

Implementasi *Genetic Algorithm* Dan *Artificial Neural Network* Untuk Deteksi Dini Jenis *Attention Deficit Hyperactivity Disorder*

Brilliant Aristyo Rahadian⁽¹⁾, Candra Dewi⁽²⁾, Bayu Rahayudi⁽³⁾

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ⁽¹⁾ brilliantarc@gmail.com, ⁽²⁾ dewi_candra@ub.ac.id, ⁽³⁾ ubay1@ub.ac.id

Abstrak

ADHD (*Attention Deficit Hyperactivity Disorder*) adalah gangguan motorik yang penderitanya sulit untuk berkonsentrasi dan melakukan sesuatu secara berlebihan. Deteksi jenis ADHD bisa dilakukan oleh pakar seperti dokter, perawat maupun orang psikolog yang menguasai bidang tersebut dan mereka memberikan solusi untuk penyembuhan anak yang terkena ADHD. Namun karena terbatasnya pakar maka cukup sulit untuk konsultasi ke seorang pakar. Maka dari itu bisa dibuat sistem untuk deteksi dini ADHD. Dalam penelitian ini dibuat implementasi metode GA-LVQ2 untuk deteksi dini jenis ADHD. Tahapan dari implementasinya adalah inisialisasi populasi, *crossover*, *mutation*, evaluasi, seleksi *elitism*, dan pelatihan LVQ2. Representasi solusi yang digunakan adalah *real coded genetic algorithm*. Panjang kromosom pada penelitian ini sebanyak 45 yang merupakan gejala ADHD. Hasil pengujian yang telah dilakukan adalah didapatkan akurasi tertinggi sebesar 95% pada pengujian dengan 20 data uji dengan nilai parameter ukuran populasi 10, *crossover rate* 0.9, *mutation rate* 0.1, banyak generasi 40, *learning rate* 0.1, pengurang *learning rate* 0.1, nilai konstanta ϵ 0.35. Keluaran sistem berupa bobot LVQ terbaik yang telah diuji dan memiliki akurasi tinggi.

Kata kunci: *attention deficit hyperactivity disorder* (ADHD), *genetic algorithm* (GA), *artificial neural network* (ANN), *learning vector quantization* (LVQ), LVQ2, GA-LVQ2

Abstract

ADHD (*Attention Deficit Hyperactivity Disorder*) is a psychomotor disorder that the patient is difficult to concentrate and do something excessively. Types of ADHD detection can be done by experts such as doctors, nurses and psychologists who has mastered and give solutions for therapy who affected by ADHD. However due to the limited expertise it's quite difficult to consultancy with an experts. Therefore can be made a system for early detection of ADHD. In this research, the implementation of GA-LVQ2 methods for early detection of ADHD types. Stages of implementation are population initialization, *crossover*, *mutation*, evaluation, *elitism* selection, and LVQ2 training. Using *real coded genetic algorithm* as the representation of solution. Chromosome length in this study was 45, which is a symptom of ADHD. The result of the testing has been done is the highest accuracy reached 95% in the test with 20 data testing with the parameter value of population size 10, *crossover rate* 0.9, *mutation rate* 0.1, generations 40, *learning rate* 0.1, the *learning rate reducer* 0.1, the constant value ϵ 0.35. System output is the best LVQ weights that have been tested and have high accuracy.

Keywords: *attention deficit hyperactivity disorder* (ADHD), *genetic algorithm* (GA), *artificial neural network* (ANN), *learning vector quantization* (LVQ), LVQ2, GA-LVQ2

1. PENDAHULUAN

Pertumbuhan adalah proses perubahan pada ukuran, besar, jumlah atau dimensi pada suatu individu yang sifatnya kuantitatif artinya bisa

diukur dengan satuan. Perkembangan adalah perubahan fungsi dan struktur tubuh menjadi lebih baik dan merupakan hasil kematangan saraf dengan organ tubuh dan sifatnya kualitatif

(Chamidah, 2009).

Masa pertumbuhan dan perkembangan manusia yang sangat rentan adalah ketika berusia 0-8 tahun. Biasanya anak akan suka berlari-lari dan menanyakan sesuatu yang ada disekitarnya. Maka dari itu perlu perhatian lebih untuk memastikan apakah perilaku yang dilakukan anak tersebut berlebihan atau normal (Fadila, 2016).

Ada perilaku abnormal yang biasa disebut dengan ADHD atau kepanjangannya *Attention Deficit Hyperactivity Disorder*. ADHD adalah gangguan motorik yang penderitanya sulit untuk berkonsentrasi dan melakukan sesuatu secara berlebihan (Fadila, 2016).

Deteksi untuk mengecek ADHD bisa dilakukan oleh pakar seperti dokter, perawat maupun orang psikolog yang paham di bidang tersebut. Namun, karena terbatasnya pakar maka cukup sulit untuk konsultasi ke seorang pakar. Maka dari itu bisa dibuat sistem untuk deteksi dini ADHD. Tujuannya adalah memudahkan masyarakat untuk deteksi dini agar anak yang terkena ADHD bisa diberikan penanganan sesuai masalah yang ada dengan cepat.

Penelitian yang dilakukan oleh Arifien (2016) membuat sistem deteksi jenis ADHD menggunakan metode *Learning Vector Quantization*. Didapatkan akurasi sebesar 70% dengan *learning rate* 0.1, pengali *learning rate* 0.1, jumlah data latih sebanyak 10% dari total data, maksimum *epoch* 3, minimum *alpha* 0.01. Selain itu penelitian yang dilakukan oleh Rifqi (2016) yang melakukan optimasi vektor bobot pada *Learning Vector Quantization* untuk klasifikasi tingkat resiko penyakit *stroke*. Optimasi tersebut menggunakan algoritma genetika. Akurasi mencapai 93% dengan parameter yang digunakan yaitu banyak iterasi GA 40, *crossover rate* 0.3, *mutation rate* 0.7, banyak iterasi LVQ 4, laju pembelajaran *alpha* 0.1, laju pembelajaran minimal 0.00000000001, laju pengurang pembelajaran 0.6 dan jumlah individu 100.

Learning Vector Quantization (LVQ) adalah teknik klasifikasi berpola dimana setiap unit keluaran merepresentasikan kelas atau kategori. Vektor bobot ke unit keluaran merupakan referensi dari vektor ke suatu kelas. Bobot diberi aturan selama melakukan pelatihan (Kusumadewi, 2003). Kelebihan metode LVQ adalah mampu mengatur nilai bobot pada pembelajaran terawasi untuk memperkirakan hasil klasifikasi (Fausett, 1994).

Genetic Algorithm (GA) merupakan tipe

dari *Evolutionary Algorithm* (EA) yang merupakan teknik optimasi paling populer. GA mampu menyelesaikan masalah yang kompleks. GA menggunakan kromosom sebagai pengkodean solusi untuk pemecahan masalah. Untuk membangkitkan solusi baru, GA menggunakan *crossover* sebagai operator utama dan *mutation* sebagai operator tambahan (Mahmudy, 2013).

Banyak keunggulan dari GA seperti bisa digunakan pada berbagai data, menyelesaikan masalah yang kompleks dan GA cukup fleksibel untuk digabungkan dengan algoritma lainnya. Banyak solusi yang didapatkan karena GA melakukan pencarian dengan sangat luas dan dilakukan secara cepat (Mahmudy, 2013).

2. LANDASAN KEPUSTAKAAN

2.1. *Attention Deficit Hyperactivity Disorder* (ADHD)

Gangguan pada perkembangan anak-anak dalam aktifitas motoriknya yang menyebabkan tingkah laku anak menjadi tidak normal atau berlebihan adalah ADHD. Tanda-tanda ADHD seperti sulit untuk diam, perasaan gelisah dan tidak dapat duduk dengan tenang. Anak yang terkena ADHD akan terganggu perkembangan kognitif, perilaku, sosialisasi dan komunikasi yang akan berakibat mengganggu kegiatan akademis sehingga mendapatkan nilai yang kurang padahal anak tersebut mempunyai intelegensi yang baik (Putri, 2013).

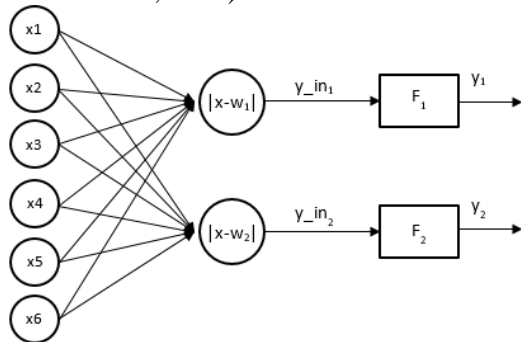
2.2. *Artificial Neural Network* (ANN)

Artificial Neural Network (ANN) atau Jaringan Syaraf Tiruan adalah sistem yang bisa mengolah informasi dengan karakteristik mirip jaringan syaraf manusia. Mampu menyerupai otak manusia untuk melakukan pembelajaran dan menyimpan pengetahuan yang telah didapat. Data dipelajari sehingga bisa memberikan kesimpulan pada data lain. Terdapat 3 hal pada ANN yaitu arsitektur jaringan, metode *training*, dan fungsi aktivasi (Pattiserlihun, 2011).

2.3. *Learning Vector Quantization 2* (LVQ2)

Pada Jaringan Syaraf Tiruan terdapat metode untuk klasifikasi yaitu LVQ. Tipe arsitektur dari LVQ adalah *Single-layer Feedforward* yang terdapat vektor input dan unit output. LVQ melakukan pembelajaran pada lapisan kompetitif. Terdapat *Reference Vector* yang menjadikan unit *output* sebagai acuan pada sebuah kelas yang mewakili *output*. Perhitungan

untuk mengelompokkan vektor *input* adalah dengan menghitung jaraknya dengan vektor bobot menggunakan *Euclidean Distance* (Kusumadewi, 2003).



Gambar 1. Arsitektur Jaringan LVQ

Ada 3 lapisan pada LVQ yaitu:

1. Lapisan *input* adalah lapisan pertama yang menerima masukan untuk diproses.
2. Lapisan kompetitif adalah lapisan untuk memproses masukan berdasarkan jarak kedekatannya.
3. Lapisan *output* adalah lapisan yang menghasilkan keluaran dari proses *Learning Vector Quantization*.

Pada penelitian ini variasi LVQ yang digunakan adalah LVQ2. Perbedaan variasi LVQ terletak pada cek *update* bobot. Pada LVQ2 cek *update* bobot dengan ketentuan sebagai berikut (Fausett, 1994):

1. Data latih ke i memiliki kelas yang sama dengan y_r .
2. Data latih ke i masuk ke *window*, syaratnya adalah Persamaan 1.

$$\frac{d_c}{d_r} > 1 - \epsilon \quad \text{dan} \quad \frac{d_r}{d_c} < 1 + \epsilon \quad (1)$$

Jika kedua kondisi tersebut terpenuhi maka melakukan *update* nilai bobot sesuai dengan Persamaan 2.

$$\begin{aligned} y_c(t+1) &= y_c(t) - \alpha(t) * [x(t) - y_c(t)] \\ y_r(t+1) &= y_r(t) + \alpha(t) * [x(t) - y_r(t)] \end{aligned} \quad (2)$$

Keterangan:

x adalah data latih ke i

y_c adalah *reference vector* terdekat

y_r adalah *reference vector* terdekat kedua

d_c adalah jarak antara data latih i dengan y_c

d_r adalah jarak antara data latih i dengan y_r

window adalah kondisi dengan perhitungan tertentu

ϵ adalah nilai batasan yang ditentukan untuk perhitungan *window*

2.4. Genetic Algorithm (GA)

Genetic Algorithm (GA) adalah salah satu tipe dari *Evolutionary Algorithm* (EA) yang sangat dikenal. GA memiliki kemampuan untuk menyelesaikan masalah yang kompleks pada berbagai bidang. GA akan memanipulasi kode dari parameter yang ada atau disebut dengan kromosom. Kemudian melakukan pencarian dari beberapa titik dalam populasi dengan menggunakan syarat dari tujuan. Digunakan operator acak yang sifatnya menggunakan teori peluang (Mahmudy, 2013).

2.5. Struktur Genetic Algorithm

Cara GA dalam memecahkan masalah adalah dengan memetakan masalah menjadi kromosom string. Kromosom ini terdiri dari gen yang merupakan variabel untuk mendapatkan solusi. Fungsi *fitness* untuk menggambarkan kualitas individu tersebut apakah merupakan solusi yang optimal. Kualitas dari solusi yang optimal ditentukan oleh representasi solusi yang merupakan kromosom (Mahmudy, Marian & Luong, 2012b). Dipilih *Real-coded Genetic Algorithm* untuk merepresentasikan kromosom pada penelitian ini karena data yang digunakan memiliki nilai *real*.

2.5.1. Inisialisasi Populasi

Tahap awal dari GA adalah inisialisasi populasi yang melakukan pembuatan individu (kromosom) awal secara acak sesuai dengan solusi yang ditentukan.

2.5.2. Reproduksi Crossover

Reproduksi *crossover* dilakukan dengan memilih secara acak 2 individu dengan metode *extended intermediate crossover*. Pembangkitan *offspring* dari 2 individu yang terpilih dihitung dengan Persamaan 3.

$$\begin{aligned} C_1 &= P_1 + \alpha (P_2 - P_1) \\ C_2 &= P_2 + \alpha (P_1 - P_2) \end{aligned} \quad (3)$$

Nilai α dibangkitkan secara acak pada range -0.25 sampai 1.25.

2.5.3. Reproduksi Mutation

Reproduksi *mutation* dilakukan dengan memilih secara acak 1 individu dengan metode *random mutation*. Nilai gen dari *offspring*

dihitung pada Persamaan 4.

$$x'_i = x_i + r (max_i - min_i) \quad (4)$$

Nilai r dibangkitkan secara acak pada *range* -0.1 sampai 0.1.

2.5.4. Evaluasi

Evaluasi setiap kromosom dilakukan untuk setiap generasi dengan menghitung nilai *fitness* yang bisa menggambarkan kualitasnya.

2.5.5. Seleksi *Elitism*

Seleksi *Elitism* adalah seleksi yang dilakukan dengan memilih beberapa individu sebanyak populasi dengan nilai *fitness* yang tertinggi dari semua individu yang ada.

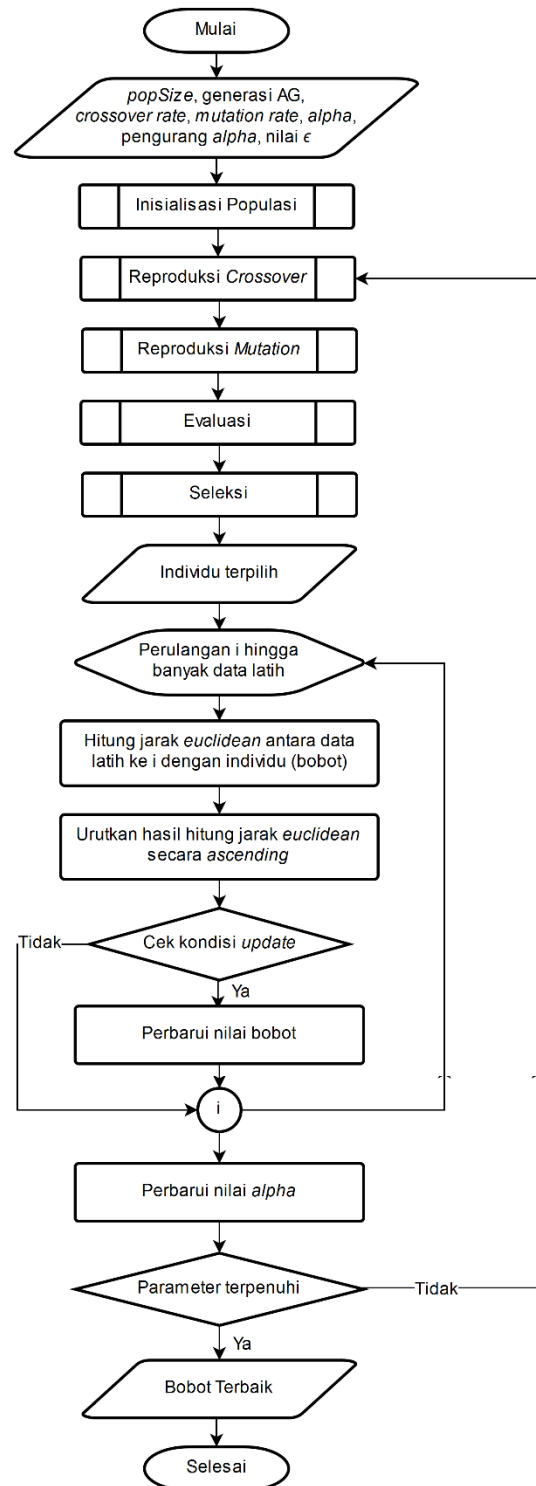
3. METODE

3.1. Deskripsi Data

Data yang digunakan sebanyak 100 dengan 45 gejala yang tampak seseorang mengalami ADHD. Untuk setiap gejala, ada 3 opsi nilai yaitu nilai 50 untuk tingkat kemunculan gejala dengan intensitas tinggi (selalu), nilai 35 untuk tingkat kemunculan gejala dengan intensitas menengah (kadang-kadang), dan nilai 15 untuk tingkat kemunculan gejala dengan intensitas rendah (tidak pernah). Pembobotan tersebut merupakan hasil konsultasi oleh Arifien (2016) dengan seorang pakar psikologi di House of Fatima yaitu Bapak Suyanto, S.Psi., M.Psi.

3.2. Perancangan Algoritma

Metode yang diimplementasikan adalah GA-LVQ2. Inisialisasi parameter-nya adalah ukuran populasi (*popSize*), banyak generasi GA, *crossover rate*, *mutation rate*, *learning rate*, pengurang *learning rate*, dan nilai ϵ . Gambar 2 adalah *flowchart* dari perancangan algoritma GA-LVQ2.



Gambar 2. *Flowchart* Perancangan Algoritma

Tahap pertama dari metode yang diimplementasikan adalah inisialisasi populasi yaitu dengan menghasilkan beberapa individu awal. Pada penelitian ini, banyaknya bobot yang dihasilkan sebanyak kelas yang ada pada data dan menghasilkan beberapa bobot LVQ dari data setiap kelasnya secara acak. Panjang kromosom sebanyak 45.

Reproduksi *crossover* dilakukan dengan memilih 2 individu acak dari individu awal dan membangkitkan nilai α secara acak. Banyak *offspring* yang dihasilkan sebanyak *crossover rate* * ukuran populasi.

Reproduksi *mutation* dilakukan dengan memilih secara acak 1 individu untuk diproses dan membangkitkan nilai r secara acak. Banyak *offspring* yang dihasilkan sebanyak *mutation rate* * ukuran populasi.

Evaluasi adalah proses menghitung nilai *fitness* dari setiap individu. Nilai *fitness* pada tahap ini digantikan dengan nilai akurasi sesuai dengan Persamaan 5.

$$\text{Akurasi} = \frac{\text{banyak kelas sama}}{\text{banyak data}} \times 100\% \quad (5)$$

Seleksi *elitism* yaitu proses untuk memilih individu terbaik dari sejumlah individu sebanyak *popSize* dari individu *parent* maupun *offspring*.

Pelatihan LVQ2 dilakukan untuk individu terpilih dengan melatih bobot dengan 80 data latih yang telah disiapkan. Menghitung jarak *Euclidean* antara bobot dengan data latih, kemudian mengurutkan hasil hitung jarak secara *ascending*, selanjutnya cek kondisi *update* sesuai dengan ketentuan algoritma LVQ2.

Kemudian *update* nilai α sesuai dengan Persamaan 6.

$$\alpha = \alpha - (\alpha * \text{pengurang } \alpha) \quad (6)$$

Selanjutnya adalah cek parameter terpenuhi, apakah sudah mencapai generasi maksimum, jika sudah maka berhenti dan jika belum kembali ke proses Reproduksi *Crossover*.

4. HASIL DAN PEMBAHASAN

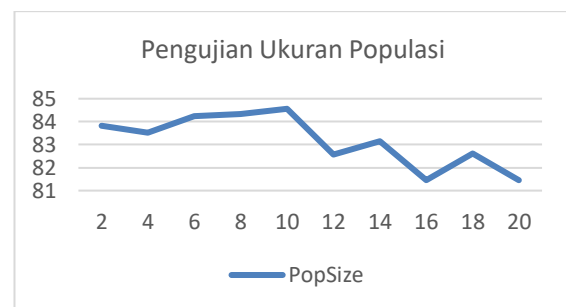
Berikut adalah hasil dari implementasi GA-LVQ2 beserta pembahasan dari pengujian yang telah dilakukan.

4.1. Pengujian Ukuran Populasi

Pengujian ukuran populasi bertujuan untuk mengetahui ukuran populasi yang dapat menghasilkan solusi optimal. Pengujian akan dilakukan 10 kali uji coba dengan ukuran populasi 2 sampai 20. Pengujian dilakukan pada 1 generasi, nilai cr 0.5, nilai mr 0.5, *learning rate* 0.1, pengurang *learning rate* 0.1, dan konstanta ϵ 0.35. Hasil pengujian ukuran populasi dijabarkan pada Tabel 1 dan Gambar 3.

Tabel 1. Rataan *Fitness* Pengujian Ukuran Populasi

PopSize	Rataan <i>Fitness</i>
2	83.812
4	83.531
6	84.249
8	84.339
10	84.561
12	82.569
14	83.138
16	81.456
18	82.621
20	81.453



Gambar 3. Grafik Pengujian Ukuran Populasi

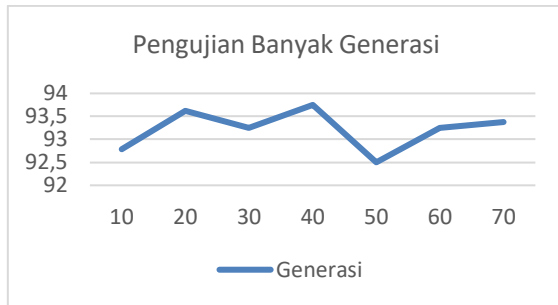
Dilihat dari hasil pengujian tersebut, hasil pengujian tidak stabil dikarenakan pembentukan individu awal dengan cara acak maka akurasi yang didapatkan tergantung dengan inisialisasi awal. Ukuran populasi 10 memiliki nilai rata-rata *fitness* yang tinggi dibandingkan dengan ukuran populasi lainnya. Maka untuk pengujian selanjutnya digunakan ukuran populasi 10.

4.2. Pengujian Banyak Generasi

Pengujian banyak generasi bertujuan untuk mengetahui banyak generasi yang dapat menghasilkan solusi optimal pada penelitian ini. Pengujian dilakukan 10 kali pada setiap banyak generasi dengan banyak generasi 10 sampai dengan 70. Pengujian dilakukan dengan ukuran populasi 10, nilai cr 0.5, nilai mr 0.5, nilai *learning rate* 0.1, nilai konstanta ϵ 0.35. Hasil pengujian banyak generasi dijabarkan pada Tabel 2 dan Gambar 4.

Tabel 2. Rataan *Fitness* Pengujian Banyak Generasi

Generasi	Rataan <i>Fitness</i>
10	92.7875
20	93.625
30	93.25
40	93.75
50	92.5
60	93.25
70	93.375



Gambar 4. Grafik Pengujian Banyak Generasi

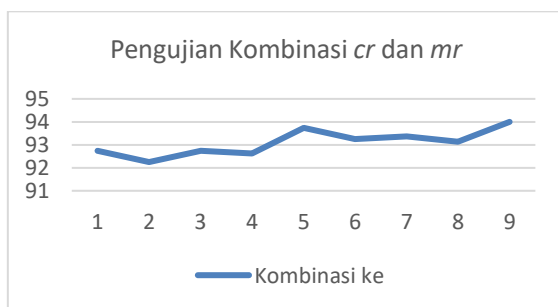
Dilihat dari hasil tersebut, grafik cenderung naik dari banyak generasi 10 hingga generasi 40 dan turun pada banyak generasi 50 kemudian naik pada generasi 60 dan 70. Banyak generasi 40 memiliki rata-rata nilai *fitness* tertinggi sebesar 93.75% sedangkan banyak generasi 50 memiliki rata-rata *fitness* rendah sebesar 92.5%. Maka banyak generasi 40 dipilih untuk digunakan pada pengujian selanjutnya karena mampu menghasilkan rata-rata *fitness* tinggi.

4.3. Pengujian Kombinasi *cr* dan *mr*

Pengujian akan dilakukan 10 kali dengan nilai *cr* dan *mr* 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 dan 0.9. Pengujian dilakukan untuk 50 generasi dengan ukuran populasi 10, nilai *learning rate* 0.1, nilai pengurang *learning rate* 0.1 dan nilai konstanta ϵ 0.35. Hasil pengujian kombinasi *crossover rate* dan *mutation rate* dijabarkan pada Tabel 3 dan Gambar 5.

Tabel 3. Rataan *Fitness* Pengujian Kombinasi *cr* dan *mr*

<i>cr</i>	<i>mr</i>	Rataan <i>Fitness</i>
0.1	0.9	92.75
0.2	0.8	92.25
0.3	0.7	92.75
0.4	0.6	92.625
0.5	0.5	93.75
0.6	0.4	93.25
0.7	0.3	93.375
0.8	0.2	93.125
0.9	0.1	94

Gambar 5. Grafik Pengujian Kombinasi *cr* dan *mr*

Dilihat dari hasil pengujian tersebut, grafik cenderung naik dari kombinasi pertama sampai kombinasi ke 5 kemudian turun pada kombinasi 6 dan naik lagi pada kombinasi ke 9. Kombinasi *cr* dan *mr* dengan nilai 0.9 dan 0.1 memiliki nilai rata-rata *fitness* yang tertinggi dibandingkan dengan kombinasi *cr* dan *mr* lainnya dengan pengujian yang sama sebesar 94. Maka untuk pengujian selanjutnya digunakan kombinasi *cr* dan *mr* dengan nilai 0.9 dan 0.1.

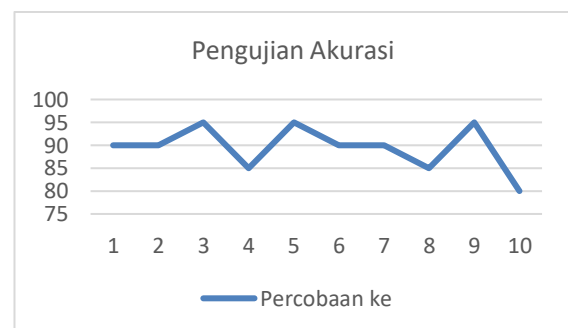
4.4. Pengujian Akurasi GA-LVQ2

Pengujian akurasi adalah pengujian yang dilakukan untuk menguji individu hasil optimasi dengan data uji yang telah disiapkan. Data uji yang sudah memiliki kelas digunakan sebagai pembandingan hasil klasifikasi. Nilai parameter GA yang digunakan untuk pengujian akurasi ini adalah ukuran populasi 10, *crossover rate* 0.9, *mutation rate* 0.1 dan banyak generasi 40.

Hasil pengujian akurasi GA-LVQ2 dipilih akurasi tertinggi dari beberapa individu yang ada sebanyak *popSize*. Tabel 4 dan Gambar 6 menjabarkan hasil pengujian akurasi GA-LVQ2 yang dilakukan sebanyak 10 kali.

Tabel 4. Hasil Pengujian Akurasi GA-LVQ2

Percobaan ke	Akurasi
1	90%
2	90%
3	95%
4	85%
5	95%
6	90%
7	90%
8	85%
9	95%
10	80%



Gambar 6. Grafik Pengujian Akurasi

Dilihat dari grafik tersebut tampak bahwa hasil pengujian akurasi masih fluktuatif. Ini dikarenakan pemberian nilai bobot awal sangat berpengaruh pada hasil pengujian. Dari 10 kali

percobaan pengujian akurasi GA-LVQ2 bisa dihitung rata-rata akurasi sebesar 89.5%. Akurasi tertinggi didapatkan pada pengujian tersebut sebesar 95%. Ini artinya implementasi GA-LVQ2 mampu mengenali 19 data dari 20 data uji dengan kondisi tertentu.

5. KESIMPULAN

Dari hasil yang didapatkan pada penelitian implementasi *genetic algorithm* dan *artificial neural network* untuk deteksi dini jenis *attention deficit hyperactivity disorder* dapat diambil kesimpulan:

1. Implementasi GA-LVQ2 untuk deteksi dini jenis ADHD berhasil dilakukan sesuai dengan perancangan yang dibuat dengan data latih sebanyak 80 dan data uji sebanyak 20.
2. Implementasi GA-LVQ2 memiliki hasil akurasi tertinggi sebesar 95% dan rata-rata akurasi sebesar 89.5% untuk 10 kali percobaan pada pengujian dengan 20 data uji. Adapun nilai parameter yang digunakan adalah :
 - a. Ukuran populasi = 10
 - b. *Crossover rate* = 0.9
 - c. *Mutation rate* = 0.1
 - d. Banyak generasi = 40
 - e. *Learning rate* = 0.1
 - f. Pengurang *learning rate* = 0.1
 - g. Nilai konstanta $\epsilon = 0.35$

6. DAFTAR PUSTAKA

- Arifien, Z., 2016. *Deteksi Jenis Attention Deficit Hyperactivity Disorder (ADHD) pada Anak Usia Dini menggunakan Metode Learning Vector Quantization*. Malang: Universitas Brawijaya.
- Chamidah, A. N., 2009. *Deteksi Dini Gangguan Pertumbuhan dan Perkembangan Anak*. [Online] Available at: <https://core.ac.uk/download/pdf/11062365.pdf> [Accessed 28 January 2017].
- Fadila, P. N., 2016. *Identifikasi Jenis Attention Deficit Hyperactivity Disorder (ADHD) pada Anak Usia Dini menggunakan Metode Neighbor Weighted K-Nearest Neighbor (NWKNN)*. Malang: Universitas Brawijaya.
- Fausett, L., 1994. *Fundamentals of Neural Networks : Architectures, Algorithms, and Applications*. New Jersey: Prentice Hall Inc.
- Kusumadewi, S., 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- Mahmudy, W. F., 2013. *Modul Kuliah Algoritma Evolusi*. Malang: Universitas Brawijaya.
- Mahmudy, W. F., Marian, R. & Luong, L., 2012b. Solving part type selection and loading problem in flexible manufacturing system using real coded genetic algorithms – Part I: Modeling. *International Conference on Control, Automation and Robotics*, pp. 699-705.
- Pattiserlihun, A., 2011. *Aplikasi Jaringan Saraf Tiruan (Artificial Neural Network) pada Pengenalan Pola Tulisan*. DIY, Prosiding Pertemuan Ilmiah XXV HFI Jateng & DIY.
- Putri, I. K., 2013. *Rancang Bangun Game sebagai Terapi Cognitive Behaviour pada Penderita Attention Deficit Hyperactivity Disorder (ADHD) menggunakan Kinect*. Malang: Universitas Brawijaya.
- Rifqi, M., 2016. *Optimasi Vektor Bobot Learning Vector Quantization (LVQ) dengan Algoritma Genetika untuk Deteksi Dini Penyakit Stroke*. Malang: Universitas Brawijaya.