Information System Journal (INFOS)
Vol. 8, No. 1, Mei 2025, pp. 40-50
E-ISSN : 2655-142X , P-ISSN : 2655-190X, DOI:https://doi.org/10.24076/infosjournal.2025v8i01.2118

IMPLEMENTASI LEARNING VECTOR QUANTIZATION 2 DAN INFORMATION GAIN UNTUK KLASIFIKASI PENYAKIT GINJAL KRONIS

Fayat Zabihullah[1)], Elvia Budianita[2)], Fadhilah Syafria[3)], Iis Afrianty[4)]
[1) 2) 3) 4)] Program Studi Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim Riau

email: 12150112066@students.uin-suska.ac.id[1)], elvia.budianita@uin-suska.ac.id[2)], fadhilah.syafria@uin-suska.ac.id[3)], iis.afrianty@uin-suska.ac.id[4)]

INFO ARTIKEL

Riwayat Artikel:

Diterima Mei, 2025
Revisi Mei, 2025
Terbit Mei, 2025

ABSTRAK

Penyakit ginjal kronis terjadi ketika ginjal gagal mempertahankan metabolisme dan keseimbangan tubuh, serta memiliki risiko kematian yang tinggi. Analisis dan prediksi menggunakan teknik klasifikasi data dapat membantu mengurangi risiko tersebut. Penelitian ini bertujuan untuk mengklasifikasikan penyakit ginjal kronis dengan menggabungkan metode seleksi fitur *Information Gain* dan algoritma *Learning Vector Quantization 2* (*LVQ* 2). *Dataset* yang digunakan terdiri dari 1659 data dengan 53 atribut dan 1 label kelas. Tahapan penelitian meliputi *preprocessing*, seleksi fitur, normalisasi, dan klasifikasi. Seleksi fitur dilakukan berdasarkan nilai *Information Gain* dengan *threshold* tertentu. Model diuji dengan kombinasi parameter *learning rate* dan *window*, serta dievaluasi menggunakan akurasi, presisi, *recall*, dan *F1-score*. Hasil terbaik diperoleh tanpa seleksi fitur dengan akurasi 93.98%. Setelah seleksi fitur, akurasi menurun sedikit menjadi 93.37%. Kombinasi *SMOTE* dan seleksi fitur meningkatkan presisi, *recall*, dan *F1-score*, namun menurunkan akurasi hingga menjadi 80.00% pada *threshold* 0.7 dengan fitur terpilih 33.

Kata Kunci :
Information Gain; *Learning Vector Quantization* 2; Ginjal Kronis; Seleksi Fitur

ABSTRACT

Chronic kidney disease (CKD) occurs when the kidneys fail to maintain metabolic balance, posing a serious health risk. This study aims to classify CKD using data classification techniques by combining Information Gain for feature selection and the Learning Vector Quantization 2 (LVQ2) algorithm. The dataset includes 1659 records with 53 attributes and one class label. The research process involves data preprocessing, feature selection, normalization, and classification. Features are selected based on their Information Gain scores with a specified threshold. The model is tested with various learning rate and window size combinations, and evaluated using accuracy, precision, recall, and F1-score. The highest accuracy of 93.98% is achieved without feature selection. After applying feature selection, accuracy slightly drops to 93.37%. However, when SMOTE is combined with feature selection, precision, recall, and F1-score improve, though accuracy decreases to 80.00% at a threshold of 0.7 with 33 selected features.

***Keywords*:**
Information Gain; *Learning Vector Quantization 2*; *Chronic Kidney Disease*; *Feature Selection*

Penulis Korespondensi:

Elvia Budianita
Program Studi Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim Riau

Email:
elvia.budianita@uin-suska.ac.id

1. PENDAHULUAN

Penyakit ginjal kronis merupakan kerusakan pada ginjal yang bersifat permanen, sehingga tubuh gagal mempertahankan metabolisme dan keseimbangan. Peluang pemulihan penyakit ini pun rendah, sementara biaya pengobatannya sangat tinggi [1]. Faktor gangguan psikologis hingga jenis terapi dialis yang digunakan, sosiodemorafi, dan status klinis menjadi faktor yang dapat mempengaruhi kualitas hidup pasien penyakit ginjal kronis [2]. Penyakit ginjal kronis mempunyai risiko kematian dan biaya perawatan yang tinggi. Berdasarkan data dari *World Health Organization* (*WHO*) sebanyak 697.5 juta pasien gagal ginjal kronis pada tahun 2017 dan sebanyak 1.2 juta meninggal dunia. Lalu menurut Data Riset Kesehatan Dasar Kementerian Kesehatan Republik Indonesia (Kemenkes RI) pada 2018, sebanyak 739.208 orang atau sekitar 3.8% masyarakat di Indonesia mengalami penyakit ginjal kronis. Prevalensi ini meningkat dari data Riskesdas pada 2013 yang hanya 2 persen.

Penyakit ginjal kronis biasanya diidentifikasi melalui skrining rutin, seperti profil kimia serum dan analisis urin, atau ditemukan secara insidental. Pasien yang diketahui atau dicurigai biasanya menunjukkan gejala seperti *hematuria* besar, *urine* berbusa (*albuminuria*), *nokturia*, nyeri pinggang, atau penurunan produksi urin. Beberapa gejala tambahan yang dapat menunjukkan penyebab sistemik adalah *hemoptisis*, ruam, *limfadenopati*, gangguan pendengaran, atau neuropati [3].

Terdapat faktor-faktor yang mempengaruhi terjadinya penyakit ginjal kronis diantaranya umur, *diabetes mellitus*, riwayat keluarga dengan gagal ginjal kronis, riwayat hipertensi, riwayat merokok dan kebiasaan mengkonsumsi alkohol. Memahami faktor-faktor penyebab penyakit ginjal kronis dapat membantu dalam meningkatkan pengobatan sejak dini dan pencegahan untuk mengurangi kasus penyakit ginjal kronis dimasyarakat [4]. Penderita penyakit ginjal kronis didominasi oleh masyarakat berusia 65 hingga 74 tahun berdasarkan rentang usia, maka diperlukan upaya pencegahan, salah satunya dengan melakukan penelitian terkait penyakit ginjal kronis untuk memecahkan masalah menggunakan pendekatan *data mining*. Menganalisa dan memprediksi faktor-faktor yang mempengaruhi penyakit ginjal kronis dapat membantu mengurangi risiko terkena penyakit ginjal kronis [5].

Data Mining merupakan suatu proses pencarian pola terhadap *dataset* sehingga menghasilkan tingkat akurasi yang tinggi [6]. Pada penelitian *data mining* penyakit ginjal dengan algoritma *KNN* menggunakan *Backward Elimination* menghasilkan nilai akurasi sebesar 99.25%, sensitifitas sebesar 99.5%, dan spesifitas sebesar 98.745% [7]. Pada penelitian lainnya dalam mendiagnosis penyakit ginjal kronis menggunakan algoritma *C4.5* dengan data yang terdiri dari 24 atribut yaitu umur, tekanan darah, gravitas, *albumin*, *sugar*, sel darah merah, *pussel*, *puscell*, bakteri, *gds*, *ureum*, *kretinin*, *natrium*, *kalium*, *hemoglobin*, *mvc*, sel darah putih, jumlah sel darah merah, hipertensi, diabetes, *cad*, nafsu makan, *edema*, *anemia* dan 1 label kelas klasifikasi memperoleh hasil pengujian tingkat akurasi dari *confusion matrix* sebesar sebesar 96.67% dan didapatkan *error classification* sebesar 3.33% [8].

Salah satu cabang ilmu komputer yang juga dapat membantu dalam memecahkan masalah dalam mendeteksi penyakit ginjal kronis adalah adanya jaringan syaraf tiruan yang meniru fungsi otak manusia dengan melakukan generalisasi model matematis pada sebuah kasus, salah satu metodenya yaitu klasifikasi menggunakan metode *Learning Vector Quantization (LVQ)*. Pada penelitian *LVQ* untuk pengenalan *barcode* barang mendapatkan hasil akurasi 90% [9]. Variasi *LVQ*, seperti *LVQ* 1, *LVQ* 2, dan *LVQ* 3, muncul sebagai hasil dari kemajuan dalam teknik *LVQ*.[10]. Menggunakan metode *LVQ* 2 dalam mengklasifikasi ukuran pakaian dengan total data yang digunakan berjumlah 50 data, dibagi 35 data untuk pelatihan dan 15 data untuk pengujian. Sistem dapat mengenali semua *training data*, untuk *testing data* dengan parameter *learning rate* sebesar 0.1, *window* sebesar 0.8, *minimum learning rate* sebesar 0.001 memperoleh akurasi sebesar 93.33% [11]. Pada penelitian lain dalam mendeteksi penyakit *tuberkulosis* paru menggunakan metode *LVQ* 2 menghasilkan akurasi 87.5% [12].

Beberapa penelitian klasifikasi juga menerapkan *Feature Selection* untuk mengoptimalkan kinerja dari pengklasifikasian. Pada penelitian ini menggunakan data yang mempunyai 52 atribut dan 1 kelas, karena banyaknya atribut maka dilakukan seleksi untuk mengetahui atribut berpengaruh pada terjadinya risiko penyakit ginjal kronis, maka penelitian ini perlu menggunakan *Information Gain Feature Selection*. Salah satu penelitian membuktikan Seleksi Fitur *Information Gain* membantu dalam memprediksi penyakit diabetes dengan metode *K-Nearest Neighbor* yang memiliki data dengan 17 atribut, hasil menunjukkan akurasi tanpa *Information Gain* 69.11%, sementara dengan *Information Gain* mencapai akurasi tertinggi sebesar 72.93%

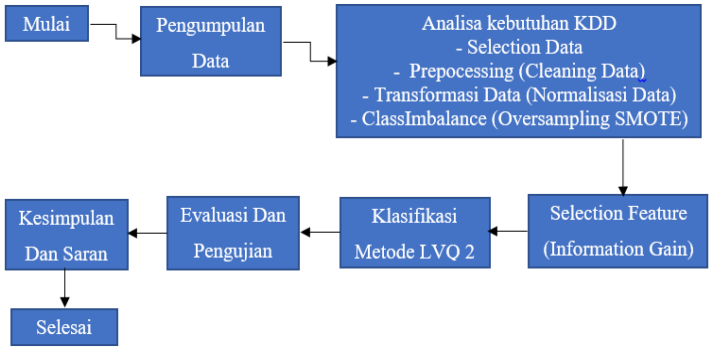
Information System Journal (INFOS)
Vol. 8, No. 1, Mei 2025, pp. 40-50
E-ISSN : 2655-142X , P-ISSN : 2655-190X, DOI:https://doi.org/10.24076/infosjournal.2025v8i01.2118

[13]. Hasil penelitian menggunakan *feature selection information gain* pada analisis sentimen maskapai penerbangan mendapatkan akurasi 86.5%, sedangkan tanpa *information gain* hanya mendapatkan akurasi 81% [14]. Lalu pada penelitian lain menggunakan seleksi fitur *Information Gain* pada metode *LVQ* mendapatkan akurasi sebesar 92.01%, hasil akurasi ini lebih tinggi dari yang tidak menggunakan seleksi fitur yang mendapatkan akurasi sebesar 91.39% [15].

Berdasarkan uraian tersebut, maka penelitian ini menerapkan seleksi fitur *Information Gain* dan *Learning Vector Quantization* 2 (*LVQ* 2) dalam klasifikasi penyakit ginjal kronis. Dengan menerapkan seleksi fitur *Information Gain* sebelum proses klasifikasi menggunakan *LVQ* 2, diharapkan agar bisa membantu mendapatkan hasil klasifikasi yang akurat serta mempercepat proses identifikasi penyakit ginjal kronis. Penelitian ini diharapkan dapat membantu dibidang Jaringan syaraf tiruan untuk meningkatkan akurasi dengan teknik *Learning Vector Quantization* 2 (*LVQ2*) ataupun teknik yang berbeda.

2. METODOLOGI PENELITIAN

Penelitian ini terdiri dari beberapa proses meliputi pengumpulan data hingga evaluasi dan pengujian, alur penilitian yang digunakan direpresentasikan pada Gambar 1.



Gambar 1. Metodologi Penelitian.

2.1 Pengumpulan Data

Penulis menggunakan metode pengumpulan data sekunder untuk mengumpulkan data yang digunakan dalam pelaksanaan penelitian ini. Sumber data sekunder terdiri dari data yang diperoleh melalui media seperti buku, jurnal, dan berbagai penelitian sebelumnya. Sedangkan untuk data penelitian akan diambil melalui situs *Kaggle dataset* https://www.kaggle.com/datasets/rabieelkharoua/chronic-kidney-disease-dataset-analysis/.

2.2 *Selection* Data

Tujuan dari pemilihan atribut untuk *dataset* penyakit ginjal kronis saat ini adalah untuk mengurangi *noise* dan berfokus pada pola-pola penting untuk meningkatkan kapasitas model yang terkait dengan penyakit ginjal kronis.

2.3 *Preprocessing Data*

Tahapan *prepocessing data* dilakukan untuk membersihkan data (*data cleaning*) agar sesuai dengan kebutuhan analisis. Pada penelitian ini data yang digunakan memiliki 53 atribut dan 1659 data, dilakukan pengecekan data yang hilang (data duplikat) dan data yang kosong (*missing value)*. Hal ini bertujuan untuk menigkatkan peforma dan kinerja model LVQ 2.

2.4 Transformasi Data

Transformasi data merupakan pengubahan atau penggabungan data kedalam format tertentu. Atribut data yang digunakan pada penelitian ini sudah berbentuk numerik, namun ada 36 atribut yang akan dilakukan penskalaan karena terdapat nilai pada atribut yang tidak memiliki rentang 0 hingga 1 dengan menggunakan

rumus normalisasi. Normalisasi data adalah mengukur rentang data menggunakan metode *min-max*, tujuannya adalah untuk meningkatkan nilai *dataset* sehingga memiliki distribusi atau skala yang konsisten.

$$Xnorm = \frac{X - Xmin}{Xmax - Xmin} \quad (1)$$

Keterangan:
$Xnorm$ = nilai yang telah dinormalisasi
X = nilai asli
$Xmax$ = nilai max
$Xmin$ = nilai min

2.5 *SMOTE*

SMOTE adalah metode penyeimbangan distribusi data sampel pada kelas minoritas dengan memilih sampel hingga jumlah sampel seimbang dengan kelas mayoritas. *Overfitting* dapat terjadi ketika metode *SMOTE* digunakan karena data kelas minoritas diduplikasi sehingga adanya data latih yang sama. Proses *SMOTE* dimulai dengan menghitung jarak antara data minoritas, menemukan nilai presentasi *SMOTE*, dan kemudian menemukan jumlah *k* terdekat dan terakhir.

2.6 *Information Gain Feature Selection*

Penelitian ini menggunakan metode *Information Gain* untuk membantu dalam memilih fitur agar meningkatkan akurasi dan efisiensi model dengan mengurangi dimensi data dan berkonsentrasi pada fitur yang paling berdampak pada hasil klasifikasi.

Information Gain merupakan teknik pemilihan fitur yang dapat mengukur seberapa banyak informasi dalam keputusan klasifikasi yang benar dalam kategori apapun yang memengaruhi ada atau tidaknya [14]. Penentuan fitur paling relevan dimulai dari mencari nilai *entropy*, karena dapat menjadi petunjuk untuk menentukan fitur-fitur yang nilainya mencukupi untuk digunakan dalam proses klasifikasi [16]. Dalam menghitung nilai *Information Gain* dibutuhkan beberapa tahapan sebagai berikut:

1. Menghitung nilai *entropy*, *entropy* adalah ukuran ketidakpastian kelas yang memanfaatkan kemungkinan peristiwa atau atribut tertentu.

$$Entropy\ (S) = \sum_i^n = 1 - Pi\ log2\ Pi \quad (2)$$

2. Melakukan perhitungan *Information Gain* menggunakan rumus.

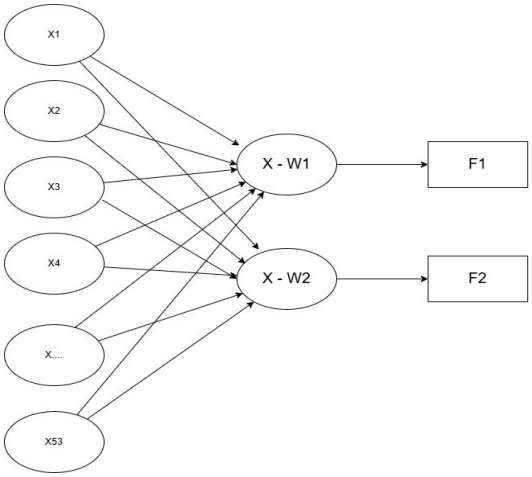
$$Gain\ (S.A) = Entropy(S) - \sum value\ (A) \frac{[Sv]}{[S]} Entropy(Sv) \quad (3)$$

Keterangan:
$Gain\ (S.A)$ = *information gain* atribut A
$Entropy(S)$ = total *entropy* semua kriteria
$Entropy(Sv)$ = *entropy* untuk masing-masing kriteria nilai v
S = himpunan kasus
A = atribut

2.7 *Learning Vector Quantization* 2

Jaringan *Learning Vector Quantization* bertujuan untuk melakukan pembelajaran pada lapisan kompetitif yang diawasi dan memiliki arsitektur jaringan berlayer tunggal. Lapisan kompetitif ini hanya menghasilkan kelas berdasarkan jarak antara vektor *input*. Jika dua vektor *input* berada di dekat satu sama lain, lapisan kompetitif akan memasukkan kedua vektor *input* tersebut ke dalam kelas yang sama. Berikut arsitektur jaringan *Learning Vector Quantization* pada Gambar 2.

Information System Journal (INFOS)
Vol. 8, No. 1, Mei 2025, pp. 40-50
E-ISSN : 2655-142X , P-ISSN : 2655-190X, DOI:https://doi.org/10.24076/infosjournal.2025v8i01.2118



Gambar 2. Arsitektur Jaringan *Learning Vector Quantization*.

Gambar 2., menunjukkan arsitektur sederhana dari jaringan *LVQ*. Setiap *input* (*x1, x2, ..., x53*) dihubungkan ke *node* kompetitif (*X-W1* dan *X-W2*), yang mewakili bobot (*W1* dan *W2*) dari dua prototipe (kelas). *Node* kompetitif ini menghitung jarak antara *input* (*X*) dan bobot (*W*) masing-masing. Hasil dari perhitungan ini menentukan *output* ke kelas *F1* dan *F2*. Jarak terdekat antara *X* dan *W* akan menentukan kelas mana (*F1* atau *F2*) yang dipilih sebagai hasil klasifikasi.

Learning Vector Quantization 2 memperbarui vektor pemenang dan *runner-up* jika persyaratan terpenuhi. Sementara itu, *Learning Vector Quantization 1* hanya memperbarui vektor referensi yang dekat dengan *input*. Tahapan *Learning Vector Quantization 2* sebagai berikut:

1. Vektor pelatihan (*Xi*), target (*T*), dan parameter bobot awal (*Wj*) diberikan dari data pelatihan yang telah dipetakan sesuai dengan kenyataan. Selain itu, *learning rate* (α), *window* (ε) pengurangan (α), dan nilai *minimum* (α) ditetapkan.
2. Perhatikan nilai (α), jika nilai (α) lebih besar dari nilai minimal (min α), maka proses dan tahapan berikutnya akan dilanjutkan. Sebaliknya, jika nilai (α) lebih rendah dari nilai min (α), Selanjutnya, nilai bobot akhir akan diterima.
3. Membaca *input*.
4. Setelah membaca *input* data, lakukan perhitungan dengan menghitung jarak antara bobot *Xi* dan *Wj* dan kemudian menghitung jarak geometris. Untuk menghitungnya, digunakan persamaan *Euclidean Distance* sebagaimana (4).

$$C = \sqrt{(x1 - w1)^2 + \cdots + (xn - wn)^2} \tag{4}$$

5. Cari jarak terkecil antara *Xi* dan *Wj* (*dc1*), yang dicari dengan menggunakan indeks vektor bobot sebagai *Cj*.
6. Memperbaiki *Wj* dengan aturan berikut ini:
   - Jika T = Cj, maka berlaku persamaan (5), sedangkan Jika T ≠ Cj, maka persamaan (6) yang akan digunakan.

$$Wj = Wj + a\,(Xi = Wj) \tag{5}$$

$$D1 > (1 - \varepsilon) * D2\ AND\ D2 < (1 + \varepsilon)D1 \tag{6}$$

   - Jika *True*, maka W yang tidak termasuk vektor X akan di perbaharui menggunakan persamaan (7), Sedangkan W yang termasuk maka vektor X akan di perbaharui dengan persamaan (8).

$$YCj(t + 1) = YCj(t) - \alpha(t)[X(t) - YCj(t)] \tag{7}$$

$$YCj(t + 1) = YCj(t + -\alpha(t)[\,X(t) - YCj\,t)] \tag{8}$$

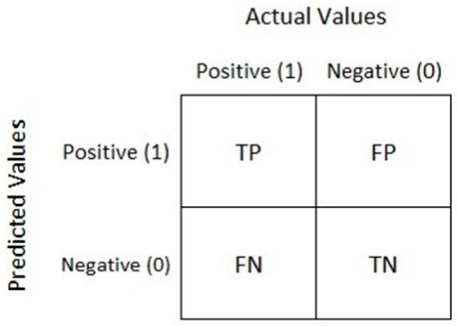
7. Setelah diperoleh *Wj* baru, apabila hasilnya *False*, maka persamaan (9) akan berlaku untuk kembali memperbarui nilai *Wj*.

$$Wj = Wj - a(Xi - Wj) \quad (9)$$

8. Lakukan perulangan α.
9. Proses dihentikan jika kondisi tertentu terpenuhi dan hasil pelatihan diberi nilai akhir. Jika tidak terpenuhi, maka proses perlu kembali ke langkah 2.

2.8 Evaluasi dan Pengujian Model

Evaluasi hasil pengujian penelitian ini menggunakan *Confusion Matrix*, yaitu suatu tabel yang menyajikan keseluruhan data yang dikategorikan dengan benar maupun salah. Penggunaan *confusion matrix* memudahkan dalam menilai tingkat ketepatan sebuah model klasifikasi. *Confusion matrix* terdiri dari empat komponen utama yaitu nilai *True Positive* (*TP*) menunjukkan keseluruhan jumlah data yang memiliki label positif dan berhasil ditetapkan sebagai positif, nilai *True negative* (*TN*) menunjukkan keseluruhan jumlah data yang memiliki label negatif dan ditetapkan dengan benar sebagai negatif, nilai *False Positive* (*FP*) merupakan keseluruhan jumlah data yang sebenarnya berlabel positif tetapi salah ditetapkan sebagai negatif, dan *False Negative* (*FN*) merupakan total data yang seharusnya label negatif namun salah ditetapkan sebagai positif. Representasi *confusion matrix* dapat dilihat pada Gambar 3.



Gambar 3. *Confusion Matrix*.

Dari *confusion matrix*, sejumlah metrik evaluasi dapat dihitung seperti berikut:

1. Akurasi adalah ukuran seberapa sering model membuat prediksi yang benar dari semua data yang diuji.

$$Akurasi = \frac{TN+TP}{TP+TN+FP+FN} \quad (10)$$

2. Presisi menunjukkan seberapa tepat model saat mengatakan suatu data termasuk dalam kelas tertentu.

$$Presisi = \frac{TP}{TP+FP} \quad (11)$$

3. *Recall* adalah ukuran seberapa lengkap model dalam menangkap semua data yang seharusnya dikategorikan positif.

$$Recall = \frac{TP}{TP+FN} \quad (12)$$

4. *F1 Score* adalah nilai gabungan antara presisi dan *recall*, terutama digunakan saat ingin menyeimbangkan antara keduanya.

$$F1\ Score = \frac{2\times(presisi\times recall)}{presisi\times recall} \quad (13)$$

Information System Journal (INFOS)
Vol. 8, No. 1, Mei 2025, pp. 40-50
E-ISSN : 2655-142X , P-ISSN : 2655-190X, DOI:https://doi.org/10.24076/infosjournal.2025v8i01.2118

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Penelitian ini menggunakan *dataset* Penyakit Ginjal Kronis yang diperoleh dari situs *kaggle* https://www.kaggle.com/datasets/rabieelkharoua/chronic-kidney-disease-datasetanalysis/data, *dataset* yang digunakan dalam penelitian ini terdiri dari 1659 data dengan 53 atribut dan satu label kelas, yaitu terkena penyakit ginjal kronis dan tidak terkena penyakit ginjal kronis. Seluruh tahapan penelitian dilakukan di *Google Collab* dengan Bahasa pemograman *Python*, mulai dari *preprocessing* data, seleksi fitur, pelatihan model dan pengujian model.

Tabel 1. Dataset Penyakit Ginjal Kronis.

Patient ID	Age	Gender	…	Diagnosis	Doctor InCharge
1	71	0	…	1	Confidential
2	34	0	…	1	Confidential
3	80	1	…	1	Confidential
4	40	0	…	1	Confidential
…	…	…	…	…	…
1659	34	1	…	1	Confidential

3.2 *Selection* Data

Pada langkah ini dilakukan penghapusan atribut *Patient ID* dan *DoctorInCharge*, atribut dihapus karena tidak relevan dalam membantu prediksi pengujian, maka atribut data yang berjumlah 53 menjadi 51 atribut, dengan target kelas yaitu *Diagnosis*. Hasil seleksi data setelah dilakukan seleksi data dapat dilihat pada Tabel 2.

Tabel 2. Hasil Seleksi Data.

Age	Gender	Ethnicity	…	Diagnosis
71	0	0	…	1
34	0	0	…	1
80	1	1	…	1
40	0	2	…	1
…	…	…	…	…
34	1	1	…	1

3.3 *Preprocessing Data*

Pada langkah ini dilakukan pengecekan data yang hilang (data duplikat) dan data yang kosong (*missing value*), hasil yang didapatkan bahwa tidak ada data yang hilang (data duplikat) dan maupun data yang kosong (*missing value*).

3.4 Transformasi Data

Pada langkah ini tidak dilakukan transformasi karena data sudah berbentuk numerik, namun pada 36 atribut terdapat nilai data yang tidak memiliki rentang 0 hingga 1, maka dilakukan normalisasi untuk memastikan seluruh data berada pada rentang yang sama. Hasil normalisasi data pada Tabel 3.

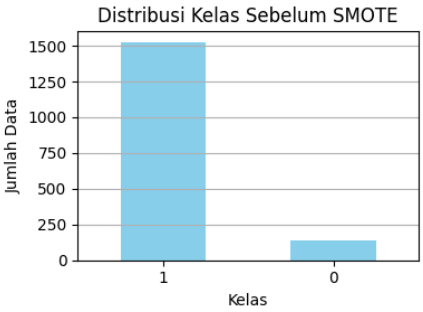
Tabel 3. Hasil Normalisasi Data.

Age	Ethnicity	SocioeconomicStatus	…	Health Literacy
0.728571	0.000000	0.0	…	0.987756
0.200000	0.000000	0.5	…	0.716498
0.857143	0.333333	0.0	…	0.735806
0.285714	0.666667	0.0	…	0.663224
…	…	…	…	…
0.200000	0.333333	0.0	…	0.455404

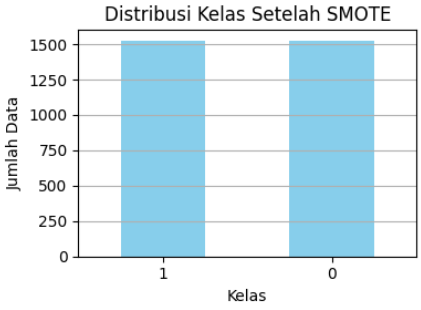
3.5 *SMOTE*

Penelitian ini memiliki kelas data tidak seimbang, kelas 1 berjumlah 1524 data dan kelas 0 berjumlah 135 data, dapat dilihat pada Gambar 4.

Data tidak seimbang dapat mempengaruhi hasil dari pengujian, maka perlu dilakukan penyeimbangan menggunakan *SMOTE* agar kelas 0 dan 1 seimbang, dimana mendapatkan hasil kelas 0 sebanyak 1524 data dan kelas 1 sebanyak 1524 data, hasil *SMOTE* dapat dilihat pada Gambar 5.



Gambar 4. Kelas Data Sebelum *SMOTE*.



Gambar 5. Kelas Data Setelah *SMOTE*.

3.6 *Information Gain Feature Selection*

Seleksi fitur menggunakan *Information Gain* dilakukan dengan menghitung nilai *entropy*, lalu dilanjutkan menghitung nilai *information gain*. Hasil yang didapatkan pada *information gain* diurutkan dari yang terbesar hingga yang terkecil. Maka ditampilkan hasil pada Tabel 4.

Tabel 4. Hasil Normalisasi Data.

No	Fitur	Information Gain
1	BMI	0.4070
2	AlcoholConsumption	0.4070
3	PhysicalActivity	0.4070
4	DietQuality	0.4070
…	…	…
51	ACEInhibitors	0.0000

Pemilihan fitur dilakukan menggunakan *information gain* berdasarkan nilai *threshold* yang akan menentukan seberapa kuat hubungan antara setiap fitur dengan target klasifikasi. Semakin tinggi nilai *threshold*, semakin ketat seleksi pada fitur, sehingga hanya fitur dengan nilai *information gain* yang lebih tinggi dari *threshold* yang akan digunakan.

3.7 Klasifikasi Metode *Learning Vector Quantization* 2

Pada tahap klasifikasi metode *Learning Vector Quantization* 2 dilakukan pelatihan dan pengujian. Data dibagi menggunakan *ratio* dimana data latih dan data uji 90:10; 80:20; dan 70:30. Data latih digunakan untuk melatih model agar memahami pola, lalu data uji digunakan untuk evaluasi performa.

3.8 Evaluasi dan Pengujian

Pengujian dilakukan dengan empat skenario yaitu menggunakan model *LVQ* 2, menggunakan teknik *SMOTE* dan model *LVQ* 2, menggunakan seleksi fitur *threshold* 0.3 dan model *LVQ* 2, menggunakan teknik *SMOTE* dengan seleksi fitur *threshold* 0.3, 0.7 dan model *LVQ* 2. Parameter *learning rate* yang digunakan dalam setiap pengujian adalah 0.1, 0.3, 0.6, dan 0.9. nilai *learning rate* yang digunakan sebesar 0.001, dan setiap *learning rate* akan diuji dengan parameter *window* sebesar 0.2, 0.3, dan 0.4. Adapun evaluasi peforma model akan dihitung berdasarkan beberapa metrik populer yaitu akurasi, presisi, *recall*, dan *F1 score*.

Pengujian menggunakan model *LVQ* 2, akurasi tertinggi didapatkan pada *leaning rate* 0.9, dengan parameter *window* sebesar 0.2, 0.3, 0.4, serta dengan *ratio* pembagian data 90:10. Hasil akurasi yang diperoleh adalah sebesar 93.98%. Pada pengujian yang menggunakan *ratio* 80:20 akurasi tertinggi diperoleh sebesar 92.77%, dan pada *ratio* 70:30 akurasi tertinggi yang diperoleh sebesar 90.96%. Hasil akurasi tertiggi pengujian menggunakan model *LVQ* 2 secara rinci dapat dilihat pada Tabel 5.

Information System Journal (INFOS)
Vol. 8, No. 1, Mei 2025, pp. 40-50
E-ISSN : 2655-142X , P-ISSN : 2655-190X, DOI:https://doi.org/10.24076/infosjournal.2025v8i01.2118

Tabel 5. Hasil Pengujian *LVQ* 2 Tanpa *Information Gain* dan Menggunakan *Ratio Data* 90:10.

Learning rate (α)	Window (ε)	Akurasi	Presisi	Recall	F1 score
0.1	0.2, 0.3, 0.4	80.66%	80.89%	80.61%	80.60%
0.3	0.2, 0.3, 0.4	79.02%	80.37%	78.91%	78.74%
0.6	0.2, 0.3	79.02%	80.37%	78.91%	78.74%
	0.4	74.75%	75.14%	74.69%	74.62%
0.9	0.2	49.51%	24.75%	50.00%	33.11%
	0.3, 0.4	60.33%	64.82%	60.59%	57.40%

Pengujian menggunakan teknik *SMOTE* dan model *LVQ* 2, akurasi tertinggi sebesar 80.66% didapatkan pada pengujian yang menerapkan *leaning rate* 0.1, *window* 0.2, 0.3, 0.4, dengan *ratio* pembagian data sebesar 90:10, sedangkan pada *ratio* 80:20 mendapatkan akurasi tertinggi sebesar 78.20%, dan pada *ratio* data 70:30 mendapatkan akurasi tertinggi 75.96%. Hasil akurasi tertiggi pengujian menggunakan teknik *SMOTE* dan model *LVQ* 2 pada Tabel 6.

Tabel 6. Hasil Pengujian *LVQ* 2 Menggunakan SMOTE dan Menggunakan dengan *Ratio Data* 90:10.

Learning rate (α)	Window (ε)	Akurasi	Presisi	Recall	F1 score
0.1	0.2, 0.3, 0.4	80.66%	80.89%	80.61%	80.60%
0.3	0.2, 0.3, 0.4	79.02%	80.37%	78.91%	78.74%
0.6	0.2, 0.3	79.02%	80.37%	78.91%	78.74%
	0.4	74.75%	75.14%	74.69%	74.62%
0.9	0.2	49.51%	24.75%	50.00%	33.11%
	0.3, 0.4	60.33%	64.82%	60.59%	57.40%

Pengujian menggunakan seleksi fitur *threshold* 0.3 dengan fitur terpilih 30 dan model *LVQ* 2 mendapatkan akurasi tertinggi pada *leaning rate* 0.9, *window* 0.2, dengan *ratio* data 90:10, yaitu sebesar 93.37%, untuk *ratio* pembagian data 80:20 akurasi tertinggi diperoleh sebesar 92.47% dan penggunaan *ratio* pembagian data 70:30 mendapatkan akurasi tertinggi 90.76%. Hasil akurasi tertiggi pengujian menggunakan seleksi fitur dan model *LVQ* 2 pada Tabel 7.

Tabel 7. Hasil Pengujian *LVQ* 2 Menggunakan *Information Gain* dengan *Threshold* 0.3 dan *Ratio Data* 90:10.

Learning rate (α)	Window (ε)	Akurasi	Presisi	Recall	F1 score
0.1	0.2, 0.3, 0.4	86.14%	63.20%	79.91%	66.59%
0.3	0.2, 0.3, 0.4	84.34%	59.31%	70.50%	61.37%
0.6	0.2, 0.3	81.33%	57.51%	68.89%	58.59%
	0.4	87.35%	54.18%	55.22%	54.58%
0.9	0.2	86.14%	57.60%	63.02%	59.08%
	0.3, 0.4	93.37%	71.95%	54.22%	55.97%
0.1	0.2, 0.3, 0.4	86.75%	50.58%	50.67%	50.60%

Pengujian menggunakan teknik *SMOTE* dengan seleksi fitur *threshold* 0.3, 0.7 dan model *LVQ* 2 mendapatkan akurasi tertinggi pada *threshold* 0.7. Fitur terpilih sebanyak 33, dengan *ratio* data 90:10, yaitu 80,00%, untuk *ratio* 80:20 akurasi tertinggi 74.59% dan pada *ratio* 70:30 akurasi tertinggi diperoleh sebesar 75.19%. Sedangkan pada penggunaan *threshold* 0.3, fitur terpilih sebanyak 36, dan menghasilkan nilai akurasi tertinggi sebesar 77.38% pada *ratio* 90:10, 72.79% menggunakan *ratio* 80:20 dan 73.01% dengan *ratio* 70:30. Akurasi tertinggi dari kedua pengujian ini didapatkan pada saat mengimplementasikan parameter *leaning rate* 0.1, dengan *window* 0.2, 0.3, dan 0.4. Hasil rincian nilai akurasi tertiggi dari pengujian yang menggunakan *SMOTE* dengan seleksi fitur dan model *LVQ* 2 dapat dilihat pada Tabel 8.

Tabel 8. Hasil Pengujian *LVQ* 2 Menggunakan *SMOTE* dengan kombinasi *Information Gain* dengan *Threshold* 0.7 dan *Ratio Data* 90:10.

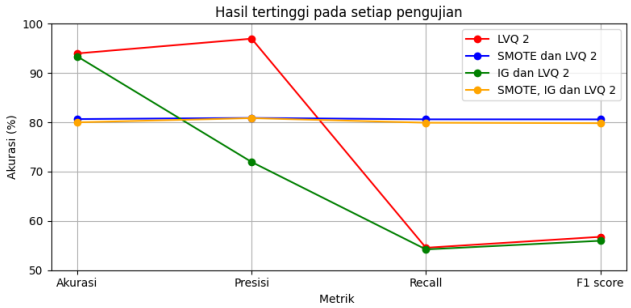
Learning rate (α)	Window (ε)	Akurasi	Presisi	Recal	F1 score
0.1	0.2, 0.3, 0.4	80.00%	80.85%	79.92%	79.83%
0.3	0.2, 0.3, 0.4	74.10%	74.88%	74.01%	73.85%
0.6	0.2, 0.3, 0.4	75.08%	76.24%	74.98%	74.75%
0.9	0.2	49.51%	24.75%	50.00%	33.11%
	0.3, 0.4	72.46%	72.71%	72.51%	72.41%

Hasil dari setiap pengujian dapat disimpulkan bahwa pengujian dengan model *LVQ* 2 dan menggunakan seleksi fitur model *LVQ* 2, mendapatkan akurasi tertinggi pada parameter yang sama, yaitu *learning rate* 0.9, min *learning rate* 0.001, *window* 0.2, dan *ratio* 90:10. Pengujian menggunakan *SMOTE* dengan model *LVQ* 2 dan pengujian menggunakan *SMOTE* dengan seleksi fitur dan model *LVQ* 2 mendapatkan akurasi tertinggi pada parameter yang sama, yaitu *learning rate* 0.1, min *learning rate* 0.001, window 0.2, 0.3, dan 0.4, serta menggunakan *ratio* 90:10. Perbandingan akurasi tertinggi pada setiap pengujian dapat dilihat pada Tabel 9.

Tabel 9. Hasil Akurasi Tertinggi pada Setiap Pengujian.

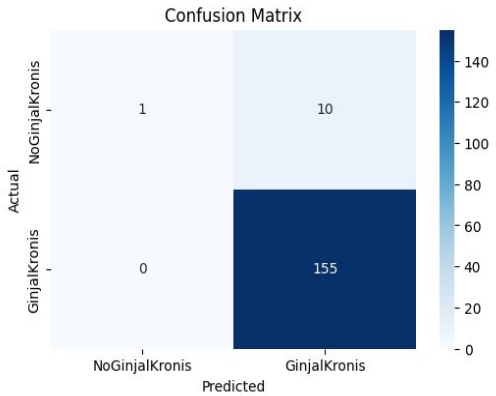
Pengujian	Akurasi	Presisi	Recall	F1 score
LVQ 2	93.98%	96.97%	54.55%	56.77%
SMOTE + *LVQ* 2	80.66%	80.89%	80.61%	80.60%
Information Gain + *LVQ* 2	93.37%	71.95%	54.22%	55.97%
SMOTE + *Information Gain* + *LVQ* 2 (*thresholds* 0.7)	80.00%	80.85%	79.92%	79.83%

Tabel 9., menunjukkan bahwa akurasi tertinggi didapatkan pada pengujian menggunakan model *LVQ* 2. Namun saat menggunakan *SMOTE*, akurasi menurun tetapi presisi, *recall* dan *F1 score* meningkat. Saat *LVQ* 2 dikombinasikan dengan seleksi fitur, akurasi kembali meningkat, tetapi sedikit lebih rendah dibandingkan pengujian model *LVQ*. Pada pengujian *SMOTE* dengan kombinasi seleksi fitur dan model *LVQ* 2, mendapatkan akurasi lebih rendah dari pada saat menggunakan *SMOTE*, namun threshold 0.7 dengan fitur terpilih sebanyak 33, mendapatkan akurasi lebih tinggi dibandingkan *threshold* 0.3 dengan fitur terpilih sejumlah 36. Grafik perbandingan nilai akurasi tertinggi pada setiap pengujian dapat dilihat pada Gambar 6.



Gambar 6. Grafik Hasil Akurasi Tertinggi pada Setiap Pengujian.

Evaluasi dilakukan setelah pengujian menggunakan *confusion matrix* untuk menganalisis jumlah prediksi yang benar dan salah. Hasil pengujian *LVQ* 2 tanpa *information gain* dengan akurasi tertinggi menunjukan bahwa model berhasil mengklasifikasikan 1 data kelas yang tidak terkena penyakit ginjal kronis (PGK) dengan benar (*True Positif*), namun terdapat 10 data kelas yang tidak terkena PGK yang salah klasifikasi sebagai yang terkena PGK (*False Positif*). Dan untuk 155 data kelas yang terkena PGK berhasil diklasifikasi semua (*True Negative*), dimana tidak ada data kelas yang terkena PGK salah klasifikasi (*False Negative*). Hasil *confusion matrix* dapat dilihat pada Gambar 7.



Gambar 7. *Confusion Matrix* Model *LVQ* 2 Tanpa *Information Gain*.

4. KESIMPULAN

Penelitian ini mengatasi tantangan klasifikasi penyakit ginjal kronis yang memiliki risiko kematian dan biaya perawatan tinggi, dengan fokus pada *dataset* berjumlah 1.659 data dan 53 atribut yang memerlukan seleksi fitur untuk mengidentifikasi atribut paling berpengaruh. Langkah penyelesaiannya melibatkan *implementasi Learning Vector Quantization* 2 (*LVQ* 2) untuk klasifikasi dan *Information Gain* untuk seleksi fitur, melalui tahapan pengumpulan data, analisis kebutuhan KDD (termasuk seleksi data, *preprocessing*, normalisasi, dan penanganan ketidakseimbangan kelas dengan *SMOTE*), seleksi fitur, klasifikasi *LVQ* 2, serta evaluasi menggunakan akurasi, presisi, *recall*, dan *F1 score*. Hasilnya menunjukkan akurasi tertinggi pada model *LVQ* 2 tanpa seleksi fitur (93.98%), namun dengan presisi, *recall*, dan *F1 score* rendah akibat ketidakseimbangan data. Penggunaan *SMOTE* berhasil meningkatkan ketiga metrik tersebut meski akurasi turun menjadi 80.66%. Seleksi fitur *Information Gain* (*threshold* 0.3) mencapai akurasi 93.37%, hampir setara dengan *LVQ* 2 murni, sementara kombinasi *SMOTE* dan seleksi fitur (*threshold* 0.7, menghasilkan 33 fitur) memperoleh akurasi sebesar 80.00% dengan metrik lain yang seimbang. Implikasi dari penelitian ini adalah *LVQ* 2 terbukti mampu mengklasifikasikan penyakit ginjal kronis pada data tidak seimbang dengan akurasi stabil. *SMOTE* efektif meningkatkan presisi, *recall*, dan *F1 score*, dan *Information Gain* dapat mengefisienkan model. Penelitian selanjutnya disarankan menggunakan metode penyeimbangan data lain atau model berbeda, serta memperhatikan metrik evaluasi selain akurasi untuk penilaian performa model yang lebih komprehensif pada data tidak seimbang.

DAFTAR PUSTAKA

[1] C. J. G. Paath, G. Masi, and F. Onibala, "Study Cross Sectional : Dukungan Keluarga Dengan Kepatuhan Hemodialisa Pada Pasien Gagal Ginjal Kronis," J. Keperawatan, vol. 8, no. 1, p. 106, 2020, doi: 10.35790/jkp.v8i1.28418.
[2] S. Anggraini and Z. Fadila, "Kualitas Hidup Pasien Gagal Ginjal Kronik Dengan Dialisis Di Asia Tenggara : a Systematic Review," Hearty, vol. 11, no. 1, p. 77, 2022, doi: 10.32832/hearty.v11i1.7947.
[3] V. K. Gliselda, "Diagnosis dan Manajemen Penyakit Ginjal Kronis (PGK)," J. Med. Hutama, vol. 2, no. 04 Juli, pp. 1135–1141, 2021.
[4] U. Hasanah, N. R. Dewi, L. Ludiana, A. T. Pakarti, and A. Inayati, "Analisis Faktor-Faktor Risiko Terjadinya Penyakit Ginjal Kronik Pada Pasien Hemodialisis," J. Wacana Kesehat., vol. 8, no. 2, p. 96, 2023, doi: 10.52822/jwk.v8i2.531.
[5] A. Kurniadi Hermawan, A. Nugroho, and Edora, "Analisa Data Mining Untuk Prediksi Penyakit Ginjal Kronik Dengan Algoritma Regresi Linier," Bull. Inf. Technol., vol. 4, no. 1, pp. 37–48, 2023, doi: 10.47065/bit.v4i1.475.
[6] M. F. S. Wibowo, N. F. Puspitasari, and B. Satya, "Penerapan Data Mining Dan Algoritma Naïve Bayes Untuk Pemilihan Konsentrasi Mahasiswa Menggunakan Metode Klasifikasi," J. Inf. Syst. Manag., vol. 3, no. 2, pp. 39–45, 2022, doi: 10.24076/joism.2022v3i2.680.
[7] I. Wisnuadji Gamadarenda and I. Waspada, "Implementation of Data Mining for the Detection of Chronic Kidney Disease (Ckd) Using K-Nearest Neighbor (Knn) With Backward Elimination," J. Teknol. Inf. dan Ilmu Komput., vol. 7, no. 2, pp. 417–426, 2020, doi: 10.25126/jtiik.202071896.
[8] N. Ismail and S. Lestari, "Mendiagnosis Penyakit Ginjal Kronis Mengunakan Algoritme C4.5," Semin. Nas. Has. Penelit. dan Pengabdi. Masy. 2023, pp. 25–31, 2023.
[9] J. Gea, "Implementasi Algoritma Learning Vector Quantization Untuk Pengenalan Barcode Barang," J. Informatics, Electr. Electron. Eng., vol. 2, no. 1, pp. 1–4, 2022, doi: 10.47065/jieee.v2i1.385.
[10] 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.
[11] M. K. Khairy, S. H. Sitorus, and D. M. Midyanti, "Klasifikasi Ukuran Pakaian Menggunakan Metode Learning Vector Quantization 2," vol. 07, no. 03, 2019.
[12] L. A. Widyasari, P. S. Sasongko, Sutikno, Suhartono, and E. Reynaldhi, "The early detection system of pulmonary tuberculosis disease using learning vector quantization 2 (lvq2)," J. Phys. Conf. Ser., vol. 1217, no. 1, 2019, doi: 10.1088/1742-6596/1217/1/012120.
[13] P. N. Sabrina and A. Komarudin, "Prediksi Penyakit Diabetes Dengan Metode K-Nearest Neighbor (Knn) Dan Seleksi Fitur Information Gain," vol. 8, no. 6, pp. 11320–11326, 2024.
[14] A. Bijaksana, P. Negara, H. Muhardi, and I. M. Putri, "Analisis Sentimen Maskapai Penerbangan Menggunakan Metode Naive Bayes Dan Seleksi Fitur Information Gain Sentiment Analysis on Airlines Using Naïve Bayes Method and Feature Selection Information Gain," Jtiik, vol. 7, no. 3, pp. 599–606, 2020, doi: 10.25126/jtiik.202071947.
[15] A. Aziz, F. Insani, J. Jasril, and F. Syafria, "Implementasi Metode Learning Vector Quantization (LVQ) Untuk Klasifikasi Keluarga Beresiko Stunting," Build. Informatics, Technol. Sci., vol. 5, no. 1, pp. 12–21, 2023, doi: 10.47065/bits.v5i1.3478.
[16] U. Mutmainnah, B. D. Setiawan, and C. Dewi, "Pengaruh Seleksi Fitur Information Gain pada K-Nearest Neighbor untuk Klasifikasi Tingkat Kelancaran Pembayaran Kredit Kendaraan," J. Pengemb. Teknol. Inf. dan Ilmu Komput., vol. 3, no. 9, pp. 8882–8888, 2019.