga *offspring* yang dihasilkan akan dievaluasi akurasinya untuk diseleksi hingga didapatkan vektor bobot (individu) dengan akurasi yang tertinggi.
2. Berdasarkan hasil pengujian parameter LVQ dan algoritma genetika dapat disimpulkan bahwa akurasi terbaik sebesar 93% dengan detail parameter yang optimal yaitu jumlah iterasi algoritma genetika = 40, nilai *cross rate* = 0.3, nilai *mutation rate* = 0.7, jumlah iterasi LVQ = 14, laju pembelajaran α = 0,1, laju pembelajaran minimal =

0.0000000001, laju pengurang pembelajaran = 0,6 dan jumlah individu 100.

4.1. Saran

Saran yang diberikan adalah masih dibutuhkan data yang banyak untuk melakukan pelatihan agar menghasilkan tingkat akurasi yang lebih maksimal, untuk pengembangan sistem dalam penelitian selanjutnya yaitu pada pemilihan metode *cost-sensitive learning* yang digunakan untuk mengetahui tingkat *error rate* yang dihasilkan. Perlu dilakukan penelitian lebih lanjut untuk menstabilkan proses pelatihan pada LVQ serta meningkatkan akurasi tanpa mengalami penurunan pada tiap iterasi.

DAFTAR PUSTAKA

- [1] Holistic Health Solution, 2011, "Stroke di Usia Muda". Jakarta. PT Gramedia widiasarana Indoneisa.
- [2] Suharjo, J.B. dan Cahyono, B. 2008. "Gaya Hidup & Penyakit Modern". Jakarta. Kanisius.
- [3] Anies, Dr. dr Mkes PKK. 2006. "Waspada Ancaman Penyakit Tidak Menular Solusi Pencegahan dari Aspek Perilaku dan Lingkungan". Jakarta. PT Elex Media Komputindo.
- [4] Wuryandari, Maharani Dessy dan Afrianto, Irawan. 2012. "Perbandingan Metode Jaringan Saraf Tiruan Backpropagation Dan Learning Vector Quantization Pada Pengenalan Wajah". Jurnal Komputer dan Informatika.
- [5] Chen, Ning. Ribeiro, Bernardete. Vieira, Armando S. Duarte, Joao dan Neves, Joao. 2010. "Hybrid Genetic Algorithm and Learning Vector Quantization Modeling for Cost-Sensitive Bankruptcy Prediction". Second International Conference on Machine Learning and Computing.
- [6] Fausett, L. 1994, "Fundamentals of Neural Network : Architectures, Algorithms, and Applications", Prentice-Hall, Inc., New Jersey.
- [7] Mahmudy, Wayan Firdaus. 2013. Algoritma Evolusi. Program Teknologi Informasi dan Ilmu Komputer. Universitas Brawijaya. Malang.