



TESIS - SM 142501

PERBANDINGAN METODE *K-NEAREST NEIGHBOR* (K-NN) DAN *LEARNING VECTOR QUANTIZATION* (LVQ) UNTUK PERMASALAHAN KLASIFIKASI TINGKAT KEMISKINAN

SANTOSO
NRP. 1213201039

Dosen Pembimbing:
Prof. Dr. M. Isa Irawan, MT

PROGRAM MAGISTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016



THESIS - SM 142501

**COMPARISON OF K-NEAREST NEIGHBOR (K-NN)
AND LEARNING VECTOR QUANTIZATION (LVQ)
METHODS FOR POVERTY LEVEL
CLASSIFICATION PROBLEM**

SANTOSO
NRP. 1213201039

Supervisor:
Prof. Dr. M. Isa Irawan, MT

MASTER'S DEGREE
MATHEMATICS DEPARTMENT
FACULTY OF MATHEMATICS AND NATURAL SCIENCES
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2016

**PERBANDINGAN METODE *K-NEAREST NEIGHBOR* (K-NN)
DAN *LEARNING VECTOR QUANTIZATION* (LVQ) UNTUK
PERMASALAHAN KLASIFIKASI TINGKAT KEMISKINAN**

Tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Sains (M.Si.)

di

Institut Teknologi Sepuluh Nopember Surabaya

Oleh:

SANTOSO

NRP. 1213 201 039

Tanggal Ujian : 14 Januari 2016

Periode Wisuda : Maret 2016

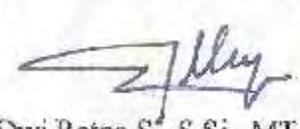
Disetujui oleh :


Prof. Dr. M. Isa Irawan, MT.
NIP. 19631225 198903 1 001

(Pembimbing)


Dr. Imam Mukhlash, S.Si., MT.
NIP. 19700831 199403 1 003

(Penguji)


Dr. Dwi Ratna S., S.Si., MT.
NIP. 19690405 199403 2 003

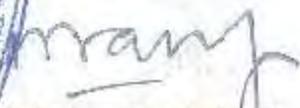
(Penguji)


Dr. Budi Setiyono, S.Si., MT.
NIP. 19720207 199702 1 001

(Penguji)



Direktur Program Pascasarjana


Prof. Ika Djauhar Manfaat, M.Sc., Ph.D.

NIP. 19601202 198701 1 001

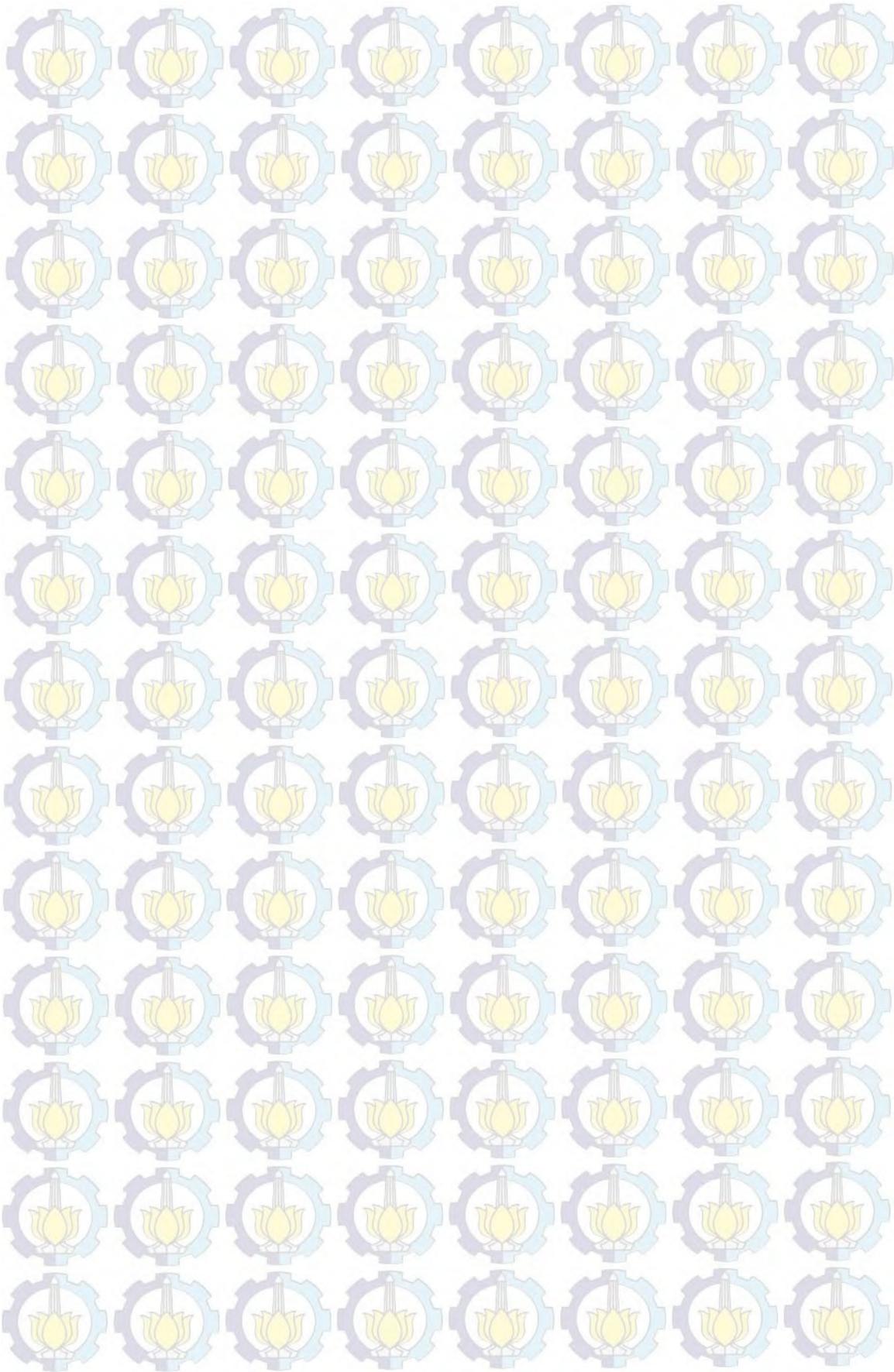
DAFTAR ISI

HALAMAN JUDUL	
LEMBAR PENGESAHAN.....	i
ABSTRAK	iii
ABSTRACT	v
DAFTAR ISI	vii
DAFTAR TABEL	ix
DAFTAR GAMBAR	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian	3
BAB 2 TINJAUAN PUSTAKA DAN DASAR TEORI	5
2.1 Hasil Penelitian Sebelumnya.....	5
2.2 Klasifikasi.....	5
2.3 Metode <i>K-Nearest Neighbor</i> (K-NN)	6
2.3.1 Algoritma <i>K-Nearest Neighbor</i> (K-NN)	8
2.4 Metode <i>Learning Vector Quantization</i> (LVQ).....	8
2.4.1 Arsitektur Jaringan <i>Learning Vector Quantization</i> (LVQ)	9
2.4.2 Algoritma <i>Learning Vector Quantization</i> (LVQ)	10
2.5 Kemiskinan.....	13
2.5.1 Indikator Kemiskinan	15
BAB 3 METODE PENELITIAN	17
3.1 Tahapan Penelitian.....	17
3.2 Diagram Alur Proses Penelitian.....	18
BAB 4 HASIL DAN PEMBAHASAN.....	19
4.1 Pengambilan dan Pengolahan Data.....	19
4.1.1 Pengambilan Data	19

4.1.2 Pengolahan Data.....	20
4.2 Implementasi Metode.....	23
4.2.1 Implementasi Metode KNN.....	24
a. Proses <i>Training</i> dan Testing Metode KNN.....	26
4.2.2 Pelatihan KNN dengan <i>Toolbox</i> Matlab.....	27
4.2.3 Implementasi Metode LVQ.....	28
a. Proses <i>Training</i> LVQ.....	32
b. Proses <i>Testing</i> LVQ.....	33
4.2.4 Pelatihan LVQ dengan <i>Toolbox</i> Matlab.....	34
4.3 Analisa dan Pembahasan.....	35
4.3.1 Metode KNN.....	35
4.3.2 Metode LVQ.....	37
4.4 Perbandingan Metode KNN dan LVQ.....	41
BAB 5 KESIMPULAN DAN SARAN.....	43
5.1 Kesimpulan.....	43
5.2 Saran.....	44
DAFTAR PUSTAKA.....	45
LAMPIRAN	

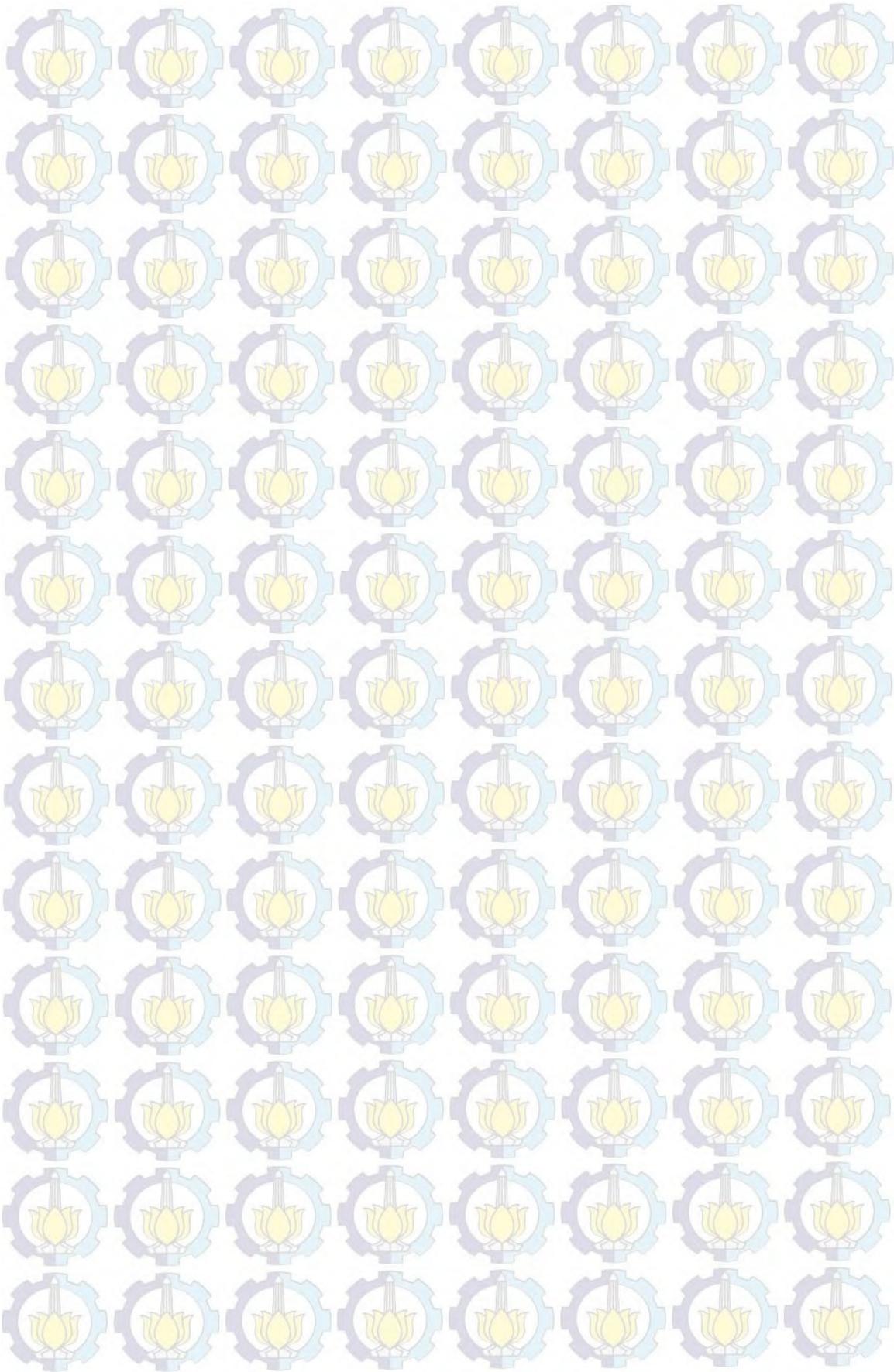
DAFTAR TABEL

	Hal
Tabel 2.1 Indikator Kemiskinan	15
Tabel 4.1 Kriteria Kemiskinan.....	19
Tabel 4.2 Hasil Transformasi Data dalam Bentuk Numerik	22
Tabel 4.3 Hasil Normalisasi 10 Dataset RTS.....	22
Tabel 4.4 Contoh Data <i>Training</i> untuk Proses KNN	24
Tabel 4.5 Hasil Urutan Jarak	26
Tabel 4.6 Contoh Data <i>Training</i> untuk Proses LVQ.....	30
Tabel 4.7 Hasil Akurasi Klasifikasi Metode KNN dengan 216 Data <i>Testing</i> yang Ditinjau dari Parameter k	36
Tabel 4.8 Hasil Akurasi Klasifikasi Metode KNN dengan 54 Data <i>Testing</i> yang Ditinjau dari Parameter k	36
Tabel 4.9 Hasil Akurasi Metode LVQ dengan 216 Data <i>Testing</i> yang Ditinjau dari Jumlah Iterasi dan <i>Learning rate</i>	38
Tabel 4.10 Hasil Akurasi Metode LVQ dengan 54 Data <i>Testing</i> yang Ditinjau dari Jumlah Iterasi dan <i>Learning rate</i>	40
Tabel 4.11 Perbandingan Akurasi Hasil Klasifikasi Metode KNN dan LVQ.....	42



DAFTAR GAMBAR

	Hal
Gambar 2.1 Arsitektur Jaringan LVQ	10
Gambar 3.1 Diagram Alur Proses Penelitian.....	18
Gambar 4.1 Diagram Alir Proses KNN.....	27
Gambar 4.2 Arsitektur Jaringan LVQ Permasalahan Tingkat Kemiskinan	29
Gambar 4.3 Diagram Alir Proses <i>Training</i> LVQ.....	33
Gambar 4.4 Diagram Alir Proses <i>Testing</i> LVQ.....	34
Gambar 4.5 Grafik keakurasian metode KNN ditinjau dari Parameter k	37
Gambar 4.6 Grafik Keakurasian Metode LVQ dengan 216 Data <i>Training</i> ditinjau dari Jumlah Iterasi	39
Gambar 4.7 Grafik Keakurasian Metode LVQ dengan 162 Data <i>Training</i> ditinjau dari Jumlah Iterasi	40



KATA PENGANTAR

Puji syukur Alhamdulillah penulis panjatkan kehadirat Allah SWT yang telah melimpahkan Rahmat, Taufiq, dan Hidayah-Nya, serta junjungan Beliau Rasulullah SAW atas suri teladan yang dibawanya sehingga penulis dapat menyelesaikan Tesis yang berjudul **“Perbandingan Metode *K-Nearest Neighbor* (K-NN) dan *Learning Vector Quantization* (LVQ) untuk Permasalahan Klasifikasi Tingkat Kemiskinan”** ini tepat pada waktunya. Tesis ini merupakan sebagian persyaratan kelulusan dalam memperoleh gelar magister di Program Studi Magister Matematika, Fakultas MIPA, Institut Teknologi Sepuluh Nopember.

Penyusunan Tesis ini tidak lepas dari bimbingan, bantuan, dan dukungan moral dari banyak pihak. Oleh sebab itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. M. Isa Irawan, MT., selaku dosen pembimbing yang telah meluangkan waktu untuk membimbing, memberikan masukan dan mendorong penulis dalam menyelesaikan tesis ini.
2. Dr. Dwi Ratna S., S.Si., MT., selaku dosen wali dan penguji yang telah memberikan motivasi, arahan, dan bimbingan selama penulis menempuh kuliah.
3. Dr. Imam Mukhlash, S.Si., MT, selaku Ketua Jurusan Matematika FMIPA ITS Surabaya dan sebagai dosen penguji yang telah memberikan masukan dan juga motivasi bagi penulis sehingga tesis ini dapat selesai.
4. Dr. Budi Setiyono, S.Si., MT, selaku dosen penguji yang telah bersedia memberikan kritik, saran dan arahan demi kesempurnaan tesis ini.
5. Dr. Subiono, M.Sc., selaku Koordinator Program Studi Pascasarjana Jurusan Matematika MIPA ITS Surabaya yang telah memberikan masukan dan motivasi dalam menyelesaikan tesis ini.
6. Seluruh dosen Matematika yang telah memberikan bekal dan ilmu pengetahuan serta staf administrasi Program Studi Magister Matematika atas segala bantuannya.

7. Istriku Novia Narulita dan anakku Ibrahim Alfaris S., yang telah memberikan do'a, motivasi dan semangat dalam menyelesaikan tesis ini.

8. Ibu, Bapak, beserta keluarga dan Mertua yang selalu memberikan dukungan dan doa agar penulis dapat menyelesaikan tesis ini.

9. Teman-teman seperjuangan mahasiswa Pascasarjana Jurusan Matematika FMIPA ITS Surabaya angkatan 2013 yang telah memberikan semangat.

Semoga Allah SWT memberikan anugerah dan karunia-Nya kepada semua pihak yang telah membantu menyelesaikan tesis ini.

Penulis menyadari bahwa dalam penulisan tesis ini masih banyak kekurangan, sehingga kritik dan saran dari pembaca sangat penulis harapkan untuk perbaikan kedepannya. Harapan penulis semoga tesis ini dapat bermanfaat bagi pembaca, khususnya mahasiswa Institut Teknologi Sepuluh Nopember.

Surabaya, Januari 2016

Penulis

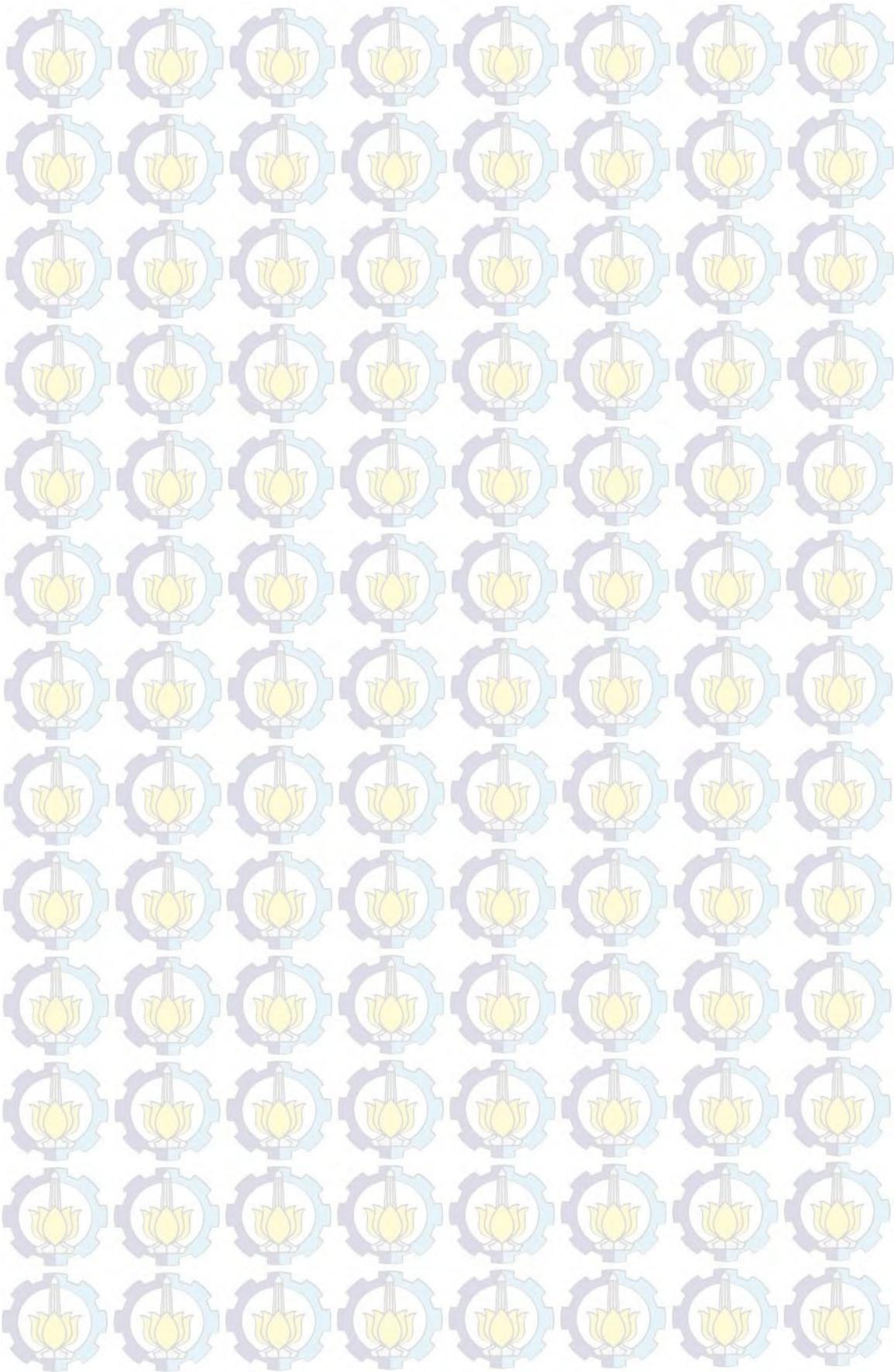
PERBANDINGAN METODE *K-NEAREST NEIGHBOR* (K-NN) DAN *LEARNING VECTOR QUANTIZATION* (LVQ) UNTUK PERMASALAHAN KLASIFIKASI TINGKAT KEMISKINAN

Nama Mahasiswa : Santoso
NRP : 1213 2010 39
Pembimbing : Prof. Dr. M. Isa Irawan, MT

ABSTRAK

Kemiskinan merupakan ketidakmampuan individu dalam memenuhi kebutuhan dasar minimal untuk hidup layak. Masalah kemiskinan merupakan salah satu persoalan mendasar yang menjadi pusat perhatian pemerintah daerah. Upaya pemerintah untuk menanggulangi kemiskinan melalui beberapa program pengentasan kemiskinan. Pemerintah sering kesulitan memilah tingkat-tingkat kemiskinan dalam masyarakat, untuk itu diperlukan suatu penelitian yang dapat membantu pemerintah mengidentifikasi tingkat kemiskinan agar bantuan pemerintah tidak salah sasaran. Penelitian ini menampilkan dua metode klasifikasi yaitu *K-Nearest Neighbor* dan *Learning Vector Quantization*. Tujuan dari penelitian ini adalah membandingkan nilai keakurasian dari kedua metode untuk hasil klasifikasi tingkat kemiskinan. Data yang digunakan sebagai atribut yaitu data yang mencirikan kemiskinan diantaranya meliputi aspek perumahan, kesehatan, pendidikan, ekonomi dan penghasilan. Dari hasil uji coba dengan menggunakan kedua metode menunjukkan nilai akurasi menggunakan KNN sebesar 93.52 %, sedangkan nilai akurasi menggunakan LVQ sebesar 75.93%. Sehingga dapat disimpulkan bahwa klasifikasi tingkat kemiskinan menggunakan metode KNN memberikan kinerja yang lebih baik dibanding metode LVQ.

Kata kunci : Kemiskinan, *K-Nearest Neighbor*, *Learning Vector Quantization*.



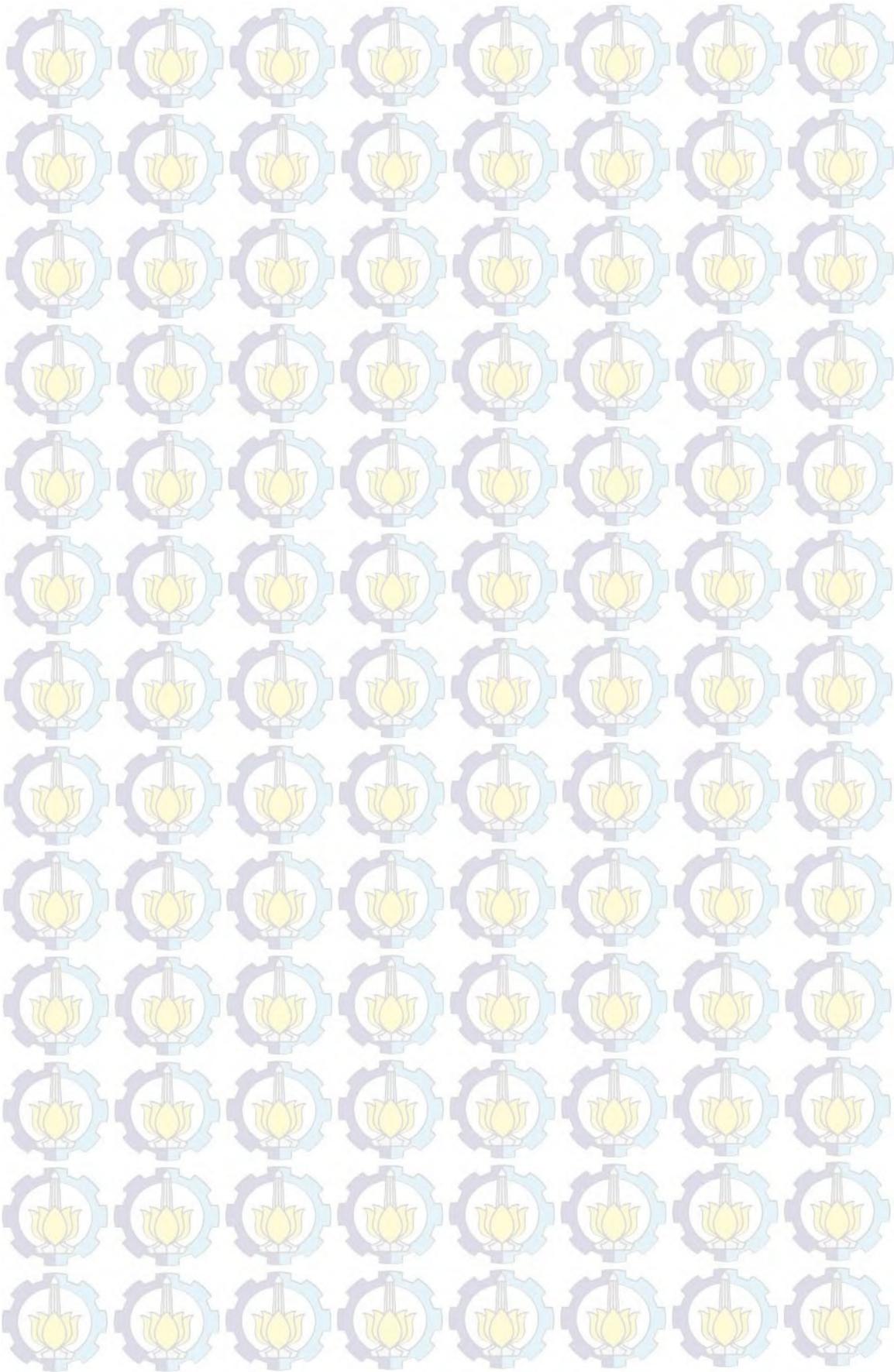
COMPARISON OF K-NEAREST NEIGHBOR (K-NN) AND LEARNING VECTOR QUANTIZATION (LVQ) METHODS FOR POVERTY LEVEL CLASSIFICATION PROBLEM

By : Santoso
Student Identity Number : 1213 2010 39
Supervisor : Prof. Dr. M. Isa Irawan, MT

ABSTRACT

Poverty is the inability of individuals to fulfill the minimum basic needs for a decent life. The problem of poverty is one of the fundamental problems that become the center of attention of local government. Government efforts to overcome poverty through poverty alleviation programs. Government is often difficult to sort out of the poverty levels in the society, therefore is necessary to a study that may help the government to identify of the poverty level in order that aid the government did not mistargeted. The aim is featured two classification methods of KNN and LVQ. The purpose of this study was to compare the accuracy of the value of both methods for classification results in poverty levels. The data is used as an attribute that is data that characterizes poverty among others include aspects of housing, health, education, economics and income. From the test results using both methods shows an accuracy value uses KNN is 93.52%, and an accuracy value uses LVQ is 75.93%. It can be concluded that the classification of poverty level using KNN method gives better performance than Learning Vector Quantization.

Keywords : K-Nearest Neighbor, Learning Vector Quantization, Poverty.



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Masalah kemiskinan merupakan salah satu persoalan mendasar yang menjadi pusat perhatian pemerintah di negara manapun. Aspek penting untuk mendukung strategi penanggulangan kemiskinan adalah tersedianya data kemiskinan yang akurat dan tepat sasaran. Pengukuran kemiskinan yang dapat dipercaya dapat menjadi instrumen tangguh bagi pengambil kebijakan dalam memfokuskan perhatian pada suatu daerah dengan kondisi hidup orang miskin. Data kemiskinan yang baik dapat digunakan untuk mengevaluasi kebijakan pemerintah terhadap kemiskinan serta menentukan target penduduk miskin dengan tujuan untuk memperbaiki kondisi mereka (BPS, 2008).

Salah satu upaya pemerintah untuk mengentas kemiskinan yaitu melalui beberapa program penanggulangan kemiskinan. Dalam hal ini pemerintah sering kesulitan untuk memilah tingkat-tingkat kemiskinan dalam masyarakat khususnya rumah tangga miskin sehingga penyaluran bantuan terkadang tidak tepat sasaran. Untuk menunjang keberhasilan pelaksanaan program terutama yang berkaitan dengan penanggulangan kemiskinan, diperlukan suatu penelitian yang dapat membantu pemerintah dalam mengidentifikasi dan mengklasifikasi rumah tangga miskin yang mempunyai ciri-ciri atau karakteristik kemiskinan yang hampir sama. Dengan mengetahui informasi mengenai kriteria kemiskinan dari masing-masing kelas diharapkan program kebijakan pemerintah daerah dapat disusun secara lebih terarah sesuai target atau sasaran yang ingin dicapai.

Permasalahan klasifikasi telah banyak dibahas oleh para peneliti dalam banyak konteks dan disiplin ilmu yang mencerminkan manfaat dan daya tarik yang luas sebagai salah satu langkah dalam analisis data. Kecermatan dan ketepatan dalam hal klasifikasi data merupakan hal yang sangat penting. Beberapa metode klasifikasi yang sering digunakan, antara lain *rule based*, *neural network*, *support vector machine*, *naive bayes*, dan *nearest neighbour*.

Dengan adanya beberapa metode, permasalahan yang sering muncul adalah jenis metode yang harus dipilih. Penelitian yang telah dilakukan sebelumnya yaitu tentang *Classification of Heart Disease Using K-Nearest Neighbor and Genetic Algorithm* oleh Jabbar (Jabbar, M.A., dkk, 2013). Penelitian ini memberikan hasil bahwa penerapan metode GA kedalam KNN untuk klasifikasi penyakit jantung memiliki tingkat akurasi yang lebih baik jika dibandingkan dengan metode KNN tanpa GA. Pada tahun 2013, Farsiah, dkk, melakukan kasifikasi gambar berwarna menggunakan *k-nearest neighbor* (KNN) dan *support vektor machine* (SVM). Selain itu, juga ada penelitian tentang komparasi hasil klasifikasi penyakit diabetes mellitus menggunakan jaringan syaraf tiruan *backpropagation* dan *learning vector quantization* (LVQ) oleh Nurkhozin (Nurkhozin, dkk, 2011).

Berdasarkan uraian diatas, dalam penelitian kali ini penulis akan mengkaji dan membandingkan kinerja metode *k-Nearest Neighbor* (KNN) dan *Learning Vektor Quantization* (LVQ) untuk permasalahan klasifikasi pada tingkat kemiskinan.

1.2 Rumusan Masalah

Berdasarkan uraian di atas maka dapat dirumuskan suatu permasalahan dalam bentuk pertanyaan sebagai berikut: Bagaimana perbandingan akurasi dan performansi metode *K-Nearest Neighbor* (KNN) dan *Learning Vector Quantization* (LVQ) untuk hasil klasifikasi tingkat kemiskinan?

1.3 Batasan Masalah

Agar penyelesaian permasalahan di atas lebih fokus dan untuk menghindari perbedaan persepsi, berikut ini diberikan beberapa batasan.

1. Data yang digunakan dalam penelitian ini adalah data sekunder yang diperoleh dari Badan Pusat Statistik (BPS) Kabupaten Situbondo. Data ini berupa dataset Rumah Tangga Sasaran (RTS) dengan 14 variabel indikator kemiskinan.
2. Tingkat kemiskinan yang dimaksud pada penelitian ini dikategorikan menjadi tiga kelas yaitu kelas pertama yang mengkategorikan rentan miskin, kelas

kedua yang mengkategorikan miskin dan kelas ketiga yang mengkategorikan sangat miskin.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, tujuan dari penelitian ini adalah menganalisis dan membandingkan performansi dan keakurasian metode KNN dan LVQ untuk permasalahan klasifikasi tingkat kemiskinan.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah untuk menambah wawasan keilmuan mengenai teknik klasifikasi menggunakan metode *k-Nearest Neighbor* (KNN) dan *Learning Vektor Quantization* (LVQ) untuk permasalahan klasifikasi tingkat kemiskinan pada suatu daerah serta dapat memberikan referensi metode yang mempunyai akurasi paling akurat pada permasalahan klasifikasi. Selain itu hasil klasifikasi ini dapat menjadi bahan pertimbangan atau acuan pemerintah daerah khususnya Kabupaten Situbondo dalam upaya menanggulangi kemiskinan rumah tangga sasaran yang masih berada dibawah garis kemiskinan.



"Halaman ini sengaja dikosongkan"

BAB 2 TINJAUAN PUSTAKA DAN DASAR TEORI

Pada bab ini dijelaskan beberapa tinjauan pustaka dan dasar teori yang dikaji untuk digunakan dalam pembahasan selanjutnya.

2.1 Hasil Penelitian Sebelumnya

Penelitian yang telah dilakukan sebelumnya yaitu tentang *Classification of Heart Disease Using K-Nearest Neighbor and Genetic Algorithm* oleh Jabbar (Jabbar, M.A., dkk, 2013). Penelitian ini memberikan hasil bahwa penerapan metode GA kedalam KNN untuk klasifikasi penyakit jantung memiliki tingkat akurasi yang lebih baik jika dibandingkan dengan metode KNN tanpa GA. Farsiah, dkk, (2013) melakukan kasifikasi gambar berwarna menggunakan *k-nearest neighbor* (KNN) dan *support vektor machine* (SVM). Hasil penelitian ini menunjukkan bahwa tingkat akurasi gambar berwarna menggunakan metode KNN sama baiknya dengan metode SVM.

Selain itu, juga ada penelitian tentang komparasi hasil klasifikasi penyakit diabetes mellitus menggunakan jaringan syaraf tiruan *backpropagation* dan *learning vector quantization* (LVQ) oleh Nurkhozin (Nurkhozin, dkk, 2011). Penelitian ini menunjukkan hasil klasifikasi data diabetes menggunakan LVQ memberikan tingkat akurasi lebih tinggi atau akurat dalam membaca pola jika dibandingkan klasifikasi data menggunakan jaringan *Backpropagation*. Lusiyanti, D. (2014) melakukan perbandingan metode *learning vector quantization* (LVQ) dan *support vektor machine* (SVM) untuk prediksi penyakit jantung koroner. Pada penelitian ini digunakan parameter yang berbeda untuk tiap metode, sehingga tingkat akurasi yang diperoleh tampak bahwa metode SVM memberikan akurasi yang lebih baik dibandingkan dengan metode LVQ.

2.2 Klasifikasi

Klasifikasi adalah sebuah proses untuk menemukan model yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat

memperkirakan kelas dari suatu objek yang kelasnya tidak diketahui (Tan dkk, 2004). Di dalam klasifikasi diberikan sejumlah *record* yang dinamakan *training set*, yang terdiri dari beberapa atribut, atribut dapat berupa kontinyu ataupun kategoris, salah satu atribut menunjukkan kelas untuk *record*. Tujuan dari klasifikasi adalah untuk:

1. Menemukan model dari *training set* yang membedakan *record* kedalam kategori atau kelas yang sesuai, model tersebut kemudian digunakan untuk mengklasifikasikan *record* yang kelasnya belum diketahui sebelumnya pada *test set*.
2. Mengambil keputusan dengan memprediksikan suatu kasus, berdasarkan hasil klasifikasi yang diperoleh .

Klasifikasi dibedakan dalam dua tahap yaitu pelatihan dan prediksi. Pada tahap pelatihan dibentuk sebuah model domain permasalahan dari setiap dataset/kasus yang ada. Penentuan model tersebut berdasarkan analisis pada sekumpulan data pelatihan, yaitu data yang label kelasnya sudah diketahui. Pada tahap klasifikasi, dilakukan prediksi kelas pada kasus/data uji baru yang telah dibuat pada tahap pelatihan.

Dalam mengevaluasi performansi sebuah model yang dibangun oleh algoritma klasifikasi dapat dilakukan dengan menghitung jumlah dari *test record* yang di prediksi secara benar (akurasi) atau salah (*error rate*) oleh model tersebut. Akurasi didefinisikan sebagai berikut.

$$\text{Akurasi} = \frac{\text{Jumlah prediksi benar}}{\text{Jumlah total prediksi}}$$

2.3 Metode *K-Nearest Neighbor* (KNN)

Algoritma *K-Nearest Neighbor* (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap obyek berdasarkan data pembelajaran yang jaraknya paling dekat dengan obyek tersebut. Prinsip kerja dari *K-Nearest Neighbor* (KNN) adalah mencari jarak terdekat antara data yang akan dievaluasi dengan *k* tetangga (*neighbor*) terdekatnya dalam data pelatihan (Sreemathy, 2012). Algoritma *k-Nearest Neighbor* menggunakan metode pembelajaran

supervised. Perbedaan antara *supervised learning* dengan *unsupervised learning* adalah pada *supervised learning* bertujuan untuk menemukan pola baru dalam data dengan menghubungkan pola data yang sudah ada dengan data yang baru. Sedangkan pada *unsupervised learning*, data belum memiliki pola apapun, dan tujuan *unsupervised learning* untuk menemukan pola dalam sebuah data (Krisandi, N. dkk., 2013).

Pada fase pembelajaran, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi dari data pembelajaran. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk data test. Jarak dari vektor yang baru ini terhadap seluruh vektor data pembelajaran dihitung, dan sejumlah k buah yang paling dekat diambil. Nilai k yang terbaik untuk algoritma ini tergantung pada data. Secara umum, nilai k yang tinggi akan mengurangi efek *noise* pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur. Nilai k yang baik dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan *cross-validation*. Kasus khusus di mana klasifikasi diprediksikan berdasarkan data pembelajaran yang paling dekat (dengan kata lain, $k = 1$) disebut algoritma *k-nearest neighbor*.

Kelebihan dan kelemahan pada metode KNN adalah sebagai berikut:

1. Kelebihan

KNN memiliki kelebihan yaitu tangguh terhadap *training* data yang *noise* dan efektif apabila data latih nya besar.

2. Kelemahan

- a. KNN perlu menentukan nilai dari parameter k (jumlah dari tetangga terdekat).
- b. Pembelajaran berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil yang terbaik.
- c. Biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap sample uji pada keseluruhan sample *training*.

Tujuan dari algoritma KNN adalah untuk mengklasifikasi objek baru berdasarkan atribut dan *training samples*, dimana hasil dari sampel uji yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada k tetangga terdekat.

Pada proses pengklasifikasian, algoritma ini tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Algoritma KNN menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari sampel uji yang baru. Menurut Sreemathy (2012), peringkat untuk k tetangga terdekat berdasarkan nilai kesamaan dihitung menggunakan jarak *Euclidean* (*Euclidean Distance*) yang didefinisikan sebagai berikut:

$$D(X, Y) = \sqrt{\sum_{i=1}^p (X_i - Y_i)^2}$$

Dengan

$D(X, Y)$: Jarak *Euclidean* (*Euclidean Distance*)

X_i : sampel data

Y_i : data uji

p : dimensi data

i : variabel data

2.3.1 Algoritma *K-Nearest Neighbor*

Algoritma *K-Nearest Neighbor* dapat dituliskan sebagai berikut (Karegowda dkk, 2012) :

1. Tentukan : k , tetangga terdekat
2. Hitung jarak data baru (*input*) dengan data training

Ukuran jarak yang digunakan adalah jarak *Euclidean*:

$$D(X, Y) = \sqrt{\sum_{i=1}^p (X_i - Y_i)^2}$$

3. Urutkan jarak dari yang terdekat
4. Periksa kelas k tetangga terdekat
5. Kelas data baru = kelas mayoritas tetangga terdekat.

2.4 Metode *Learning Vector Quantization* (LVQ)

Jaringan LVQ (*Learning Vector Quantization*) pertama kali diperkenalkan oleh Tuevo Kohonen. LVQ merupakan salah satu jaringan Saraf tiruan yang melakukan pembelajaran pada lapisan kompetitif yang terawasi. Suatu

lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor *input* secara berkelompok ke dalam kelas yang sudah didefinisikan melalui jaringan yang telah dilatih. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif hanya bergantung pada jarak antara vektor-vektor *input*. Jika dua vektor *input* mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor *input* tersebut ke dalam kelas yang sama.

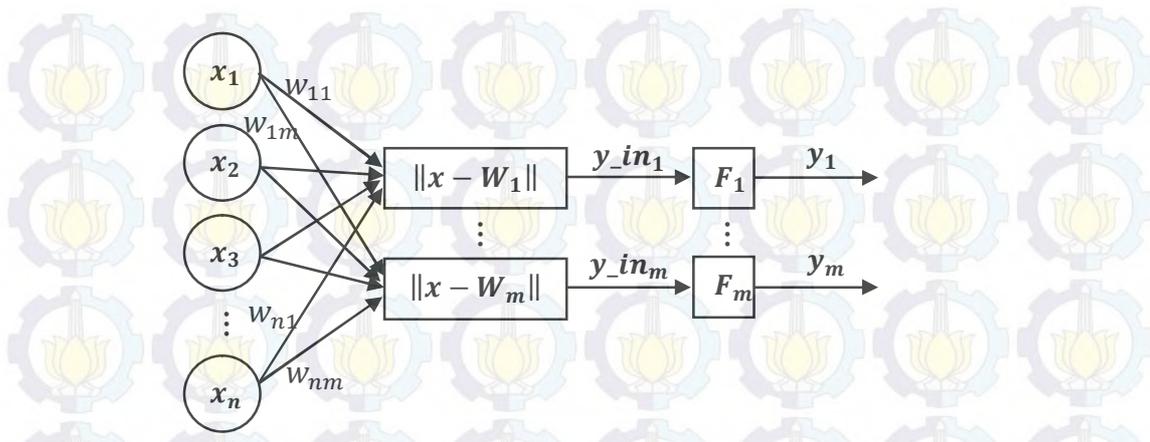
Jaringan LVQ adalah jaringan yang mengklasifikasi pola sehingga setiap unit keluaran menyatakan suatu kelas atau kategori. Vektor bobot untuk unit keluaran sering disebut vektor referensi untuk kelas yang dinyatakan oleh unit tersebut. Selama pelatihan unit keluaran dicari posisinya dengan mengatur bobotnya lewat pelatihan terawasi (Fausett, 1994).

Kelebihan dan kelemahan pada metode LVQ antara lain:

1. Kelebihan
 - a. Dapat meringkas dataset yang besar menjadi vektor codebook berukuran kecil untuk klasifikasi
 - b. Dimensi dalam codebook tidak dibatasi seperti pada teknik nearest neighbor
 - c. Model yang dihasilkan dapat diperbarui secara bertahap
2. Kelemahan
 - a. Diperlukan perhitungan jarak untuk seluruh atribut
 - b. Akurasi model dengan bergantung pada inisialisasi model serta parameter yang digunakan (learning rate, iterasi).
 - c. Akurasi juga dipengaruhi distribusi kelas pada data training.
 - d. Sulit untuk menentukan jumlah codebook vektor untuk masalah yang diberikan.

2.4.1 Arsitektur Jaringan *Learning Vektor Quantization* (LVQ)

Arsitektur jaringan *Learning Vektor Quantization* digambarkan pada Gambar 2.1 dibawah ini.



Gambar 2.1 Arsitektur Jaringan LVQ

Deskripsi dari Gambar 2.1 adalah sebagai berikut:

x : *vector* pelatihan masukan

y : *vector output* keluaran

F : fungsi aktivasi

W_j : *vector* bobot penghubung *output* ke- j

$\|x - w_j\|$: jarak *Euclidean* antara *vector* bobot dan unit *output* dimana:

$$\|x - w_j\| = \sqrt{\sum_{i=1}^n (x_i - w_j)^2} ; \text{ dengan } j = 1, 2, \dots, m$$

Misal pada Gambar 2.1 merupakan contoh struktur jaringan LVQ yang memiliki empat *input layers* dengan dua unit *neuron* pada *output layers*. W_1 dan W_2 merupakan bobot yang menghubungkan *input layers* ke *output layers*. Setiap fungsi aktivasi F melakukan pemetaan y_{in} ke y_1 atau y_2 . Fungsi aktivasi F_1 akan memetakan y_{in_1} ke $y_1 = 1$ apabila $|x - w_1| < |x - w_2|$ dan $y_2 = 0$. Demikian pula dengan fungsi aktivasi F_2 , akan memetakan y_{in_2} ke $y_2 = 1$ apabila $|x - w_1| > |x - w_2|$ dan $y_1 = 0$.

2.4.2 Algoritma *Learning Vector Quantization* (LVQ)

Berikut ini adalah algoritma *Learning Vector Quantization* (LVQ).

1. Tetapkan:

- a. Bobot awal variabel input ke- j menuju ke kelas (cluster ke- i : W_{ij} dengan $i = 1, 2, \dots, k$: dan $j = 1, 2, \dots, m$

- b. Masukkan epoch : $MaxEpoch$
- c. Parameter learning rate : α
- d. Pengurangan learning rate : $Deca$
- e. Minimum learning rate yang diperbolehkan: $Min\alpha$

2. Masukkan:

- a. Data input : x_{ij} dengan $i = 1, 2, \dots, n$ dan $j = 1, 2, \dots, m$
- b. Target berupa kelas : T_k dengan $k = 1, 2, \dots, n$

3. Tetapkan kondisi awal : $epoch = 0$

4. Kerjakan jika : $(epoch \leq MaxEpoch)$ dan $(\alpha \geq Min\alpha)$

a. $Epoch = epoch + 1$

b. Kerjakan untuk $i = 1$ sampai n

i. Tentukan J sedemikian hingga $|x_i - W_j|$ minimum dengan $j = 1, 2, \dots, k$

ii. Perbaiki W_j dengan ketentuan:

Jika $C_j = T$ maka

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

Jika $C_j \neq T$ maka

$$W_j(\text{baru}) = W_j(\text{lama}) - \alpha(x - W_j)$$

5. Reduksi nilai *learning rate*

Pengurangan nilai α bisa dilakukan dengan: $\alpha - Deca$; atau dengan cara
 $\alpha = \alpha - \alpha * Deca$

Contoh pengerjaan menggunakan LVQ.

Metode LVQ dalam mengklasifikasikan 5 vektor menjadi 2 kelas. Pada contoh ini, dua vektor referensi akan digunakan vektor *input* yang mewakili dua kelas 1 dan 2:

	Vektor <i>input</i>	Kelas
1.	(1, 1, 0, 0)	1
2.	(0, 0, 0, 1)	2

3. (0, 0, 1, 1) 2
4. (1, 0, 0, 0) 1
5. (0, 1, 1, 0) 2

Dua vektor yang pertama, yaitu:

- (1, 1, 0, 0) 1
- (0, 0, 0, 1) 2

merupakan dua vektor referensi yang akan digunakan untuk menginisialisasi bobot awal. Unit output yang pertama mewakili kelas 1 dan yang kedua mewakili kelas 2 (yang dilambangkan dengan $C_1 = 1$ dan $C_2 = 2$). Sedangkan tiga vektor sebagai vektor training, yaitu:

- (0, 0, 1, 1) 2
- (1, 0, 0, 0) 1
- (0, 1, 1, 0) 2

Di bawah ini ditunjukkan satu iterasi sebagai berikut:

Step 0: Inisialisasi bobot awal (dari vektor referensi)

$$w_1 = (1, 1, 0, 0)$$

$$w_2 = (0, 0, 0, 1)$$

Inisialisasi *learning rate* $\alpha = 0.1$

Step 1: Mulai komputasi

Step 2: Untuk vektor input $x = (0, 0, 1, 1)$ dengan $T = 2$ kerjakan Step 3 – 4

Step 3: $J = 2$, sejak x lebih dekat ke w_2 dari pada w_1

Step 4: Karena $T = 2$ dan $C_2 = 2$ *update* w_2 sebagai berikut:

$$w_2 = (0, 0, 0, 1) + 0.1[(0, 0, 1, 1) - (0, 0, 0, 1)]$$

$$= (0, 0, 0.1, 1)$$

Step 2: Untuk vektor input $x = (1, 0, 0, 0)$ dengan $T = 1$ kerjakan Step 3 – 4

Step 3: $J = 1$

Step 4: Karena $T = 1$ dan $C_1 = 1$ *update* w_1 sebagai berikut:

$$w_1 = (1, 1, 0, 0) + 0.1[(1, 0, 0, 0) - (1, 1, 0, 0)]$$

$$= (1, 0.9, 0, 0)$$

Step 2: Untuk vektor input $x = (0, 1, 1, 0)$ dengan $T = 2$ kerjakan Step 3 – 4

Step 3: $J = 1$

Step 4: Karena $T = 2$ tetapi $C_1 = 1$ update w_1 sebagai berikut:

$$\begin{aligned}w_1 &= (1, 0.9, 0, 0) - 0.1[(0, 1, 1, 0) - (1, 0.9, 0, 0)] \\ &= (1.1, 0.89, -0.1, 0)\end{aligned}$$

Step 5: *Training* telah lengkap dilakukan untuk 1 iterasi

Diperoleh w_1, w_2 yang baru tersebut digunakan untuk proses berikutnya.

Reduksi *learning rate*. Kondisi stop dipenuhi (Fausett, 1994).

2.5 Kemiskinan

Definisi kemiskinan yang digunakan di berbagai negara bermacam-macam. Kemiskinan sering dipandang sebagai ketidakmampuan untuk membayar biaya hidup minimal walaupun beberapa ahli berpendapat bahwa kemiskinan juga merupakan kurangnya akses terhadap jasa-jasa seperti pendidikan, kesehatan, informasi, serta kurangnya akses masyarakat terhadap partisipasi pembangunan dan politik.

Badan Perencanaan Pembangunan Nasional (Bappenas) mendefinisikan kemiskinan sebagai kondisi dimana seseorang atau sekelompok orang tidak mampu memenuhi hak-hak dasarnya untuk mempertahankan dan mengembangkan kehidupan yang bermartabat. Hak-hak dasar antara lain: (1) terpenuhinya kebutuhan pangan; (2) kesehatan, pendidikan, pekerjaan, perumahan, air bersih, pertanahan, sumberdaya alam dan lingkungan; (3) rasa aman dari perlakuan atau ancaman tindak kekerasan; (4) hak untuk berpartisipasi dalam kehidupan sosial-politik.

Badan Pusat Statistik (BPS) mendefinisikan kemiskinan merupakan ketidakmampuan individu dalam memenuhi kebutuhan dasar minimal untuk hidup layak. Yaitu kondisi yang berada dibawah garis nilai standar kebutuhan minimum yang disebut garis kemiskinan (*poverty line*) atau batas kemiskinan (*poverty threshold*). Nilai garis kemiskinan yang digunakan untuk menentukan kemiskinan mengacu pada kebutuhan minimum 2.100 kalori per orang per hari ditambah

dengan kebutuhan minimum non pangan. Individu yang pengeluarannya lebih rendah dari garis kemiskinan tersebut dikategorikan miskin (BPS, 2008).

Kemiskinan merupakan salah satu persoalan mendasar yang menjadi pusat perhatian pemerintah di negara manapun. Salah satu aspek penting untuk mendukung strategi penanggulangan kemiskinan adalah tersedianya data kemiskinan yang akurat dan tepat sasaran. Pengukuran kemiskinan yang tepat dipercaya dapat menjadi pedoman tangguh bagi pengambil kebijakan dalam memfokuskan perhatian pada suatu daerah dengan kondisi hidup orang miskin. Data kemiskinan yang baik dapat digunakan untuk mengevaluasi kebijakan pemerintah terhadap kemiskinan, membandingkan kemiskinan antar daerah, serta menentukan target penduduk miskin dengan tujuan untuk memperbaiki kondisi mereka (BPS, 2008).

Data kemiskinan yang bersumber dari BPS sering menjadi dasar dalam implementasi program penanggulangan kemiskinan oleh pemerintah. Seperti yang diketahui, BPS mengeluarkan dua jenis data kemiskinan, yaitu data kemiskinan makro dan data kemiskinan mikro. Kedua data ini memiliki kriteria, pengukuran, dan cakupan kemiskinan yang berbeda. Kriteria kemiskinan yang berbeda ini tentunya akan menimbulkan penafsiran yang berbeda pula mengenai jumlah penduduk miskin dan tingkat penanganan terhadap persoalan kemiskinan. Pendekatan pertama, yaitu kemiskinan makro yang dikeluarkan oleh BPS adalah data kemiskinan yang bersumber dari Survei Sosial Ekonomi Nasional (Susenas). Kemiskinan makro dihitung dengan menggunakan pendekatan kebutuhan dasar yang mencakup kebutuhan dasar makanan dan bukan makanan. Dari kebutuhan dasar ini dihitung suatu garis yang disebut garis kemiskinan. Selanjutnya, yang dikategorikan penduduk miskin adalah penduduk yang pengeluarannya ada di bawah garis kemiskinan (BPS, 2012).

Pendekatan kedua adalah kemiskinan mikro yang penghitungannya menggunakan pendekatan non moneter. Di Indonesia, pendataan kemiskinan mikro sudah dilakukan tiga kali. Pertama, data kemiskinan mikro yang bersumber dari Pendataan Sosial Ekonomi 2005 (PSE05), kedua pada tahun 2008 dengan diselenggarakannya Pendataan Program Perlindungan Sosial 2008 (PPLS08), dan yang terbaru adalah Pendataan Program Perlindungan Sosial (PPLS) 2011.

Jika data kemiskinan yang bersumber dari Susenas hanya mampu menyajikan jumlah dan persentase penduduk miskin di suatu wilayah, maka data mikro mampu menyediakan informasi mengenai penduduk miskin sampai dengan nama dan alamat penduduk miskin tersebut. Perbedaan yang terjadi selain metode dan pendekatan adalah juga cakupan. Pada kemiskinan makro hanya mencakup penduduk miskin, sedangkan pada kemiskinan mikro selain penduduk miskin juga mencakup penduduk hampir miskin (BPS, 2012).

2.5.1 Indikator Kemiskinan

Menurut BPS (2008), aspek yang menjadi dasar acuan dalam menentukan rumah tangga sasaran (RTS) antara lain perumahan, kesehatan, pendidikan, ekonomi dan penghasilan. Aspek tersebut diturunkan menjadi indikator-indikator kemiskinan seperti dalam Tabel 2.1 berikut:

Tabel 2.1 Indikator Kemiskinan

Aspek	Indikator
Perumahan	<ul style="list-style-type: none"> a. Luas lantai rumah/bangunan b. Jenis lantai rumah/bangunan c. Jenis dinding rumah/bangunan d. Penerangan rumah yang digunakan
Kesehatan	<ul style="list-style-type: none"> a. Kemampuan berobat ke puskesmas/poliklinik b. Kebiasaan membeli/konsumsi daging/ayam/susu c. Sumber air minum d. Fasilitas tempat buang air besar
Pendidikan	Pendidikan tertinggi kepala rumah tangga
Ekonomi	<ul style="list-style-type: none"> a. Kemampuan membeli pakaian b. Bahan bakar yang digunakan
Penghasilan	<ul style="list-style-type: none"> a. Pendapatan perbulan b. Lapangan pekerjaan kepala rumah tangga c. Kepemilikan aset

Sumber : BPS, 2008

Perumahan atau tempat tinggal merupakan tempat istirahat dan berlindung. Rumah yang baik harus memenuhi syarat kelayakan sebagai tempat tinggal. Beberapa variabel yang sering digunakan untuk menilai kelayakan sebuah tempat tinggal adalah luas, jenis lantai dan jenis dinding. Semakin bagus kualitas dari lantai mencerminkan kondisi sosial ekonomi yang tinggi.

Tingkat kesejahteraan di bidang kesehatan digambarkan oleh adanya sumber air minum, fasilitas tempat buang air, kemampuan berobat dan konsumsi makanan. Sumber air yang dipergunakan untuk minum haruslah air yang bersih. Air bersih adalah salah satu jenis sumber daya berbasis air yang bermutu baik dan biasa dimanfaatkan oleh manusia untuk dikonsumsi atau dalam melakukan aktivitas mereka sehari-hari. Air bersih sangat dipengaruhi oleh sumber air minum tersebut. Semakin jauh dari sumber pencemar, maka air tersebut akan semakin bersih dan sehat untuk dikonsumsi. Air yang bersih dan layak untuk diminum adalah air yang bersumber, seperti dari air kemasan, air isi ulang, leding, sumur bor, sumur terlindung, dan mata air terlindung. Kesehatan lingkungan di sekitar tempat tinggal juga dipengaruhi oleh fasilitas pembuangan akhir dari kotoran. Fasilitas pembuangan akhir tinja yang tidak baik akan memperburuk kesehatan lingkungan.

Tingkat kesejahteraan rumah tangga dalam aspek penghasilan dapat dilihat dari aset atau barang yang dimiliki dan pendapatan perkapita kepala rumah tangga. Semakin besar aset yang dimiliki maka akan semakin tinggi pula tingkat kesejahtraannya. Demikian pula jika pendapatan yang diperoleh semakin besar maka semakin tinggi pula tingkat perekonomiannya. Faktor yang mempengaruhi pendapatan kepala rumah tangga adalah lapangan pekerjaan yang juga dipengaruhi oleh tingkat pendidikan tertinggi kepala rumah tangga. Sedangkan tingkat kesejahteraan dalam aspek ekonomi dapat digambarkan dari segi kemampuan membeli pakaian dan pemakaian bahan bakar untuk memasak.

BAB 3

METODE PENELITIAN

Pada bagian ini diuraikan beberapa metode penelitian yang akan digunakan untuk mencapai tujuan penelitian.

3.1 Tahapan Penelitian

Tahap 1. Studi Literatur

Pada tahap pertama ini dilakukan identifikasi permasalahan yang dibahas yaitu mencari literatur-literatur yang berhubungan dengan bahan penelitian.

Tahap 2. Pengumpulan dan Pengolahan Data

Pada tahap ini dilakukan pengumpulan data dan penggalian sumber informasi yang berkaitan dengan penelitian ini. Data yang digunakan berupa data sekunder dari BPS Kabupaten Situbondo. Data ini berupa dataset Rumah Tangga Sasaran (RTS) yang didasarkan pada indikator kemiskinan. Data yang diperoleh yaitu berupa data karakteristik rumah tangga yang meliputi: luas lantai, jenis lantai, sumber air minum, fasilitas tempat buang air besar, kepemilikan aset, pendapatan, pendidikan kepala rumah tangga, dan konsumsi.

Pada tahap pengolahan data dilakukan transformasi data dengan merubah data kategori menjadi numerik. Selanjutnya dilakukan normalisasi data berdasarkan variabel sehingga data berada pada selang [0,1]. Untuk normalisasi data digunakan rumus:

$$x_* = \frac{x_{data} - x_{min}}{x_{max} - x_{min}}$$

Kemudian dilakukan pembagian data menjadi data *training* dan data *testing*.

Tahap 3. Implementasi Metode

Pada implementasi metode ini dilakukan pembagian data menjadi data *training* dan data *testing*, selanjutnya dilakukan proses *training* dan *testing* data menggunakan dua metode klasifikasi yaitu metode KNN

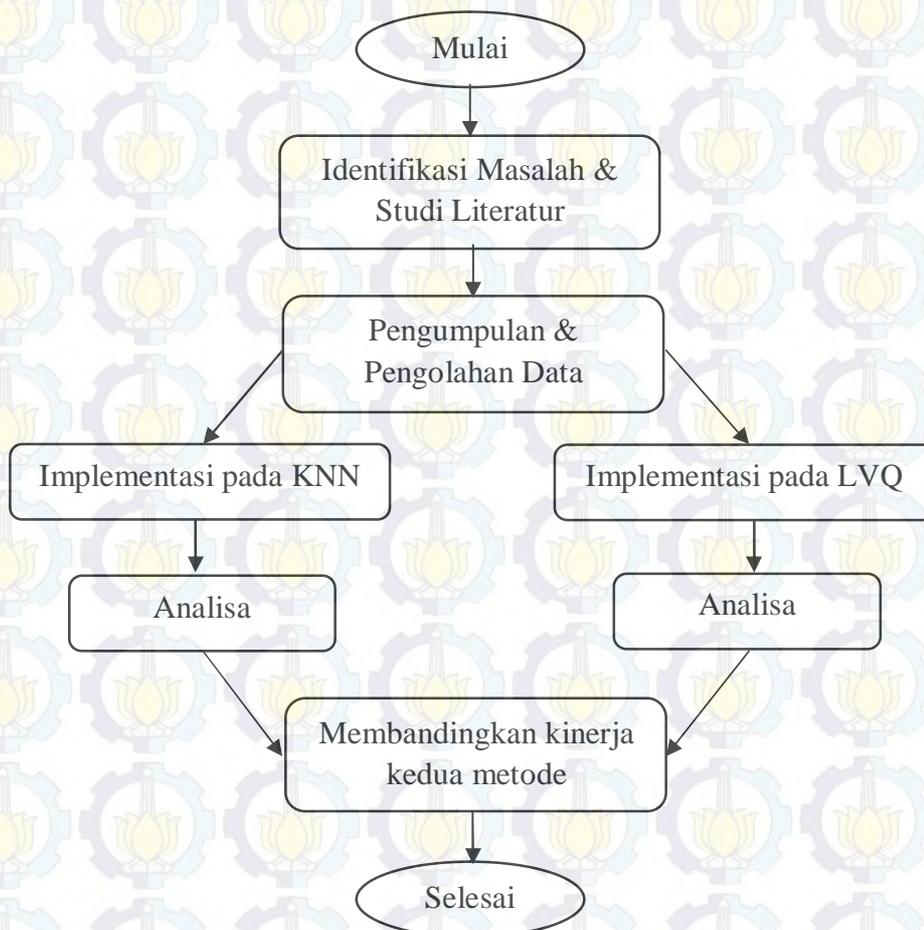
dan LVQ. Proses implementasi pengkodean untuk *training* dan *testing* data dibuat dengan menggunakan *toolbox* KNN dan LVQ yang terdapat pada *software* MATLAB R2010a. Selanjutnya hasil program dijalankan dengan menggunakan data yang telah diolah sebelumnya.

Tahap 4. Analisa dan Pembahasan

Pada tahap ini dianalisis dan dibahas hasil klasifikasi yang telah diperoleh dari *output* program dengan dilihat nilai akurasi pada kedua metode tersebut. Selanjutnya, ditarik kesimpulan dari hasil penelitian.

3.2 Diagram Alur Proses Penelitian

Kerangka dasar proses penelitian ini digambarkan dalam diagram alur yang disajikan pada Gambar 3.1 sebagai berikut:



Gambar 3.1 Diagram Alur Proses Penelitian

BAB 4 HASIL DAN PEMBAHASAN

4.1 Pengambilan dan Pengolahan Data

4.1.1 Pengambilan Data

Pengambilan data dalam tesis ini dilakukan di kantor Badan Pusat Statistik Kabupaten Situbondo. Sumber data yang digunakan adalah data rumah tangga sasaran Pendataan Program Perlindungan Sosial (PPLS) tahun 2011 dengan mengambil sampel data sebanyak 216 rumah tangga. Data rumah tangga sasaran yang diperoleh didasarkan pada data Pendataan Sosial Ekonomi (PSE) sebelumnya. PSE dimaksudkan untuk mendapatkan data kemiskinan mikro berupa rumah tangga penerima BLT (Bantuan Langsung Tunai) yang selanjutnya digunakan sebagai dasar penentuan rumah tangga sasaran. Metode pengambilan data sampel ini adalah selain mengambil data yang sudah tersedia di BPS tersebut, juga dilakukan wawancara terhadap kepala instansi untuk mengetahui gambaran secara umum faktor-faktor yang mempengaruhi kemiskinan. Penentuan rumah tangga sasaran (RTS) didasarkan pada pendekatan karakteristik rumah tangga miskin yang terdiri dari 14 kriteria/indikator kemiskinan diantaranya seperti pada Tabel 4.1 berikut:

Tabel 4.1 Kriteria Kemiskinan

Atribut	Keterangan
1	Luas lantai
2	Jenis lantai
3	Jenis dinding
4	Sumber penerangan
5	Bahan bakar memasak
6	Sumber air minum
7	Jenis jamban/WC
8	Kepemilikan aset
9	Pendapatan

10	Pendidikan
11	Pekerjaan
12	Kemampuan berobat
13	Konsumsi
14	Kemampuan membeli pakaian

4.1.2 Pengolahan Data

Pada Tabel 4.1 yang berisi kriteria kemiskinan merupakan atribut-atribut yang dijadikan variabel inputan untuk proses klasifikasi menggunakan metode KNN dan LVQ. Tabel tersebut menjelaskan atribut-atribut dataset indikator kemiskinan yang terdiri dari 14 atribut yaitu $x_1, x_2, x_3, \dots, x_{14}$. Dari 14 atribut dataset tersebut, terdapat data yang ditransformasi terlebih dahulu kedalam bentuk numerik agar dapat dijadikan inputan pada proses *training* maupun *testing*. Berikut diberikan contoh hasil transformasi data ke-1 sampai data ke-10 (contoh 10 data awal dapat dilihat pada lampiran 1) pada Tabel 4.2 berikut dengan keterangan untuk setiap variabelnya adalah:

x_1 = Luas lantai

0 = lebih dari 8 m²

1 = kurang/sama dengan 8 m²

x_2 = Jenis lantai

0 = Keramik

1 = Plester

2 = Tanah/kayu

x_3 = Jenis dinding

0 = Tembok

1 = Kayu

2 = Bambu/lainnya

x_4 = Sumber penerangan

0 = Listrik

1 = Non-listrik

x_5	= Bahan bakar memasak
0	= Gas
1	= Minyak tanah
2	= Kayu
x_6	= Sumber air bersih/minum
0	= Air PDAM
1	= Sumur
2	= Sungai
x_7	= Jenis jamban /WC
0	= Sendiri
1	= Bersama/Umum
2	= Tidak ada
x_8	= Kepemilikan aset
0	= Punya
1	= Tidak punya
x_9	= Pendapatan (total pendapatan perbulan)
x_{10}	= Pendidikan
0	= \geq SMA/Sederajat
1	= SMP/Sederajat
2	= SD/Sederajat
3	= Tidak tamat SD
x_{11}	= Pekerjaan
0	= Pegawai
1	= Wiraswasta
2	= Bangunan
3	= Petani
4	= Buruh/tidak bekerja
x_{12}	= Kemampuan berobat
x_{13}	= Konsumsi
x_{14}	= Kemampuan membeli pakaian

Tabel 4.2 Hasil Transformasi Data dalam Bentuk Numerik

No	Atribut Dataset													
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
1	1	2	1	0	2	2	2	1	352.5	1	1	0	0	0
2	0	1	1	1	1	1	1	0	378.7	2	0	0	0	0
3	0	2	1	0	2	2	2	1	235.5	2	4	0	0	0
4	1	1	1	1	1	1	1	1	220.8	1	4	1	1	0
5	0	1	1	0	1	2	1	1	361	0	0	0	0	0
6	0	1	1	1	1	1	1	0	233	2	4	0	0	1
7	0	0	0	0	0	0	0	0	354	0	0	0	1	0
8	0	1	1	0	1	1	1	1	390.6	2	1	0	1	0
9	0	2	2	0	2	2	2	1	210.9	3	4	1	1	1
10	0	0	0	1	0	0	0	0	237	0	4	0	1	0

Setelah semua data bernilai numerik, tahap berikutnya adalah dilakukan normalisasi data yaitu dengan menggunakan persamaan sebagai berikut:

$$x_* = \frac{x_{data} - x_{min}}{x_{max} - x_{min}}$$

Setelah dilakukan normalisasi diperoleh data yang nantinya digunakan sebagai variabel input seperti pada Tabel 4.3 berikut:

Tabel 4.3 Hasil Normalisasi 10 Dataset RTS

No	Atribut Dataset													
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
1	1	1	0.5	0	1	0.5	1	1	0.46	0.33	0.25	0	0	0
2	0	0.5	0.5	1	0.5	0.5	0.5	0	0.51	0.67	0	0	0	0
3	0	1	0.5	0	1	1	1	1	0.23	0.67	1	0	0	0
4	1	0.5	0.5	1	0.5	0.5	0.5	1	0.20	0.33	1	1	1	0
5	0	0.5	0.5	0	0.5	1	0.5	1	0.48	0	0	0	0	0
6	0	1	0.5	1	0.5	0.5	0.5	0	0.22	0.67	1	0	0	1
7	0	0	0	0	0	0	0	0	0.46	0	0	0	1	0
8	0	0.5	0.5	0	0.5	0.5	0.5	1	0.54	0.67	0.25	0	1	0
9	0	1	1	0	1	1	1	1	0.18	1	1	1	1	1
10	0	0	0	1	0	0	0	0	0.23	0	1	0	1	0

Tabel 4.3 menunjukkan hasil normalisasi data yang digunakan sebagai variabel input dalam proses *training* dan *testing*. Sedangkan unit *output* berupa

target kelas yang diklasifikasikan ke dalam tiga kelas kategori yaitu kelas 1 yang mengkategorikan rentan miskin, kelas 2 yang mengkategorikan miskin dan kelas 3 yang mengkategorikan sangat miskin.

4.2 Implementasi Metode

Pada tahap ini yang terpenting adalah membaca permasalahan dan menerjemahkannya ke dalam algoritma pada kedua metode yang digunakan. Pertama adalah membangun model dari set data *training*, menentukan data-data *input* dan menentukan nilai parameter, menginisialisasi bobot, *update* bobot, dan lain-lain. Kedua adalah melakukan *training* dan *testing* data-data *input* yang diperoleh dan ditransformasi sebelumnya sehingga didapatkan perbandingan keakurasiannya.

Proses implementasi dengan menggunakan kedua metode ini dibagi menjadi dua tahap. Tahap pertama adalah menentukan pembagian data *training* dan data *testing*, selanjutnya dilakukan proses *training* yang diperlukan untuk mengenali pola. Penelitian ini dilakukan dengan menggunakan jumlah kesekuruhan data sebanyak 216 data sebagai data *training* dan *testing*. Selain itu, juga dilakukan uji coba dengan jumlah data *training* yang lain. Untuk menentukan jumlah data *training* peneliti mengambil data sebanyak 75 persen dari keseluruhan data, sedangkan sisa data digunakan sebagai data *testing*. Sebelum dilakukan proses *training* perlu menentukan nilai parameter *k*, *learning rate*, jumlah iterasi dan jumlah data *training*.

Tahap kedua yaitu proses *testing* yaitu untuk menentukan kelas hasil klasifikasi data dan menguji tingkat akurasi model. Setelah dilakukan *testing*, selanjutnya dihitung persentase keakurasian hasil klasifikasi. Penentuan keakurasian ini adalah untuk mengetahui tingkat akurasi klasifikasi dari kedua metode. Dalam hal ini dapat ditentukan dengan persamaan:

$$\text{Akurasi} = \frac{\text{Jumlah data benar}}{\text{Jumlah data uji}} \times 100\%$$

4.2.1 Implementasi Metode KNN

Pengklasifikasian dengan metode *K-Nearest Neighbor* (KNN) dibagi menjadi dua proses, yaitu proses *training* dan *testing*. Pada proses *training* KNN menggunakan data sampel yang terdiri dari variabel-variabel dan kelas target yang diambil dari banyaknya kelas klasifikasi sebagai *input*. Sedangkan pada proses *testing*, KNN menggunakan nilai perhitungan jarak untuk atribut-atribut dari setiap data uji terhadap seluruh atribut pada data *training* dengan rumus jarak eucliden. Selanjutnya dihasilkan sejumlah nilai *k* tetangga terdekat, dimana hasil dari data uji yang baru diklasifikasikan berdasarkan mayoritas dari kelas kategori pada *k* tetangga terdekat. Berikut diberikan contoh implementasi metode KNN dengan menggunakan 6 dataset dan kelas klasifikasi yang masing-masing diwakili oleh 3 dataset.

Tabel 4.4 Contoh Data *Training* untuk Proses KNN

No	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	Kls
1	1	1	0.5	0	1	0.5	1	1	0.46	0.33	0.25	0	0	0	2
2	0	0.5	0.5	1	0.5	0.5	0.5	0	0.51	0.67	0	0	0	0	1
3	0	1	0.5	0	1	1	1	1	0.23	0.67	1	0	0	0	3
4	1	0.5	0.5	1	0.5	0.5	0.5	1	0.20	0.33	1	1	1	0	3
5	0	0.5	0.5	0	0.5	1	0.5	1	0.48	0	0	0	0	0	1
6	0	1	0.5	1	0.5	0.5	0.5	0	0.22	0.67	1	0	0	1	2
7	0	0	0	0	0	0	0	0	0.46	0	0	0	1	0	1

Dari Tabel 4.4 yang terdiri dari 7 data dengan 6 data sebagai data *training* dan 14 variabel yaitu x_1 sampai x_{14} pada setiap data serta 3 kelas target yang mewakili kelas klasifikasi. Selanjutnya ditentukan nilai *k* tetangga terdekat dan sampel data baru sebagai data uji. Pada permasalahan ini, dipilih nilai $k = 3$ dan data uji data ke-7 adalah $(0,0,0,0,0,0,0,0,0,0.46,0,0,0,1,0)$ yang berupa vektor. Tahap berikutnya dihitung jarak data *training* dengan data baru menggunakan persamaan jarak eucliden diperoleh nilai sebagai berikut:

Untuk data *training* 1 dengan data baru:

$$D_{1,7} = \sqrt{(1-0)^2 + (1-0)^2 + (0.5-0)^2 + (0-0)^2 + (1-0)^2 + (0.5-0)^2 + (1-0)^2 + (1-0)^2 + (0.46-0.46)^2 + (0.33-0)^2}$$

$$+ (0.25 - 0)^2 + (0 - 0)^2 + (0 - 1)^2 + (0 - 0)^2 = 2.5829$$

Untuk data *training* 2 dengan data baru:

$$D_{2,7} = \frac{\sqrt{(0 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + (1 - 0)^2 + (0.5 - 0)^2}}{\sqrt{(0.5 - 0)^2 + (0.5 - 0)^2 + (0 - 0)^2 + (0.51 - 0.46)^2 + (0.67 - 0)^2}} \\ \frac{\sqrt{(0 - 0)^2 + (0 - 0)^2 + (0 - 1)^2 + (0 - 0)^2}}{\sqrt{(0 - 0)^2 + (0 - 0)^2 + (0 - 1)^2 + (0 - 0)^2}} = 1.9239$$

Untuk data *training* 3 dengan data baru:

$$D_{3,7} = \frac{\sqrt{(0 - 0)^2 + (1 - 0)^2 + (0.5 - 0)^2 + (0 - 0)^2 + (1 - 0)^2}}{\sqrt{(1 - 0)^2 + (1 - 0)^2 + (1 - 0)^2 + (0.23 - 0.46)^2 + (0.67 - 0)^2}} \\ \frac{\sqrt{(1 - 0)^2 + (1 - 0)^2 + (1 - 1)^2 + (0 - 0)^2}}{\sqrt{(1 - 0)^2 + (1 - 0)^2 + (1 - 1)^2 + (0 - 0)^2}} = 2.7842$$

Untuk data *training* 4 dengan data baru:

$$D_{4,7} = \frac{\sqrt{(1 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + (1 - 0)^2 + (0.5 - 0)^2}}{\sqrt{(0.5 - 0)^2 + (0.5 - 0)^2 + (1 - 0)^2 + (0.2 - 0.46)^2 + (0.33 - 0)^2}} \\ \frac{\sqrt{(1 - 0)^2 + (1 - 0)^2 + (1 - 1)^2 + (0 - 0)^2}}{\sqrt{(1 - 0)^2 + (1 - 0)^2 + (1 - 1)^2 + (0 - 0)^2}} = 2.5351$$

Untuk data *training* 5 dengan data baru:

$$D_{5,7} = \frac{\sqrt{(0 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + (0 - 0)^2 + (0.5 - 0)^2}}{\sqrt{(1 - 0)^2 + (0.5 - 0)^2 + (1 - 0)^2 + (0.48 - 0.46)^2 + (0 - 0)^2}} \\ \frac{\sqrt{(0 - 0)^2 + (0 - 0)^2 + (0 - 1)^2 + (0 - 0)^2}}{\sqrt{(0 - 0)^2 + (0 - 0)^2 + (0 - 1)^2 + (0 - 0)^2}} = 2.0001$$

Untuk data *training* 6 dengan data baru:

$$D_{6,7} = \frac{\sqrt{(0 - 0)^2 + (1 - 0)^2 + (0.5 - 0)^2 + (1 - 0)^2 + (0.5 - 0)^2}}{\sqrt{(0.5 - 0)^2 + (0.5 - 0)^2 + (0 - 0)^2 + (0.22 - 0.46)^2 + (0.67 - 0)^2}} \\ \frac{\sqrt{(1 - 0)^2 + (0 - 0)^2 + (0 - 1)^2 + (1 - 0)^2}}{\sqrt{(1 - 0)^2 + (0 - 0)^2 + (0 - 1)^2 + (1 - 0)^2}} = 2.5508$$

Setelah jarak data uji ke setiap data *training* diperoleh, tahap selanjutnya adalah mengurutkan setiap nilai jarak tersebut dari yang terkecil (terdekat). Kemudian dilakukan pemeriksaan setiap kelas untuk $k = 3$ tetangga terdekat.

Hasil pengurutan jarak dapat dilihat pada Tabel 4.5 berikut:

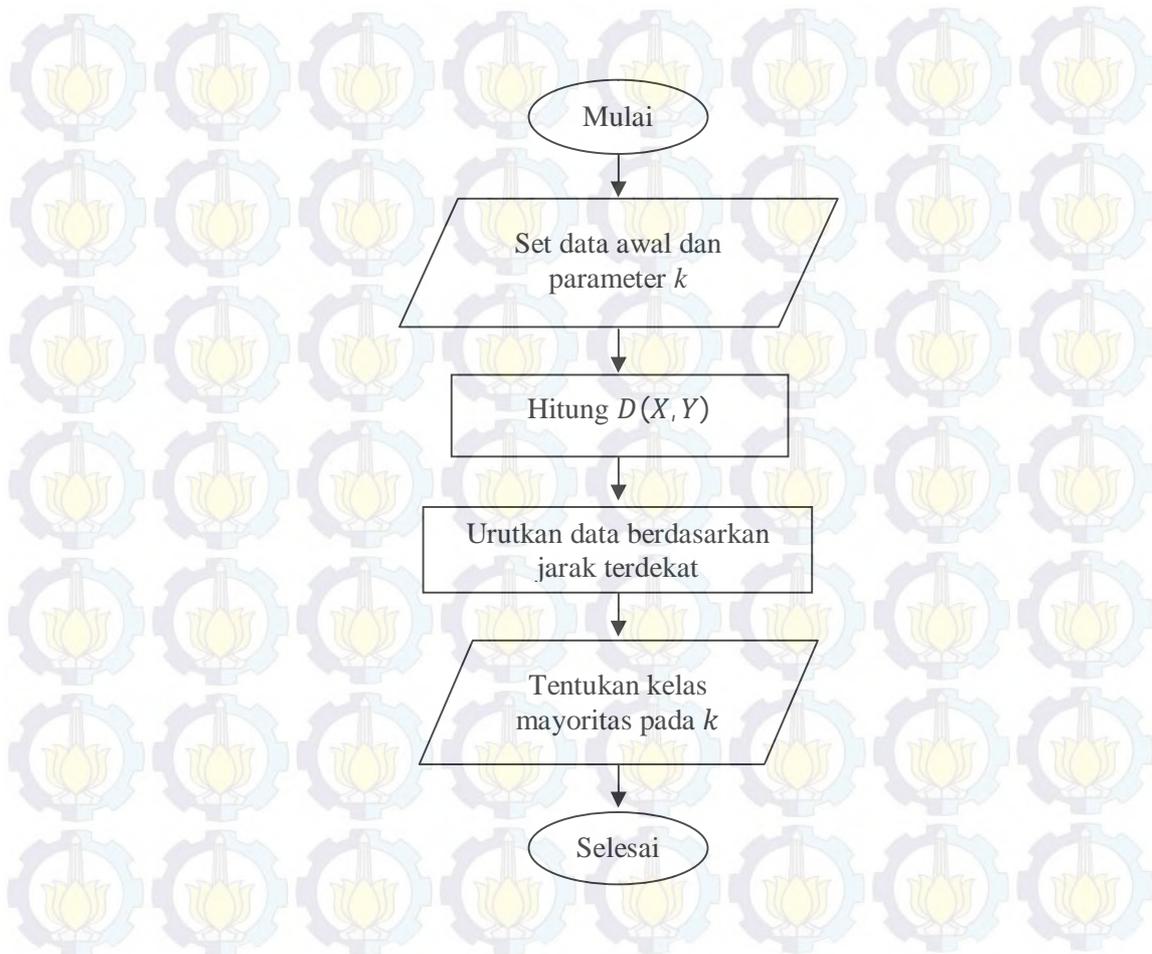
Tabel 4.5 Hasil Urutan Jarak

Data ke-	Nilai Jarak	Kelas	Urutan
1	2.5829	2	5
2	1.9239	1	1
3	2.7842	3	6
4	2.5351	3	3
5	2.0001	1	2
6	2.5508	2	4

Pada Tabel 4.5 diperoleh urutan jarak dari yang terkecil (terdekat) berturut-turut yaitu data ke-2, 5, 4, 6, 1 dan ke-3. Selanjutnya dilakukan pemeriksaan terhadap kelas sesuai dengan urutan jarak terdekat diperoleh urutan kelas yaitu 1, 1, 3, 2, 2 dan 3. Langkah terakhir yaitu menentukan kelas dari data baru/uji yang merupakan kelas mayoritas tetangga terdekat. Pada contoh permasalahan ini dengan parameter $k = 3$, diperoleh kelas hasil klasifikasi untuk data uji (*testing*) yaitu pada kelas 1 yang merupakan mayoritas dari target kelas yang dipilih.

a. Proses *Training* dan *Testing* Metode KNN

Sebelum dilakukan proses *training* perlu menentukan nilai-nilai parameter k , jumlah data *training*, kelas data *training* dan data *testing* sebagai langkah awal penyimpanan vektor-vektor atribut dan kelas klasifikasi dari data. Tujuan dari proses *testing* adalah mendapatkan output target kelas yang hasilnya dicocokkan dengan kelas klasifikasi pada data *testing* untuk mengetahui presentasi keakurasiannya. Berikut adalah diagram alir proses KNN.



Gambar 4.1 Diagram Alir Proses KNN

4.2.2 Pelatihan KNN dengan *Toolbox* Matlab

Pada bagian ini disajikan pelatihan algoritma *K-Nearest Neighbor* dengan perintah-perintah Matlab. Pada tahapan inialisasi data, diperlukan suatu penyimpanan vektor-vektor atribut dan kelas klasifikasi dari data *training* dan data *testing*. Setelah proses penyimpanan, selanjutnya dilakukan proses klasifikasi untuk menentukan kelas dari data testing dengan menentukan nilai k tetangga terdekat dan menghitung jarak data testing dengan data training menggunakan *eucliden distance*.

```

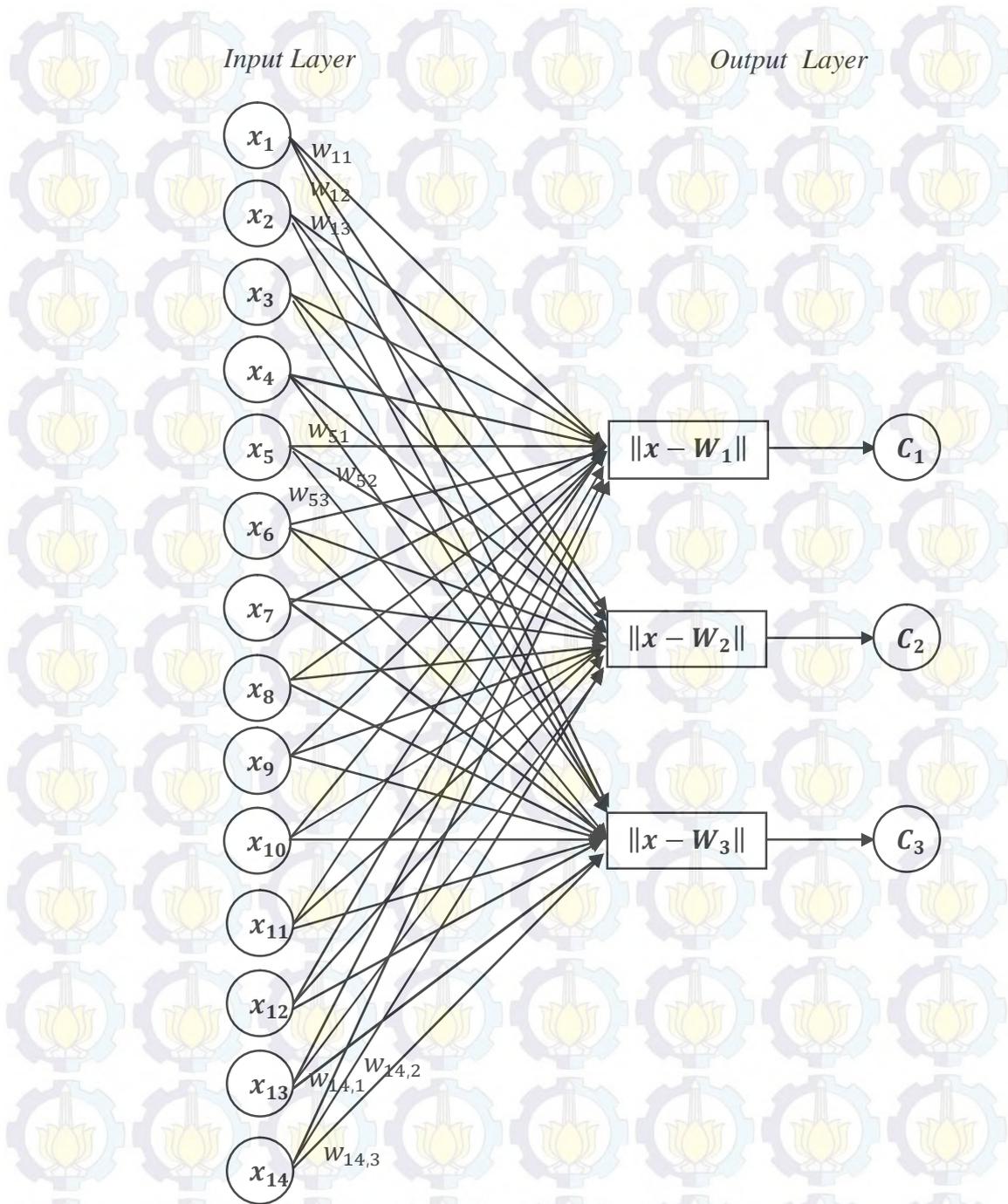
dataTraining = xlsread('data.xlsx', 1, 'lengthvektor');
dataTesting  = xlsread('data.xlsx', 1, 'lengthvektor');
Cls_KNN     = xlsread('data.xlsx', 1, 'lengthclass');
k           = 'nilai k tetangga terdekat';
Class      = knnclassify(dataSample,dataTraining,
                        Cls_KNN,k,eucliden);
  
```

Setelah diperoleh hasil kelas klasifikasi, dilakukan perbandingan dengan kelas klasifikasi awal dari data *testing*. Selanjutnya dilakukan perhitungan nilai akurasi antara hasil kelas klasifikasi dari hasil program dengan kelas awal data *testing*.

```
j=length(cls_awal);  
for i=1:j,  
    k(i)=0;  
    if cls_awal(i)~=Class(i),  
        k(i)=k(i)+0;  
    else  
        k(i)=k(i)+1;  
    end;  
end;  
  
jum=sum(k);  
disp(strcat('Data Dikenali=',int2str(jum)))  
disp(strcat('Data Uji=',int2str(j)))  
b=(jum/j)*100;  
b1=num2str(b);  
disp(strcat('Tingkat Akurasi=',b1))
```

4.2.3 Implementasi Metode LVQ

Learning Vector Quantization (LVQ) merupakan jaringan *single layer* yang terdiri atas dua lapisan *input* dan *output*. Lapisan *input* terdiri dari 8 unit *input* yang diambil dari variabel kriteria kemiskinan, sedangkan unit *output* terdiri atas dua unit *output* yang diambil dari banyaknya kelas klasifikasi. Arsitektur jaringan LVQ dalam penelitian ini adalah sebagai berikut:



Gambar 4.2 Arsitektur Jaringan LVQ Permasalahan Tingkat Kemiskinan

Deskripsi dari Gambar 4.2 adalah sebagai berikut:

x_1 sampai x_{14} : adalah vektor training dari 14 variabel dengan;

x_1 = Luas lantai

x_8 = Kepemilikan aset

x_2 = Jenis lantai

x_9 = Pendapatan

x_3 = Jenis dinding x_{10} = Pendidikan
 x_4 = Sumber penerangan x_{11} = Pekerjaan
 x_5 = Bahan bakar memasak x_{12} = Kemampuan berobat
 x_6 = Sumber air minum x_{13} = Konsumsi
 x_7 = Jenis jamban /WC x_{14} = Kemampuan membeli

$w_{1,1}$ sampai $w_{14,3}$: adalah vektor bobot untuk unit *output* ke-1, 2 dan 3.
 C_1, C_2 dan C_3 : adalah kategori/kelas hasil komputasi oleh unit *output* yang terdiri dari tiga kelas yaitu;

C_1 = kelas 1 mewakili kategori rentan miskin
 C_2 = kelas 2 mewakili kategor miskin
 C_3 = kelas 3 mewakili kategor sangat miskin

$\|x - w_j\|_{j=1,2,3}$: adalah jarak Euclidean antara vektor *input* dengan vektor bobot untuk unit *output*.

Dalam menentukan bobot awal pada metode LVQ digunakan pola-pola yang sudah ada, yang diambil dari salah satu data yang mewakili kelas masing-masing. Selanjutnya bobot tersebut diubah (*di-update*) tergantung dari kelas vektor masukan sesuai dengan kelas yang dinyatakan sebagai neuron pemenang. Jika sesuai, maka vektor bobot *di-update* sehingga jaraknya semakin dekat dengan vektor masukan. Bobot awal ini diambil dari data ke-1, 2 dan ke-3 (lampiran 1), yang harus dibawa ke dalam bentuk vektor. Berikut diberikan contoh implementasi metode LVQ dengan menggunakan 6 dataset yang masing-masing mewakili kelas kategori. Data tersebut disajikan dalam Tabel 4.6 berikut:

Tabel 4.6 Contoh Data *Training* untuk Proses LVQ

No	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	Kls
1	1	1	0.5	0	1	0.5	1	1	0.46	0.33	0.25	0	0	0	2
2	0	0.5	0.5	1	0.5	0.5	0.5	0	0.51	0.67	0	0	0	0	1
3	0	1	0.5	0	1	1	1	1	0.23	0.67	1	0	0	0	3
4	1	0.5	0.5	1	0.5	0.5	0.5	1	0.20	0.33	1	1	1	0	3
5	0	0.5	0.5	0	0.5	1	0.5	1	0.48	0	0	0	0	0	1
6	0	1	0.5	1	0.5	0.5	0.5	0	0.22	0.67	1	0	0	1	2

Dari 6 dataset pada Tabel 4.6 tersebut, diambil tiga data yang merupakan vektor referensi yang mewakili masing-masing kelas untuk menginisialisasi bobot awal diantaranya data ke-1, 2 dan data ke-3. Karena harus berupa vektor, maka bentuk bobot awal dapat dituliskan sebagai berikut:

$$w_1 = (1, 1, 0.5, 0, 1, 0.5, 1, 1, 0.46, 0.33, 0.25, 0, 0, 0)$$

$$w_2 = (0, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 0, 0.51, 0.67, 0, 0, 0, 0)$$

$$w_3 = (1, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 1, 0.2, 0.33, 1, 1, 1, 0)$$

Sedangkan tiga vektor untuk data ke-4, 5 dan data ke-6 sebagai data *training*. Di bawah ini ditunjukkan satu iterasi sebagai langkah proses *training* sebagai berikut:

1. Inisialisasi bobot (dari vektor referensi)

$$w_1 = (1, 1, 0.5, 0, 1, 0.5, 1, 1, 0.46, 0.33, 0.25, 0, 0, 0)$$

$$w_2 = (0, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 0, 0.51, 0.67, 0, 0, 0, 0)$$

$$w_3 = (0, 1, 0.5, 0, 1, 1, 1, 1, 0.23, 0.67, 1, 0, 0, 0)$$

Inisialisasi *learning rate* $\alpha = 0.1$

2. Mulai komputasi

3. Untuk vektor *input* $x = (1, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 1, 0.2, 0.33, 1, 1, 1, 0)$ dengan $T = 3$ kerjakan 4-5.

4. $J = 1$

5. Karena $T = 3$ tetapi $C_1 = 1$ *update* w_1 sebagai berikut:

$$w_1 = (1, 1, 0.5, 0, 1, 0.5, 1, 1, 0.46, 0.33, 0.25, 0, 0, 0)$$

$$- 0.1[(1, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 1, 0.2, 0.33, 1, 1, 1, 0)$$

$$- (1, 1, 0.5, 0, 1, 0.5, 1, 1, 0.46, 0.33, 0.25, 0, 0, 0)]$$

$$= (1, 1.05, 0.5, -0.1, 1.05, 0.5, 1.05, 1, 0.486, 0.33, 0.175, -0.1, -0.1, 0)$$

3. Untuk vektor *input* $x = (0, 0.5, 0.5, 0, 0.5, 1, 0.5, 1, 0.48, 0, 0, 0, 0, 0)$ dengan $T = 1$ kerjakan 4-5

4. $J = 1$

5. Karena $T = 1$ dan $C_1 = 1$ *update* w_1 sebagai berikut:

$$w_1 = (1, 1.05, 0.5, -0.1, 1.05, 0.5, 1.05, 1, 0.486, 0.33, 0.175, -0.1, -0.1, 0)$$

$$+ 0.1[(0, 0.5, 0.5, 0, 0.5, 1, 0.5, 1, 0.48, 0, 0, 0, 0, 0)$$

$$- (1, 1.05, 0.5, -0.1, 1.05, 0.5, 1.05, 1, 0.486, 0.33, 0.175, -0.1, -0.1, 0)]$$

$$= (0.9, 0.995, 0.5, -0.09, 0.995, 0.55, 0.995, 1, 0.4854, 0.297, 0.1575, -0.09, -0.09, 0)$$

3. Untuk vektor *input* $x = (0, 1, 0.5, 1, 0.5, 0.5, 0.5, 0, 0.22, 0.67, 1, 0, 0, 1)$ dengan $T = 2$ kerjakan 4-5

4. $J = 2$

5. Karena $T = 2$ dan $C_2 = 2$ *update* w_2 sebagai berikut:

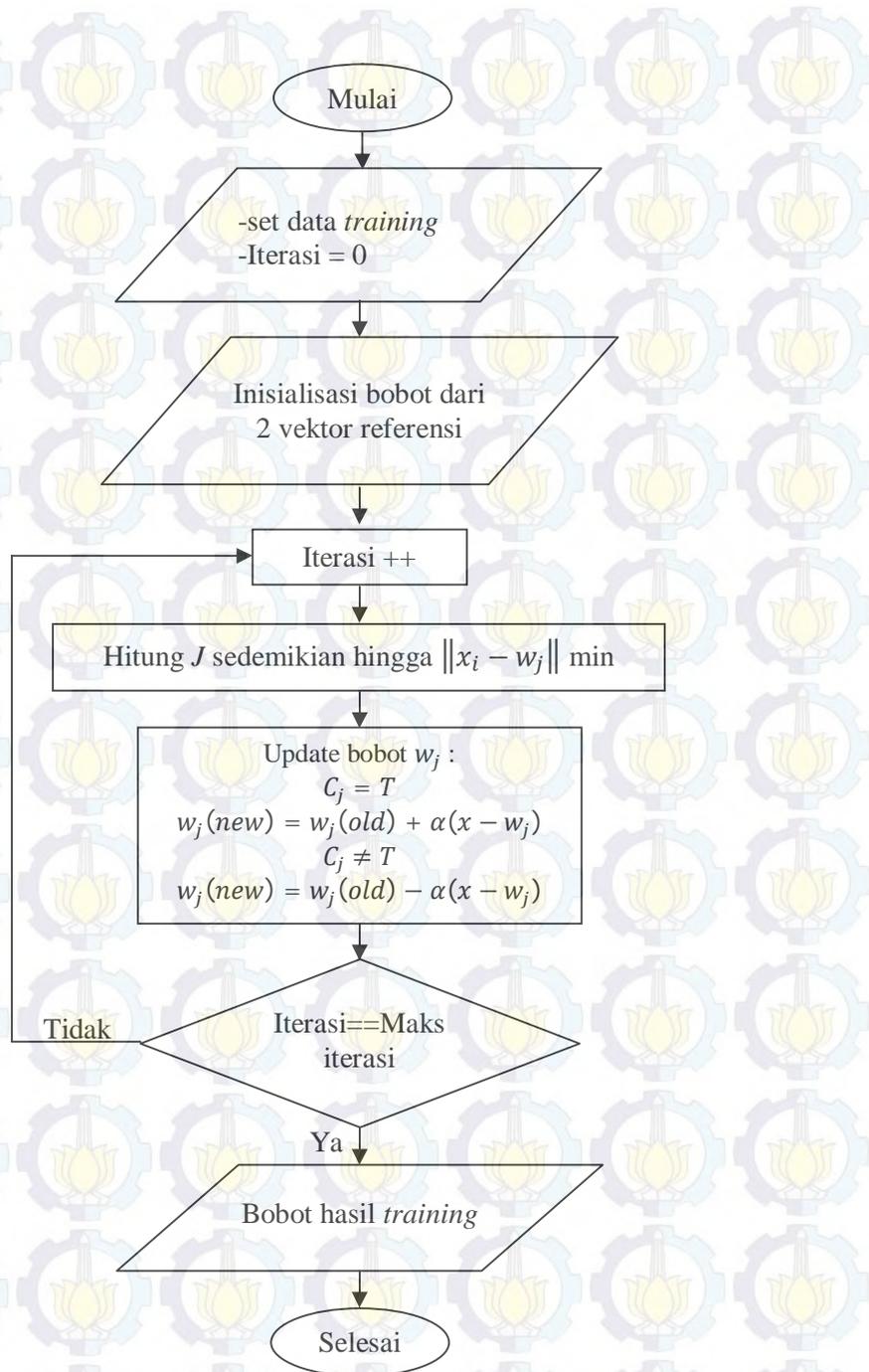
$$\begin{aligned} w_2 &= (0, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 0, 0.51, 0.67, 0, 0, 0, 0) \\ &+ 0.1[(0, 1, 0.5, 1, 0.5, 0.5, 0.5, 0, 0.22, 0.67, 1, 0, 0, 1) \\ &- (0, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 0, 0.51, 0.67, 0, 0, 0, 0)] \\ &= (0, 0.55, 0.5, 1, 0.5, 0.5, 0.5, 0, 0.481, 0.67, 0.1, 0, 0, 0.1) \end{aligned}$$

6. *Training* telah lengkap dilakukan untuk 1 iterasi

Diperoleh w_1, w_2 dan w_3 yang baru tersebut digunakan untuk proses berikutnya. Kondisi stop dipenuhi.

a. Proses *Training* LVQ

Sebelum dilakukan proses *training* perlu menentukan nilai-nilai parameter *learning rate*, jumlah iterasi (*epoch*) dan jumlah data *training*. Proses *training* bertujuan untuk memperoleh *output* berupa nilai bobot-bobot yang optimal dari *training* terakhir yang kemudian disimpan untuk selanjutnya digunakan dalam proses *testing*. Berikut adalah diagram alir proses *training* LVQ.

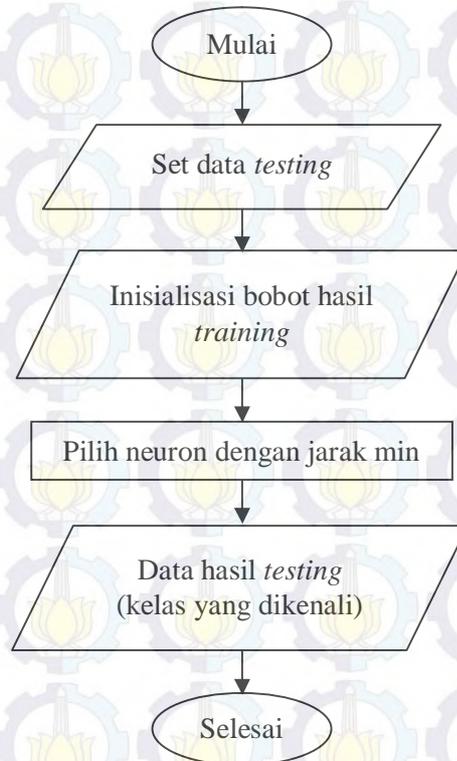


Gambar 4.3 Diagram Alir Proses *Training* LVQ

b. Proses *Testing* LVQ

Algoritma yang digunakan pada proses *testing* hampir sama dengan algoritma *training*, tetapi bobot awal yang digunakan adalah bobot akhir dari hasil *training*. Tujuan dari proses *testing* adalah mendapatkan output jaringan yang

hasilnya dicocokkan dengan target kelas klasifikasi dan untuk mengetahui presentasi keakurasiannya. Berikut ini adalah diagram alir proses *testing* LVQ.



Gambar 4.4 Diagram Alir Proses *Testing* LVQ

4.2.4 Pelatihan LVQ dengan *Toolbox* Matlab

Pada bagian ini akan disajikan pelatihan algoritma LVQ dengan perintah-perintah MatLab. Pada tahapan inisialisasi jaringan, diperlukan inisialisasi data *training*, target kelas, nilai *learning rate* dan bobot awal yang digunakan untuk mendapatkan nilai bobot akhir pada proses *training*.

```

T = ind2vec(target');
LearningRate = 0.01;
net = newlvq(datatrain',3,[.4 .3 .3],
            LearningRate);
net.IW{1,1} = % inisialisasi bobot awal
net.trainParam.epochs = 100; % banyak iterasi
net = train(net, datatrain',T);
  
```

Setelah dilakukan *training*, diperoleh bobot-bobot akhir yang nantinya digunakan untuk melakukan pengujian terhadap data *testing* yang telah ditentukan. Proses penentuan kelas klasifikasi dari data *testing* dilakukan sebagai berikut:

```
net.IW{1,1}
hasil_LVQ   = sim(net,datatesting');
cls_LVQ     = vec2ind(hasil_LVQ);
```

Setelah diperoleh hasil kelas klasifikasi dari program, dilakukan perbandingan dengan kelas klasifikasi awal dari data *testing*. Selanjutnya dilakukan perhitungan nilai akurasi antara hasil kelas klasifikasi dari hasil program dengan kelas awal data *testing* sebagai berikut:

```
j=length(classTest);
for i=1:j;
    k(i)=0;
    if classTest(i)~=cls_LVQ(i),
        k(i)=k(i)+0;
    else
        k(i)=k(i)+1;
    end;
end;
jum=sum(k);
disp(strcat('Data Dikenali=',int2str(jum)))
disp(strcat('Data Uji=',int2str(j)))
b=(jum/j)*100;
b1=num2str(b);
disp(strcat('Tingkat Akurasi=',b1))
```

4.3 Analisa dan Pembahasan

Tujuan penelitian ini adalah untuk mencari perbandingan tingkat akurasi dari metode KNN dan LVQ. Di bawah ini akan diberikan nilai keakurasian dari hasil klasifikasi dengan menggunakan kedua metode tersebut.

4.3.1 Metode KNN

Keakurasian dari hasil uji coba menggunakan metode KNN ditinjau dari dua parameter yaitu k tetangga terdekat dan jumlah data *training*. Uji coba metode KNN dilakukan dengan cara menentukan nilai k dan jumlah data *training* yang digunakan. Hasil akurasi klasifikasi yang ditinjau dari parameter $k = 3, 4, \dots, 10$ untuk jumlah data *training* dan *testing* sebanyak 216 data disajikan pada Tabel 4.7 sebagai berikut:

Tabel 4.7 Hasil Akurasi Klasifikasi Metode KNN dengan 216 Data *Testing* yang Ditinjau dari Parameter k

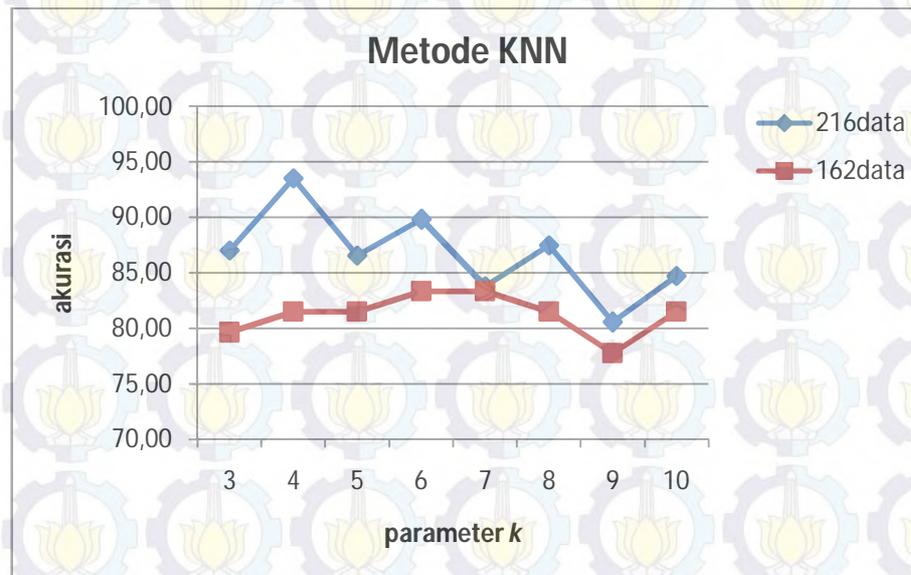
Parameter	Akurasi (%)	<i>Elapsed time</i> (s)
$k = 3$	87.037	6.21
$k = 4$	93.5185	4.09
$k = 5$	86.5741	3.93
$k = 6$	89.8148	3.95
$k = 7$	83.7963	3.91
$k = 8$	87.5	3.85
$k = 9$	80.5556	3.90
$k = 10$	84.7222	3.89

Tabel 4.7 menunjukkan hasil uji coba dengan menggunakan metode KNN. Uji coba metode ini dilakukan dengan menggunakan nilai parameter k yang diambil adalah $k = 3, 4, \dots, 10$. Berdasarkan nilai k yang digunakan, hasil akurasi tertinggi terlihat pada nilai $k = 4$ yaitu 93.5185%. Uji coba selanjutnya dengan menggunakan data *training* sebanyak 162 data dan sisanya sebanyak 54 data digunakan sebagai data *testing*. Hasil keakurasian dari uji coba yang ditinjau dari parameter $k = 3, 4, \dots, 10$ untuk 54 data *testing* dapat dilihat pada Tabel 4.8 berikut :

Tabel 4.8 Hasil Akurasi Klasifikasi Metode KNN dengan 54 Data *Testing* yang Ditinjau dari Parameter k

Parameter	Akurasi (%)	<i>Elapsed time</i> (s)
$k = 3$	79.6296	4.36
$k = 4$	81.4815	4.09
$k = 5$	81.4815	4.08
$k = 6$	83.3333	4.06
$k = 7$	83.3333	3.93
$k = 8$	81.4815	3.69
$k = 9$	77.7778	3.91
$k = 10$	81.4815	3.86

Tabel 4.8 menunjukkan hasil uji coba dengan menggunakan metode KNN. Uji coba metode ini dilakukan dengan menggunakan nilai parameter k yang diambil adalah $k = 3, 4, \dots, 10$ dengan jumlah data *training* sebanyak 162 data. Selanjutnya hasil uji coba dengan menggunakan metode KNN tersebut disajikan dalam bentuk grafik pada Gambar 4.3 berikut:



Gambar 4.5 Grafik Keakurasian Metode KNN ditinjau dari Parameter k

Gambar 4.5 menunjukkan tingkat akurasi dari metode KNN dengan menggunakan jumlah data *training* masing-masing sebanyak 216 dan 162 data. Grafik diatas menunjukkan akurasi tertinggi yaitu dengan parameter $k = 4$ dengan jumlah data *training* sebanyak 216 data. Terlihat pula bahwa pada jumlah data training sebanyak 216 data, nilai akurasi mengalami penurunan untuk setiap parameter $k = 5, 7$ dan 9 .

4.3.2 Metode LVQ

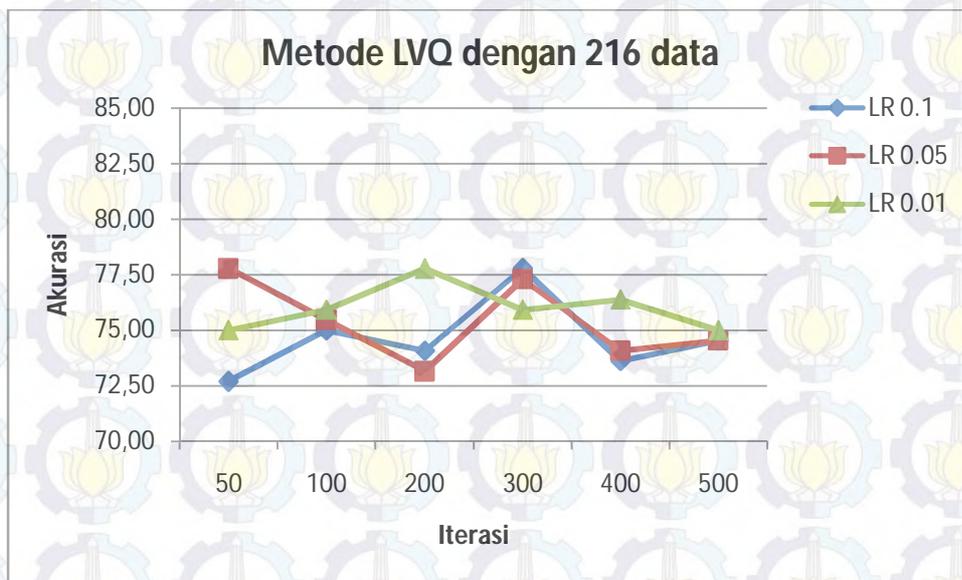
Keakurasian dari hasil uji coba menggunakan metode LVQ ditinjau dari parameter *learning rate*, jumlah iterasi dan jumlah data *training*. Uji coba metode LVQ dilakukan dengan cara mengubah nilai *learning rate* yang digunakan yaitu 0.01, 0.05 dan 0.1 dengan jumlah iterasi yang digunakan dari 50 sampai 500 iterasi. Uji coba pertama dilakukan dengan menggunakan jumlah data *training*

dan *testing* sebanyak 216 data. Hasil keakurasian klasifikasi dapat dilihat pada Tabel 4.9 berikut ini.

Tabel 4.9 Hasil Akurasi Metode LVQ dengan 216 Data *Testing* yang Ditinjau dari Jumlah Iterasi dan *Learning rate*

Iterasi	<i>Learning rate</i> (α)		
	0.1	0.05	0.01
50	72.6852	77.7778	75
100	75	75.463	75.9259
200	74.0741	73.1481	77.7778
300	77.7778	77.3148	75.9259
400	73.6111	74.0741	76.3889
500	74.537	74.537	75

Tabel 4.9 menunjukkan hasil uji coba menggunakan metode LVQ. Uji coba dilakukan dengan cara mengubah nilai *learning rate* yang diambil adalah 0.01, 0.05 dan 0.1 serta jumlah iterasi antara 50 sampai 500 iterasi. Hasil akurasi pada Tabel 4.9 didapat dari uji coba *running* program dengan *mentraining* data sebanyak 216 data dengan menggunakan *learning rate* = 0.1, 0.05 dan 0.01. Berikut adalah grafik nilai keakurasian menggunakan metode LVQ untuk jumlah data *training* sebanyak 216 data yang disajikan pada Gambar 4.6 di bawah ini.



Gambar 4.6 Grafik Keakurasian Metode LVQ ditinjau dari Jumlah Iterasi

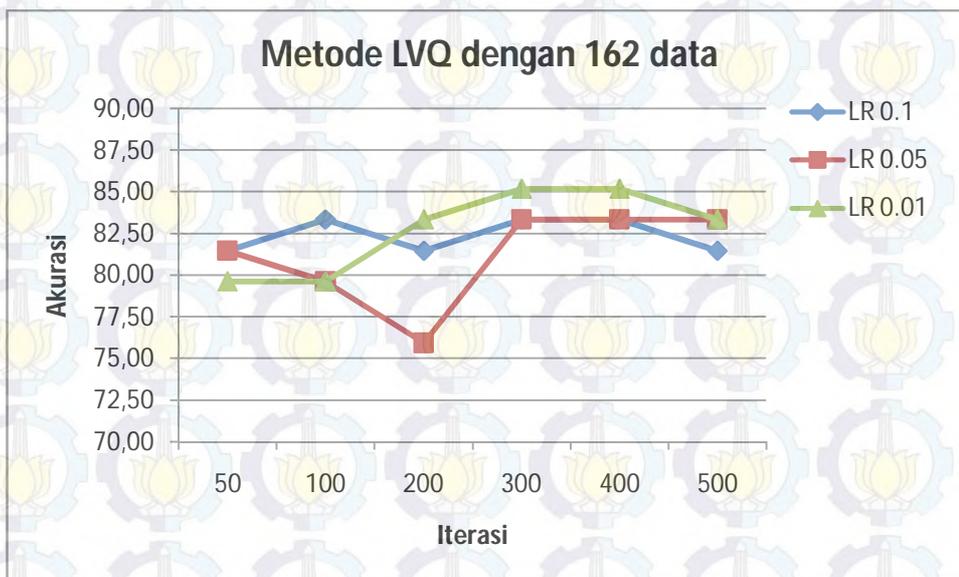
Gambar 4.6 menunjukkan tingkat akurasi dari metode LVQ dengan menggunakan jumlah data *training* sebanyak 216 data. Grafik diatas menunjukkan akurasi dengan *learning rate* 0.1, 0.05 dan 0.01 pada iterasi antara 50 sampai 500 iterasi. Pada grafik di atas terlihat bahwa nilai akurasi untuk *learning rate* 0.01 mengalami penerunan ketika jumlah iterasi lebih dari 200 iterasi. Sedangkan untuk *learning rate* 0.1 dan 0.05 nilai keakurasiannya tidak stabil pada setiap jumlah iterasi yang digunakan.

Uji coba selanjutnya dengan metode LVQ dilakukan menggunakan data *training* sebanyak 162 data dan sisa data sebanyak 54 data digunakan sebagai data *testing*. Hasil keakurasian klasifikasi dengan 162 data *training* dapat dilihat pada Tabel 4.10 berikut:

Tabel 4.10 Hasil Akurasi Metode LVQ dengan 54 Data *Testing* yang Ditinjau dari Jumlah Iterasi dan *Learning rate*

Iterasi	<i>Learning rate</i> (α)		
	0.1	0.05	0.01
50	81.4815	81.4815	79.6296
100	83.3333	79.6296	79.6296
200	81.4815	75.9259	83.3333
300	83.3333	83.3333	85.1852
400	83.3333	83.3333	85.1852
500	81.4815	83.3333	83.3333

Tabel 4.10 memperlihatkan akurasi hasil klasifikasi menggunakan metode LVQ menggunakan 162 data *training* dan 54 data *testing*. Uji coba dilakukan dengan cara mengubah nilai *learning rate* yang diambil adalah 0.01, 0.05 dan 0.1 serta jumlah iterasi antara 50 sampai 500 iterasi. Hasil akurasi pada Tabel 4.10 didapat dari uji coba *running* program dengan *mentraining* data sebanyak 162 data dengan menggunakan *learning rate* = 0.1, 0.05 dan 0.01. Berikut juga disajikan grafik nilai keakurasian menggunakan metode LVQ untuk jumlah data *training* sebanyak 162 data pada Gambar 4.7 di bawah ini :



Gambar 4.7 Grafik Keakurasian Metode LVQ ditinjau dari Jumlah Iterasi

Gambar 4.7 menunjukkan tingkat akurasi dari metode LVQ dengan menggunakan jumlah data *training* sebanyak 162 data. Grafik diatas menunjukkan akurasi dengan *learning rate* 0.1, 0.05 dan 0.01 pada iterasi antara 50 sampai 500 iterasi. Pada grafik di atas terlihat bahwa nilai akurasi untuk *learning rate* 0.01 mengalami kenaikan yang kosntan ketika jumlah iterasi lebih dari 100 iterasi. Namun ketika jumlah iterasi lebih dari 300 iterasi, nilai keakurasiannya semakin kecil. Jika dibanding dengan parameter learning rate yang lain, nilai learning rate 0.01 menghasilkan nilai akurasi yang cukup baik.

4.4 Perbandingan Metode KNN dan LVQ

Metode KNN dan LVQ merupakan metode yang digunakan dalam penyelesaian permasalahan klasifikasi dengan algoritma yang berbeda. KNN merupakan metode untuk melakukan klasifikasi terhadap obyek berdasarkan data pembelajaran yang jaraknya paling dekat dengan obyek tersebut, yaitu mencari jarak terdekat antara data yang dievaluasi dengan k tetangga (*neighbor*) terdekatnya dalam data pelatihan. Sedangkan LVQ merupakan jaringan *single layer* yang mempunyai lapisan *input* dan *output*. Dalam jaringan ini setiap neuron yang terdapat dalam lapisan *input* selalu terhubung dengan setiap neuron yang terdapat pada lapisan *output*.

Dalam perbandingan dua metode, parameter yang digunakan dalam proses *training* berdeda. Hal ini terlihat pada metode KNN digunakan parameter k tetangga tedekat, sedangkan pada LVQ digunakan parameter *learning rate* dan jumlah iterasi. Dari hasil uji coba, perbandingan dua metode ini didasarkan pada hasil klasifikasi dengan jumlah data *training* yang digunakan dan untuk selanjutnya diambil nilai rata-rata akurasi tertinggi pada setiap parameter yang ditentukan guna memperoleh hasil perbandingan dari kedua metode.

Setelah diperoleh parameter-parameter dan jumlah iterasi yang memberikan nilai akurasi tertinggi pada penelitian ini. Uji coba selanjutnya dilakukan pada jumlah data *training* yang sama yaitu sebanyak 216 data dengan parameter-parameter $k = 4$, learning rate sebesar 0.01 dan 300 iterasi. Hasil yang diperoleh dari proses *training* dan *testing* dengan menggunakan jumlah data sebanyak 216 data pada kedua metode diperlihatkan pada Tabel 4.11 berikut:

Tabel 4.11 Perbandingan Akurasi Hasil Klasifikasi Metode KNN dan LVQ

Metode	Parameter	Hasil sesuai	Akurasi (%)	<i>Elapsed time</i> (sec)
KNN	$k = 4$	202	93.5185	4.09
LVQ	$epoch = 300$ $\alpha = 0.01$	164	75.9259	110.2

Pada Tabel 4.11 menunjukkan perbandingan akurasi hasil klasifikasi menggunakan metode KNN dan LVQ. Uji coba dilakukan dengan menggunakan data sebanyak 216 data sebagai data *training* dan *testing*. Hasil uji coba dengan metode KNN menunjukkan bahwa dari keseluruhan data *testing*, diperoleh hasil yang sesuai kelas sebanyak 202 data sehingga nilai akurasinya sebesar 93.5185%. Sedangkan uji coba dengan metode LVQ diperoleh jumlah data yang sesuai kelas sebanyak 164 data, sehingga nilai akurasi yang diperoleh sebesar 75.9259%. Jika ditinjau dari performansi dalam melakukan proses *running* program, pada Tabel 4.11 terlihat bahwa metode KNN jauh lebih cepat jika dibandingkan dengan menggunakan metode LVQ. Hal ini dikarenakan ketika menggunakan metode LVQ diperlukan waktu iterasi untuk memperoleh bobot akhir selama proses iterasi. Sedangkan pada metode KNN hanya dilakukan pengukuran jarak terhadap dataset sehingga waktu yang digunakan untuk *running* program cukup singkat.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan pada Bab 4, dapat ditarik kesimpulan sebagai berikut:

1. Metode *K-Nearest Neighbor* (KNN)

Nilai keakurasian tertinggi hasil klasifikasi yang ditinjau dari parameter $k = 3$ sampai $k = 10$, untuk jumlah data *training* sebanyak 216 data adalah 93.5185% yaitu pada parameter $k = 4$. Sedangkan untuk nilai akurasi tertinggi hasil klasifikasi menggunakan 162 data *training* adalah 83.3333% pada parameter $k = 6$ dan $k = 7$.

2. Metode *Learning Vector Quantization* (LVQ)

Nilai keakurasian tertinggi hasil klasifikasi yang ditinjau dari parameter *learning rate* sebesar 0.01, 0.05 dan 0.1 dan jumlah iterasi antara 50 sampai 500 iterasi untuk 216 data *training* adalah 77.7778% pada *learning rate* 0.01 dengan jumlah iterasi sebanyak 200 iterasi. Sedangkan untuk 162 data *training*, nilai keakurasian tertinggi adalah sebesar 85.1852% yaitu dengan parameter *learning rate* 0.01 pada jumlah iterasi sebanyak 300 iterasi.

3. Hasil klasifikasi berdasarkan jumlah data *training* yang sama untuk parameter $k = 4$, $\alpha = 0.01$ dan 300 iterasi diperoleh nilai akurasi sebesar 93.5185% pada KNN, sedangkan nilai akurasi pada LVQ sebesar 75.9259%. Jika ditinjau dari performansi kedua metode klasifikasi, metode KNN lebih cepat dalam melakukan proses *running* program jika dibandingkan dengan metode LVQ. Dari uraian diatas dapat disimpulkan bahwa metode KNN lebih baik jika dibandingkan dengan metode LVQ dalam kaitannya dengan permasalahan klasifikasi tingkat kemiskinan.

5.2 Saran

Dalam rangka pengembangan penelitian lebih lanjut dapat dilakukan dengan merubah jenis jarak (*distance*) yang digunakan serta parameter-parameter yaitu *k*, *learning rate* dan jumlah iterasi. Selain itu, penggunaan jenis data pada penelitian ini kurang sesuai untuk diterapkan pada metode KNN ataupun LVQ sehingga memungkinkan untuk diterapkan dalam kasus dengan jenis data lain.

LAMPIRAN 1 : Data Rumah Tangga Sasaran

NO	RTS	LL	JL	JD	SP	BB	SAM	JJ	AS	PND (*000)	PEND	PKJ	KB	KS	KM	KELAS
1	DARMAN	1	Tanah	kayu	0	kayu	sumur	Tdk	1	352,5	SMP	Wiraswasta	0	0	0	M
2	BAHTIAR	0	Plester	kayu	1	minyak	Sumur	Umum	0	378,7	SD	Pegawai	0	0	0	RM
3	SURIADI	0	Tanah	kayu	0	kayu	sungai	Tdk	1	235,5	SD	buruh/tidak	0	0	0	SM
4	YASMIN	1	Plester	kayu	1	minyak	Sumur	Umum	1	220,8	SMP	buruh/tidak	1	1	0	SM
5	AGUS	0	Plester	kayu	0	minyak	sungai	Umum	1	361	>SMA	Pegawai	0	0	0	RM
6	YAHYA	0	tanah	kayu	1	minyak	Sumur	Umum	0	233	SD	buruh/tidak	0	0	1	M
7	KARNAIN	0	Keramik	tembok	0	Gas	PDAM	Sendiri	0	354	>SMA	Pegawai	0	1	0	RM
8	DARWIN	0	Plester	kayu	0	minyak	Sumur	Umum	1	390,6	SD	Wiraswasta	0	1	0	M
9	SIRAM	0	Tanah	bambu	0	kayu	sungai	Tdk	1	210,9	TDK	buruh/tidak	1	1	1	SM
10	MILAH	0	Keramik	tembok	1	Gas	PDAM	Sendiri	0	237	>SMA	buruh/tidak	0	1	0	RM
11	MUHLI	1	Plester	bambu	0	kayu	sungai	Tdk	1	321	TDK	bangunan	0	1	0	SM
12	ZAINAL M	1	Tanah	bambu	0	kayu	sungai	Tdk	1	324,6	SD	bangunan	0	1	0	SM
13	GUSTIANI	1	Keramik	tembok	1	Gas	PDAM	Sendiri	0	376	SD	Pegawai	0	0	0	RM
14	ALIMIN A	1	Keramik	tembok	0	Gas	PDAM	Sendiri	0	338,5	SD	bangunan	0	0	0	RM
15	AMRAN	1	Keramik	tembok	0	Gas	PDAM	Sendiri	0	366,4	SD	Wiraswasta	0	0	0	RM
16	DJAINI	1	Tanah	bambu	0	kayu	sungai	Tdk	0	210	TDK	buruh/tidak	1	0	1	SM
17	RAWI	1	Plester	bambu	1	kayu	sungai	Tdk	1	200,6	SMP	buruh/tidak	1	1	1	SM
18	NAFI'AH	0	Plester	kayu	0	minyak	Sumur	Sendiri	0	298	SD	bangunan	0	0	0	RM
19	SARDIN	0	Tanah	bambu	0	kayu	sungai	Tdk	1	450,7	>SMA	Pegawai	0	1	0	M
20	SATNAWI	0	Tanah	bambu	1	kayu	sungai	Tdk	0	365	SD	Petani	0	0	0	M
21	MARDIN	1	Plester	bambu	1	minyak	sungai	Sendiri	0	375,5	SD	Pegawai	0	0	0	RM

22	YUSUP	1	Plester	bambu	0	kayu	sungai	Tdk	0	388,5	SD	Wiraswasta	0	0	0	M
23	JAPAR USMAN	0	Tanah	bambu	0	kayu	sungai	Tdk	1	430,1	>SMA	Wiraswasta	0	0	0	M
24	SOPIAN SALIM	1	Tanah	bambu	1	kayu	sungai	Tdk	0	387,2	SMP	Petani	0	1	0	M
25	EFENDI HASIM	1	Plester	kayu	1	minyak	PDAM	Sendiri	0	377,5	>SMA	Pegawai	0	1	0	RM
26	KARDION	0	Tanah	bambu	0	kayu	sungai	Tdk	1	397,3	>SMA	Wiraswasta	0	0	0	M
27	JULINA	0	Plester	bambu	1	kayu	sungai	Tdk	0	465,6	SD	Wiraswasta	0	0	0	M
28	SUBNA YUSUP	1	Plester	bambu	0	kayu	sungai	Sendiri	0	432	SD	Petani	0	0	0	RM
29	BAHRI	1	Tanah	bambu	0	kayu	sungai	Tdk	1	264	SMP	buruh/tidak	0	1	0	SM
30	HAMDAN	1	Keramik	tembok	1	Gas	PDAM	Sendiri	0	325	SD	bangunan	0	0	0	RM
31	JEMAIN	0	Plester	kayu	0	minyak	Sumur	Sendiri	0	374,5	SD	Wiraswasta	0	0	0	RM
32	KURNIADI	1	Keramik	tembok	0	Gas	PDAM	Sendiri	0	228,4	SD	buruh/tidak	1	0	0	M
33	ISKANDAR	0	Plester	bambu	0	kayu	sungai	Tdk	0	399,7	SD	Petani	0	0	0	RM
34	SAWAL	0	Plester	bambu	0	kayu	sungai	Tdk	0	267,5	TDK	buruh/tidak	0	0	0	M
35	ZAILANI	1	Keramik	tembok	0	Gas	sungai	Sendiri	0	387,8	SD	Wiraswasta	0	0	0	RM
36	NURHANI	1	Plester	kayu	0	minyak	PDAM	Sendiri	0	403,5	SD	Pegawai	0	0	0	RM
37	SUHAIMI	1	Plester	kayu	0	minyak	Sumur	Tdk	0	364,7	TDK	Petani	0	1	0	M
38	AMRAN	1	Plester	bambu	0	kayu	sungai	Tdk	1	305,4	SMP	bangunan	0	0	0	M
39	SALEH	0	Plester	bambu	0	kayu	sungai	Tdk	1	431,5	SMP	Petani	0	1	0	M
40	JUNAIDI	0	Plester	bambu	0	kayu	sungai	Tdk	0	406,9	SD	Wiraswasta	0	0	0	RM
41	NURSILAH	0	Plester	bambu	0	kayu	sungai	Tdk	0	379,6	SD	Petani	0	0	0	RM
42	AGUS BUDIMAN	1	Plester	kayu	0	minyak	Sumur	Sendiri	0	307,7	SD	bangunan	0	0	0	RM
43	JULAIHA	0	Plester	bambu	0	kayu	sungai	Tdk	0	350,6	TDK	Petani	0	1	0	M
44	SAINEN	0	Plester	bambu	0	kayu	sungai	Tdk	0	356,5	SD	Petani	0	1	0	M



45	MAHMUD	0	Plester	bambu	0	kayu	sungai	Tdk	0	348	SD	bangunan	0	0	0	M
46	ISKANDAR	0	Plester	bambu	0	kayu	sungai	Tdk	1	366,6	SD	Petani	0	0	0	M
47	BUDIONO	0	Plester	kayu	0	minyak	sungai	Sendiri	0	341,4	SD	bangunan	0	0	0	RM
48	BASTIRIN	0	Plester	bambu	0	kayu	sungai	Tdk	0	200,6	TDK	buruh/tidak	1	0	1	M
49	NORMAN	0	Plester	bambu	0	kayu	sungai	Tdk	0	377,5	TDK	Petani	0	0	0	RM
50	MAISURAH	0	Plester	bambu	0	kayu	sungai	Tdk	0	122,5	TDK	buruh/tidak	1	0	1	M
51	EFFINDI	1	Plester	bambu	0	kayu	sungai	Tdk	0	190,2	SD	buruh/tidak	1	0	1	SM
52	RIDO'I	1	Tanah	bambu	0	kayu	sungai	Tdk	1	379,6	SD	Petani	0	1	0	M
53	BUDIATI	0	Tanah	bambu	1	kayu	sungai	Tdk	0	357	SD	Petani	0	0	0	M
54	REPENDI	1	Plester	kayu	1	kayu	sungai	Sendiri	0	278,8	SD	bangunan	0	0	0	M
55	YAMAN	1	Tanah	bambu	0	kayu	sungai	Tdk	1	355,5	TDK	Petani	0	0	0	M
56	NORMALA	1	Plester	bambu	1	minyak	sungai	Sendiri	0	424,5	SD	Pegawai	0	0	0	RM
57	SALAM	1	Plester	kayu	0	minyak	sungai	Sendiri	0	309,3	SD	bangunan	0	0	0	RM
58	RUSDI	1	Plester	bambu	0	minyak	Sumur	Sendiri	1	501	>SMA	Wiraswasta	0	0	0	RM
59	PAHRUM	1	Plester	bambu	0	kayu	sungai	Sendiri	0	326,5	TDK	bangunan	0	1	0	M
60	PUKIMA	1	Plester	kayu	0	minyak	sungai	Sendiri	1	395,5	SD	Petani	0	0	0	RM
61	LARAM	1	Plester	bambu	0	kayu	sungai	Sendiri	0	521,6	SD	Petani	0	0	0	RM
62	ABDUSAMAD	1	Plester	bambu	0	minyak	Sumur	Sendiri	0	477,3	SD	Wiraswasta	0	1	0	M
63	BU'ALI	0	Plester	kayu	0	minyak	Sumur	Sendiri	0	350	SD	bangunan	0	0	0	RM
64	ARDIMAN	1	Plester	kayu	0	minyak	Sumur	Sendiri	1	355,6	SD	Wiraswasta	0	0	0	RM
65	SAHUDI	1	Plester	bambu	0	kayu	sungai	Tdk	1	221,7	TDK	buruh/tidak	1	1	0	SM
66	MUDARI	1	Plester	kayu	0	minyak	Sumur	Umum	1	253	SD	buruh/tidak	0	1	0	SM
67	SALIMAN	1	Plester	kayu	0	kayu	sungai	Umum	0	457,6	TDK	Petani	0	0	0	RM
68	MARTILAN	1	Plester	kayu	0	kayu	sungai	Umum	0	399,5	SD	Petani	0	0	0	RM

69	MUTARI	1	Plester	kayu	0	kayu	sungai	Umum	0	437,5	SD	Petani	0	0	0	RM
70	MISWAR	0	Plester	kayu	0	kayu	Sumur	Tdk	0	346	SD	bangunan	0	0	0	M
71	MATSALEH	1	Tanah	bambu	0	kayu	sungai	Tdk	1	350,6	TDK	Petani	0	1	0	M
72	ASMAD	1	Plester	kayu	1	kayu	Sumur	Tdk	0	287,4	TDK	bangunan	0	0	0	SM
73	HANMID	0	Plester	bambu	0	kayu	sungai	Sendiri	1	346	SD	bangunan	0	0	0	M
74	MURSIDI	0	Tanah	bambu	0	kayu	sungai	Tdk	0	388,7	SD	Petani	0	0	0	M
75	MAT SAHRI	0	Plester	bambu	1	kayu	sungai	Sendiri	0	365	SD	Petani	0	0	0	RM
76	MAYURI	1	Plester	bambu	0	minyak	Sumur	Sendiri	0	403,5	SD	Pegawai	0	0	0	RM
77	SUDAI	0	Plester	bambu	0	kayu	sungai	Sendiri	0	398,4	SD	Petani	0	0	0	RM
78	HANAFI	1	Tanah	bambu	0	kayu	sungai	Tdk	1	338,9	TDK	bangunan	0	1	0	SM
79	ASMUDI	1	Plester	kayu	1	minyak	Sumur	Sendiri	0	426,5	SD	Pegawai	0	0	0	RM
80	JAWADI	1	Plester	kayu	0	minyak	Sumur	Umum	0	277,4	SD	bangunan	0	0	0	M
81	ASMAN	1	Keramik	tembok	0	Gas	PDAM	Sendiri	0	285	SD	bangunan	0	0	0	M
82	JULIAH	0	Plester	kayu	0	minyak	sungai	Umum	1	375,6	SD	Petani	0	0	0	RM
83	KONI	1	Tanah	bambu	0	kayu	sungai	Tdk	1	221,4	>SMA	buruh/tidak	1	1	0	SM
84	ADMONADI	0	Tanah	bambu	1	kayu	sungai	Tdk	1	320,4	TDK	bangunan	0	0	0	M
85	BANI	0	Tanah	bambu	1	kayu	sungai	Tdk	1	370,5	TDK	Petani	0	0	0	M
86	MAHLI	1	Plester	kayu	1	minyak	Sumur	Umum	0	458	SD	Wiraswasta	0	0	0	M
87	KASUM	1	Plester	kayu	0	minyak	Sumur	Umum	0	433	SD	Wiraswasta	0	0	0	RM
88	AMALIAH	1	Tanah	bambu	0	kayu	sungai	Tdk	1	270,5	TDK	bangunan	0	0	0	M
89	ASWAN	1	Plester	kayu	1	minyak	Sumur	Umum	1	220,6	TDK	buruh/tidak	1	1	0	SM
90	ARBAAH	1	Keramik	tembok	0	Gas	sungai	Sendiri	0	388,6	SD	Petani	0	0	0	RM
91	ISMAWANDI	1	Plester	bambu	0	kayu	sungai	Tdk	0	335	SD	bangunan	0	0	0	M
92	INDRAWATI	1	Plester	kayu	0	minyak	PDAM	Sendiri	0	348,4	SD	bangunan	0	0	0	RM



93	MASRI	1	Keramik	tembok	0	Gas	PDAM	Sendiri	0	227	SD	buruh/tidak	1	0	0	M
94	DAPID	1	Keramik	tembok	0	Gas	Sumur	Sendiri	0	347	SD	bangunan	0	0	0	RM
95	JAYADI	1	Keramik	tembok	0	Gas	PDAM	Sendiri	0	336,8	SD	bangunan	0	0	0	RM
96	HAMSAR	1	Keramik	tembok	0	Gas	PDAM	Sendiri	0	320,6	SD	wiraswasta	0	0	0	RM
97	MULIADI	1	Keramik	tembok	0	Gas	Sumur	Sendiri	0	407,5	SD	Wiraswasta	0	0	0	RM
98	RUSMINA	1	Plester	kayu	0	minyak	Sumur	Umum	0	448,6	SD	Wiraswasta	0	0	0	M
99	JAMANI	0	Plester	kayu	0	minyak	Sumur	Umum	0	454,5	SD	Wiraswasta	0	0	0	RM
100	HAIRANI	1	Plester	kayu	0	kayu	sungai	Sendiri	0	302,8	SD	bangunan	0	0	0	RM
10	RASAD	1	Plester	bambu	0	minyak	sungai	Sendiri	0	438,2	SD	Petani	0	0	0	RM
102	JAUWADI	1	Plester	bambu	0	minyak	Sumur	Umum	0	326,4	SD	bangunan	0	0	0	M
103	MUNADIN	1	Tanah	bambu	0	kayu	sungai	Tdk	1	275,5	SD	bangunan	0	0	0	M
104	ASNI	1	Tanah	bambu	1	kayu	sungai	Tdk	1	180,6	SD	buruh/tidak	1	1	1	SM
105	RUSNAH	1	Plester	kayu	1	minyak	Sumur	Umum	1	358,2	SD	Petani	0	0	0	M
106	JUSNIYANTO	1	Plester	bambu	0	kayu	Sumur	Tdk	1	350,5	SD	Petani	0	0	0	M
107	HEIRANI	1	Tanah	bambu	0	kayu	Sumur	Tdk	1	364,2	SD	Petani	0	0	0	M
108	SARBANI	0	Plester	kayu	1	minyak	Sumur	Sendiri	0	403,5	SD	Petani	0	0	0	RM
109	PAIMAH	1	Plester	kayu	0	minyak	PDAM	Sendiri	0	406,6	SD	Wiraswasta	0	0	0	M
110	HAMDANI	1	Plester	kayu	0	minyak	Sumur	Umum	0	375,4	SD	Wiraswasta	0	0	0	RM
111	DARMAN	1	Tanah	bambu	0	kayu	sungai	Tdk	1	194,5	SD	buruh/tidak	1	1	1	SM
112	ARDIMANSYAH	1	Keramik	tembok	1	Gas	PDAM	Sendiri	0	560,4	SD	Wiraswasta	0	0	0	RM
113	ASNAN	0	Tanah	bambu	0	kayu	sungai	Tdk	1	340,5	>SMA	bangunan	0	0	0	M
114	ISKANDAR	1	Plester	kayu	1	minyak	Sumur	Umum	0	354	SD	Wiraswasta	0	0	0	M
115	FAISAL	1	Plester	kayu	0	minyak	Sumur	Umum	0	375	SD	Wiraswasta	0	0	0	M
116	PARIMAH	1	Tanah	bambu	0	kayu	sungai	Tdk	0	264,5	SD	buruh/tidak	0	1	0	SM

117	ARISADIKIN	1	Tanah	bambu	1	kayu	Sumur	Umum	1	358,5	>SMA	Wiraswasta	0	1	0	SM
118	MOHER	1	Keramik	tembok	1	Gas	PDAM	Sendiri	0	438,2	SD	Pegawai	0	0	0	RM
119	HAIDA	1	Tanah	bambu	0	kayu	Sumur	Tdk	0	405,5	SD	Petani	0	0	0	M
120	MURIANTO	1	Plester	kayu	1	minyak	Sumur	Umum	0	375,5	SD	Petani	0	0	0	M
121	SUHAIRI	1	Tanah	bambu	0	kayu	sungai	Tdk	1	359	SD	Petani	0	0	0	M
122	HANA	1	Plester	bambu	1	kayu	sungai	Tdk	1	390,5	SD	Petani	0	0	0	M
123	M.RABUDIN	1	Plester	kayu	0	minyak	Sumur	Umum	0	410,4	SD	Pegawai	0	0	0	M
124	P.MISAN	1	Plester	bambu	0	kayu	sungai	Tdk	0	390,5	SD	Petani	0	0	0	M
125	MUSMIN	0	Tanah	bambu	0	kayu	sungai	Tdk	1	354	SD	Petani	0	0	0	M
126	ATIMIN	1	Plester	kayu	1	minyak	Sumur	Umum	0	365,5	SD	Wiraswasta	0	0	0	M
127	H.IDUS	1	Keramik	tembok	0	Gas	PDAM	Sendiri	0	340,5	SD	bangunan	0	0	0	M
128	ASMAYADI	1	Keramik	tembok	0	Gas	PDAM	Sendiri	0	400,5	SD	Pegawai	0	0	0	RM
129	SAPIAH	1	Tanah	bambu	0	kayu	Sumur	Tdk	1	448,2	SD	Petani	0	0	0	M
130	LUKMAN HAKIM	1	Plester	bambu	1	kayu	Sumur	Tdk	1	275,5	SD	buruh/tidak	0	0	0	SM
131	HENDRA	0	Plester	bambu	0	minyak	Sumur	Sendiri	0	349	SD	bangunan	0	0	0	RM
132	SATIAH	1	Tanah	bambu	0	kayu	sungai	Tdk	1	175,5	SD	buruh/tidak	1	1	1	SM
133	ARIANTO	1	Keramik	tembok	1	Gas	Sumur	Sendiri	1	345,5	SD	bangunan	0	0	0	M
134	SUGITO	0	Plester	kayu	0	minyak	Sumur	Sendiri	0	348	SD	bangunan	0	0	0	RM
135	BOWO ARISNO	0	Plester	kayu	0	minyak	Sumur	Sendiri	0	350	SD	bangunan	0	1	0	M
136	WAKIJO	1	Keramik	tembok	0	Gas	PDAM	Sendiri	0	335,6	SD	bangunan	0	0	0	RM
137	ABDUL KADIR	1	Keramik	tembok	0	Gas	sungai	Sendiri	0	265,5	SD	buruh/tidak	0	1	0	M
138	NGATIRAN TP	0	Tanah	bambu	0	kayu	Sumur	Tdk	1	375,7	>SMA	Wiraswasta	0	1	0	M
139	KIRNO WIYOTO	1	Plester	bambu	1	kayu	sungai	Sendiri	1	405,3	SD	Petani	0	0	0	M



140	RUSMANTO	0	Keramik	tembok	0	Gas	PDAM	Sendiri	0	349	SD	bangunan	0	0	0	RM
141	MULIADI	0	Tanah	bambu	0	kayu	Sumur	Tdk	0	335,8	>SMA	bangunan	0	0	0	M
142	MUSNAWAR	0	Keramik	tembok	1	Gas	PDAM	Sendiri	0	305,5	SD	Wiraswasta	0	1	0	M
143	SOPIAN	1	Plester	bambu	0	kayu	Sumur	Tdk	1	358,5	SD	Petani	0	0	0	M
144	AGUS SUPAR	1	Plester	kayu	0	minyak	Sumur	Umum	0	275,5	SD	bangunan	0	1	0	M
145	TASIMAN	0	Keramik	tembok	0	Gas	PDAM	Sendiri	1	335	SD	wiraswasta	0	0	0	RM
146	JAILANI	1	Tanah	bambu	0	kayu	sungai	Tdk	1	355,9	SD	Petani	0	1	0	SM
147	ARMIA	1	Tanah	bambu	1	kayu	sungai	Tdk	1	180,5	SD	buruh/tidak	1	1	1	SM
148	ASMADI	0	Keramik	tembok	1	Gas	Sumur	Sendiri	0	325,5	SD	bangunan	0	0	0	RM
149	JUNIAH	1	Keramik	tembok	0	Gas	PDAM	Sendiri	1	400,5	>SMA	Pegawai	0	0	0	RM
150	HENDRI	1	Tanah	bambu	0	kayu	sungai	Tdk	1	355,7	>SMA	Petani	0	0	0	M
151	JOHARI	0	Plester	bambu	1	kayu	Sumur	Tdk	0	345,5	SD	bangunan	0	0	0	M
152	HASMAN	1	Keramik	tembok	0	Gas	PDAM	Sendiri	0	342,5	>SMA	wiraswasta	0	0	0	RM
153	RAWIYA	1	Keramik	tembok	0	Gas	PDAM	Sendiri	0	345,5	SD	Petani	0	0	0	RM
154	AKIM AHMAT	0	Tanah	bambu	0	kayu	sungai	Tdk	1	349	>SMA	bangunan	0	0	0	M
155	JASIMA	0	Tanah	bambu	1	kayu	sungai	Tdk	1	305,5	SMP	bangunan	0	0	0	M
156	SAMSUDIN	1	Tanah	bambu	1	kayu	sungai	Tdk	1	354,5	>SMA	Petani	0	1	0	SM
157	SABUDIN	0	Plester	bambu	1	kayu	Sumur	Tdk	1	280,3	SD	buruh/tidak	0	1	0	SM
158	INDRA	0	Plester	bambu	0	kayu	Sumur	Tdk	0	341,5	SD	buruh/tidak	0	0	0	M
159	SUHENDI	1	Tanah	bambu	0	kayu	sungai	Tdk	1	260,5	SD	buruh/tidak	0	1	0	SM
160	AMAD	0	Keramik	tembok	1	Gas	Sumur	Sendiri	0	325,5	SD	bangunan	0	0	0	RM
161	SUYOTO	0	Keramik	tembok	0	Gas	PDAM	Sendiri	0	348	>SMA	bangunan	0	0	0	RM
162	JUNAIDI	0	Keramik	tembok	0	Gas	PDAM	Sendiri	0	342,5	>SMA	bangunan	0	1	0	RM
163	LAMRAN	1	Tanah	bambu	0	kayu	Sumur	Tdk	1	375,5	>SMA	Petani	0	0	0	M

164	PARDIR	1	Keramik	tembok	1	Gas	PDAM	Sendiri	0	335,8	SD	bangunan	0	0	0	RM
165	YURYANTO	1	Keramik	tembok	0	Gas	PDAM	Sendiri	0	342	>SMA	wiraswasta	0	1	0	RM
166	EKO P	1	Keramik	tembok	0	Gas	Sumur	Sendiri	0	325,5	SD	bangunan	0	0	0	RM
167	SOPIA	0	Keramik	tembok	0	Gas	PDAM	Sendiri	0	331,7	SMP	bangunan	0	1	0	RM
168	SULEMAN	1	Tanah	bambu	0	kayu	Sumur	Tdk	0	406,5	>SMA	Petani	0	0	0	M
169	SAINI	1	Keramik	tembok	1	Gas	PDAM	Sendiri	1	349	>SMA	wiraswasta	0	0	0	RM
170	INDRA	0	Tanah	bambu	0	kayu	sungai	Tdk	0	325,7	>SMA	wiraswasta	0	0	0	M
171	SAMAI	1	Tanah	bambu	1	kayu	Sumur	Tdk	1	295,5	>SMA	bangunan	0	1	0	SM
172	SUPENDI	1	Tanah	bambu	1	kayu	Sumur	Tdk	0	344	>SMA	bangunan	0	0	0	M
173	SUPARDI	0	Plester	kayu	1	kayu	sungai	Sendiri	1	375,5	SD	Petani	0	1	0	M
174	IDRUS	1	Keramik	tembok	0	Gas	PDAM	Sendiri	0	504,5	>SMA	Pegawai	0	0	0	RM
175	SAHARI	1	Plester	kayu	0	minyak	Sumur	Umum	0	622,5	>SMA	Pegawai	0	0	0	RM
176	SUPNI	1	Tanah	bambu	0	kayu	sungai	Tdk	1	304,4	>SMA	wiraswasta	0	0	0	M
177	SUHERMAN	0	Tanah	bambu	1	kayu	Sumur	Tdk	0	335	SD	bangunan	0	0	0	M
178	RABUDIN	0	Tanah	bambu	1	kayu	sungai	Tdk	1	210,5	SMP	buruh/tidak	1	0	1	SM
179	DAUT	0	Keramik	tembok	1	Gas	Sumur	Sendiri	1	325	SD	wiraswasta	0	1	0	M
180	RAFA'I	1	Keramik	tembok	0	Gas	PDAM	Sendiri	0	345,7	>SMA	wiraswasta	0	0	0	RM
181	HAMIJI	1	Plester	bambu	0	kayu	Sumur	Tdk	0	379,5	SD	Petani	0	0	0	M
182	BAHTIAR	0	Tanah	bambu	0	kayu	Sumur	Tdk	1	405,5	>SMA	Pegawai	0	0	0	M
183	SAPARIAH	0	Tanah	bambu	1	kayu	sungai	Tdk	1	188,4	SD	buruh/tidak	1	0	1	SM
184	SIMAH	0	Tanah	kayu	1	kayu	PDAM	Umum	0	315,5	>SMA	wiraswasta	0	0	0	M
185	TAHIR	1	Keramik	tembok	1	Gas	PDAM	Sendiri	0	342,5	>SMA	wiraswasta	0	0	0	RM
186	HAMIDI	0	Keramik	tembok	0	Gas	PDAM	Sendiri	0	335,1	SD	wiraswasta	0	0	0	RM
187	KARTINI	0	Keramik	tembok	0	Gas	PDAM	Sendiri	0	290,5	SMP	wiraswasta	0	0	0	RM

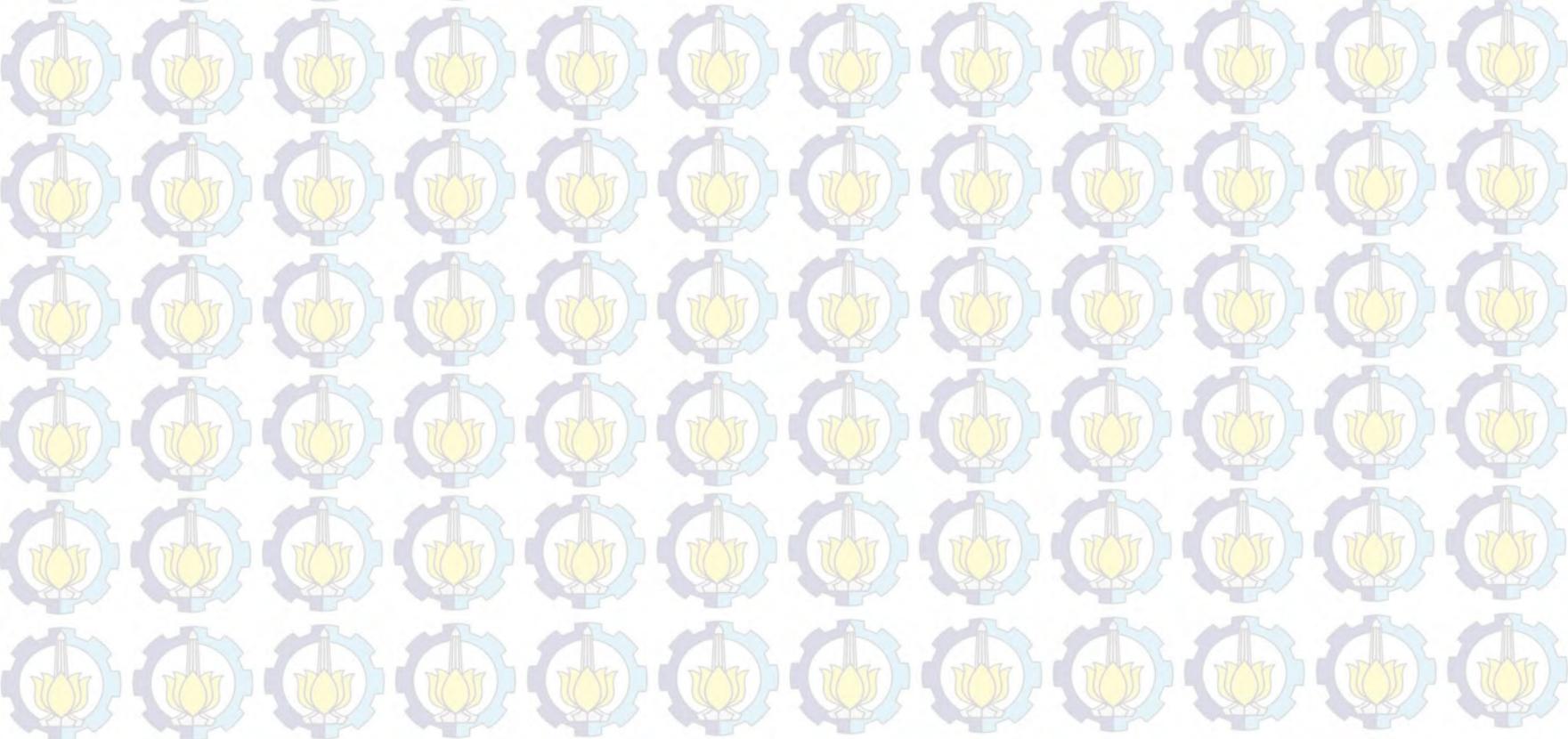


188	RUSLI	1	Plester	kayu	0	minyak	Sumur	Umum	0	275,5	SD	wiraswasta	0	0	0	M
189	MADAN	0	Plester	kayu	0	minyak	Sumur	Umum	0	345,5	SD	wiraswasta	0	0	0	M
190	ABDUL AZIS	1	Plester	kayu	0	minyak	PDAM	Sendiri	0	250,6	SD	buruh/tidak	0	0	0	M
191	MAI MUNAH	1	Plester	kayu	0	minyak	Sumur	Umum	1	175,5	SD	buruh/tidak	1	0	1	SM
192	HASAN	0	Keramik	tembok	0	Gas	PDAM	Sendiri	0	345,5	>SMA	wiraswasta	0	1	0	RM
193	MASTOR	0	Tanah	bambu	0	kayu	sungai	Tdk	0	280,6	SMP	bangunan	0	1	0	M
194	EFENDI ABDULAH	0	Tanah	bambu	1	kayu	sungai	Tdk	0	324,5	>SMA	wiraswasta	0	0	0	M
195	SUMUHDI	1	Tanah	bambu	1	kayu	sungai	Tdk	1	280	SMP	bangunan	0	0	0	SM
196	JULIANSYAH	0	Tanah	bambu	1	kayu	sungai	Tdk	1	190,5	SD	buruh/tidak	1	1	1	SM
197	RIDWAN	0	Tanah	bambu	1	kayu	sungai	Tdk	0	225,5	SD	buruh/tidak	1	1	0	SM
198	ISMAIL	1	Plester	kayu	1	minyak	PDAM	Sendiri	0	450,5	>SMA	Pegawai	0	0	0	RM
199	SURIYADI	0	Plester	kayu	0	minyak	Sumur	Sendiri	0	421,5	>SMA	Pegawai	0	1	0	RM
200	HADANAN	0	Keramik	tembok	0	Gas	PDAM	Sendiri	1	335	SMP	wiraswasta	0	1	0	RM
20	SYARIFAH	1	Plester	kayu	0	minyak	Sumur	Sendiri	0	295,5	SMP	wiraswasta	0	0	0	M
202	KARTO S	1	Keramik	tembok	0	Gas	PDAM	Sendiri	1	325,5	SMP	wiraswasta	0	0	0	RM
203	SUPIANTO	0	Tanah	bambu	0	kayu	sungai	Tdk	0	390	SMP	Petani	0	1	0	M
204	ABDUL H	0	Tanah	bambu	1	kayu	Sumur	Tdk	0	405,5	>SMA	Petani	0	0	0	M
205	ASMADI	0	Keramik	tembok	1	Gas	PDAM	Sendiri	0	305,7	SD	wiraswasta	0	1	0	M
206	SUHARDI	1	Plester	kayu	0	minyak	Sumur	Sendiri	0	425	>SMA	Wiraswasta	0	0	0	RM
207	M.NAWAWI	0	Plester	kayu	0	minyak	Sumur	Sendiri	0	403,5	>SMA	Pegawai	0	0	0	RM
208	RAMLI	0	Plester	kayu	0	minyak	PDAM	Sendiri	0	349	SMP	wiraswasta	0	0	0	M
209	MURDIANA	0	Plester	bambu	0	kayu	Sumur	Tdk	1	250,5	SD	buruh/tidak	0	0	0	M
210	M. YENURI	0	Plester	bambu	0	kayu	sungai	Tdk	1	375,8	SMP	Petani	0	1	0	M





211	ADI SANUSI	0	Tanah	bambu	0	kayu	Sumur	Tdk	1	211,5	SD	buruh/tidak	1	1	1	SM
212	MUTIARDI	1	Tanah	bambu	1	kayu	sungai	Tdk	1	357,5	SMP	Petani	0	0	0	M
213	HUSIN	1	Plester	bambu	1	kayu	Sumur	Tdk	0	395,1	>SMA	Petani	0	0	0	M
214	ROKI'IN	0	Plester	kayu	0	minyak	Sumur	Umum	1	345,5	SMP	wiraswasta	0	0	0	M
215	MARIANA	1	Tanah	bambu	0	kayu	sungai	Tdk	1	195	SMP	buruh/tidak	1	0	1	SM
216	SAPAWI	1	Tanah	bambu	1	kayu	Sumur	Tdk	1	210,5	SD	buruh/tidak	1	1	1	SM



LAMPIRAN 2 : Listing Program Metode K-Nearest Neighbor

Source Code

```
clc
clear all
tic
tcpu=cputime

Training= xlsread('data_proses.xlsx', 1, 'B3:O218');%data
training 216(1)
Cls_KNN= xlsread('data_proses.xlsx', 1, 'P3:P218');
Sample = xlsread('data_proses.xlsx', 1, 'B3:O218');
K      = 4
class  = knnclassify(Sample, Training, Cls_KNN,K);

cls_awal=xlsread('data_proses.xlsx', 1, 'P3:P218');

%HASIL
[~,~,raw]=xlsread('data_proses.xlsx','A3:A218');
n=length(cls_awal);
for i=1:n;
    raw{i,2}=cls_awal(i);
    raw{i,3}=class(i);
end;
disp('-----');
disp('  Data | Cls_Awal | Class  ');
disp('-----');
disp(raw);

j=length(cls_awal);
for i=1:j,
    k(i)=0;
    if cls_awal(i)~=class(i),
        k(i)=k(i)+0;
    else
        k(i)=k(i)+1;
    end;
end;
jum=sum(k);
disp('-----');
disp('----- AKURASI KNN -----');
disp('-----');
disp(strcat('Data Dikenali=',int2str(jum)))
disp(strcat('Data Uji=',int2str(j)))
b=(jum/j)*100;
b1=num2str(b);
disp(strcat('Tingkat Akurasi=',b1))
```

```
% %PLOT
%R=rand(2,n);
load R
Y= R';
initial = ([[1:n]]',Y);
gscatter(initial(:,2)+cls_awal,initial(:,3)+class,cls_awal,
'r','o*+');
hold on;
gscatter(initial(:,2)+cls_awal,initial(:,3)+class,class,'b'
,'o*+');

legend('Target awal 1', 'Target awal 2','Target awal
3','KNN 1','KNN 2','KNN 3');
xlabel('Initial X');
ylabel('Initial Y');

ttcpu=cputime-tcpu;
fprintf('Waktu Testing:%s\n',ttcpu);
toc;
```

LAMPIRAN 3 : Listing Program Metode Learning Vector Quantization

Source Code

```

clc
clear all
tic
tcpu=cputime
datatrain = xlsread('data_proses.xlsx', 1,
'B3:O218');
target = xlsread('data_proses.xlsx', 1,
'P3:P218');
T = ind2vec(target');
LearningRate = 0.01;
net = newlvq(datatrain',3,[.4 .3
.3],LearningRate);
net.IW{1,1} =
[1,1,0.5,0,1,0.5,1,1,0.46,0.33,0.25,0,0,0;0,0.5,0.5,1,0.5,0
.5,0.5,0,0.51,0.67,0,0,0,0;0,1,0.5,0,1,1,1,1,0.23,0.67,1,0,
0,0];
net.IW{1,1};
net.trainParam.epochs = 50; %banyak iterasi
net = train(net, datatrain',T);
%load ans
net.IW{1,1}
test = xlsread('data_proses.xlsx', 1,
'B3:O218');
target2test = xlsread('data_proses.xlsx', 1,
'P3:P218');
%net.IW{1,1}
hasil_LVQ = sim(net,test');
cls_LVQ = vec2ind(hasil_LVQ);

%HASIL TRAINING
[~,~,raw]=xlsread('data_proses.xlsx','A3:A218');
n=length(target2test);
for i=1:n;
    raw{i,2}=target2test(i);
    raw{i,3}=cls_LVQ(i);
end;
disp('-----');
disp('Data      Target      Class LVQ');
disp('-----');
disp(raw);

j=length(target2test);
for i=1:j;
    k(i)=0;
    if target2test(i)~=cls_LVQ(i),

```

```

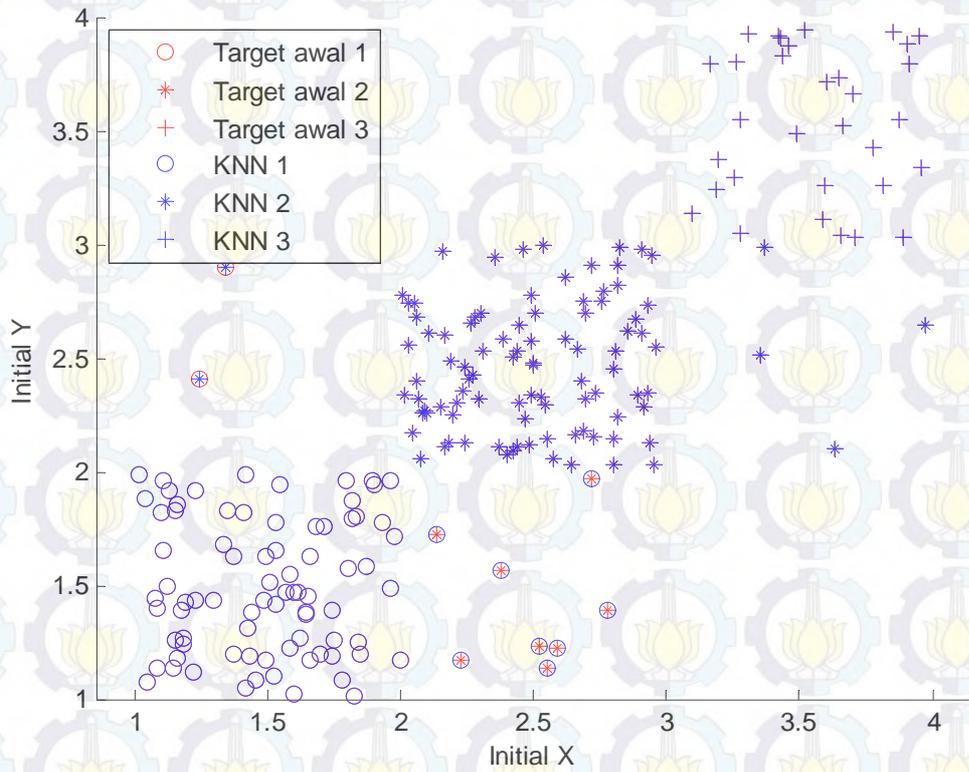
        k(i)=k(i)+0;
    else
        k(i)=k(i)+1;
    end;
end;
jum=sum(k);
disp('-----');
disp('----- AKURASI LVQ -----');
disp('-----');
disp(strcat('Data Dikenali=',int2str(jum)))
disp(strcat('Data Uji=',int2str(j)))
b=(jum/j)*100;
b1=num2str(b);
disp(strcat('Tingkat Akurasi=',b1))

% %PLOT
R=rand(2,n);
load R
Y= R';
initial=[[1:n]',Y];
gscatter(initial(:,2)+target2test,initial(:,3)+cls_LVQ',target2test,'r','o*+');
hold on;
gscatter(initial(:,2)+target2test,initial(:,3)+cls_LVQ',cls_LVQ,'b','o*+');
legend('Target awal 1', 'Target awal 2','Target awal 3','LVQ 1','LVQ 2','LVQ 3');
xlabel('Initial X');
ylabel('Initial Y');
ttcpu=cputime-tcpu;
fprintf('Waktu Testing:%s\n',ttcpu);
toc;

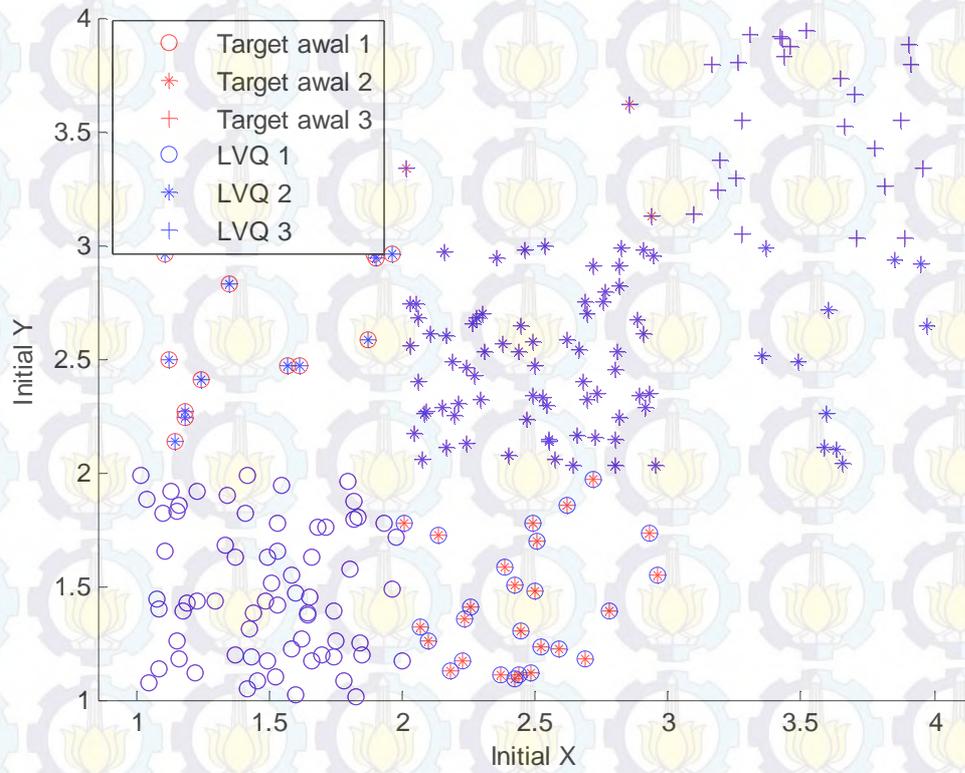
```

LAMPIRAN 4 : Plot Hasil *Testing* Metode KNN dan LVQ dengan 216 Data

1. Plot Hasil *Testing* Kelas Klasifikasi Metode KNN



2. Plot Hasil *Testing* Kelas Klasifikasi Metode LVQ



DAFTAR PUSTAKA

BPS, (2008), *Analisis Dan Perhitungan Tingkat Kemiskinan 2008*, Badan Pusat Statistik, Jakarta.

BPS, (2012), *Analisis Data Kemiskinan Berdasarkan Data Pendataan Program Perlindungan Sosial (PPLS) 2011*, Badan Pusat Statistik, Jakarta.

Farsiah, L., Abidin, T.F., dan Munadi, K, (2013), *Kasifikasi Gambar Berwarna Menggunakan K-Nearest Neighbor dan Support Vektor Machine*, FMIPA USK, Banda Aceh.

Fausett, L, (1994), *Fundamental Of Neural Network : Architectures, Algorithms, and Aplications*, Prentice Hall Inc, New Jersey.

Jabbar, M.A., Deekshatulu, B.L., dan Chandra, P, (2013), “ Classification of Heart Disease Using K-Nearest Neighbor and Genetic Algorithm”, *International Conference on (CIMTA), Procedia Technology*,10, 85 – 94.

Karegowda, G.A, Jayaram, M.A, dan Manjunath, A.S, (2012), “ Cascading K-means Clustering and K-Nearest Neighbor Classifier for Categorization of Diabetic Patients”, *International Journal of Engineering and Advanced Technology (IJEAT)*, Vol.1, No.3, 147-151.

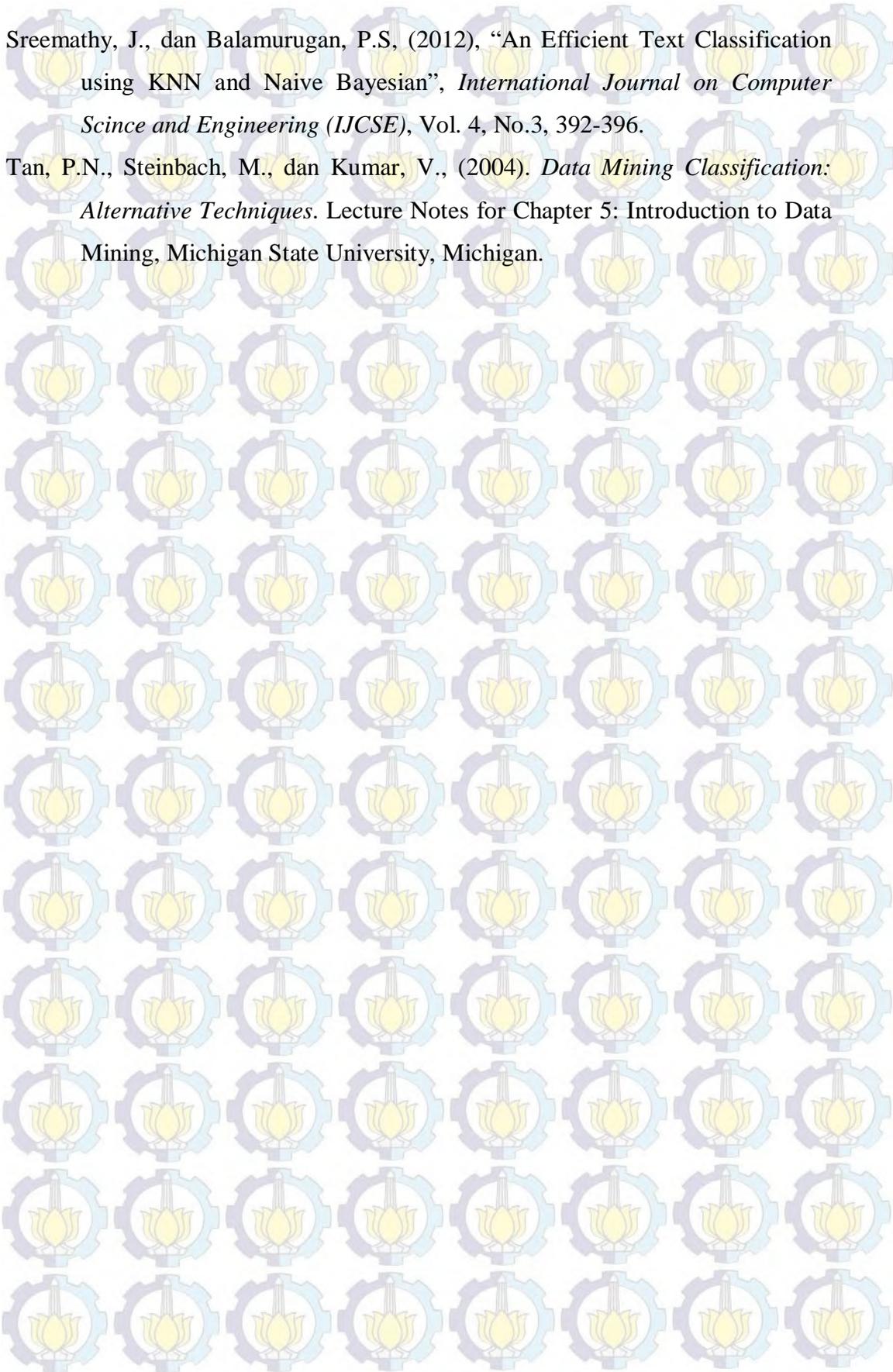
Krisandi, N., Helmi, dan Prihandono, B, (2013), “Algoritma K-Nearest Neighbor dalam Klasifikasi Data Hasil Produksi Kelapa Sawit pada PT. Minamas Kecamatan Parindu”, *Buletin Ilmiah Matematika Statistik dan Terapannya (Bimaster)*, 02(1), 33-38.

Lusiyanti, D, (2014), *Perbandingan Metode Learning Vector Quantization (LVQ) dan Support Vector Machine (SVM) untuk Deteksi Penyakit Jantung Koroner*, Pascasarjana Matematika ITS, Surabaya.

Nurkhozin, A., Irawan, M.I., dan Mukhlas, I, (2011), *Komparasi Hasil Klasifikasi Penyakit Diabetes Mellitus Menggunakan Jaringan Syaraf Tiruan Backpropagation dan Learning Vector Quantization (LVQ)*, Pascasarjana FMIPA ITS, Surabaya.

Sreemathy, J., dan Balamurugan, P.S, (2012), “An Efficient Text Classification using KNN and Naive Bayesian”, *International Journal on Computer Science and Engineering (IJCSE)*, Vol. 4, No.3, 392-396.

Tan, P.N., Steinbach, M., dan Kumar, V., (2004). *Data Mining Classification: Alternative Techniques*. Lecture Notes for Chapter 5: Introduction to Data Mining, Michigan State University, Michigan.



BIODATA PENULIS



Penulis yang memiliki nama lengkap Santoso lahir di Sukoharjo, 17 Maret 1984. Penulis telah menempuh pendidikan formal mulai dari SDN VIII Dawuhan Situbondo, SMP Negeri 1 Situbondo, dan SMA Negeri 1 Situbondo. Penulis melanjutkan studi S1 di Universitas Negeri Jember jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam dan menyelesaikan studi pada tahun 2010. Penulis melanjutkan studi Pra-S2 di Jurusan Matematika

Institut Teknologi Sepuluh Nopember Surabaya pada tahun 2012 melalui jalur Beasiswa calon dosen Pra-S2 selama satu tahun dan melanjutkan studi S2 melalui jalur beasiswa BPPDN Dikti dan tercatat sebagai mahasiswa Program Pascasarjana Institut Teknologi Nopember Surabaya pada tahun 2013 dan lulus pada semester Gasal periode 2015/2016. Segala kritik dan saran yang membangun dapat dikirimkan melalui alamat email santoso17march@gmail.com.

