

ANALISIS IMPLEMENTASI ALGORITMA *LEARNING VECTOR QUANTIZATION* 3 TERHADAP KLASIFIKASI PENENTUAN PENERIMA BANTUAN LANGSUNG TUNAI DESA

Aidil Dzakwan Alfari Islami, Fajri Profesio Putra

Teknik Informatika, Politeknik Negeri Bengkalis

Jalan Bathin Alam, Sei. Alam, Bengkalis

aidildzakwan6@gmail.com

ABSTRAK

Program Bantuan Langsung Tunai (BLT) oleh pemerintah Indonesia bertujuan mengurangi kemiskinan dan meningkatkan kesejahteraan. Namun, proses seleksi penerima sering menghadapi bias sosial yang mengurangi objektivitas. Penelitian ini mengembangkan sistem klasifikasi berbasis algoritma *Learning Vector Quantization* (LVQ) 3.0 untuk mendukung proses seleksi yang lebih adil. Data yang digunakan mencakup 100 calon penerima dari Desa Kelapapati, yang dibagi 80% untuk pelatihan dan 20% untuk pengujian. Parameter yang digunakan meliputi usia, tingkat pendidikan, kepemilikan rumah, penggunaan listrik, dan sumber air. Evaluasi dilakukan menggunakan teknik *K-Fold Cross Validation* untuk mengukur akurasi algoritma. Sistem diuji dengan berbagai nilai *Learning Rate* (0,01; 0,025; 0,05; 0,075; dan 0,1) dan *Window* (0,2; 0,3; 0,4; dan 0,5). Hasil menunjukkan bahwa algoritma LVQ 3.0 mampu mengklasifikasikan data dengan akurasi 86,22%, sehingga layak diterapkan dalam seleksi penerima BLT. Penambahan variasi data disarankan untuk meningkatkan kemampuan sistem dalam mengenali pola secara lebih baik lagi, karena semakin bervariasi data yang digunakan, maka semakin tinggi pula tingkat akurasi sistem dalam mengidentifikasi pola yang ada.

Kata kunci : *Bantuan Langsung Tunai, Machine Learning, Jaringan Syaraf Tiruan, Learning Vector Quantization 3, K-Fold Cross Validation.*

1. PENDAHULUAN

Pembangunan bangsa erat kaitannya dengan upaya pemenuhan kebutuhan masyarakat, terutama dalam mewujudkan kesejahteraan bagi kelompok miskin [1].

Salah satu langkah strategis yang dilakukan pemerintah untuk mencapai tujuan adalah melalui program Bantuan Langsung Tunai (BLT), yang bertujuan untuk mengurangi angka kemiskinan, meningkatkan daya beli masyarakat, serta menjaga stabilitas ekonomi nasional, khususnya di tingkat desa sebagai unit terkecil pemerintahan.

Namun demikian, dalam praktiknya, proses seleksi calon penerima BLT masih menghadapi beberapa tantangan, terutama dalam aspek objektivitas. Mekanisme seleksi masih mengandalkan musyawarah desa yang rentan terhadap bias subjektif, terutama ketika jumlah penerima melebihi batas akolasi dana yang menimbulkan dilema dalam penentuan prioritas penerima bantuan.

Peraturan Menteri Desa No. 13 Tahun 2023 telah mengatur kriteria untuk penerima BLT Dana Desa [2].

Meski demikian, keputusan akhir tetap berada di tangan pihak desa yang sering kali menghadapi dilema moral yang rumit dalam menentukan prioritas penerima bantuan. Misalnya, perbandingan antara keluarga janda lansia yang memiliki rumah sendiri dengan keluarga muda yang menyewa rumah dan sama-sama miskin bisa menimbulkan perbedaan pendapat yang tajam antar warga maupun perangkat desa.

Untuk mengurangi potensi bias dan meningkatkan akurasi dalam proses pengambilan keputusan, pendekatan berbasis data merupakan opsi yang menarik untuk diteliti. Teknologi *machine learning* dapat memberikan solusi yang lebih objektif, konsisten, dan efisien. Salah satu algoritma yang relevan untuk tugas ini adalah *Learning Vector Quantization* (LVQ), yaitu metode pembelajaran terawasi (*supervised learning*) yang efektif dalam tugas klasifikasi berdasarkan pola data [3].

Algoritma LVQ secara umum bekerja dengan menghitung jarak antara data masukan dan prototipe tiap kelas, kemudian memperbarui bobot prototipe tersebut melalui proses pembelajaran sehingga mencapai klasifikasi yang optimal. Algoritma LVQ 3 merupakan pengembangan dari versi sebelumnya dengan penyempurnaan pada pembaruan bobot, sehingga lebih stabil dalam proses pembelajaran.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk menganalisis dan mengimplementasikan algoritma *Learning Vector Quantization* 3 dalam proses klasifikasi calon penerima Bantuan Langsung Tunai (BLT) Dana Desa. Dengan memanfaatkan pendekatan ini, diharapkan proses penyaluran bantuan dapat dilakukan secara lebih objektif, adil, dan berbasis data, sehingga meningkatkan kepercayaan masyarakat terhadap mekanisme distribusi bantuan sosial serta mendukung terwujudnya kesejahteraan yang lebih merata.

2. TINJAUAN PUSTAKA

Penelitian terdahulu terkait analisis penggunaan algoritma LVQ 3 sebagai dasar pembangunan model klasifikasi telah banyak dilakukan. Penelitian Khurun Ain et al. yang berjudul “Klasifikasi status gizi pada lansia menggunakan learning vector quantization 3 (LVQ 3)” menggunakan algoritma LVQ 3 untuk mengklasifikasikan status gizi pada lansia. Dari pengujian yang dilakukan, nilai akurasi tertinggi adalah 86,67% dengan *learning rate* = 0,2 dan *window* = 0,4 [4].

Penelitian selanjutnya oleh M. Deni Alfani dan Ilman Zuhri Yadi berjudul “Learning vector quantization 3 (LVQ 3) usage to determine recipients of the family hope program (case study: Tanjung Lubuk District)” menghasilkan sistem klasifikasi yang digunakan untuk mengklasifikasikan penerima program keluarga harapan. Pengujian dilakukan sebanyak 5 kali untuk mengurangi bias. Dengan *learning rate* = 0,1; 0,3; 0,5; dan 0,7 serta *window* = 0,1; dan 0,5. Didapatkan rata-rata akurasi 94% dengan 94,4% sebagai nilai tertinggi [5].

Berdasarkan penelitian-penelitian sebelumnya ini, dapat disimpulkan bahwa algoritma LVQ 3 memiliki performa yang baik untuk dalam menyelesaikan permasalahan klasifikasi, khususnya pada data yang memiliki kriteria terstruktur.

2.1. Bantuan Langsung Tunai (BLT)

Bantuan Langsung Tunai atau biasa disingkat BLT, menurut Peraturan Menteri Desa, Pembangunan Daerah Tertinggal, dan Transmigrasi Republik Indonesia Nomor 13 Tahun 2023 BAB 1 Pasal 1 Nomor 9 mendefinisikan BLT sebagai kegiatan pemberian langsung dalam bentuk dana tunai kepada keluarga penerima manfaat dan diputuskan melalui musyawarah desa. Menurut Pasal 4, besaran tunai yang diberikan sebesar Rp.300.000,00 setiap bulan dalam satu tahun. Karena BLT bersifat sementara program ini beberapa kali diturunkan pada keadaan tertentu, misalnya krisis BBM dan pandemi COVID-19 di masa silam. Sedangkan, BLT desa difungsikan untuk memberikan bantuan uang tunai kepada keluarga kurang mampu untuk memenuhi kebutuhan pokok mereka, hal ini dilakukan untuk meningkatkan daya beli di daerah-daerah pedesaan, karena jika ekonomi di lapisan masyarakat terendah dapat bergerak, maka pada akhirnya perekonomian pada tingkat nasional juga akan meningkat [6].

2.2. Jaringan Syaraf Tiruan (JST)

Jaringan Syaraf Tiruan (JST) merupakan sistem pemrosesan informasi yang meniru mekanisme jaringan syaraf biologis. JST terdiri dari unit-unit sederhana yang disebut neuron, yang saling terhubung dan bekerja secara paralel untuk mengenali serta mempelajari pola dari data [7].

Model ini belajar dari data latih dan membentuk representasi internal untuk digunakan dalam prediksi data baru.

Secara umum, JST terdiri dari tiga jenis lapisan: lapisan input yang menerima data mentah, lapisan tersembunyi yang mempelajari pola atau hubungan antar data, dan lapisan output yang menghasilkan keluaran akhir. Pada algoritma seperti *Learning Vector Quantization* (LVQ), data input dapat langsung terhubung ke lapisan output untuk proses klasifikasi berbasis jarak.

2.3. Learning Vector Quantization

Learning Vector Quantization (LVQ) adalah algoritma pembelajaran terawasi (*supervised learning*) berbasis Jaringan Syaraf Tiruan (JST) yang dirancang khusus untuk tugas klasifikasi. Pada algoritma LVQ, setiap kelas direpresentasikan oleh vektor prototipe. Vektor ini diperbarui secara iteratif selama proses pelatihan dengan tujuan mendekatkannya pada pola distribusi data input.

Proses klasifikasi dilakukan dengan menghitung jarak input ke seluruh prototipe, kemudian data tersebut dimasukkan ke dalam kelas yang memiliki yang memiliki prototipe dengan jarak terdekat. Tujuan utama LVQ adalah memperoleh prototipe yang mampu memisahkan kelas dengan akurasi optimal [8].

2.4. Learning Vector Quantization 3

Metode LVQ 3 merupakan pengembangan dari metode yang sudah ada, yaitu LVQ 1 dan LVQ 2.1. LVQ 3 melakukan pengembangan terhadap metode LVQ 2.1 untuk memastikan vektor agar selalu mendekati distribusi dari kelas. Metode ini juga mengubah pengkondisian dan penentuan kondisi *window* [8].

2.5. Euclidian Distance

Euclidian distance merupakan metode yang digunakan untuk mengukur jarak antara dua titik yang pada kasus ini merupakan data input dan prototipe kelas. Semakin dekat jarak antara data input dengan prototipe kelas, semakin besar kemungkinan data tersebut termasuk dalam kelas tersebut [9].

$$j = \sqrt{\sum_{i=1}^n (X_i - W_j)^2} \quad (1)$$

Keterangan: j = jarak dari data X_i ke W_j
 X_i = elemen ke- i dari data X_i
 W_j = elemen ke- i dari data W_j
 n = jumlah elemen dari data X_i dan data W_j

2.6. K-fold Cross Validation

K-fold Cross Validation adalah metode validasi yang digunakan untuk mengevaluasi performa model secara efisien, terutama saat jumlah data terbatas [10].

Teknik ini membagi dataset menjadi k bagian (*fold*) yang sama besar. Model dilatih dan diuji sebanyak k kali, di mana pada setiap iterasi satu *fold* digunakan sebagai data uji, dan sisanya sebagai data

latih. Nilai evaluasi akhir diperoleh dari rata-rata hasil setiap iterasi [11].

Metode ini dinilai lebih objektif dibandingkan teknik konvensional seperti *train-test split*, karena seluruh data digunakan bergiliran sebagai data uji. Dalam praktik umum, nilai k yang direkomendasikan adalah 5 atau 10, karena memberikan keseimbangan antara akurasi evaluasi dan efisiensi komputasi [12].

3. METODE PENELITIAN

Dalam mengimplementasikan *machine learning* dalam pengambilan keputusan, diperlukan pembangunan model klasifikasi terlebih dahulu. Model ini membutuhkan dataset yang akan digunakan untuk melatih model dan nantinya akan diuji seberapa baik performa dari model klasifikasi. Penelitian ini menggunakan algoritma *Learning Vector Quantization 3* (LVQ 3) sebagai algoritma dari model klasifikasi yang ingin dibangun, serta metode *K-fold Cross Validation* digunakan untuk memvalidasi performa model klasifikasi, sehingga dataset yang dipakai akan digunakan secara maksimal dan pembagian tugas (data latih dan data uji) akan bergantian pada setiap iterasi. Hasil akurasi akhir merupakan rata-rata akurasi untuk seluruh iterasi [11]. Dalam mendukung proses pelatihan dan pengujian model, dibangun sebuah sistem yang mempermudah integrasi algoritma serta evaluasi secara efisien.

3.1. Dataset

Dataset yang digunakan pada penelitian ini bersumber dari data calon penerima BLT Desa Kelapapati. Pada awalnya, jumlah total sampel yang tersedia adalah sebanyak 92 data. Namun, untuk mendukung validasi model menggunakan teknik *K-fold Cross Validation* dengan nilai $K = 5$ dan $K = 10$, maka jumlah data disesuaikan menjadi 90 data. Penyesuaian ini dilakukan agar data dapat dibagi secara merata kedalam setiap *fold* tanpa menyisakan data yang tidak terpakai. Dengan demikian, proses validasi dapat dilakukan secara optimal dan konsisten pada setiap iterasi pengujian.

Model klasifikasi yang dibangun dalam penelitian ini dilatih menggunakan dataset yang terdiri dari dua komponen utama: fitur (input) dan label (output). Fitur merupakan sekumpulan variabel yang merepresentasikan karakteristik calon penerima. Fitur-fitur ini dipilih berdasarkan kriteria sosial ekonomi yang umum digunakan dalam penentuan kelayakan penerima bantuan sosial di Desa Kelapapati, seperti: *usia*, *pendidikan kepala keluarga*, *kepemilikan rumah*, *daya listrik terpasang*, dan *sumber air minum*. Label adalah variabel target yang menjadi hasil akhir dari proses klasifikasi. Dalam penelitian ini, label yang digunakan adalah: *Terima* (memenuhi kriteria) dan *Tidak Terima* (Tidak memenuhi kriteria).

3.2. K-fold Cross Validation

K-fold Cross Validation merupakan metode validasi yang dipakai dalam penelitian ini untuk mengukur performa model klasifikasi yang dibangun. Teknik ini dipilih karena mampu memaksimalkan pemanfaatan dataset yang tersedia, sekaligus dapat memberikan estimasi akurasi model yang lebih stabil dan dapat diandalkan.

Secara umum, *K-fold Cross Validation* bekerja dengan cara membagi dataset menjadi K bagian (*fold*) yang berukuran sama besar. Pada setiap iterasi proses validasi, satu *fold* digunakan sebagai data uji (*testing data*), sedangkan *fold* lainnya sebanyak $K - 1$ digunakan sebagai data latih (*training data*). Proses ini kemudian diulang sebanyak K kali, dengan pergantian *fold* yang berperan sebagai data uji pada setiap iterasi, sehingga seluruh data berkesempatan digunakan secara merata baik sebagai data latih maupun data uji [11].

Sebagai ilustrasi, jika jumlah data pada dataset adalah 90 dan nilai K yang digunakan adalah 5, maka dataset akan dibagi kedalam 5 *fold*, masing-masing berisikan 18 data. Pada setiap iterasi, satu *fold* (18 data) digunakan sebagai data uji, sedangkan 72 data lainnya ($K - 1$ *fold*) digunakan untuk melatih model. Dengan demikian, rasio data latih terhadap data uji dalam skenario ini adalah 4:1. Sebaliknya, jika nilai K yang digunakan adalah 10, maka dataset akan dibagi kedalam 10 *fold*, masing-masing terdiri dari 9 data. Pada setiap iterasi, 9 data akan digunakan sebagai data uji dan 81 data sebagai data latih, menghasilkan rasio 9:1. Proses ini diulang sebanyak 10 kali hingga setiap *fold* pernah berperan sebagai data uji.

Setiap iterasi pelatihan akan menghasilkan model klasifikasi beserta vektor prototipe kelas yang dibentuk berdasarkan data latih. Model ini kemudian akan diuji terhadap data uji pada *fold* yang bersangkutan untuk menghitung tingkat akurasi pada iterasi tersebut. Nilai akurasi akhir model didapatkan berdasarkan jumlah rata-rata dari nilai akurasi di setiap iterasi. Nilai rata-rata ini mencerminkan performa keseluruhan dari model terhadap dataset dan dianggap sebagai nilai representatif.

3.3. Tahap Pelatihan

Pada fase ini, proses utama yang dilakukan adalah pembaruan atau *update* vektor prototipe kelas menggunakan data latih hingga mencapai nilai yang optimal. Sebelum proses pelatihan dimulai, beberapa parameter penting akan diinisialisasi terlebih dahulu, seperti *learning rate* (α), *window*, dan jumlah *fold* (K).

Selama proses pelatihan, setiap data latih diukur jaraknya terhadap masing-masing prototipe kelas menggunakan rumus *euclidian distance*. Berdasarkan aturan dalam algoritma LVQ 3, prototipe yang relevan akan diperbarui berdasarkan hasil perbandingan tersebut dan digunakan sebagai acuan untuk data berikutnya.

Setelah seluruh data latih dalam satu siklus pelatihan (*epoch*) selesai diproses, nilai *epoch* akan bertambah dan proses pelatihan diulang menggunakan prototipe yang telah diperbarui pada *epoch* sebelumnya. Dengan demikian, vektor prototipe akan terus berkembang secara iteratif berdasarkan data yang digunakan dan aturan aturan pembaruan dalam algoritma. Proses ini berlangsung secara berulang hingga mencapai kondisi berhenti yang telah ditentukan sebelumnya, yaitu ketika jumlah maksimum *epoch* telah tercapai. Nilai prototipe akhir dari proses ini kemudian disimpan dan akan digunakan dalam fase pengujian untuk mengklasifikasikan data uji.

Secara umum, proses pelatihan ini bertujuan untuk memungkinkan model mempelajari pola dan distribusi data latih secara bertahap. Melalui pembaruan vektor prototipe yang dilakukan secara iteratif, diharapkan model dapat menggeneralisasi dengan baik dan menghasilkan prediksi yang akurat terhadap data uji.

Langkah-langkah pelatihan data pada metode LVQ 3 adalah sebagai berikut [8]:

- Inisialisasi vektor prototipe awal (W_j), *window* (ϵ), *alfa* (α), *Reduction alfa*, dan *MaxEpoch*.
- Masukkan data latih (X_i) beserta targetnya (T).
- Cek kondisi berhenti:
 - $Epoch < MaxEpoch$
- Selama kondisi berhenti masih salah, kerjakan langkah 4-8.
- Untuk setiap data latih, hitung jarak prototipe (W) dan data latih (X) menggunakan rumus:

$$j = \sqrt{\sum_{i=1}^n (X_i - W_j)^2} \quad (2)$$

- Tentukan jarak terdekat pertama (D_c) dengan c sebagai kelasnya dan jarak terdekat kedua (D_r) dengan r sebagai kelasnya.
- Perbarui w_j sebagai berikut:
 - Jika $c = T$ dan $T \neq r$ maka hanya prototipe dari jarak terdekat pertama yang diubah

$$W_c(\text{baru}) = W_c(\text{lama}) + \alpha [X_i - W_c(\text{lama})] \quad (3)$$

- Jika $c \neq r$ dan $r = T$ maka, tentukan kondisi *window* dengan rumus berikut:

$$\text{Min} \left(\frac{D_c}{D_r}, \frac{D_r}{D_c} \right) > (1 - \epsilon)(1 + \epsilon) \quad (4)$$

Apabila masuk kedalam *window*, maka prototipe diperbarui dengan persamaan:

$$W_c(\text{baru}) = W_c(\text{lama}) - \alpha [X_i - W_c(\text{lama})] \quad (5)$$

$$W_r(\text{baru}) = W_r(\text{lama}) + \alpha [X_i - W_r(\text{lama})] \quad (6)$$

Sedangkan, apabila tidak masuk kedalam *window* ataupun $c \neq r$ dan $r \neq T$, maka pembaruan prototipe tidak perlu dilakukan.

- Kurangi nilai α dan $Epoch+1$
- Setelah iterasi selesai ulangi langkah 1-7, dengan catatan bahwa nilai prototipe terbaru setelah diperbarui akan dijadikan sebagai nilai prototipe awal (W_j) untuk iterasi selanjutnya.
- Tes kondisi berhenti

3.4. Tahap Pengujian

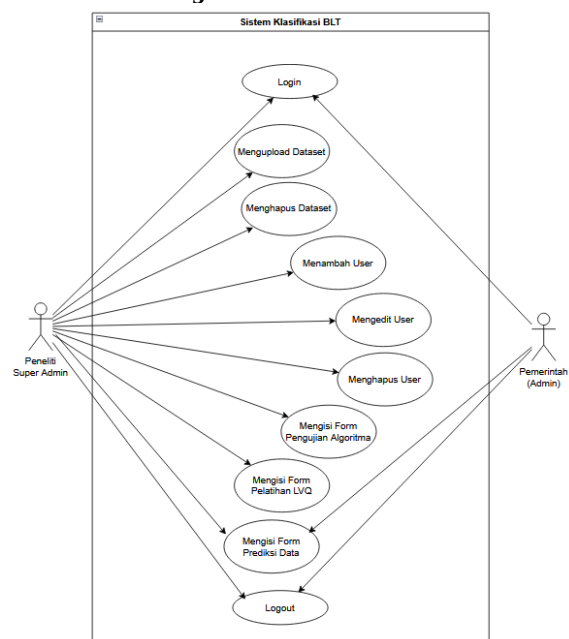
Pada fase pengujian, prototipe akhir yang diperoleh dari hasil proses pelatihan akan digunakan untuk mengukur akurasi model klasifikasi. Tujuan dari tahap ini adalah untuk mengevaluasi sejauh mana model yang telah dilatih mampu melakukan prediksi yang akurat terhadap data yang belum pernah dilihat sebelumnya, yaitu data uji. Proses pengujian dilakukan dengan cara menghitung jarak antara setiap vektor data uji terhadap seluruh vektor prototipe kelas yang telah terbentuk pada tahap pelatihan. Setelah jarak dihitung, sistem akan menentukan kelas prediksi untuk data uji tersebut berdasarkan prototipe dengan jarak terdekat.

Setelah semua data uji diprediksi, sistem kemudian menghitung akurasi model sebagai ukuran performa. Nilai akurasi diperoleh dengan membandingkan hasil prediksi model terhadap kelas sebenarnya yang dimiliki oleh data uji.

Langkah-langkah pengujian data pada metode LVQ 3 adalah sebagai berikut [8]:

- Inisialisasi prototipe (W_j) dari hasil proses pelatihan sebelumnya dan vektor input (X_i), yakni data pengujian.
- Hitung jarak antara W_j dan X_i menggunakan rumus *euclidian distance*.
- Bandingkan hasil prediksi kelas dengan kelas sebenarnya.

3.5. Usecase Diagram



Gambar 1. Usecase diagram

Sistem ini dirancang untuk mendukung proses pelatihan dan pengujian terhadap algoritma LVQ 3, serta untuk melakukan klasifikasi data secara otomatis dan efisien. Oleh karena itu, sistem ini harus dilengkapi dengan fitur-fitur utama yang mendukung fleksibilitas dan kemudahan dalam pelaksanaan proses tersebut. Salah satu fitur yang wajib dimiliki adalah kemampuan untuk mengunggah (*upload*) dataset ke dalam sistem. Selain itu, sistem juga harus menyediakan antarmuka yang memudahkan pengguna dalam mengatur parameter-parameter penting yang dibutuhkan dalam proses pelatihan dan pengujian algoritma, seperti nilai *learning rate* (*alfa*), parameter *window*, jumlah epoch, serta jenis validasi yang digunakan.

Terdapat dua aktor pada sistem ini, yaitu super admin dan admin. Super Admin memiliki hak akses penuh terhadap sistem, sedangkan admin hanya dapat melihat hasil akurasi model dan melakukan uji satuan.

4. HASIL DAN PEMBAHASAN

4.1. Halaman Login

Gambar 2. Halaman login

Sebelum masuk kedalam sistem, pengguna perlu login terlebih dahulu. Pengguna perlu memasukkan username dan password sesuai dengan akun yang telah teregistrasi.

4.2. Halaman Beranda

Gambar 3. Halaman beranda

Halaman Beranda yang berisikan kalimat pengenalan sistem.

4.3. Halaman Pengaturan User

Halaman ini berfungsi untuk manajemen user dan hanya bisa diakses oleh user yang berstatus super admin. Pada halaman ini, user dapat ditambah, diedit, dan dihapus.

Gambar 4. Halaman pengaturan user

4.4. Halaman Data

ID	Usia	Pendidikan Kepala Keluarga	Kepemilikan Rumah	Daya Listrik Terpasang	Sumber Air Minum	Output
1	0.5	0.25	1	1	1	1
2	0.5	0.75	0.33	1	1	1
3	0.5	0.75	0.66	1	1	1
4	1	0.75	1	1	1	1
5	1	0.5	0.66	1	1	1
6	1	0.75	1	1	1	1
7	1	0.25	1	0.5	1	1
8	1	0.5	0.66	1	1	1
9	0.5	0.25	0.66	1	1	2
10	0.5	0.5	1	0.5	1	1

Gambar 5. Halaman data

Halaman ini berisikan dataset yang akan digunakan oleh algoritma. Halaman menampilkan dataset, selain itu juga terdapat fitur pencarian, pagination/filter, dan penghapusan dataset.

Gambar 6. Halaman data - upload

Halaman upload akan tampil apabila tidak ada dataset yang terupload. File akan dibaca dan dimasukkan ke array data. Format file yang diperbolehkan adalah Excel dan CSV. Menangani proses upload file dengan validasi yang ketat dan memenuhi kebutuhan sistem, seperti format file (Excel dan CSV), setelah diupload file tersebut akan dipindahkan ke direktori 'file_upload/'.

4.5. Halaman Pengujian Algoritma

Gambar 7. Halaman pengujian algoritma

Halaman Form untuk menginputkan parameter untuk keperluan pengujian algoritma, yaitu dengan tujuan untuk menguji apakah penerapan algoritma telah sesuai dengan menampilkan proses secara detail.

LVQ 3.0 - KLASIFIKASI BLT						
Pengujian Algoritma LVQ 3.0						
No.	Uraian	Pendidikan Kepala Keluarga	Kepemilikan Rumah	Daya Listrik Terpasang	Sumber Air Minum	Output
1	0.5	0.75	0.33	1	1	1
2	0.5	0.75	0.66	1	1	1
3	1	0.75	1	1	1	1
4	1	0.5	0.66	1	1	1
5	1	0.75	1	1	1	1
6	1	0.25	1	0.5	1	1
7	1	0.5	0.66	1	1	1
8	0.5	0.5	1	0.5	1	1
9	1	1	1	0.5	1	1
10	1	0.75	1	1	1	1

Gambar 8. Halaman pengujian algoritma - hasil

Halaman ini merupakan ekstensi dari halaman 'Pengujian Algoritma' yang menampilkan detail seperti, dataset latih yang digunakan, prototipe awal yang dipilih, proses pelatihan, proses pengujian, dan vektor prototipe pada setiap *epoch*. Halaman ini juga menampilkan grafik tren vektor, yaitu gambaran perubahan *epoch* untuk masing-masing *epoch*.

4.6. Halaman LVQ

Gambar 9. Halaman lvq

Halaman Form untuk menginputkan parameter untuk keperluan pengujian akurasi. Halaman ini berbeda dari pada proses yang diinisialisasi dari halaman 'Pengujian Algoritma' halaman ini juga mengimplementasikan metode *K-fold Cross Validation* dan merupakan halaman yang menjadi dasar untuk pengujian pada penelitian ini.

4.7. Halaman Hasil Akurasi

LVQ 3.0 - KLASIFIKASI BLT						
Hasil Akurasi - LVQ 3.0						
No.	Jumlah Data	K-Fold	Alpha	Window	Epoch	Tanggal
1	100	10	0.1	0.5	500	2024-12-30
2	100	10	0.1	0.4	500	2024-12-30
3	100	10	0.1	0.3	500	2024-12-30
4	100	10	0.1	0.2	500	2024-12-30
5	100	10	0.075	0.5	500	2024-12-30
6	100	10	0.075	0.4	500	2024-12-30
7	100	10	0.075	0.3	500	2024-12-30
8	100	10	0.075	0.2	500	2024-12-30
9	100	10	0.05	0.5	500	2024-12-30

Gambar 10. Halaman hasil akurasi

Halaman ini menampilkan hasil dari pengujian akurasi dan terhubung dengan halaman 'LVQ'. Data dari hasil pengujian akan disimpan di database sebagai entri dan vektor yang dihasilkan dari pengujian dapat diuji coba untuk mengklasifikasikan sebuah data dengan menekan tombol 'Uji LVQ' pada masing-masing entri.

4.8. Halaman Uji Satuan

Gambar 11. Halaman uji satuan

Halaman ini menampilkan form untuk menguji vektor yang dihasilkan dari sebuah entri yang dihasilkan dari halaman 'LVQ' dengan menekan tombol 'Uji LVQ'.

Gambar 12. Halaman uji satuan - notifikasi

4.9. Pengujian Kebenaran Implementasi Algoritma LVQ 3

Pengujian ini dilakukan sebagai bagian dari proses verifikasi untuk memastikan bahwa algoritma LVQ 3 telah diimplementasikan dengan benar dan sesuai pada sistem. Tujuan utama dari pengujian ini adalah untuk memvalidasi kebenaran implementasi logika algoritma dalam sistem dengan membandingkan hasil vektor kelas yang diperoleh secara manual dengan hasil yang dihasilkan oleh sistem dan memastikan tidak adanya perbedaan.

Berikut merupakan langkah dari pengujian yang dilakukan:

- Lakukan perhitungan klasifikasi secara manual menggunakan nilai $\alpha = 0,1$ dan nilai $window = 0,2$.
- Lakukan perhitungan klasifikasi pada sistem di halaman 'Pengujian Algoritma'. Pilih nilai $\alpha = 0,1$ dan nilai $window = 0,2$ pada form halaman.
- Bandingkan vektor dari ketiga *epoch* pertama untuk masing-masing hasil perhitungan sistem dengan manual.
- Pastikan tidak ada perbedaan.

Tabel 1. Vektor hasil sistem

"Sistem Klasifikasi BLT Desa"						
W1	Epoch-1	0,9077	0,6246	0,4264	0,8766	0,9159
	Epoch-2	0,8998	0,6190	0,4215	0,8871	0,9162
	Epoch-3	0,8991	0,6179	0,4205	0,8874	0,9164
W2	Epoch-1	0,5691	0,2793	0,6558	0,7442	0,8622
	Epoch-2	0,5729	0,3161	0,6530	0,7171	0,8463
	Epoch-3	0,5728	0,3191	0,6527	0,7149	0,8449

Tabel 2. Vektor hasil manual

Perhitungan Manual						
W1	Epoch-1	0,9077	0,6246	0,4264	0,8766	0,9159
	Epoch-2	0,8998	0,6190	0,4215	0,8871	0,9162
	Epoch-3	0,8991	0,6179	0,4205	0,8874	0,9164
W2	Epoch-1	0,5691	0,2793	0,6558	0,7442	0,8622
	Epoch-2	0,5729	0,3161	0,6530	0,7171	0,8463
	Epoch-3	0,5728	0,3191	0,6527	0,7149	0,8449

Karena tidak adanya perbedaan dari kedua hasil, maka Algoritma LVQ 3 telah diimplementasikan dengan benar pada sistem.

4.10. Pengujian Akurasi Model Klasifikasi Algoritma LVQ 3

Pengujian dalam penelitian ini menggunakan sistem Klasifikasi BLT yang telah dibangun. Pengujian dalam penelitian ini dilakukan dengan membagi dataset menjadi dua bagian utama, yaitu data latih (*training data*) dan data uji (*testing data*). Pada tahap awal, proses pelatihan dilakukan terhadap prototipe kelas menggunakan data latih untuk membentuk model klasifikasi. Setelah proses pelatihan selesai, model yang telah terbentuk kemudian diuji menggunakan data uji guna mengevaluasi performanya dalam mengklasifikasikan data baru. Evaluasi ini bertujuan untuk mengukur tingkat akurasi model dalam mengenali pola dan membuat prediksi yang tepat.

Pengujian dilakukan dengan menerapkan berbagai kombinasi variasi parameter untuk mengetahui pengaruh masing-masing konfigurasi terhadap performa klasifikasi. Pengujian yang divariasikan meliputi nilai *learning rate* atau *alfa* (dengan nilai 0,01; 0,025; 0,05; 0,075; dan 0,1), parameter *window* (dengan nilai 0,2; 0,3; 0,4; dan 0,5), dan jenis *fold* (5-fold dan 10-fold). Setiap kombinasi parameter diuji secara menyeluruh dan hasil pengujian dari masing-masing kombinasi tersebut dihitung nilai rata-ratanya untuk memperoleh gambaran umum mengenai kinerja model.

Sebagai upaya untuk meningkatkan reliabilitas hasil pengujian, metode *K-fold Cross Validation* diterapkan. Pendekatan ini bertujuan untuk menghindari *overfitting* dan memastikan bahwa model yang dikembangkan memiliki generalisasi yang baik terhadap data yang belum pernah dilihat sebelumnya.

		AKURASI														Total	
		5-fold validation						10-fold validation									
		Alfa						Alfa									
Window	0.2	0.01	0.025	0.05	0.075	0.1	0.01	0.025	0.05	0.075	0.1	0.01	0.025	0.05	0.075	0.1	83%
	0.3	77.77%	85.55%	85.55%	83.33%	83.33%	83.33%	86.66%	82.22%	83.33%	85.55%	83.33%	83.33%	83.33%	84.5%	84.5%	
	0.4	82.22%	90%	86.66%	88.88%	90%	78.88%	90%	86.66%	90%	91.11%	80.55%	82.22%	83.33%	84.5%	90.1%	
	0.5	82.22%	91.11%	88.88%	91.11%	92.22%	81.11%	84.44%	88.88%	91.11%	91.11%	80.55%	82.22%	83.33%	84.5%	90.1%	
	Total	80.55%	89.16%	86.1%	87.22%	87.55%	81.66%	86.66%	85.27%	86.94%	87.77%	80.55%	89.16%	86.1%	87.22%	87.55%	
Rata-rata		85.88%															

Gambar 13. Tabel akurasi

Dengan rata-rata akurasi keseluruhan sebesar 85,09%, maka performa akurasi sistem klasifikasi masuk dalam penilaian Baik.

5. KESIMPULAN DAN SARAN

Penelitian ini menunjukkan bahwa algoritma *Learning Vector Quantization 3* (LVQ 3) dapat diimplementasikan secara efektif dalam proses klasifikasi calon penerima Bantuan Langsung Tunai (BLT) di tingkat desa, khususnya pada kasus di Desa Kelapapati. Penggunaan metode *K-fold Cross Validation* sebagai teknik validasi untuk mengukur performa model secara menyeluruh memungkinkan evaluasi yang lebih akurat terhadap kinerja algoritma dengan membagi dataset menjadi beberapa bagian untuk pelatihan dan pengujian secara bergantian. Penerapan sistem klasifikasi ini memberikan solusi alternatif yang berbasis teknologi dalam distribusi bantuan sosial. Dengan demikian, sistem ini tidak hanya berpotensi meningkatkan efisiensi dan transparansi dalam penyaluran bantuan, tetapi juga mendukung upaya pemerintah dalam mewujudkan keadilan sosial dan pemerataan kesejahteraan bagi masyarakat yang membutuhkan.

Untuk pengembangan penelitian selanjutnya, disarankan untuk menambah jumlah sampel dataset guna memperluas pola data yang dapat dipelajari model, menambahkan variabel atau atribut data untuk meningkatkan variasi dataset, mengimplementasikan variasi lain dari algoritma *Learning Vector Quantization* (LVQ), menggabungkan algoritma LVQ dengan metode lain, serta melakukan uji coba lapangan guna menguji kemampuan generalisasi model dalam kondisi nyata.

DAFTAR PUSTAKA

- [1] S. Rizayani, S. Syaharuddin, M. R. N. Handy, E. W. Abbas, and J. Jumriani, "Kebijakan Pengentasan Kemiskinan melalui Program Keluarga Harapan di Kota Banjarbaru," *PAKIS (Publikasi Berk. Pendidik. Ilmu Sos.*, vol. 2, no. 1, pp. 74–82, 2022, doi: 10.20527/pakis.v2i1.5209.
- [2] Menteri Sosial Republik Indonesia, "Peraturan Menteri Sosial Republik Indonesia No. 3 Tahun 2021 tentang Pengelolaan Data Terpadu Kesejahteraan Sosial," pp. 1–22, 2021.
- [3] E. A. Sianipar *et al.*, "Optimasi Klasifikasi Penyakit Diabetes Dengan Algoritma Learning Vector Quantization (LVQ)," *J. Comput. Sci. Informatics Eng.*, vol. 04, no. 2, pp. 72–84, 2025.
- [4] K. A. Muzaqi, A. Junaidi, and W. A. Saputra, "Klasifikasi Status Gizi Pada Lansia Menggunakan Learning Vector Quantization 3 (LVQ 3)," *J. Dinda Data Sci. Inf. Technol. Data Anal.*, vol. 2, no. 1, pp. 28–36, 2022, doi: 10.20895/dinda.v2i1.272.
- [5] M. D. Alfani and I. Z. Yadi, "Learning Vector Quantization 3 (LVQ3) Usage To Determine Recipients of the Family Hope Program (Case Study: Tanjung Lubuk District)," *J. Inf. Syst. Informatics*, vol. 4, no. 4, pp. 1030–1041, 2022, doi: 10.51519/journalisi.v4i4.374.
- [6] Peraturan Menteri Sosial Republik Indonesia

- Nomor 4 Tahun 2023 Tentang Pelaksanaan Program Sembako, “Pelaksanaan Program Sembako,” no. 572, pp. 1–13, 2023.
- [7] V. Avenazh, S. Hawari, and I. Purwanto, “Membandingkan Metode Jaringan Syaraf Tiruan backpropagation dan Learning Vector Quantization,” vol. 1, no. 1, pp. 107–114, 2022.
- [8] Darmila, “Evaluasi Perbandingan Performansi Lvq 1, Lvq 2, Dan Lvq 3 Dalam Klasifikasi Jenis Kelamin Menggunakan Tulang Tengkorak Darmila 1*, Iis Afrianty 2, Suwanto Sanjaya 3, Rahmad Abdillah 4, Iwan Iskandar 5, Fadhilah Syafria 6,” vol. 7, pp. 344–353, 2022.
- [9] A. Ultsch and J. Lötsch, “Euclidean distance-optimized data transformation for cluster analysis in biomedical data (EDOtrans),” *BMC Bioinformatics*, vol. 23, no. 1, pp. 1–18, 2022, doi: 10.1186/s12859-022-04769-w.
- [10] X. Zhang and C. A. Liu, “Model averaging prediction by K-fold cross-validation,” *J. Econom.*, vol. 235, no. 1, pp. 280–301, 2023, doi: 10.1016/j.jeconom.2022.04.007.
- [11] M. Shiddiq, F. Candra, B. Anand, and M. F. Rabin, “Neural network with k-fold cross validation for oil palm fruit ripeness prediction,” *Telkomnika (Telecommunication Comput. Electron. Control.)*, vol. 22, no. 1, pp. 164–174, 2024, doi: 10.12928/TELKOMNIKA.v22i1.24845.
- [12] I. K. Nti, “Performance of Machine Learning Algorithms with Different K Values in K-fold Cross- Validation,” no. December, 2021, doi: 10.5815/ijitcs.2021.06.05.