

**KLASIFIKASI PENERIMAAN PROGRAM KELUARGA HARAPAN  
(PKH) MENGGUNAKAN METODE *LEARNING VECTOR  
QUANTIZATION* (STUDI KASUS DESA KEDUNGJATI)**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Vidya Capristyan Pamungkas

NIM: 135150201111157



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

**PENGESAHAN**

**KLASIFIKASI PENERIMAAN PROGRAM KELUARGA HARAPAN (PKH)  
MENGUNAKAN METODE *LEARNING VECTOR QUANTIZATION*  
(STUDI KASUS DESA KEDUNGJATI)**

**SKRIPSI**

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Vidya Capristyan Pamungkas  
NIM: 135150201111157

Skripsi ini telah diuji dan dinyatakan lulus pada  
*17 Oktober 2018*  
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

  
Lailil Muflikhah, S.Kom, M.Sc  
NIP:19741113 200501 2 001

  
Randy Cahya Wihandika, S.ST., M.Kom  
NIK: 201405 880206 1001

Mengetahui  
Ketua Jurusan Teknik Informatika  
  
Tri Astoto Kurnawan, S.T., M.T., Ph.D  
NIP:19710518 200312 1 001



### PERNYATAAN ORSINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsurunsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, Oktober 2018



Vidya Capri styan Pamungkas  
Nim: 135150201111157

## KATA PENGANTAR

*Assalamu'alaikum Wr, Wb.*

Puji syukur kehadiran ALLAH SWT, yang telah memberikan rahmat dan hidayahnya kepada penulis sehingga dapat menyelesaikan skripsi dengan judul Klasifikasi Penerimaan Program Keluarga Harapan (PKH) Menggunakan Metode *Learning vector quantization*. Selama proses penyusunan skripsi ini banyak bantuan yang diterima oleh penulis dari awal penyusunan sampai dengan akhir penulisan. Oleh karena itu dalam kesempatan ini penulis ingin mengucapkan terimakasih kepada:

1. Tuhan yang Maha Esa, Allah SWT.
2. Bapak Sunyoto, Ibu Sri Setyoningsih, Ibu Supiyatun, Yuanita Budi Prastiwi, Hermawan, Adhiastha Athalha Herjuno dan semua keluarga besar yang tiada hentinya selalu memberikan semangat, doa dan dukungan kepada penulis.
3. Ibu Lailil Muflikhah, S.Kom, M.Sc selaku dosen pembimbing I yang telah membimbing, mengarahkan, memberikan saran, kritik dan juga ilmu sehingga penulis dapat menyelesaikan skripsi.
4. Bapak Randy Cahya Wihandika, S.ST., M.Kom selaku dosen pembimbing II yang telah membimbing, mengarahkan, memberikan saran, kritik dan juga ilmu sehingga penulis dapat menyelesaikan skripsi.
5. Bapak Sunyoto selaku pakar dan narasumber yang telah memberikan arahan dan ilmu terkait dengan Program Keluarga Harapan.
6. Teman-teman saya Sambalado( Shima, Vinny, Tari, Ida), Keluarga Cemara(Tony, Brillli, Iman, Rizal, Abas) Mochammad Hadi Santoso, Anggi, dan teman-teman seperjuangan TIF-2013, yang selalu memberikan semangat, ilmu dan doa selama saya kuliah sampai dengan saat ini.
7. Dosen dan Karyawan Fakultas Ilmu Komputer UB.

Dengan kerendahan hati, penulis sangat menyadari bahwa skripsi ini memiliki banyak kekurangan. Oleh karena itu kritik dan saran yang bersifat membangun sangat dibutuhkan untuk penelitian selanjutnya. Penulis berharap semoga skripsi ini dapat bermanfaat bagi diri sendiri maupun bagi banyak pihak.

Malang, Oktober 2018



Vidya Capristyan Pamungkas

Vidyacapristyanp16@gmail.com

## ABSTRAK

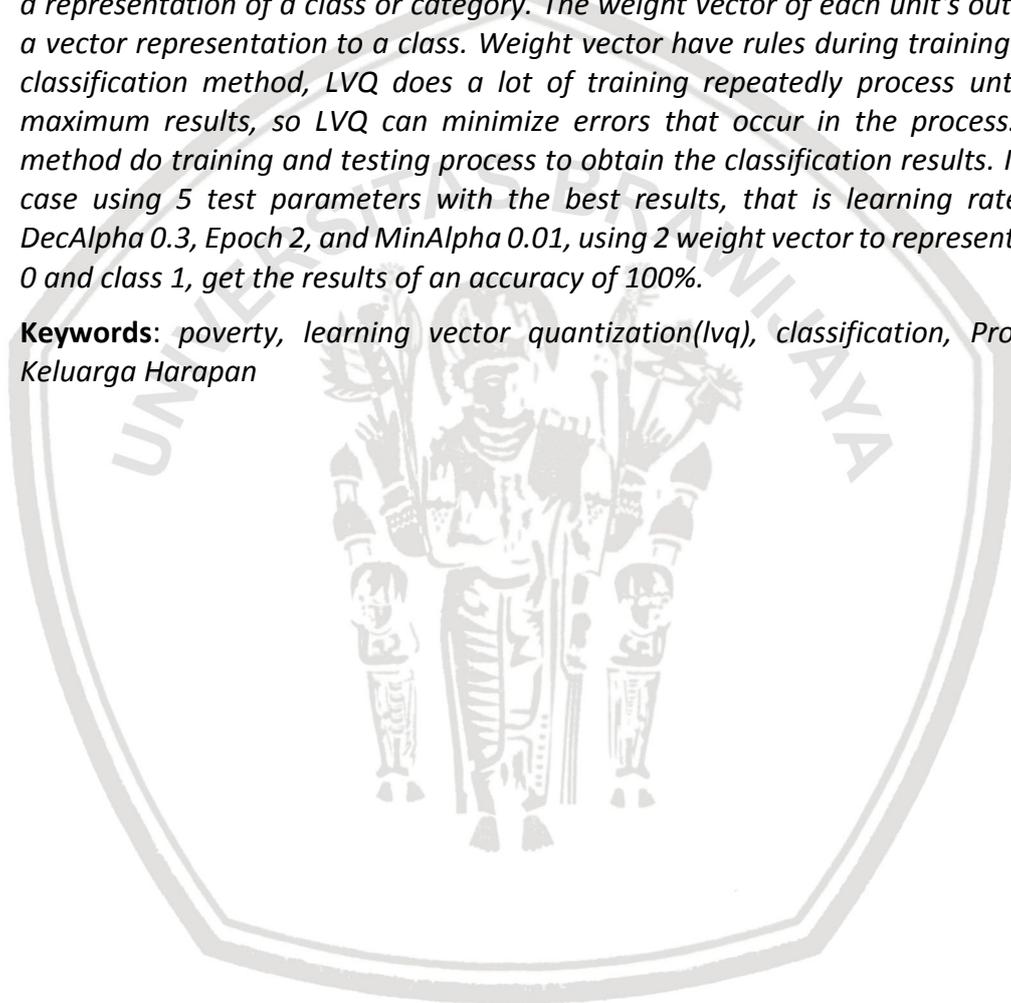
Kemiskinan adalah kondisi ketidakmampuan seseorang untuk memenuhi kebutuhan pokok untuk hidup yang layak. Angka kemiskinan yang terus meningkat, terutama di kabupaten *Jombang* dari tahun ketahun hingga tahun 2017 mencapai 131,16 jiwa, berbagai cara telah dilakukan pemerintah untuk memberantas kemiskinan, salah satunya adalah *Program Keluarga Harapan* atau PKH, didesa *Kedungjati* petugas desa yang melakukan survey kepala keluarga masih menggunakan cara yang manual, yaitu dengan mendatangi setiap kepala keluarga dan mencatat satu persatu kriteria yang dibutuhkan oleh petugas. Sistem klasifikasi penerimaan program keluarga harapan menggunakan *Learning vector quantization* (LVQ). LVQ merupakan metode klasifikasi yang memiliki pola dimana keluaran setiap unit adalah representasi dari kelas atau kategori. Vektor bobot dari keluaran setiap unit merupakan representasi dari vector ke suatu kelas. Vektor bobot memiliki aturan sendiri selama pelatihan. Sebagai metode klasifikasi, LVQ melakukan banyak pelatihan sampai berulang-ulang sampai dengan mendapatkan hasil *maksimum*, sehingga LVQ dapat meminimumkan *error* yang terjadi dalam proses. LVQ melakukan proses pelatihan dan pengujian untuk mendapatkan hasil klasifikasi, pada penelitian ini, menggunakan 5 parameter uji dengan hasil paling maksimal yaitu *learning rate* 0,7, *DecAlpha* 0,3, *Epoch* 2, dan *Minalpha* 0.01 dengan menggunakan 2 bobot yang mewakili kelas 0 dan kelas 1, mendapatkan hasil akurasi sebesar 100%.

**Kata kunci:** kemiskinan, *Learning vector quantization*(LVQ), klasifikkasi, Program Keluarga Harapan, Klasifikasi

## ABSTRACT

Poverty is a condition of someone inability to fulfill basic needs for a decent life. The poverty rate is increases, especially in Jombang Regency from year to year until 2017 reaching 131.16 people, various ways have been carried out by the government to reduce poverty, one of which is Program Keluarga Harapan or PKH, Kedungjati Village officer doing survey head of family with manual method by visiting each head of family and recording one by one the criteria. Classification system of Program Keluarga Harapan using Learning vector quantization (LVQ). LVQ is a classification method that has a pattern where the output of each unit is a representation of a class or category. The weight vector of each unit's output is a vector representation to a class. Weight vector have rules during training. As a classification method, LVQ does a lot of training repeatedly process until get maximum results, so LVQ can minimize errors that occur in the process. LVQ method do training and testing process to obtain the classification results. In this case using 5 test parameters with the best results, that is learning rate 0.7, DecAlpha 0.3, Epoch 2, and MinAlpha 0.01, using 2 weight vector to represent class 0 and class 1, get the results of an accuracy of 100%.

**Keywords:** poverty, learning vector quantization(lvq), classification, Program Keluarga Harapan



## DAFTAR ISI

PENGESAHAN .....	Error! Bookmark not defined.
PERNYATAAN ORSINALITAS .....	Error! Bookmark not defined.
KATA PENGATAR .....	Error! Bookmark not defined.
ABSTRAK .....	iv
ABSTRACT .....	vi
DAFTAR ISI .....	vii
DAFTAR TABEL .....	xi
DAFTAR GAMBAR .....	xiii
DAFTAR LAMPIRAN .....	xiv
BAB 1 PENDAHULUAN .....	2
1.1 Latar belakang .....	2
1.2 Rumusan masalah .....	3
1.3 Tujuan .....	3
1.4 Manfaat .....	3
1.5 Batasan masalah .....	3
1.6 Sistematika pembahasan .....	4
BAB I Pendahuluan .....	4
BAB II Landasan Kepustakaan .....	4
BAB III Metodologi .....	4
BAB IV Perancangan Sistem .....	4
BAB V Implementasi .....	4
BAB VI Pengujian dan Analisis .....	4
BAB VII Penutup .....	5
BAB 2 LANDASAN KEPUSTAKAAN .....	6
2.1 Kajian Pustaka .....	6
2.2 Program Keluarga Harapan (PKH) .....	8
2.2.1 Tujuan Program Keluarga Harapan (PKH) .....	8
2.2.2 Penerima Program Keluarga Harapan dan Kewajibannya .....	9
2.3 Jaringan Syaraf Tiruan .....	9
2.4 <i>Learning vector quantization</i> (LVQ) .....	9
2.4.1 Parameter-parameter Pada <i>Learning vector quantization</i> (LVQ) .....	10

2.4.2	Algoritme <i>Learning Vector Quantization</i> (LVQ)	10
1.6	Pengujian Akurasi	11
BAB 3 METODOLOGI PENELITIAN		12
3.1	Tipe Penelitian	12
3.2	Strategi Penelitian	12
3.3	Lokasi Penelitian	12
3.4	Pengumpulan Data	12
3.5	Data Penelitian	13
3.6	Implementasi Algoritme	13
3.7	Jadwal penelitian	13
BAB 4 PERANCANGAN		14
4.1	Deskripsi Proses Kerja Sistem Secara Umum	14
4.2	Proses Pelatihan <i>Learning vector quantization</i> (LVQ)	15
4.3	Pengujian <i>Learning vector quantization</i> (LVQ)	17
4.5	Pelatihan <i>Learning vector quantization</i> (LVQ)	17
4.5.1	Menghitung Jarak <i>Euclidean</i>	19
4.5.2	Cek Kondisi <i>Update</i>	20
4.5.3	Perbarui Nilai <i>Alpha</i>	23
4.5.4	Cek <i>Error</i> Minimum	23
4.6	Perhitungan Manual <i>Learning vector quantization</i> (LVQ)	23
4.6.1	Tahap Awal	23
4.6.2	Hitung Jarak <i>Euclidean</i>	23
4.7	Perancangan Antarmuka	26
4.7.1	Antarmuka Halaman Awal Sistem	26
4.7.2	Antarmuka Halaman Training dan Testing	27
4.7.3	Antarmuka Halaman Input	27
4.7.4	Antarmuka Halaman Data	28
4.7.5	Antarmuka Halaman Parameter	29
4.8	Perancangan Pengujian	29
4.8.1	Pengujian Pengaruh Nilai <i>Learning Rate</i> ( <i>Alpha</i> ) Terhadap Hasil Akurasi	29
4.8.2	Pengujian Pengaruh Pengali <i>Learning rate</i> ( <i>DecAlpha</i> ) Terhadap Hasil Akurasi	30

4.8.3	Pengujian Pengaruh Maksimum <i>Epoch</i> Terhadap Hasil Akurasi .	31
4.8.4	Pengujian Pengaruh Jumlah Data Latih Terhadap Hasil Akurasi .	31
4.8.5	Pengujian Pengaruh Minimum <i>Learning Rate</i> Terhadap Hasil Akurasi	32
BAB 5 IMPLEMENTASI		34
5.1	Perangkat Keras	34
5.2	Perangkat Lunak	34
5.3	Implementasi Program Metode <i>Learning vector quantization</i> .	35
5.3.1	Proses Pelatihan dan Pengujian <i>Learning vector quantization</i> (LVQ)	35
5.3.1.1	Proses inisialisasi bobot	35
5.3.2	Proses Pengujian <i>Learning vector quantization</i> (LVQ)	39
5.4	Implementasi Antarmuka Sistem	40
5.4.1	Antarmuka Halaman Awal Sistem	40
5.4.2	Antarmuka Halaman <i>Training</i> dan <i>Testing</i>	41
5.4.3	Antarmuka Halaman Input Data	42
5.4.4	Antarmuka Halaman Tampilkan Data	42
BAB 6 PENGUJIAN DAN ANALISIS		44
6.1	Pelatihan Metode <i>Learning vector quantization</i>	44
6.1.1	Pengaruh Nilai <i>Learning rate</i> ( <i>Alpha</i> ) Terhadap Hasil Akurasi	44
6.1.2	Pengaruh Pengali <i>Learning rate</i> ( <i>DecAlpha</i> ) Terhadap Hasil Akurasi	45
6.1.3	Pengaruh Maksimum iterasi ( <i>Epoch</i> ) Terhadap Hasil Akurasi	46
6.1.4	Pengaruh Jumlah Data Latih Terhadap Hasil Akurasi	47
6.1.5	Pengaruh Minimum <i>Learning Rate</i> Terhadap Hasil Akurasi	49
6.2	Pengujian Metode <i>Learning vector quantization</i>	50
BAB 7 PENUTUP		51
7.1	Kesimpulan	51
7.2	Saran	51
DAFTAR PUSTAKA		52
LAMPIRAN		53
LAMPIRAN A DATA LATIH		53

LAMPIRAN B DATA UJI ..... 56  
LAMPIRAN C NILAI BOBOT DATA PROGRAM KELUARGA HARAPAN ..... 58  
LAMPIRAN D SURAT KESEDIAAN PAKAR ..... 59



## DAFTAR TABEL

Tabel 2.1 Kajian pustaka .....	6
Tabel 4.1 Data latih PKH.....	18
Tabel 4.2 Bobot awal pelatihan .....	19
Tabel 4.3 Data ke-1 pelatihan .....	19
Tabel 4.4 Hasil Perhitungan jarak antara data latih dengan bobot.....	19
Tabel 4.5 Update bobot baru .....	20
Tabel 4.6 Data kedua pelatihan .....	20
Tabel 4.7 Hasil perhitungan jarak data kedua .....	20
Tabel 4.8 Hasil update bobot .....	20
Tabel 4.9 Data ketiga pelatihan .....	21
Tabel 4.10 Hasil perhitungan jarak .....	21
Tabel 4.11 Update bobot .....	21
Tabel 4.12 Data keempat pelatihan.....	21
Tabel 4.13 Hasil perhitungan jarak .....	21
Tabel 4.14 Update bobot .....	22
Tabel 4.15 Data kelima pelatihan .....	22
Tabel 4.16 Hasil perhitungan jarak .....	22
Tabel 4.17 Bobot klasifikasi LVQ.....	23
Tabel 4.18 Data kesatu pengujian.....	23
Tabel 4.19 Hasil perhitungan jarak .....	23
Tabel 4.20 Data uji kedua.....	24
Tabel 4.21 Hasil perhitungan jarak .....	24
Tabel 4.22 Data uji ketiga.....	24
Tabel 4.23 Hasil perhitungan jarak .....	24
Tabel 4.24 Data uji ke empat .....	25
Tabel 4.25 Hasil perhitungan jarak .....	25
Tabel 4.26 Data uji kelima.....	25
Tabel 4.27 Hasil perhitungan jarak .....	25
Tabel 4.28 Perancangan pengujian pengaruh learnig rate .....	30
Tabel 4.29 Perancangan pengujian pengaruh pengali <i>Learning rate</i> .....	30
Tabel 4.30 Perancangan pengujian maksimum epoch .....	31

Tabel 4.31 Perancangan pengujian pengaruh jumlah data latih ..... 32

Tabel 4.32 Perancangan pengujian nilai minimum *alpha*..... 32

Tabel 5.1 Perangkat keras ..... 34

Tabel 5.2 Perangkat lunak..... 34

Tabel 6.1 Hasil pengujian *learning rate* ..... 44

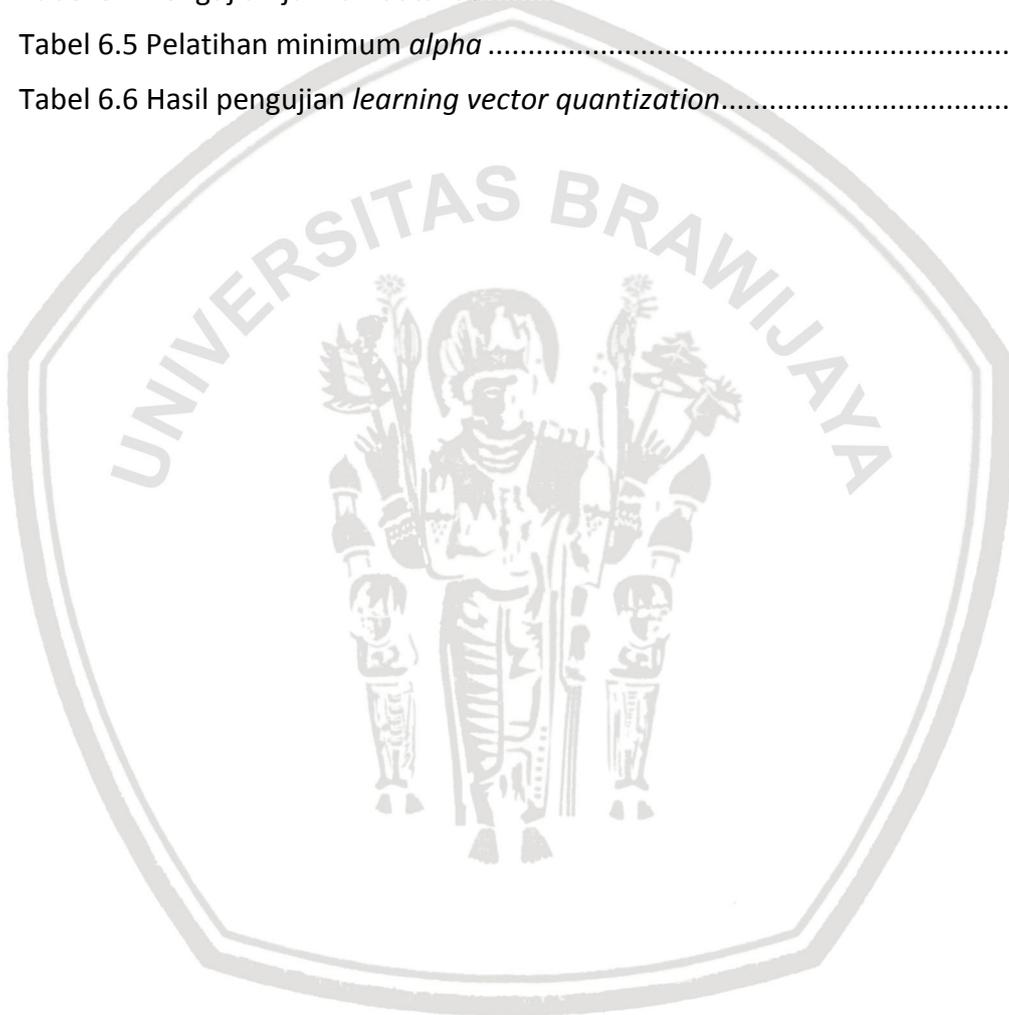
Tabel 6.2 Hasil pengujian pengali *learning rate*..... 45

Tabel 6.3 Perancangan pengujian *epoch* ..... 47

Tabel 6.4 Pengujian jumlah data latih..... 48

Tabel 6.5 Pelatihan minimum *alpha* ..... 49

Tabel 6.6 Hasil pengujian *learning vector quantization*..... 50



## DAFTAR GAMBAR

Gambar 2.1 Gambar arsitektur <i>learning vector quantization</i> .....	10
Gambar 4.1 Bagan perancangan .....	14
Gambar 4.2 Flowchart sistem .....	15
Gambar 4.3 Flowchart pelatihan <i>learning vector quantization</i> .....	16
Gambar 4.4 Flowchart pengujian <i>learning vector quantization</i> .....	17
Gambar 4.5 Perancangan antarmuka halaman awal sistem .....	26
Gambar 4.6 Perancangan antarmuka halaman login admin .....	27
Gambar 4.7 Perancangan antarmuka halaman klasifikasi .....	28
Gambar 4.8 Perancangan antarmuka halaman hasil klasifikasi .....	28
Gambar 4.9 Halaman antarmuka parameter .....	29
Gambar 5.1 Halaman awal sistem .....	40
Gambar 5.2 Halaman Training dan Testing .....	41
Gambar 5.3 Halaman input data .....	42
Gambar 5.4 Halaman tampilkan data .....	43
Gambar 6.1 Grafik hasil pelatihan <i>learning rate</i> .....	45
Gambar 6.2 Grafik pelatihan pengali <i>learning rate</i> .....	46
Gambar 6.3 Grafik pengujian maksimal epoch .....	47
Gambar 6.4 Grafik hasil pengujian jumlah data latih .....	48
Gambar 6.5 Grafik hasil pelatihan <i>MinAlpha</i> .....	49

## DAFTAR LAMPIRAN

Lampiran A Data Latih .....	57
Lampiran B Data Uji.....	60
Lampiran C Nilai Bobot Data Program Keluarga Harapan.....	62
Lampiran D Surat Kesiediaan Pakar .....	63



**DAFTAR RIWAYAT HIDUP****I. KETERANGAN PERORANGAN**

1	Nama lengkap	Vidya Capristyan Pamungkas	
2	NIK	3517166712940001	
3	Tempat Lahir / Tgl. Lahir	Jombang 27 Desember 1994	
4	Jenis Kelamin	Wanita	
5	A g a m a	Islam	
6	Perguruan Tinggi	Universitas Brawijaya	
7	Pendidikan Terakhir	SMA	
8	Program Studi	Teknik Informatika	
9	IPK	2.82	
10	Telepon/HP	081330789190	
11	Email	<a href="mailto:Vidyacapristyanp16@gmail.com">Vidyacapristyanp16@gmail.com</a>	
12	Status perkawinan	Belum kawin	
13	Alamat Rumah	a. Jalan	
		b. Kelurahan / Desa	Kedungjati
		c. Kecamatan	Kabuh
		d. Kabupaten / Kota	Jombang
		e. Propinsi	Jawa Timur
14	Kegemaran (Hobby)	Berenang	

## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Kemiskinan adalah sebuah kondisi ketidakmampuan seseorang untuk memenuhi kebutuhan dasar dalam kehidupan sehari-hari (BPS dan Depsos, 2002). Dimana individu atau masyarakat tersebut cukup atau kurang dalam memenuhi kebutuhannya. Seseorang disebut miskin juga dapat dilihat dari pendapatannya, apabila pendapatannya lebih rendah dari pada rata-rata pendapatan di daerah tersebut, sehingga masyarakat tersebut tidak dapat memenuhi kebutuhan dan tidak dapat mensejahterkan dirinya (Suryawati, 2004). Kondisi masyarakat yang dapat disebut miskin dapat diketahui berdasarkan kemampuan masyarakat tersebut dalam mengolah pendapatan untuk memenuhi standar hidup (Nugroho, 1995).

Berbagai cara telah dilakukan pemerintah untuk memberantas atau mengatasi kemiskinan, mulai dari adanya program BLT (Bantuan langsung tunai), Jaminan kesehatan masyarakat, dana BOS, beras miskin, PNPM mandiri dan juga Program Keluarga Harapan. Salah satu kebijakan sosial yang dikembangkan oleh pemerintah ini adalah Program Keluarga Harapan, yang menjadi sasaran adalah keluarga miskin dan hampir dari semua anggota keluarga bisa mendapatkan bantuan tersebut, mulai dari balita, anak sekolah, ibu hamil juga lansia. Bantuan PKH ini diberikan kepada penerima dana sebanyak 4 kali dalam 1 tahun. Sebagai imbalannya para penerima bantuan ini memiliki kewajiban selama itu, yaitu memberikan laporan kepada petugas, bukti penggunaan dana tersebut sudah sesuai dengan yang seharusnya.

Angka kemiskinan di kabupaten *Jombang* sendiri dari tahun 2012 sampai dengan 2017 terus naik hingga mencapai 131,16 jiwa (BPS Jombang, 2017). Desa *Kedungjati* merupakan suatu desa yang letaknya berada di utara kabupaten *Jombang*. Di desa *Kedungjati* juga melaksanakan program dari pemerintah yakni Program Keluarga Harapan, dalam melaksanakan tugasnya, perangkat desa *Kedungjati* mensurvey satu persatu kepala keluarga dan mendatangi kediamannya. Setelah melengkapi semua data yang dibutuhkan, selanjutnya petugas harus memilah tiap kepala rumah tangga yang sudah didata dengan menggunakan cara manual, yaitu dengan menulis satu persatu data yang dibutuhkan oleh petugas. Karena itu sering menyita banyak waktu dan tenaga. Maka dibutuhkan suatu sistem yang dapat mengklasifikasikan Program Keluarga Harapan dengan cepat.

*Learning vector quantization* (LVQ) merupakan metode klasifikasi yang memiliki pola dimana keluaran setiap unit adalah representasi dari kelas atau kategori. Vektor bobot dari keluaran setiap unit merupakan representasi dari vektor ke suatu kelas. Vektor bobot memiliki aturan sendiri selama pelatihan (Kusumadewi, 2003). Untuk klasifikasi, LVQ melakukan banyak pelatihan sampai berulang-ulang sampai dengan mendapatkan hasil *maksimum*, sehingga LVQ memiliki kelebihan yaitu meminimumkan *error* yang terjadi dalam proses. LVQ juga memiliki kelebihan lain yaitu mampu mengatur bobot pada proses pembelajaran atau pelatihannya (Fausett, 1994). Proses yang dilakukan PLVQ adalah proses pelatihan dan proses pengujian, pada proses pelatihan LVQ melakukan pencarian jarak terdekat dengan

menggunakan *Euclidean*, antara data dengan bobot 1 dan bobot 2, setelah mendapatkan jarak terdekat, maka bobot diupdate sesuai dengan ketentuan kondisi yang didapatkan. Bobot terakhir yang didapatkan akan digunakan dalam proses pengujian, proses pengujian hampir sama dengan proses pelatihan, hanya saja pada proses pengujian tidak diperlukan update bobot.

Salah satu penelitian yang membahas tentang metode *learning vector quantization* adalah penelitian yang dilakukan oleh Rio Arifando (2017), yang berjudul “Klasifikasi Calon Penerima Bantuan Keluarga Miskin Menggunakan Metode *Learning vector quantization* (LVQ) (Studi Kasus: Daerah Kecamatan Mlandingan, Situbondo)” penelitian tersebut mengklasifikasi calon penerima BKM atau Bantuan Keluarga Miskin, menggunakan 7 kriteria yang dibutuhkan untuk proses klasifikasi, 5 parameter skenario pengujian, dengan hasil pengujian adalah *learning rate* yang digunakan adalah 0,1, pengali *alpha* 0,1, *epoch* sebanyak 2, dan *minalpha* 0,01, hasil yang didapatkan sebesar 98%.

## 1.2 Rumusan masalah

1. Bagaimana mengimplementasikan metode *Learning vector quantization* (LVQ) dalam klasifikasi penerimaan Program Keluarga Harapan menggunakan metode *Learning vector quantization* (LVQ)?
2. Berapakah hasil akurasi metode *Learning vector quantization* (LVQ) untuk klasifikasi penerimaan Program Keluarga Harapan?

## 1.3 Tujuan

1. Menerapkan metode *Learning vector quantization* (LVQ) untuk klasifikasi penerimaan Program Keluarga Harapan.
2. Mengetahui tingkat akurasi metode *Learning vector quantization* (LVQ) dari hasil klasifikasi penerimaan Program Keluarga Harapan.

## 1.4 Manfaat

1. Bagi perangkat desa *kedungjati* sebagai petugas pendataan penerimaan Program Keluarga Harapan, hasil dari penelitian dapat membantu petugas untuk membandingkan hasil pendataan masyarakat penerima Program Keluarga Harapan dengan metode *learning vector quantization*.
2. Bagi peneliti, hasil dari penelitian ini dapat digunakan sebagai referensi untuk pengembangan penelitian baru dibidang yang sama atau dibidang lain.

## 1.5 Batasan masalah

1. Penelitian ini dibuat hanya untuk klasifikasi penerimaan Program Keluarga Harapan menggunakan metode *learning vector quantization*.
2. Data yang digunakan adalah data penduduk desa *kedungjati* kecamatan *kabuh*, kabupaten *jombang* yang menerima Program Keluarga Harapan.

## 1.6 Sistematika pembahasan

Sistematika penulisan laporan ditunjukkan untuk memberikan gambaran dan uraian dari proposal skripsi klasifikasi penerimaan Program Keluarga Harapan menggunakan metode *Learning vector quantization* secara garis besar yang meliputi beberapa bab, sebagai berikut:

### BAB I Pendahuluan

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan laporan penelitian klasifikasi penerimaan Program Keluarga Harapan menggunakan metode *Learning vector quantization*.

### BAB II Landasan Kepustakaan

Bab ini membahas uraian dan pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah, yang berkaitan dengan tema, masalah, atau pertanyaan penelitian klasifikasi penerimaan Program Keluarga Harapan dengan menggunakan metode *learning vector quantization* (LVQ).

### BAB III Metodologi

Bab ini membahas mengenai langkah-langkah atau metodologi penelitian untuk menyelesaikan tahap pembuatan sistem klasifikasi Program Keluarga Harapan dengan menggunakan metode *learning vector quantization* diantaranya tipe penelitian, strategi penelitian, lokasi penelitian, pengumpulan data, data penelitian, implementasi metode, peralatan pendukung dan juga jadwal penelitian.

### BAB IV Perancangan Sistem

Bab ini menjelaskan mengenai perancangan yang digunakan dalam system, meliputi deskripsi proses kerja sistem secara umum, klasifikasi *Learning vector quantization*, proses pelatihan dengan *Learning vector quantization*, proses pengujian dengan *Learning vector quantization*, perhitungan manual, dan perancangan antarmuka.

### BAB V Implementasi

Pada bab ini akan dibahas mengenai lingkungan implementasi, implementasi program, serta implementasi antarmuka untuk klasifikasi Program Keluarga Harapan Menggunakan Metode *Learning vector quantization*.

### BAB VI Pengujian dan Analisis

Bab ini membahas mengenai tahap pengujian terhadap sistem “Klasifikasi Penerimaan Program Keluarga Harapan (PKH) Menggunakan Metode *Learning vector quantization*”..

## BAB VII Penutup

Bab ini menjelaskan kesimpulan dari penelitian yang telah dilakukan dan saran untuk pengembangan lebih lanjut.



## BAB 2 LANDASAN KEPUSTAKAAN

Bab landasan kepastakaan ini berisi penjeladsan serta uraian dari kajian pustaka, konsep dan juga teori ataupun metode yang berhubungan dengan lvq atau pkh. Pada bab ini terdapat beberapa dasar teori dari beberapa sumber pustaka penelitian lain yang menunjang penelitian ini.

### 2.1 Kajian Pustaka

Beberapa penelitian yang sebelumnya telah dilakukan dan terkait dengan kasus yaitu Program Keluarga Harapan, juga metode yaitu *Learning Vector Quantization* (LVQ). Oleh karena itu berikut adalah kajian pustaka yang dijelaskan pada Tabel 2.1.

**Tabel 2.1 Kajian pustaka**

No	Nama Peneliti	Obyek	Metode	Hasil
1.	Eka Afriandi	Telapak tangan	<i>Learning Vector Quantization</i> (LVQ).	Pada penelentain dengan judul “Identifikasi Telapak Tangan Menggunakan Jaringan Syaraf Tiruan <i>Learning vector quantization</i> ” menggunakan perangkat lunak waterfall, memproses bagaimana suatu sistem dapat mengenali telapak tangan user, dengan menggunakan bahasa pemrograman matlab dan system basis datanya Microsoft Access, sistem dapat melakukan identifikasi telapak tangan antara user satu dengan yang lain dengan tingkat akurasi sebesar 74,66% (Afriandi, 2016)
2.	M’Sa’idul Umam	Citra daun tanaman kedelai	<i>Learning Vector Qauntization</i> (LVQ)	Penelitian dengan judul “Implementasi Metode <i>Learning vector quantization</i> (Lvq) Untuk Identifikasi Penyakit Pada Citra Daun Tanaman Kedelai” ini melakukan sebuah identifikasi dengan menerapkan metode lvq untuk mendeteksi penyakit pada citra daun kedelai, dengan melakukan beberapa

				<p>pegujian seperti mengubah nilai threshold (C), bobot awal, nilai alpha, nilai pengurang alpha dan jumlah data latih. Dari kelima skenario pengujian tersebut didapatkan akurasi paling maksimal adalah 93%.</p> <p>(Umam, 2015)</p>
3.	Diana Wijayanti MP	Kulit hasil perawatan	<i>Learning Vector Qauntization(LVQ)</i>	<p>Penelitian dengan judul "Identifikasi Diagnosis Perubahan Hasil Perawatan Kulit Menggunakan Metode <i>Learning vector quantization (Lvq)</i>", keluaran dari program atau teknologi ini adalah membandingkan kulit wajah sebelum dan sesudah perawatan dengan cara melihat nilai red, green dan blue(RGB) pada citra kulit orang tersebut, dengan melakukan beberapa pengujian, didapatkan hasil paling baik yaitu 92,31%.</p> <p>(Wijayanti, 2016)</p>
4.	Rifwan hamidi	Air sungai	<i>Learning vector quantization</i>	<p>Penelitian dengan judul "Implementasi <i>Learning vector quantization (LVQ)</i> untuk Klasifikasi kualitas Air Sungai' ini mengklasifikasikan kualitas air sungai, dengan menggunakan 4 kelas, pengujian dilakukan dengan menggunakan 7 parameter yang akan memberikan hasil kelas berupa apakah air tersebut memenuhi baku mutu, tercemar ringan, sedang ataukah berat. Dari ketujuh parameter didapatkan hasil paling baik yaitu 81,13%.</p> <p>(Hamidi, 2017)</p>

5.	Ivan Agustinus	Hipertensi	<i>Learning vector quantization</i>	<p>Penelitian dengan judul “Klasifikasi Risiko Hipertensi Menggunakan Metode <i>Learning vector quantization</i> (LVQ)” ini mendiagnosis penyakit hipertensi dengan membagi 4 kelas dan 12 fitur. Pengujian dilakukan dengan menggunakan 6 skenario dan hasil dari sistem adalah menghasilkan diagnosis yang dialami oleh pasien. Dari pengujian, didapatkan hasil terbaik yaitu 93,84% (Agustinus 2017).</p>
----	----------------	------------	-------------------------------------	---

## 2.2 Program Keluarga Harapan (PKH)

Program Keluarga Harapan atau sering disebut dengan PKH adalah salah satu program bantuan dari pemerintah yang melibatkan semua anggota keluarga, kepala rumah tangga harus memenuhi kualifikasi tertentu yang akan dipilih untuk mendapatkan bantuan tersebut, kualifikasi atau syarat untuk rumah tangga yang bisa mendapatkan bantuan tersebut dapat dibandingkan dengan melihat luas tanah sempit, jenis lantai bangunan, jenis dinding bangunan, wc, sumber air yang digunakan untuk sehari-hari, sumber penerangan dirumah, bahan bakar yang digunakan untuk memasak, mampu atau tidak membayar anggota keluarga ke puskesmas, memiliki atau tidak ibu hamil atau anak balita, dan apakah memiliki anak yang sedang bersekolah. Dari beberapa kriteria tersebut rumah tangga yang berhak atau memenuhi, akan mendapat bantuan sesuai dengan anggaran biaya pemerintah pertahunnya. Dana tersebut akan diberikan 4kali dalam setahun, para penerima bantuan PKH ini berhak memenuhi sebuah kewajiban setiap bulannya bahwa dana tersebut digunakan untuk semestinya.

### 2.2.1 Tujuan Program Keluarga Harapan (PKH)

Beberapa tujuan dari pemberian bantuan PKH adalah:

1. Mengurangi angka kemiskinan dengan meningkatkan taraf hidup penerima dengan mengakses layanan pendidikan, kesehatan yang layak dan juga kesejahteraan sosial.

2. Mengurangi beban untuk pengeluaran rumah tangga.
3. Menciptakan kemandirian dan juga perubahan perilaku masyarakat.

### 2.2.2 Penerima Program Keluarga Harapan dan Kewajibannya.

Rumah tangga, yang mendapatkan bantuan dana program keluarga harapan, memiliki kewajiban dalam setahun tersebut untuk memberika laporan bahwa dana bantuan tersebut digunakan bagaimana semestinya, kewajiban yang harus dilakukan adalah sebagai berikut:

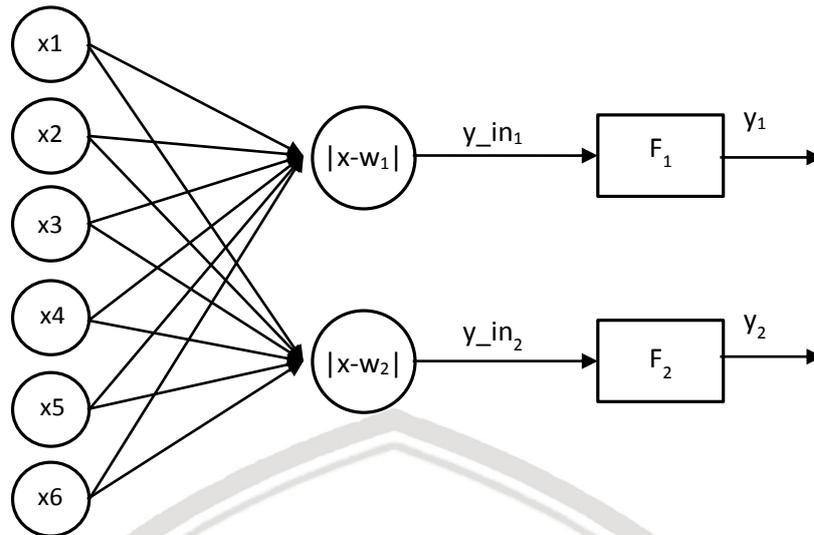
1. Untuk Ibu hamil memiliki kewajiban, yaitu memeriksakan kehamilan difaskes sebanyak minimal 3 kali dalam 3x trisemester, melahirkan di puskesmas dengan ahli tenaga kesehatan, pemeriksaan kesehatan atau imunisasi 2 kali sebelum bayi usia 1 bulan.
2. Untuk bayi, memiliki kewajiban saat diusia 0-11 bulan, imunisasi lengkap dengan pemberian suplemen vitamin A, serta pemeriksaan berat badan setiap bulan. Dan pada usia 6-11 bulan.
3. Balita, memiliki kewajiban, pada usia 1-5 tahun pemeriksaan kesehatan yang meliputi imunisasi dan juga penimbangan berat badan . Pada usia 5-6 tahun, mendapatkan vitamin sesuai dengan kebutuhan anak tersebut. Usia 6-7 tahun, imunisasi secara lengkap.
4. Pelajar, memiliki kewajiban, diusia 6-21 tahun, harus menyelesaikan pendidikan ( SD, SMP, SLTA ), dan minima kehadirannya 80%.
5. Penyandang disabilitas berat, kewajibannya, melakukan pemeriksaan kesehatan yang dapat dilakukan oleh tenaga kesehatan atau melalui kunjungan kerumah (*home care*).

### 2.3 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (JST) implementasi dari teknologi artificial intelligence. Metode ini merancang sebuah komputasi cerdas, sehingga dapat merepresentasikan otak dari manusia. Jst merancang model sistem komputasi yang dapat menirukan cara kerja jaringan saraf biologi yang diimplementasikan menggunakan program komputer dan mampu menyelesaikan proses pembelajaran (Kusumadewi, 2003).

### 2.4 Learning vector quantization (LVQ)

*Learning vector quantization* (LVQ) merupakan salah satu metode dari jaringan syaraf tiruan yang dapat digunakan dalam mengklasifikasi data, pada prosesnya terdapat proses pembelajaran dan proses pengujian. Lvq memiliki tipe arsitektur, yaitu menggunakan *Single-layer Feedforward* yang terdapat vektor input dan unit output. *Euclidean Distance* digunakan untuk perhitungan pengelompokan suatu vector input dan output (Kusumadewi, 2003). Arsitektur *learning vector quantization* dapat dilihat pada Gambar 2.1.



**Gambar 2.1** Gambar arsitektur *learning vector quantization*

Sumber : Kusumadewi, 2003

Dari gambar diatas, dapat dilihat bahwa X adalah input, w adalah vector bobot dan y adalah hasil dari proses. L<sub>vq</sub> melakukan pencarian jarak terdekat dengan menggunakan *eulidean distance*.

#### 2.4.1 Parameter-parameter Pada *Learning vector quantization* (LVQ)

Parameter-parameter yang digunakan pada metode LVQ ini adalah sebagai berikut:

1.  $\alpha$  (*Learning rate*) *Alpha* parameter yang digunakan untuk pengendalian bobot yang digunakan untuk proses perhitungan.
2. *DecAlpha* adalah Pengkali *Learning rate*, digunakan pada setiap iterasi pada proses pelatihan.
3. *MinAlpha* (*Minimum Learning rate*) Yaitu kondisi untuk berhentinya iterasi, dengan kondisi  $\alpha < \text{MinAlpha}$ .
4. *MaxEpoch* (*Maksimum epoch*) Yaitu jumlah iterasi yang menjadi batas dalam perulangan selama proses pelatihan.

#### 2.4.2 Algoritme *Learning Vector Quantization* (LVQ)

Dalam prosesnya metode LVQ memiliki 2 tahapan, yang pertama adalah proses pelatihan, yang kedua adalah proses pengujian. Tahapan pelatihan pada metode LVQ ini dapat dituliskan dalam bentuk algoritme sebagai berikut (Fauset, 1994) :

Step 0. Inisialisasi data latih serta parameter awal, *Alpha*, *DecAlpha*, *Epoch*, *MinAlpha* dan juga bobot awal yang didapatkan dari data yang mewakili kelas 1 dan kelas 0.

Step 1. Perulangan hingga kondisi terpenuhi lakukan langkah 2-6

- Step 2. Untuk tiap data latih (X) yang ada lakukan langkah 3-4
- Step 3. Hitung jarak minimum dengan menggunakan Euclidean  $||x - W_j||$
- Step 4. Update  $W_j$  dengan ketentuan :

Jika  $T = C_j$  maka :

$$W_j(new) = W_j(old) + \alpha [X - W_j(old)] \tag{2.1}$$

Jika  $T \neq C_j$  maka :

$$W_j(new) = W_j(old) - \alpha [X - W_j(old)] \tag{2.2}$$

Step 5. Update nilai  $\alpha$  dengan cara kalikan  $\alpha$  dengan pengkali  $\alpha$ .

Step 6. Kondisi berhentinya perulangan berdasarkan iterasi maksimal atau  $\alpha < MinAlpha$

Keterangan dari algoritma yaitu :

- $\alpha$  = Learning Rate
- X = Data (x1 ,x2, .....xn)
- T = Kelas dari label yang sudah ada.
- $W_j$  = Bobot yang sudah ditentukan untuk output ke-j
- $C_j$  = Hasil kelas yang dihasilkan dari data ke-j
- $|| X - W_j ||$  = Jarak *Euclidean* antara data latih dengan bobot vector ke-j

### 1.6 Pengujian Akurasi

Pengujian yang dilakukan ini bertujuan untuk melihat berapa besar keberhasilan sistem dalam mengklasifikasi. Untuk mendapatkan akurasi dapat dihitung melalui presentase kebenaran dimana perbandingan antara jumlah data benar dari hasil keluaran sistem yang dibuat dengan jumlah total data yang digunakan lalu dikalikan 100. Akurasi dapat dinyatakan dalam persamaan 2.4.

$$Akurasi = \frac{\text{Jumlah data benar hasil program}}{\text{jumlah total data}} \times 100\% \tag{2.3}$$



## BAB 3 METODOLOGI PENELITIAN

Bab ini membahas mengenai langkah-langkah atau metodologi penelitian untuk menyelesaikan tahap pembuatan sistem klasifikasi Program Keluarga Harapan dengan menggunakan metode *learning vector quantization* diantaranya tipe penelitian, strategi penelitian, lokasi penelitian, pengumpulan data, data penelitian, implementasi metode, peralatan pendukung dan juga jadwal penelitian.

### 3.1 Tipe Penelitian

Tipe penelitian yang digunakan pada penelitian ini, adalah penelitian non-implimentatif analitik. Tipe penelitian ini merupakan tipe penelitian yang dilakukan berdasarkan permasalahan atau fenomena tertentu yang selanjutnya akan diteliti dan dianalisis sehingga menghasilkan penyelesaian dari masalah tersebut. Pada penelitian ini, mengangkat permasalahan yang ada di desa *kedungjati*, yaitu dalam pengklasifikasian Program Keluarga Harapan yang masih manual, maka penelitian ini membuat suatu sistem dengan menggunakan lvq untuk mengklasifikasi penerimaan Program Keluarga Harapan. Jenis penelitian ini menggunakan data kualitatif, dimana data yang digunakan adalah hasil survey yang dilakukan petugas desa ke setiap rumah tangga yang ada di desa *kedungjati*.

### 3.2 Strategi Penelitian

Untuk strategi penelitian ini, penulis menggunakan metode *learning vector quantization* (lvq) untuk mengklasifikasi penerimaan Program Keluarga Harapan, untuk dapat menghasilkan klasifikasi, lvq melakukan proses pembelajaran atau pelatihan, dengan menggunakan 5 parameter, yaitu *learnig rate*, pengkali *learnig rete*, *epoch*, jumlah data latih dan juga *minalpha*. Diperlukan bobot awal dan menentukan parameter awal, setelah itu proses pelatihan dilakukan dengan mencari jarak terdekat dari data dengan bobot 1 dan bobot 2, untuk selanjutnya mencocokkan kelas data dengan hasil bobot terdekat, kemudian mengupdate bobot terbaru, bobot terbaru ini yang akan selanjutnya digunakan untuk pelatihan data selanjutnya. Dilakukan sampai data terakhir, bobot yang didapatkan pada data terakhir, akan digunakan pada proses pengujian. Pada proses pengujian, proses hampir sama, hanya saja pada proses pengujian tidak perlu melakukan proses *update* bobot.

### 3.3 Lokasi Penelitian

Lokasi penelitian dilakukan di desa *Kedungjati*, Kecamatan *Kabuh*, Kabupaten *Jombang*, Jawa timur. Dari desa *kedungjati* penulis mendapatkan data berupa 10 kriteria perkepala rumah tangga yang dibutuhkan untuk proses klasifikasi.

### 3.4 Pengumpulan Data

Pengumpulan data pada penelitian ini didapatkan dari penerima Program Keluarga Harapantahun 2017 pada desa *kedungjati* kecamatan *kabuh*, kabupaten

*jombang*. Dari survey petugas ke setiap rumah tangga yang ada di desa *kedungjati*. Data tersebut berisi nama-nama kepala keluarga yang disurvei oleh petugas beserta 10 kriteria yang dibutuhkan untuk digunakan pertimbangan penerimaan Program Keluarga Harapan dan juga kriteria yang digunakan untuk pertimbangan pemberian Program Keluarga Harapan yaitu luas tanah sempit, jenis lantai bangunan, jenis dinding bangunan, wc, sumber air yang digunakan untuk sehari-hari, sumber penerangan dirumah, bahan bakar yang digunakan untuk memasak, mampu atau tidak membayar anggota keluarga ke puskesmas, memiliki atau tidak ibu hamil atau anak balit, dan apakah memiliki anak yang sedang bersekolah.

### 3.5 Data Penelitian

Data penelitian ini menggunakan nama-nama kepala rumah tangga dengan kriteria yang dibutuhkan untuk mengklasifikasi calon penerima bantuan. Total data yang didapatkan adalah 150 data. Data latih terdiri dari 100 dan data uji terdiri dari 50. 2 kelas yaitu berhak menerima yang mewakili kelas 1 dan tidak menerima yang mewakili kelas 0.

### 3.6 Implementasi Algoritme

Data yang didapat bersifat kualitatif, untuk dapat melakukan proses lvq, maka data tersebut diubah menjadi kuantitatif, dari pakar, penulis diberikan data yaitu dengan hasil 30, 60, 90, 50, dan 100. Untuk 30, adalah untuk kriteria paling bawah atau kriteria miskin, untuk 50 dan 60 adalah kriteria yang menengah atau cukup, 90 dan 100 merupakan kriteia dengan kata lain adalah kaya. Utuk selanjutnya diperlukan bobot awal dan parameter uji. Lvq melakukan proses pelatihan atau pembelajaran dengan menggunakan *Euclidean distance*. Yaitu dengan memproses data dengan tiap bobot. Setelah mendapatkan jarak, maka akan mengupdate bobot sesuai dengan ketentuan. Bobot yang didapat akan digunakan untuk proses pengujian.

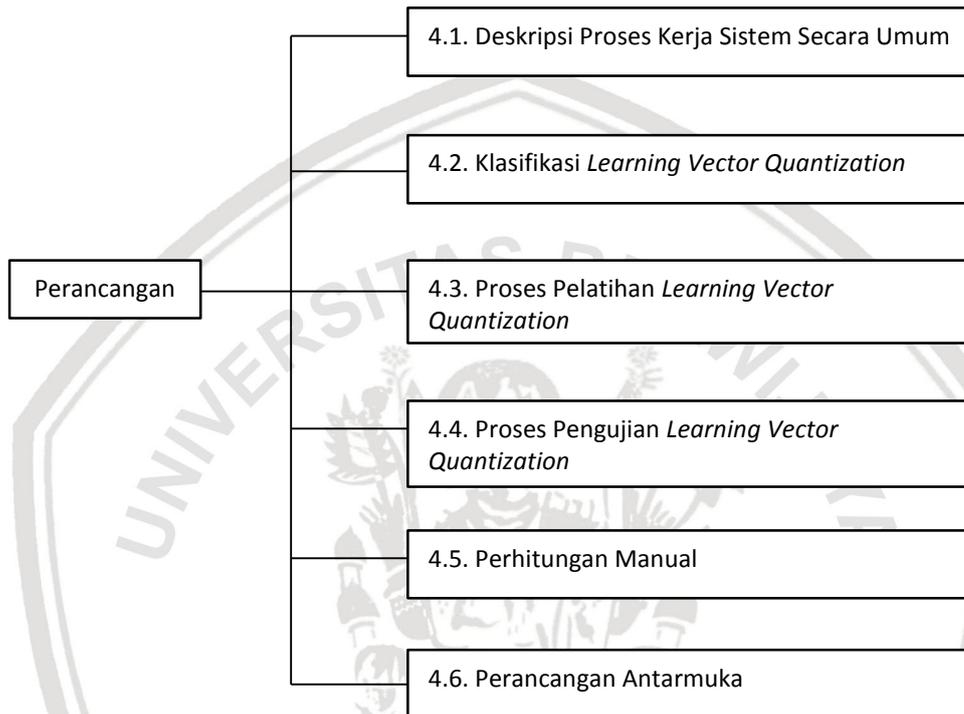
### 3.7 Jadwal penelitian

No.	Uraian	Juni				Juli				Agustus				September				Oktober			
		Minggu ke-																			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Pengumpulan data	■																			
2	Studi kepustakaan		■	■	■																
3	Perancangan					■	■														
4	Implementasi algoritme							■	■	■	■	■									
5	Pengujian dan analisis												■	■	■	■	■	■			
6	Kesimpulan dan saran																		■	■	



## BAB 4 PERANCANGAN

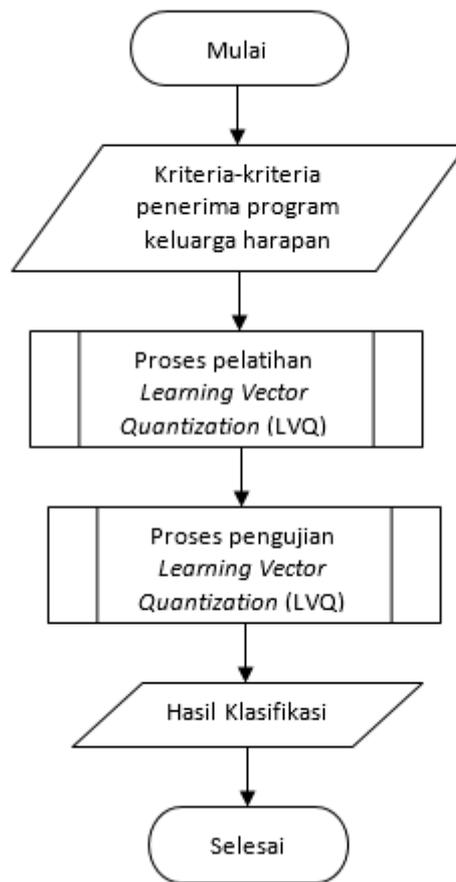
Bab perancangan menjelaskan mengenai perancangan yang digunakan dalam system, meliputi deskripsi proses kerja sistem secara umum, klasifikasi *Learning vector quantization*, proses pelatihan dengan *Learning vector quantization*, proses pengujian dengan *Learning vector quantization*, perhitungan manual, dan perancangan antarmuka. Bagan perancangan dapat dilihat pada Gambar 4.1.



Gambar 4.1 Bagan perancangan

### 4.1 Deskripsi Proses Kerja Sistem Secara Umum

Proses kerja sistem secara umum terdiri dari beberapa tahapan meliputi *input* kriteria-kriteria calon penerima Program Keluarga Harapan, yang terdiri dari 10 kriteria yang harus di *input* oleh petugas desa untuk kemudian diklasifikasikan apakah orang tersebut berhak atau tidak berhak untuk menerima bantuan Program Keluarga Harapan, proses pelatihan dengan metode *Learning Vector Quantization*, proses pengujian dengan metode *Learning vector quantization*, dan *output* hasil dari klasifikasi penerimaan dana bantuan Program Keluarga Harapan. Proses kerja sistem secara umum ditunjukkan pada Gambar 4.2.

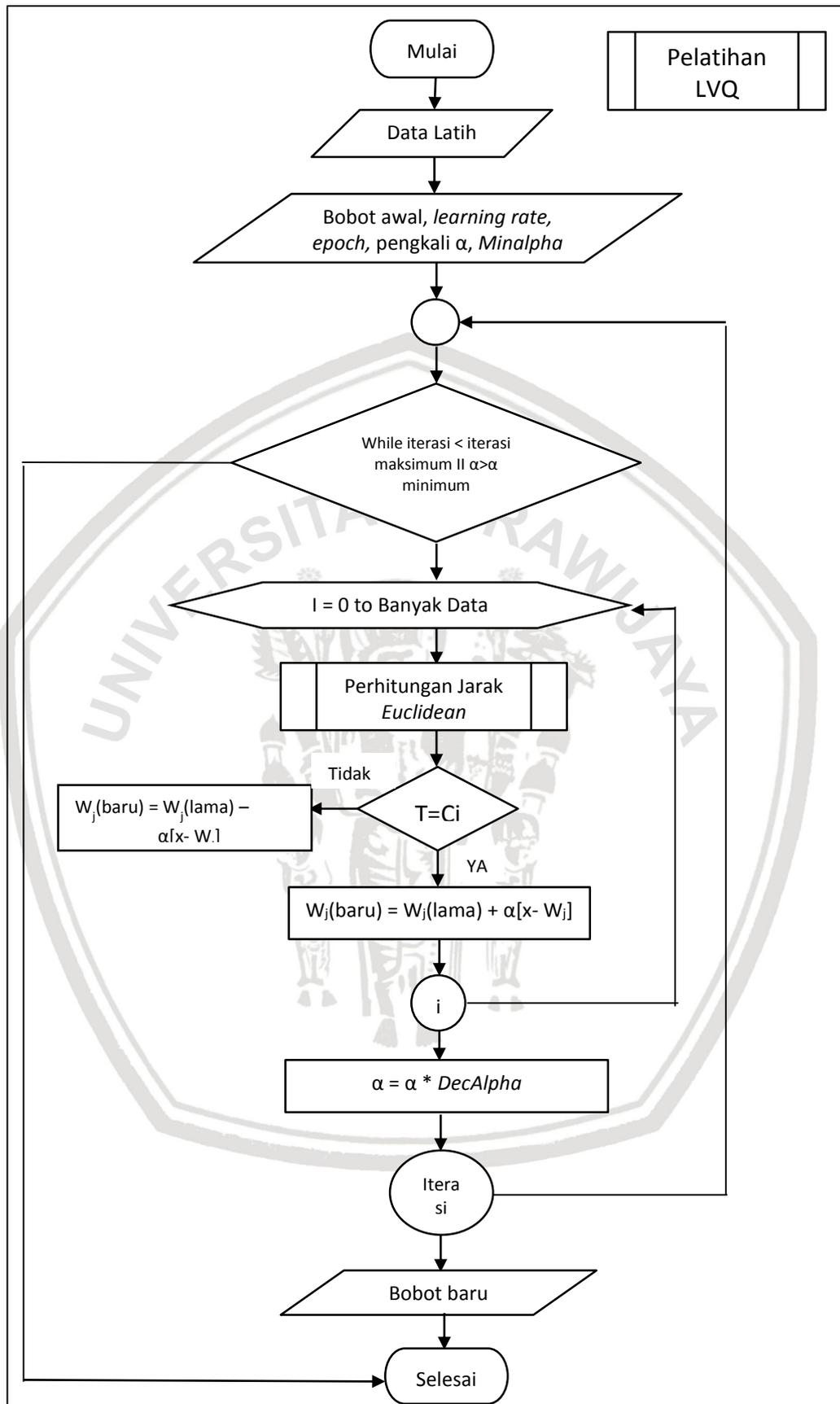


**Gambar 4.2 Flowchart sistem**

Pada Gambar 4.2 tersebut terdapat *input* berupa 10 kriteria yang digunakan petugas dalam pemilihan calon penerimaan Program Keluarga Harapan. Selanjutnya akan dilakukan proses pelatihan dengan mencari jarak dengan menggunakan *Euclidean* untuk selanjutnya mencari jarak terdekat dan akan digunakan untuk mengupdate bobot sampai didapatkan bobot akhir dengan sesuai kondisi yang sudah ditentukan, bobot akhir yang didapat akan digunakan untuk proses selanjutnya yaitu proses pengujian. Proses pengujian hampir sama dengan proses pelatihan, bedanya pada proses pengujian tidak perlu lagi untuk update bobot untuk pencarian jarak pada tiap datanya.

#### **4.2 Proses Pelatihan *Learning vector quantization* (LVQ)**

Pada proses pelatihan *Learning vector quantization* ini menggunakan 100 data latih, 2 kelas dan 5 parameter. Diantaranya *Learning rate*, pengkali *learning rate*, *epoch* dan *MinAlpha*. Selanjutnya menghitung jarak dan mencari jarak terdekat untuk selanjutnya akan diupdate dan hasil akhir bobot akan digunakan untuk proses selanjutnya. Yaitu proses pengujian. Proses pelatihan *Learning vector quantization* dapat dilihat pada Gambar 4.4.

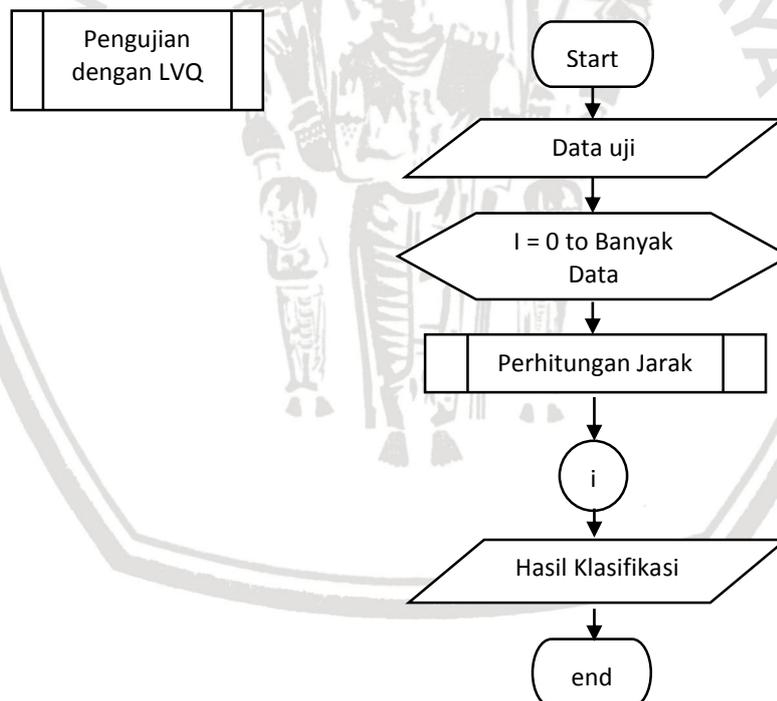


Gambar 4.3 Flowchart pelatihan *learning vector quantization*

Pada proses pelatihan LVQ, yang dibutuhkan adalah data latih, bobot awal yang didapatkan dari 100 data latih secara acak, kemudian menentukan *learning rate*, pengkali *learning rate*, *iterasi*, dan *Minalpha*. Kemudian akan mencari jarak terdekat dengan menggunakan *Euclidean*, dilihat manakah kelas bobot yang lebih dekat dengan data, selanjutnya melakukan update bobot dengan melakukan pengecekan apakah kelas data sama dengan kelas bobot yang terdekat. Bobot yang sudah diupdate akan digunakan untuk proses perhitungan jarak pada data selanjutnya. Proses terus berjalan sampai pada iterasi maksimum. Setelah mencapai iterasi maksimum, bobot terakhir yang didapatkan akan digunakan pada proses pengujian.

### 4.3 Pengujian *Learning vector quantization* (LVQ)

Dalam pengujian *Learning vector quantization*, prosesnya hampir sama dengan pelatihan *Learning vector quantization*, bedanya pada proses pengujian, bobot yang didapat diperoleh dari bobot akhir proses pelatihan *Learning vector quantization*, dan tidak perlu lagi *mengupdate* bobotnya, setelah proses perhitungan jarak akan langsung didapatkan hasil kelas. Setelah dilakukan tahap pelatihan, akan diperoleh bobot akhir ( $W$ ). Bobot ini yang akan digunakan selanjutnya untuk melakukan tahap pengujian. *Flowchart* pengujian *Learning vector quantization* dapat dilihat pada Gambar 4.5.



Gambar 4.4 Flowchart pengujian *learning vector quantization*

### 4.5 Pelatihan *Learning vector quantization* (LVQ)

Untuk melakukan pelatihan LVQ, diperlukan data latih dan menentukan nilai parameter yang ada pada LVQ, penentuan parameter didapatkan dari penelitian

sebelumnya yang menjadi acuan penelitian ini, yaitu nilai  $\alpha$  sebesar 0,1,  $Dec\alpha$  sebesar 0,1,  $Min\alpha$  0,01 dan  $maxEpoch$  2 Data latih yang digunakan dalam penelitian ini adalah 100 dari data, dapat dilihat pada Tabel 4.1.

**Tabel 4.1 Data latih PKH**

nomor	V01	V02	V03	V04	V05	V06	V07	V08	V09	V10	FINAL
1	90	90	90	100	90	90	90	100	50	50	0
2	90	90	90	100	90	90	90	100	50	100	0
3	30	60	60	100	30	60	90	50	50	50	1
4	60	30	60	100	90	60	90	50	100	50	1
5	90	90	90	100	90	90	90	100	50	50	0
6	30	60	60	100	90	90	90	50	100	100	1
7	30	30	30	50	60	90	90	50	50	50	1
8	90	90	90	100	90	90	90	100	50	50	0
9	60	60	90	50	30	90	90	50	50	50	1
10	60	60	60	50	60	90	90	100	50	50	1
11	90	90	90	100	60	90	90	100	100	50	0
12	60	30	30	50	60	60	90	50	100	50	1
13	60	60	60	50	30	60	90	50	50	50	1
14	30	30	60	50	90	90	30	50	50	50	1
15	30	60	60	100	90	90	90	50	50	50	1
16	90	90	90	100	90	90	90	100	50	100	0
17	30	60	60	100	90	90	90	50	50	50	1
18	90	90	90	100	90	90	90	100	100	50	0
19	90	90	90	100	90	90	90	100	50	50	0
20	90	90	90	100	90	90	90	100	100	50	0

Keterangan :

- V01 : luas tanah bangunan.
- V02 : Jenis Lantai bangunan rumah Terbuat dari Tanah atau Bambu.
- V03 : Jenis dinding Bangunan rumah terbuat dari bambu, kayu atau semen.
- V04 : Jamban.
- V05 : Sumber Air Untuk Keseharian Berasal dari Sumur, Sanyo atau PDAM
- V06 : Sumber Penerangan Dirumah Bukan Listrik, Gabung Tetangga atau PLN
- V07 : Bahan Bakar Yang Digunakan untuk masak berasal dari minyak tanah, kayu, atau LPG.
- V08 : Tidak Mampu Membayar Anggota Keluarga Berobat ke Pukesmas.
- V09 : Memiliki Ibu Hamil/nifas/anak balita.
- V10 : Memiliki anak yang sedang sekolah.

Inisialisasi bobot didapatkan dari data latih secara acak yang mewakili 2 kelas berbeda. Yaitu indeks ke 3 mewakili kelas 1, dan indeks ke 4 mewakili kelas 0. Inisialisasi bobot dapat dilihat pada Tabel 4.3

**Tabel 4.2 Bobot awal pelatihan**

W1	0.5	0	0.5	1	1	0	1	0	1	0
W2	1	1	1	1	1	1	1	1	0	0

Data ke-1 adalah data yang digunakan untuk inputan pelatihan. Terdiri dari 10 kriteria dan terdapat sebuah kelas yang sudah diketahui. Data ke-1 dapat dilihat pada Tabel 4.4.

**Tabel 4.3 Data ke-1 pelatihan**

1.	1	1	1	1	1	1	1	1	0	0	0
----	---	---	---	---	---	---	---	---	---	---	---

#### 4.5.1 Menghitung Jarak *Euclidean*

Menghitung jarak *Euclidean* dari data ke bobot 1 dan 2. Bobot didapatkan dari data latih dengan acak. Berikut adalah perhitungan jarak *Euclidean* antara bobot dengan data latih yang pertama.  $Jarak = \sqrt{\sum_{i=0}^{fitur} (x_i - w_i)^2}$  (4.2)

Perhitungan dengan mencari jarak pada data latih kesatu dengan bobot pertama (W1) dan bobot kedua (W2) dapat diperoleh menggunakan persamaan 2.1. Kemudian hasil dari jarak dijumlahkan untuk mendapatkan jarak paling minium, dan dianggap sebagai Kelas target. berikut Tabel 4.5 merupakan hasil perhitungan jarak antara data latih kesatu dengan W1 dan W2. Tabel hasil perhitugan jarak dapat dilihat pada Tabel 4.5.

**Tabel 4.4 Hasil Perhitungan jarak antara data latih dengan bobot**

D1W1	0.25	1	0.25	0	0	1	0	1	1	0	2.121
D1W2	0	0	0	0	0	0	0	0	0	0	0

Data latih ke-1 dengan bobot 1

$$D1W1 = \sqrt{(1 - 0.5)^2 + (1 - 0)^2 + (1 - 0.5)^2 + (1 - 1)^2 + (1 - 1)^2 + (2 - 0)^2 + (1 - 1)^2 + (0 - 1)^2 + (0 - 1)^2 + (0 - 0)^2} = 2.121$$

Data latih ke-1 dengan bobot 2

D1W2=

$$\sqrt{(1 - 1)^2 + (1 - 1)^2 + (1 - 1)^2 + (1 - 1)^2 + (1 - 1)^2 + (2 - 1)^2 + (1 - 1)^2 + (0 - 1)^2 + (0 - 1)^2 + (0 - 0)^2} = 0$$

#### 4.5.2 Cek Kondisi Update

Pada Tabel 4.5 didapatkan hasil perhitungan jarak yang paling minimum adalah  $D1 - W2$  yaitu 0 yang akan digunakan untuk update bobot dengan menggunakan persamaan 2.2 untuk  $T = Cj$ , dan persamaan 2.3  $T \neq Cj$ . Hasil update akan digunakan untuk bobot data selanjutnya yaitu data ke-2. Pada data 1 dihasilkan kelas 0 paling terdekat, maka update bobot menggunakan rumus 2.2. Tabel update bobot yang baru dapat dilihat pada Tabel 4.6.

**Tabel 4.5 Update bobot baru**

W1	0.5	0	0.5	1	1	0	1	0	1	0
W2	1	1	1	1	1	1	1	1	0	0

Setelah mendapatkan bobot baru, proses dilanjutkan kembali untuk data selanjutnya dengan menggunakan bobot terakhir yang didapatkan pada proses update bobot. Data latih kedua dapat dilihat pada Tabel 4.7.

**Tabel 4.6 Data kedua pelatihan**

2.	1	1	1	1	1	1	1	1	0	1	0
----	---	---	---	---	---	---	---	---	---	---	---

Berikut Tabel 4.8 merupakan hasil perhitungan jarak antara data latih kedua dengan W1 dan W2. Tabel hasil perhitungan jarak dapat dilihat pada Tabel 4.8

**Tabel 4.7 Hasil perhitungan jarak data kedua**

D2W1	0.25	1	0.25	0	0	1	0	1	1	1	2.345
D2W2	0	0	0	0	0	0	0	0	0	1	1

$$\sqrt{(1 - 0.5)^2 + (1 - 0)^2 + (1 - 0.5)^2 + (1 - 1)^2 + (1 - 1)^2 + (2 - 0)^2 + (1 - 1)^2 + (0 - 1)^2 + (0 - 1)^2 + (1 - 0)^2} = 2.345$$

$$\sqrt{(1 - 1)^2 + (1 - 1)^2 + (1 - 1)^2 + (1 - 1)^2 + (1 - 1)^2 + (2 - 1)^2 + (1 - 1)^2 + (0 - 1)^2 + (0 - 1)^2 + (0 - 0)^2} = 1$$

Kelas data latih yang kedua adalah 0, dan dari hasil perhitungan jarak diatas, nilai terkecil yang didapatkan adalah kelas 0, yaitu 1 yang artinya kelas target sama dengan kelas yang sudah diberikan pada data latih kedua. Selanjutnya dapat dilanjutkan pada proses data ketiga. Selanjutnya update bobot, update bobot dapat dilihat pada Tabel 4.9.

**Tabel 4.8 Hasil update bobot**

W1	0.5	0	0.5	1	1	0	1	0	1	0
W2	1	1	1	1	1	1	1	1	0	0.1

Kemudian dapat dilanjutkan dengan memproses data ketiga, data ketiga dapat dilihat pada Tabel 4.10.



**Tabel 4.9 Data ketiga pelatihan**

3.	0	0.5	0.5	1	0	0	1	0	0	0	1
----	---	-----	-----	---	---	---	---	---	---	---	---

Selanjutnya mencari jarak terdekat antara data dua dengan bobot 1 dan 2. Berikut ini merupakan hasil perhitungan jarak antara data latih kesatu dengan W1 dan W2. Tabel hasil perhitungan jarak dapat dilihat pada Tabel 4.11.

**Tabel 4.10 Hasil perhitungan jarak**

D3W1	0.25	0.25	0	0	1	0	0	0	1	0	1.581
D3W2	1	0.25	0.25	0	1	1	0	1	0	0.01	2.123

$$\sqrt{(0 - 0.5)^2 + (1 - 0.5)^2 + (1 - 0.5)^2 + (1 - 1)^2 + (1 - 0)^2 + (2 - 0)^2 + (1 - 1)^2 + (0 - 0)^2 + (0 - 0)^2 + (1 - 0)^2} = 1.581$$

$$\sqrt{(0 - 1)^2 + (1 - 1)^2 + (1 - 1)^2 + (1 - 1)^2 + (1 - 1)^2 + (2 - 1)^2 + (1 - 1)^2 + (0 - 1)^2 + (0 - 1)^2 + (1 - 0)^2} = 2.123$$

Kelas data latih yang ketiga adalah 1, dan dari hasil perhitungan jarak diatas, nilai terkecil yang didapatkan adalah kelas 1, yaitu 1.581 yang artinya kelas target sama dengan kelas yang sudah diberikan pada data latih kedua. Selanjutnya dapat dilanjutkan pada proses data keempat. Selanjutnya update bobot, update bobot dapat dilihat pada Tabel 4.12.

**Tabel 4.11 Update bobot**

W1	0.45	0.05	0.5	1	0.9	0	1	0	0.9	0
W2	1	1	1	1	1	1	1	1	0	0.1

Kemudian dapat dilanjutkan dengan memproses data keempat, data keempat dapat dilihat pada Tabel 4.13.

**Tabel 4.12 Data keempat pelatihan**

4.	0.5	0	0.5	1	1	0	1	0	1	0	1
----	-----	---	-----	---	---	---	---	---	---	---	---

Selanjutnya mencari jarak terdekat antara data empat dengan bobot 1 dan 2. Berikut ini merupakan hasil perhitungan jarak antara data latih empat dengan W1 dan W2. Tabel hasil perhitungan jarak dapat dilihat pada Tabel 4.14.

**Tabel 4.13 Hasil perhitungan jarak**

D4W1	0.002	0.002	0	0	0.01	0	0	0	0.01	0	0.158
D4W2	0.25	1	0.25	0	0	1	0	1	1	0.01	2.123



$$\sqrt{(0.5 - 0.45)^2 + (0 - 0.05)^2 + (0.5 - 0.5)^2 + (1 - 1)^2 + (1 - 0.9)^2 + (0 - 0)^2 + (1 - 1)^2 + (0 - 0)^2 + (1 - 1)^2 + (0 - 0.9)^2} = 0.158$$

$$\sqrt{(0.5 - 1)^2 + (0 - 1)^2 + (0.5 - 1)^2 + (1 - 1)^2 + (1 - 1)^2 + (0 - 1)^2 + (1 - 1)^2 + (0 - 1)^2 + (1 - 1)^2 + (0 - 0)^2} = 2.123$$

Kelas data latih yang keempat adalah 1, dan dari hasil perhitungan jarak diatas, nilai terkecil yang didapatkan adalah kelas 1, yaitu 0.158 yang artinya kelas target sama dengan kelas yang sudah diberikan pada data latih kedua. Selanjutnya dapat dilanjutkan pada proses data kelima. Selanjutnya update bobot, update bobot dapat dilihat pada Tabel 4.15.

**Tabel 4.14 Update bobot**

W1	0.455	0.045	0.5	1	0.91	0	1	0	0.91	0
W2	1	1	1	1	1	1	1	1	0	0.1

Kemudian dapat dilanjutkan dengan memproses data keempat, data keempat dapat dilihat pada Tabel 4.16.

**Tabel 4.15 Data kelima pelatihan**

5.	1	1	1	1	1	1	1	1	0	0	0
----	---	---	---	---	---	---	---	---	---	---	---

Selanjutnya mencari jarak terdekat antara data lima dengan bobot 1 dan 2. Berikut ini merupakan hasil perhitungan jarak antara data latih kelima dengan W1 dan W2. Tabel hasil perhitugan jarak dapat dilihat pada Tabel 4.17.

**Tabel 4.16 Hasil perhitungan jarak**

D5W1	0.297	0.912	0.25	0	0.008	1	0	1	0.828	0	2.072
D5W2	0	0	0	0	0	0	0	0	0	0.01	0.1

$$\sqrt{(1 - 0.455)^2 + (1 - 0.045)^2 + (1 - 0.5)^2 + (1 - 1)^2 + (1 - 0.91)^2 + (1 - 0)^2 + (1 - 1)^2 + (1 - 0)^2 + (1 - 1)^2 + (0 - 0.91)^2} = 2.072$$

$$\sqrt{(1 - 1)^2 + (1 - 1)^2 + (1 - 1)^2 + (1 - 1)^2 + (1 - 1)^2 + (1 - 1)^2 + (1 - 1)^2 + (1 - 1)^2 + (1 - 1)^2 + (0 - 0)^2} = 0.1$$

Kelas data latih yang kelima adalah 0, dan dari hasil perhitungan jarak diatas, nilai terkecil yang didapatkan adalah kelas 0, yaitu 0.1 yang artinya kelas target sama dengan kelas yang sudah diberikan pada data latih kelima. Selanjutnya dapat dilanjutkan pada proses data keenam, sampai semua data habis dan mendapatkan bobot paling akhir dari data paling akhir yang akan digunakan untuk proses pengujian data.



### 4.5.3 Perbarui Nilai *Alpha*

Dalam satu kali iterasi, nilai *alpha* akan diperbarui dengan mengalikan nilai *alpha* dengan *DecAlpha*.

### 4.5.4 Cek *Error* Minimum

Kemudian melakukan pengecekan *error* minimum sesuai dengan persamaan 2.4. Apakah nilai *alpha* sudah lebih kecil sama dengan dari *error* minimum, jika sudah maka hentikan proses LVQ, namun jika belum maka tambah *epoch* dan melanjutkan proses LVQ lagi.

## 4.6 Perhitungan Manual *Learning vector quantization* (LVQ)

Pada bab ini akan dijelaskan untuk proses perhitungan klasifikasi secara manual dengan menggunakan lvq. Yang harus disiapkan adalah data latih, bobot, *alpha*, *DecAlpha*, dan *MinAlpha*, pertama akan dicari jarak minimum antara setiap data ke bobot 1 dan juga bobot 2, selanjutnya *update* bobot sesuai kondisi. Dan hasil klasifikasi akan didapatkan.

### 4.6.1 Tahap Awal

Untuk mengklasifikasi dengan metode LVQ maka bobot *update* terakhir digunakan untuk mengelompokkan data uji. Berikut Tabel 4.18 adalah vektor bobot terakhir dari pelatihan LVQ.

**Tabel 4.17 Bobot klasifikasi LVQ**

W1	0.3115	0.330	0.419	0.593	0.731	0.680	0.770	0	0.268	0.173
W2	0.967	0.987	0.952	1.000	0.946	0.995	1	1	0.079	0.119

Kemudian dapat dilanjutkan dengan memproses data keempat, data keempat dapat dilihat pada Tabel 4.19.

**Tabel 4.18 Data kesatu pengujian**

1.	0.5	0	0.5	0	0	0	0	0	0	0	1
----	-----	---	-----	---	---	---	---	---	---	---	---

### 4.6.2 Hitung Jarak *Euclidean*

Menghitung jarak *Euclidean* antara seetiap data latih dengan bobot 1 dan 2 yang sudah disiapkan. Berikut adalah perhitungan jarak *Euclidean* antara bobot dengan data sesuai dengan persamaan 4.20.

**Tabel 4.19 Hasil perhitungan jarak**

D1W1	0.035	0.109	0.006	0.351	0.534	0.462	0.593	0	0.071	0.030	1.481
D1W2	0.218	0.974	0.205	1.001	0.894	0.991	1	1	0.006	0.014	2.511

Data uji ke1 dengan bobot 1 :

$$\sqrt{(0.5 - 0.311)^2 + (0 - 0.330)^2 + (0 - 0.419)^2 + (0 - 0.593)^2 + (0 - 1.071)^2 + (0 - 0.680)^2 + (0.5 - 0.770)^2 + (0 - 0)^2 + (0 - 0.079)^2 + (1 - 0.119)^2} = 1.4816$$

Data uji ke1 dengan bobot 2:

$$\sqrt{(0.5 - 0.967)^2 + (0 - 0.987)^2 + (0 - 0.952)^2 + (0 - 1.000)^2 + (0 - 0.946)^2 + (0 - 0.995)^2 + (0.5 - 1)^2 + (0 - 1)^2 + (0 - 0.079)^2 + (1 - 0.119)^2} = 2.511$$

Kelas data uji yang pertama adalah 1, dan dari hasil perhitungan jarak diatas, nilai terkecil yang didapatkan adalah kelas 1, yaitu 1.481 yang artinya kelas target sama dengan kelas yang sudah diberikan pada data uji pertama. Selanjutnya dapat dilakukan perhitungan data kedua dengan menggunakan bobot yang sama. Data uji kedua dapat dilihat pada Tabel 4.21.

**Tabel 4.20 Data uji kedua**

2.	1	1	1	1	1	1	1	1	0	1	0
----	---	---	---	---	---	---	---	---	---	---	---

Berikut hasil perhitungan jarak kedua, hasil perhitungan jarak kedua dapat dilihat pada Tabel 4.22.

**Tabel 4.21 Hasil perhitungan jarak**

D2W1	0.511	0.664	0.0002	0.058	0.045	0.099	0.009	1	0.338	0.905	1.906
D2W2	0	0	0	0	0.001	0	0	0	0.060	0.773	0.913

Data uji ke2 dengan bobot 1 :

$$\sqrt{(1 - 0.311)^2 + (1 - 0.330)^2 + (1 - 0.419)^2 + (1 - 0.593)^2 + (1 - 1.071)^2 + (1 - 0.680)^2 + (1 - 0.770)^2 + (1 - 0)^2 + (0 - 0.079)^2 + (1 - 0.119)^2} = 1.906$$

Data uji ke2 dengan bobot 2:

$$\sqrt{(1 - 0.967)^2 + (1 - 0.987)^2 + (1 - 0.952)^2 + (1 - 1.000)^2 + (1 - 0.946)^2 + (1 - 0.995)^2 + (1 - 1)^2 + (1 - 1)^2 + (0 - 0.079)^2 + (1 - 0.119)^2} = 0.913$$

Kelas data uji yang kedua adalah 0, dan dari hasil perhitungan jarak diatas, nilai terkecil yang didapatkan adalah kelas 0, yaitu 0.913 yang artinya kelas target sama dengan kelas yang sudah diberikan pada data uji kedua. Selanjutnya dapat dilakukan perhitungan data keselanjutnya dengan menggunakan bobot yang sama. Data uji ketiga dapat dilihat pada Tabel 4.23.

**Tabel 4.22 Data uji ketiga**

3.	0	0.5	0.5	0	0	0.5	0	0	1	0	1
----	---	-----	-----	---	---	-----	---	---	---	---	---

Berikut adalah hasil dari perhitungan jarak ketiga, perhitungan jarak ketiga dapat dilihat pada Tabel 4.24.

**Tabel 4.23 Hasil perhitungan jarak**

D3W1	0.081	0.099	0.234	0.573	0.618	0.0338	0.815	0	0.174	0.002	1.622
D4W2	1	0.25	0.25	1	0.935	0.25	1	1	0.567	0.014	2.503

Data uji ke3 dengan bobot 1 :



$$\sqrt{(0 - 0.311)^2 + (0.5 - 0.330)^2 + (0.5 - 0.952)^2 + (0 - 1.000)^2 + (0 - 0.995)^2 + (0.5 - 1)^2 + (0 - 1)^2 + (0 - 0.079)^2 + (1 - 0.119)^2} = 1.622$$

Data uji ke3 dengan bobot 2 :

$$\sqrt{(0 - 0.967)^2 + (0.5 - 0.987)^2 + (0.5 - 0.952)^2 + (0 - 1.000)^2 + (0 - 0.946)^2 + (0.5 - 0.995)^2 + (0 - 1)^2 + (0 - 1)^2 + (0 - 0.079)^2 + (1 - 0.119)^2} = 2.503$$

Kelas data uji yang ketiga adalah 1, dan dari hasil perhitungan jarak diatas, nilai terkecil yang didapatkan adalah kelas 1, yaitu 1.622 yang artinya kelas target sama dengan kelas yang sudah diberikan pada data uji ketiga. Selanjutnya dapat dilakukan perhitungan data keempat dengan menggunakan bobot yang sama. Data uji keempat dapat dilihat pada Tabel 4.25.

**Tabel 4.24 Data uji ke empat**

4.	1	1	1	1	1	1	1	1	1	0	0	0
----	---	---	---	---	---	---	---	---	---	---	---	---

Selanjutnya, hasil perhitungan jarak keempat dapat dilihat pada Tabel 4.26.

**Tabel 4.25 Hasil perhitungan jarak**

D4W1	0.474	0.448	0.336	0.165	0.072	0.102	0.052	1	0.0719	0.030	1.659
D4W2	0.001	0.000	0.002	7.62E-	0.002	1.8E-	0	0	0.006	0.0143	0.164

Data uji ke4 dengan bobot 1 :

$$\sqrt{(1 - 0.311)^2 + (1 - 0.330)^2 + (1 - 0.952)^2 + (1 - 1.000)^2 + (1 - 0.995)^2 + (1 - 1)^2 + (1 - 1)^2 + (0 - 0.079)^2 + (0 - 0.119)^2} = 1.659$$

Data uji ke4 dengan bobot 2 :

$$\sqrt{(1 - 1)^2 + (1 - 1)^2 + (1 - 1)^2 + (0 - 1)^2 + (0 - 0.967195)^2 + (0.5 - 1)^2 + (0 - 1)^2 + (0 - 1)^2 + (1 - 0.24661)^2 + (0 - 0.12073)^2} = 0.164$$

Kelas data uji yang keempat adalah 0, dan dari hasil perhitungan jarak diatas, nilai terkecil yang didapatkan adalah kelas 0, yaitu 0.164 yang artinya kelas target sama dengan kelas yang sudah diberikan pada data uji keempat. Selanjutnya dapat dilakukan perhitungan data kelima sampai data terakhir dengan menggunakan bobot yang sama. Data uji kelima dapat dilihat pada Tabel 4.27.

**Tabel 4.26 Data uji kelima**

5.	0	0	0.5	0	0.5	1	0	0	0	0	1
----	---	---	-----	---	-----	---	---	---	---	---	---

Hasil dari perhitungan jarak data kelima dapat dilihat pada Tabel 4.28.

**Tabel 4.27 Hasil perhitungan jarak**

D5W1	0.936	0.109	0.006	0.351	0.053	0.102	0.593	0	0.0719	0.030	1.501
D5W2	0.936	0.974	0.205	1.001	0.198	1.8E-05	1	1	0.006	0.014	2.310

Data uji ke5 dengan bobot 1 :

$$\sqrt{(0 - 0.311)^2 + (0 - 0.330)^2 + (0.5 - 0.952)^2 + (1 - 1.000)^2 + (0 - 0.995)^2 + (0 - 1)^2 + (0 - 1)^2 + (0 - 0.079)^2 + (0 - 0.119)^2} = 1.501$$



Data uji ke5 dengan bobot 2 :

$$\sqrt{(0 - 1)^2 + (0 - 1)^2 + (0.5 - 1)^2 + (1 - 1)^2 + (1 - 0.967195)^2 + (0 - 1)^2 + (0 - 1)^2 + (0 - 1)^2 + (0 - 0.24661)^2 + (0 - 0.12073)^2} = 2.310$$

Kelas data uji yang kelima adalah 1, dan dari hasil perhitungan jarak diatas, nilai terkecil yang didapatkan adalah kelas 1, yaitu 1.501 yang artinya kelas target sama dengan kelas yang sudah diberikan pada data uji kelima.

#### 4.7 Perancangan Antarmuka

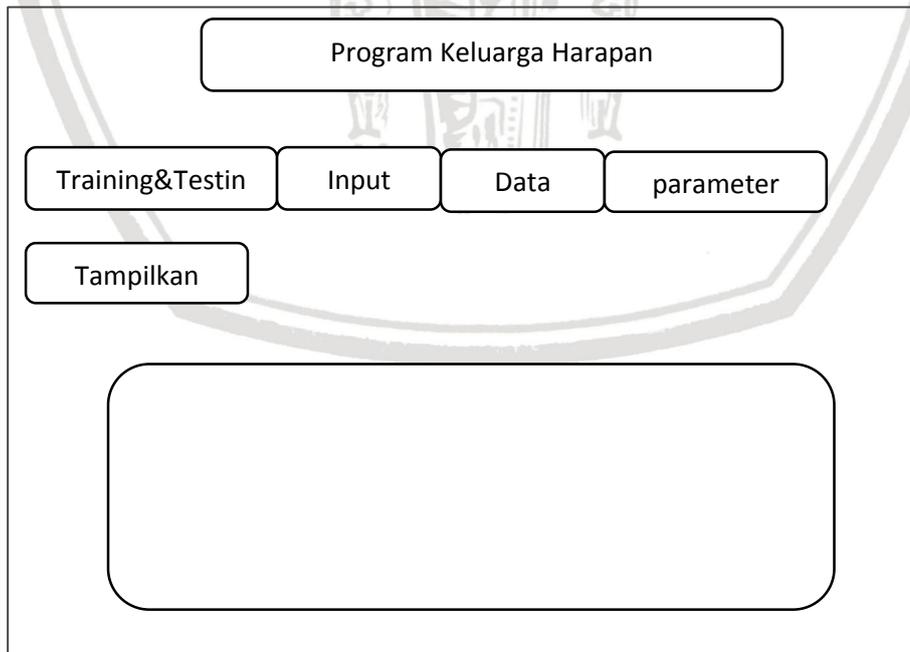
Tahap perancangan antarmuka dilakukan untuk implementasi metode LVQ dalam sistem “Klasifikasi Penerimaan Program Keluarga Harapan (PKH) dengan metode *Learning vector quantization*”.

Perancangan antar muka terdiri dari

1. Antarmuka halaman awal sistem.
2. Antarmuka halaman *training* dan *testing*.
3. Antarmuka halaman input.
4. Antarmuka halaman data.
5. Antarmuka halaman parameter.

##### 4.7.1 Antarmuka Halaman Awal Sistem

Halaman awal sistem, langsung menuju pada halaman *training* dan *testing*, karena untuk melakukan klasifikasi atau *input* data harus melakukan training data untuk mendapatkan bobot yang akan digunakan selanjutnya. perancangan antarmuka halaman tersebut dapat dilihat pada Gambar 4.5.

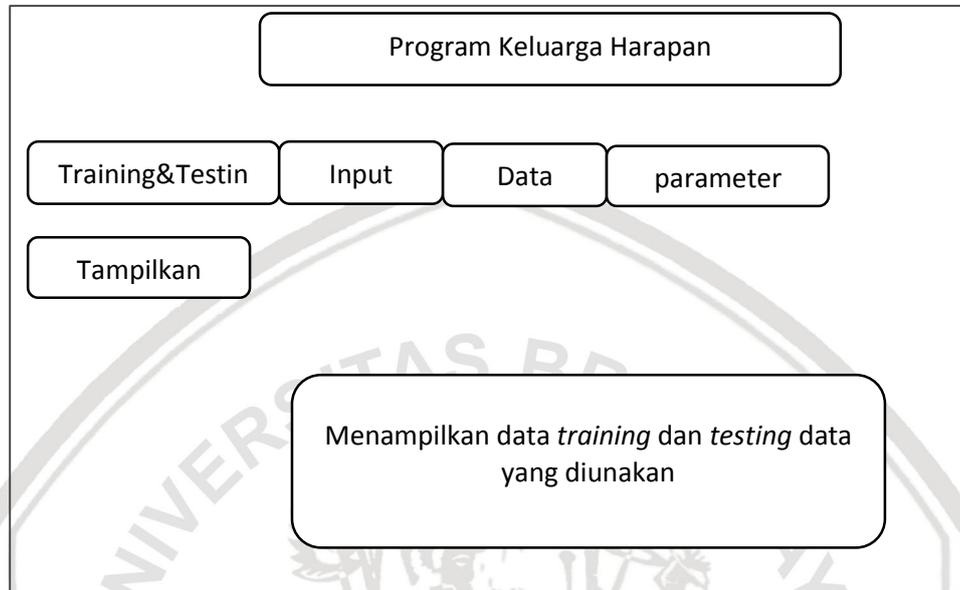


Gambar 4.5 Perancangan antarmuka halaman awal sistem



#### 4.7.2 Antarmuka Halaman Training dan Testing

Halaman *training* dan *testing*, ketika menekan tombol tampilan maka sistem akan menampilkan hasil data yang dilatih, dan menghasilkan 2 bobot akhir yang selanjutnya akan digunakan untuk proses selanjutnya, yaitu menghitung data uji. Perancangan antarmuka halaman login admin dapat dilihat pada Gambar 4.6.



Gambar 4.6 Perancangan antarmuka halaman login admin

#### 4.7.3 Antarmuka Halaman Input

Halaman ini digunakan untuk melakukan proses klasifikasi penerima dana Program Keluarga Harapan, admin harus mengisi semua data yang benar, ada nama, lalu mengisi 10 kriteria yang harus sesuai dengan masing-masing kepala keluarga, cara mengisinya adalah dengan memilih salah satu kriteria yang sesuai dengan keadaan kepala rumah tangga yang telah diketahui oleh petugas atau admin yang menggunakan. Pada setiap kriteria memiliki bobot, petugas wajib mengisi sesuai dengan kondisi calon penerima dana pkh. Setelah mengisi 10 kriteria, maka selanjutnya admin harus menekan tombol selesai untuk mengetahui hasil dari pengisian kriteria. Dari hasil tersebut, menunjukkan apakah kepala rumah tangga tersebut berhak menerima atau tidak dana program keluarga tersebut, hasil *input* yang sudah diisikan akan tersipan pada halaman data. Perancangan antarmuka halaman klasifikasi atau *input* bisa dilihat pada Gambar 4.7.

Program Keluarga Harapan

Training&Testing    Input    Data    parameter

Isi kriteria calon penerima dana program keluarga harapan

1. Kriteria 1 ..... ya/tidak
2. Kriteria 2..... ya/tidak
3. Dst samapai dengan kriteria 16

selesai

Hasil Program

**Gambar 4.7 Perancangan antarmuka halaman klasifikasi**

#### 4.7.4 Antarmuka Halaman Data

Halaman ini menampilkan nama-nama kepala rumah tangga yang sudah diisi sebelumnya pada halaman *input*. Berisikan nama dan hasil dari sistem apakah orang tersebut menerima atau tidak Program Keluarga Harapan. Data-data atau nama-nama kepala rumah tangga tersebut dapat diriset atau dihapus, apabila dirasa petugas sudah penuh atau sudah dicetak. Perancangan antarmuka halaman hasil klasifikasi dapat dilihat pada Gambar 4.9.

Program Keluarga Harapan

Training&Testin    Input    Data    parameter

No	Nama	Hasil
1.	Vidya Capristyan Pamungkas	Berhak
2.	Nurul Hidayat	Tidak
3.	Abdul M	Tidak
4.	Maya L	Berhak
dst		

**Gambar 4.8 Perancangan antarmuka halaman hasil klasifikasi**

#### 4.7.5 Antarmuka Halaman Parameter

Halaman ini digunakan untuk mengisi parameter yang akan digunakan untuk proses perhitungan yang dilakukan pada proses data latih agar mendapatkan hasil yang bagus sesuai dengan apa yang diinginkan. Halaman antarmuka parameter dapat dilihat pada Gambar 4.9.

The image shows a software interface for 'Program Keluarga Harapan'. At the top is a title bar. Below it are four tabs: 'Training&Testin', 'Input', 'Data', and 'parameter'. The 'parameter' tab is active, showing four input fields for 'Learning rate', 'DecaAlpha', 'MaxEpoch', and 'MinError'. A large watermark of the Universitas Brawijaya logo is visible in the background.

Gambar 4.9 Halaman antarmuka parameter

### 4.8 Perancangan Pengujian

Pada bab pengujian akan dilakukan beberapa skenario untuk menganalisis apakah sistem yang diimplementasikan sudah sesuai dengan metode yang sudah ditetapkan, ada 5 skenario pengujian, yaitu pengujian pengaruh nilai *learning rate*, pengujian pengali *learning rate*, pengujian pengaruh maksimum *epoch*, pengujian pengaruh jumlah data latih dan yang terakhir adalah pengaruh minimum *learning rate*.

#### 4.8.1 Pengujian Pengaruh Nilai *Learning Rate (Alpha)* Terhadap Hasil Akurasi

Skenario pengujian ini, bertujuan untuk mengetahui pengaruh nilai *learning rate* terhadap hasil akurasi, pegujian dilakukan sebanyak 10 kali percobaan dengan menggunakan parameter awal yang didapatkan dari refrensi penelitian senbelumnya. Nilai *learning rate* yang akan diuji yaitu 0,1, 0,2, 0,3, 0,4 sampai dengan 1. Nilai *learning rate* yang akan menghasilkan nilai terbaik akan digunakan untuk pengujian selanjutnya. Perancangan pengujian pengaruh *learning rate* dapat dilihat pada Tabel 4.29.

Tabel 4.28 Perancangan pengujian pengaruh learnig rate

Learningrate	Percobaan ke-										Rata-Rata
	1	2	3	4	5	6	7	8	9	10	
0,1											
0,2											
0,3											
0,4											
0,5											
0,6											
0,7											
0,8											
0,9											
1											

**4.8.2 Pengujian Pengaruh Pengali *Learning rate* (DecAlpha) Terhadap Hasil Akurasi**

Pengujian pengaruh *learning rate* dilakukan untuk mengetahui bagaimana pengaruh sebuah *learning rate* pada hasil akurasi sistem lvq. Pengujian dilakukan sebanyak 10 kali percobaan dengan menggunakan nilai pengali *learning rate* 0,1, 0,2, sampai dengan 1. Dari pengujian pengali *learning rate* sebanyak 10 kali percobaan, diambil nilai terbaik yang akan digunakan untuk proses pengujian selanjutnya. Pada proses pelatihan metode LVQ nilai pengali *learning rate* pada iterasi mencapai batas maksimum yang menjadi batas akhir dari suatu proses pelatihan. Perancangan pengujian pengaruh pegali *learning rate* dapat dilihat pada Tabel 4.30.

Tabel 4.29 Perancangan pengujian pengaruh pengali *Learning rate*

Pengali Learningrate	Percobaan ke-										Rata-Rata
	1	2	3	4	5	6	7	8	9	10	
0,1											
0,2											
0,3											
0,4											
0,5											
0,6											



0,7											
0,8											
0,9											
1											

#### 4.8.3 Pengujian Pengaruh Maksimum *Epoch* Terhadap Hasil Akurasi

Pengujian pengaruh maksimum *epoch* terhadap akurasi, menggunakan nilai 1 sampai dengan 10. Maksimum *epoch* memiliki peran atau sebagai kondisi berhentinya proses pengujian, sehingga semakin besar nilai dari *epoch* maka proses *iterasi* akan semakin banyak dan waktu yang dibutuhkan semakin lama, sebaliknya, jika nilai *epoch* semakin kecil, maka proses *iterasi* akan semakin sedikit dan waktu yang ditunjukkan juga semakin sedikit. Dari hasil pengujian nilai *epoch* ini akan diambil nilai *epoch* terbaik untuk digunakan pada proses pengujian yang selanjutnya. Perancangan pengujian maksimum *epoch* dapat dilihat pada Tabel 4.31.

Tabel 4.30 Perancangan pengujian maksimum *epoch*

<i>Epoch</i>	Percobaan ke-										Rata-Rata
	1	2	3	4	5	6	7	8	9	10	
0,1											
0,2											
0,3											
0,4											
0,5											
0,6											
0,7											
0,8											
0,9											
1											

#### 4.8.4 Pengujian Pengaruh Jumlah Data Latih Terhadap Hasil Akurasi

Pengujian pengaruh jumlah data latih ini dilakukan bertujuan untuk mengetahui pengaruh jumlah data latih yang digunakan terhadap hasil akurasi. Pengujian dilakukan sebanyak 10 kali percobaan dan menggunakan banyak jumlah data latih sebanyak 10%, 20%, 30%, 40%, 50%, sampai dengan 100% dari total 100 data. Setelah mendapat hasil paling baik, nilai terbaik yang didapatkan dari hasil pengujian ini yang akan selanjutnya digunakan pada proses pengujian yang

selanjutnya. Perancangan pengujian pengaruh jumlah data latih dapat dilihat pada Tabel 4.32.

**Tabel 4.31 Perancangan pengujian pengaruh jumlah data latih**

Banyak Jumlah Data Latih Yang digunakan	Rata-rata Hasil Akurasi
10%	
20%	
30%	
40%	
50%	
60%	
70%	
80%	
90%	
100%	

#### 4.8.5 Pengujian Pengaruh Minimum *Learning Rate* Terhadap Hasil Akurasi

Pengujian pengaruh minimum *learning rate* dilakukan untuk mengetahui pengaruh terhadap hasil akurasi dari system yang akan diimplementasikan. Pengujian dilakukan sebanyak 10 kali percobaan dengan menggunakan nilai minimum *alpha* 0,1, 0,01, 0,001, 0,0001 hingga 0,0000000001. Nilai minimum *alpha* merupakan salah satu kondisi berhentinya suatu proses pelatihan selain batas *maksimum epoch*, sehingga apabila semakin besar nilainya, maka proses pelatihan akan cepat selesai. Selanjutnya akan diambil nilai terbaik yang akan digunakan untuk proses pengujian yang selanjutnya. Rancangan antarmuka halaman pengujian pengaruh nilai minimum *alpha* dapat dilihat pada Tabel 4.33.

**Tabel 4.32 Perancangan pengujian nilai minimum *alpha***

Minimum <i>alpha</i>	Percobaan ke-										Rata-Rata
	1	2	3	4	5	6	7	8	9	10	
0,1											
0,2											
0,3											
0,4											
0,5											

0,6											
0,7											
0,8											
0,9											
1											



## BAB 5 IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi klasifikasi program keluarga harapan dengan l<sub>vq</sub> sesuai dengan perancangan yang sudah dibuat. Terdapat pembahasan jenis perangkat keras dan lunak, implementasi program dan juga implementasi antar muka.

### 5.1 Perangkat Keras

Dalam membangun sistem klasifikasi program keluarga harapan menggunakan metode *learning vector quantization* ini ada beberapa spesifikasi perangkat keras yang ditunjukkan pada Tabel 5.1.

Tabel 5.1 Perangkat keras

Perangkat Keras	Keterangan
Prosesor intel(R) 2955U (1,4 GHz, 2MB L3)	Digunakan untuk memproses algoritma yang terdapat pada aplikasi.
Memory RAM 4 GB	Digunakan untuk menampung beban saat proses berjalannya aplikasi.

### 5.2 Perangkat Lunak

Dalam membangun sistem klasifikasi program keluarga harapan menggunakan metode *Learning vector quantization* ini ada beberapa spesifikasi perangkat lunak yang ditunjukkan pada Tabel 5.2.

Tabel 5.2 Perangkat lunak

Perangkat Lunak	Keterangan
Sistem Operasi Windows 8 64-bit	Sebagai perantara antara aplikasi dengan hardware
Netbeans 8.0,2	Aplikasi yang digunakan untuk membangun aplikasi klasifikasi Program Keluarga Harapan dengan metode <i>learning vector quantization</i> .
Notepad	Digunakan untuk menyimpan data latih dan data uji yang selanjutnya akan dibaca oleh sistem klasifikasi tsb menggunakan <i>netbeans</i> .
Microsoft Word 2016	Digunakan untuk membuat dokumen skripsi
Microsoft Excel 2016	Digunakan untuk perhitungan atau manualisasi data uji dan data latih.

### 5.3 Implementasi Program Metode *Learning vector quantization*.

Sub bab ini akan menjelaskan tentang penjabaran implementasi program yang digunakan dalam membangun sistem klasifikasi Program Keluarga Harapan dengan menggunakan metode *learning vector quantization*. Implementasi program yaitu implementasi dari algoritme *learning vector quantization* yang terdiri dari pelatihan *learning vector quantization* dan pengujian *learning vector quantization*.

#### 5.3.1 Proses Pelatihan dan Pengujian *Learning vector quantization* (LVQ)

Pada tahap pelatihan ada beberapa langkah yang dilakukan metode lvq, yang pertama adalah inialisasi bobot awal yang mewakili kelas, yaitu kelas 1 dan kelas 0, kemudian mencari jarak antara data ke bobot 1 dan bobot 2 menggunakan *euclidean*. Selanjutnya adalah mencari jarak terkecil diantara kelas 1 dan 0 dan yang terakhir adalah update bobot.

##### 5.3.1.1 Proses inialisasi bobot

Ada 2 cara dalam pengambilan bobot pada lvq, yang pertama adalah dengan menentukan secara manual bobot 1 dan 2 yang diambil dari data latih, yang kedua adalah dengan cara mengacak atau *random* dalam memberikan nilainya. Pada program ini, bobot didapatkan secara random dari 100 data latih. *Source code* untuk inialisasi bobot dapat dilihat pada *Soure code* 5.1.

```

1      public void inialisasi() {
2          int in1 = 0, in2 = 0; // agar while jalan, maka di
3      pilih angka -1 , selain 0 dan 1
4          double cek1 = -1, cek2 = -1;
5
6          while (cek1 != 1) { //merandom mencari kelas 1
7              in1 = 0;
8              in1 = (int) (Math.random() * 100);
9              cek1 = datalatih[in1][10];
10         }
11
12         while (cek2 != 0) { //merandom mencari kelas 0
13             in2 = 0;
14             in2 = (int) (Math.random() * 100);
15             cek2 = datalatih[in2][10];
16         }
17         System.out.println(in1);
18         System.out.println(in2);
19         for (int j = 0; j < kolomlatih - 1; j++) {
20             bobotawal[0][j] = datalatih[3][j]; //kelas 1
21             bobotawal[1][j] = datalatih[4][j];
22         }
23
24         for (int i = 0; i < 2; i++) {
25             hasil.append("Bobot " + (i + 1) + " :");
26             for (int j = 0; j < kolomlatih - 1; j++) {
27                 hasil.append(bobotawal[i][j] + "\t");
28             }
29             hasil.append("\n");
30         }
31     }

```

**Source Code 5.1** Proses pelatihan *learning vector quantization* (LVQ)

Penjelasan *source code* 5.1:

Pada baris ke 4-9 adalah inisialisasi untuk mengacak 100 data latih, untuk mencari kelas 1 untuk mewakili kelas 1, kelas 0 untuk mewakili kelas 0. Baris 10 dan 11 adalah menampilkan hasil bobot yang ditentukan sendiri dari kelas 1 adalah indeks ke 3 dan kelas 0 adalah indeks ke 4 selanjutnya akan digunakan untuk bobot awal pada proses pelatihan.

**5.1.1.2** Proses Hitung Jarak

```

1 public void hitungjarak() {
2     int kelas[] = new int[barislatih];
3     double[] jarak = new double[2];
4     double jml = 0;
5     jarak[0] = 0;
6     jarak[1] = 0;
7     for (int i = 0; i < min.length; i++) {
8         System.out.println(min[i] + " - " + max[i]);
9     }
10    for (int a = 0; a < maxepoch; a++) {
11        hasil.append("Iterasi " + (a + 1) + "\n");
12        for (int i = 0; i < barislatih; i++) {
13            jml = 0;
14            hasil.append("Data " + (i + 1) + "\n");
15            for (int j = 0; j < 2; j++) { //data-w^2
16                for (int k = 0; k < kolomlatih - 1; k++)
17                {
18                    jml = jml
19                    +
20                    (Math.pow(datalatih[i][k] - bobotawal[j][k], 2));
21                }
22                System.out.print((Math.pow(datalatih[i][k]
23                -
24                bobotawal[j][k], 2)) + " ");
25            }
26            System.out.println(jml);
27            jarak[j] = Math.sqrt(jml); //jumlah akar
                jml = 0;
            }
            hasil.append("Jarak 1 = " + jarak[0] + "\n");
            hasil.append("Jarak 2 = " + jarak[1] + "\n");
        }
    }

```

**Source Code 5.2** Proses pelatihan *learning vector quantization* (LVQ)

Penjelasan *source code* 5.2:

Pada *Source code* 5.2 hitung jarak, dilakukan proses pengurangan antara data dengan masing-masing bobot 1 dan 0, lalu di pangkat duakan, setelah itu untuk mencari jarak paling minimum, dicari dengan menjumlahkan semua data 1 sampai dengan 10, setelah itu diakarkan, setelah mendapatkan hasilnya, dapat dibedakan anantara kelas 1 dan 0, manakah yang jaraknya paling kecil atau minimum. Setelah itu akan dilanjutkan pada proses berikutnya yaitu update bobot.

Proses Update Bobot

```

1     if (jarak[0] <= jarak[1]) { //jika kls 0(diterima)
2         if (datalatih[i][kolomlatih - 1] == 1) {
3

```

```

4         for (int j = 0; j < kolomlatih - 1;
5     j++) {
6             bobotbaru[0][j] =
7     bobotawal[0][j] + (alpha * (datalatih[i][j] -
8     bobotawal[0][j]));
9         }
10        for (int j = 0; j < kolomlatih - 1;
11    j++) {
12            bobotawal[0][j] =
13    bobotbaru[0][j];
14        }
15        } else {
16            for (int j = 0; j < kolomlatih - 1;
17    j++) {
18                bobotbaru[0][j] =
19    bobotawal[0][j] - (alpha * (datalatih[i][j] -
20    bobotawal[0][j]));
21            }
22            for (int j = 0; j < kolomlatih - 1;
23    j++) {
24                bobotawal[0][j] =
25    bobotbaru[0][j];
26            }
27        }
28        } else if (jarak[0] > jarak[1]) { //jika kls
29    1 (tdk diterima)
30            if (datalatih[i][kolomlatih - 1] == 0) {
31                for (int j = 0; j < kolomlatih - 1;
32    j++) {
33                    bobotbaru[1][j] =
34    bobotawal[1][j] + (alpha * (datalatih[i][j] -
35    bobotawal[1][j]));
36                }
37                for (int j = 0; j < kolomlatih - 1;
38    j++) {
39                    bobotawal[1][j] =
40    bobotbaru[1][j];
41                }
42            } else {
43                for (int j = 0; j < kolomlatih - 1;
44    j++) {
45                    bobotbaru[1][j] =
46    bobotawal[1][j] - (alpha * (datalatih[i][j] -
47    bobotawal[1][j]));
48                }
49                for (int j = 0; j < kolomlatih - 1;
50    j++) {
51                    bobotawal[1][j] =
52    bobotbaru[1][j];
53                }
54            }
55        }
56        if (jarak[0] < jarak[1]) { //jika 0 < 1
57            diterima
58                kelas[i] = 1;
59            } else {
60                kelas[i] = 0;
61            }
62        }

```

```

61         for (int w = 0; w < 2; w++) { //nampilkan
62 bobot baru
63             hasil.append("Bobot " + (w + 1) + " = ");
64             for (int j = 0; j < kolomlatih - 1; j++)
65             {
66                 hasil.append(bobotawal[w][j]      +
67 "\t");
68             }
69             hasil.append("\n");
70         }
71
72         jarak[0] = 0;
73         jarak[1] = 0;
74     }
75
76
77
78
79

```

**Source Code 5.3** Proses pelatihan *learning vector quantization* (LVQ)

Penjelasan *source code* 5.3:

Untuk proses update bobot pada *Source code* 5.3, baris 1 sampai dengan 18 adalah kondisi apabila kelas 1 diterima, melakukan pengecekan apakah kelas dari sistem sama dengan kelas yang sudah ada labelnya, jika benar, maka proses update bobot akan dilakukan perhitungan sesuai dengan persamaan 2.1, dan apabila salah maka proses yang dilakukan adalah persamaan 2.2.

Pada baris 19 sampai 36, sama halnya dengan sebelumnya, kondisi ini untuk kelas 1. Pengecekan hasil sistem dengan kelas yang sudah berlabel. Kemudian dilanjutkan dengan update bobot sesuai dengan kondisi, apabila benar akan diproses sesuai dengan persamaan 2.1 dan apabila tidak maka akan diproses dengan persamaan 2.2.

### 5.1.1.3 Proses Kondisi Berhenti dan Hitung Akurasi

```

1     alpha = alpha * decAlpha;
2         if (alpha < min_alpha) {
3             a = maxepoch;
4
5
6
7         }
8     }
9
10    int benar = 0;
11    for (int i = 0; i < barislatih; i++) {
12        hasil.append("Kelas data latih " + (i + 1) + " = "
13 + datalatih[i][kolomlatih - 1] + " dan kelas sistem = " +
14 kelas[i] + "\n");
15        if (datalatih[i][kolomlatih - 1] == kelas[i]) {
16            benar++;
17        }
18    }
19
20
21

```

```

22     }
23     double jmllatih = (double) barislatih;
24     double akurasi = (benar / jmllatih) * 100;
25     hasil.append("Akurasi sistem = " + akurasi + "%\n");
26     }
27 }
28 }
29 }

```

#### Source Code 5.4 Proses kondisi berhenti dan hitung akurasi

Pada Source code 5.4 pada setiap melakukan *iterasi*, nilai *alpha* dikalikan dengan pengali *alpha* untuk proses *update* bobotnya, untuk berhentinya proses, bias dilihat dalam beberapa kondisi, yang pertama iterasi dan yang kedua maksimum iterasi atau *epoch*, dapat dilihat pada baris ke 3 dan 4. Jika kelas data yang sudah ada labelnya sama dengan kelas target dari sistem maka akan disimpan sebagai benar, untuk rumus menghitung jumlah akurasi dapat dilihat pada baris ke 24 sampai dengan 27.

### 5.3.2 Proses Pengujian *Learning vector quantization* (LVQ)

```

1  double[] jarak = new double[2];
2  double jml;
3  jarak[0] = 0;
4  jarak[1] = 0;
5  for (int i = 0; i < barisuji; i++) {
6      jml = 0;
7      for (int j = 0; j < 2; j++) { //hitung data ke
8  tiap bobot
9          for (int k = 0; k < kolomuji - 2; k++) {
10         jml += Math.pow(datauji[i][k] -
11 bobotbaru[j][k], 2);
12     }
13     jarak[j] = Math.sqrt(jml); //jarak akhir
14     jml = 0;
15 }
16 hasil.append("Jarak 1 = " + jarak[0] + "\n");
17 hasil.append("Jarak 2 = " + jarak[1] + "\n");
18 if (jarak[0] < jarak[1]) { //jika 0 < 1 diterima
19     kelas[i] = 0;
20 } else {
21     kelas[i] = 1;
22 }
23 jarak[0] = 0;
24 jarak[1] = 0;
25 }
26
27 hasil.append("BOBOT\n");
28 for (int w = 0; w < 2; w++) { //nampilkan bobot baru
29     hasil.append("Bobot " + (w + 1) + " = ");
30     for (int j = 0; j < kolomlatih - 1; j++) {
31         hasil.append(bobotawal[w][j] + "\t");
32     }
33     hasil.append("\n");
34 }
35
36 int benar = 0;
37 for (int i = 0; i < datauji.length; i++) {

```

```

38     hasil.append("Kelas data uji " + (i + 1) + " = "
39 + datauji[i][kolomuji - 1] + " dan kelas sistem = " + kelas[i]
40 + "\n");
41         if (datauji[i][kolomuji - 1] == kelas[i]) {
42             benar++;
43         }}
44     double jmluji = (double) barisuji;
45     double akurasi = (benar / jmluji) * 100;
46     hasil.append("Akurasi sistem = " + akurasi + "%\n");
47

```

**Source Code 5.4** Proses pengujian *learning vector quantization*

Penjelasan source code 5.4 :

Source code 5.4 adalah proses pengujian, bedanya dengan pelatihan adalah tidak perlu adanya update bobot, bobot yang digunakan adalah hasil update bobot terakhir pada proses pelatihan, pada baris ke 1 sampai dengan 15 adalah proses perhitungan jarak antara data dengan bobot 1 dan bobot 2, data 1 dikurangkan dengan masing-masing bobot 1 dan bobot 2 lalu dipangkatkan dua. Dicari untuk jarak paling minimum, untuk dicocokkan dengan kelas yang sudah berlabel.

## 5.4 Implementasi Antarmuka Sistem

Antarmuka sistem merupakan tampilan sistem yang digunakan untuk interaksi antara pengguna dengan sistem. Pada sistem, terdapat halaman awal sistem, halaman *training* dan *testing*, halaman *input* data, dan halaman data.

### 5.4.1 Antarmuka Halaman Awal Sistem

Pada halama awal sistem, tertuju pada halaman *training* dan *testing*, karena untuk melakukan klasifikasi atau *input* data harus melakukan training data untuk mendapatkan bobot yang akan digunakan selanjutnya. perancangan antarmuka halaman tersebut dapat dilihat pada Gambar 5.1.



**Gambar 5.1** Halaman awal sistem

### 5.4.2 Antarmuka Halaman *Training* dan *Testing*

Petugas yang akan menggunakan sistem untuk menginputkan data baru, harus melakukan proses *training*, untuk mendapatkan bobot akhir yang akan digunakan untuk bobot awal pada saat input data baru. Dengan cara menekan tombol tampilan maka sistem akan menampilkan hasil data yang sudah di *training* dan juga di *testing*. Perancangan antarmuka halaman login admin dapat dilihat pada Gambar 5.2.

KLASIFIKASI PROGRAM KELUARGA HARAPAN

Pelatihan Pelatihan & Pengujian Input Data

Data Latih :

90.0	90.0	90.0	100.0	90.0	90.0	90.0	100.0	50.0	50.0	0.0
90.0	90.0	90.0	100.0	90.0	90.0	90.0	100.0	50.0	100.0	0.0
30.0	60.0	60.0	100.0	30.0	60.0	90.0	50.0	50.0	50.0	1.0
60.0	30.0	60.0	100.0	90.0	60.0	90.0	50.0	100.0	50.0	1.0
90.0	90.0	90.0	100.0	90.0	90.0	90.0	100.0	50.0	50.0	0.0
30.0	60.0	60.0	100.0	90.0	90.0	90.0	50.0	100.0	100.0	1.0
30.0	30.0	30.0	50.0	60.0	90.0	90.0	50.0	50.0	50.0	1.0
90.0	90.0	90.0	100.0	90.0	90.0	90.0	100.0	50.0	50.0	0.0
60.0	60.0	90.0	50.0	30.0	90.0	90.0	50.0	50.0	50.0	1.0
60.0	60.0	60.0	50.0	60.0	90.0	90.0	100.0	50.0	50.0	1.0
90.0	90.0	90.0	100.0	60.0	90.0	90.0	100.0	100.0	50.0	0.0
60.0	30.0	30.0	50.0	60.0	60.0	90.0	50.0	100.0	100.0	1.0
60.0	60.0	60.0	50.0	30.0	60.0	90.0	50.0	50.0	50.0	1.0
30.0	30.0	60.0	50.0	90.0	90.0	30.0	50.0	50.0	50.0	1.0
30.0	60.0	60.0	100.0	90.0	90.0	90.0	50.0	50.0	50.0	1.0
90.0	90.0	90.0	100.0	90.0	90.0	90.0	100.0	50.0	100.0	0.0
30.0	60.0	60.0	100.0	90.0	90.0	90.0	50.0	50.0	50.0	1.0
90.0	90.0	90.0	100.0	90.0	90.0	90.0	100.0	100.0	50.0	0.0
90.0	90.0	90.0	100.0	90.0	90.0	90.0	100.0	50.0	50.0	0.0
90.0	90.0	90.0	100.0	90.0	90.0	90.0	100.0	100.0	50.0	0.0
60.0	30.0	60.0	100.0	90.0	90.0	90.0	50.0	50.0	50.0	1.0
90.0	90.0	90.0	100.0	90.0	90.0	90.0	100.0	50.0	100.0	0.0
90.0	90.0	90.0	100.0	90.0	90.0	90.0	100.0	50.0	50.0	0.0
60.0	60.0	60.0	100.0	90.0	90.0	90.0	50.0	50.0	50.0	1.0
90.0	90.0	60.0	100.0	90.0	90.0	90.0	100.0	50.0	50.0	0.0

KLASIFIKASI PROGRAM KELUARGA HARAPAN

Pelatihan Pelatihan & Pengujian Input Data

Kelas data latih 77 = 1.0 dan kelas sistem = 1  
 Kelas data latih 78 = 0.0 dan kelas sistem = 0  
 Kelas data latih 79 = 0.0 dan kelas sistem = 0  
 Kelas data latih 80 = 0.0 dan kelas sistem = 0  
 Kelas data latih 81 = 0.0 dan kelas sistem = 0  
 Kelas data latih 82 = 1.0 dan kelas sistem = 1  
 Kelas data latih 83 = 1.0 dan kelas sistem = 0  
 Kelas data latih 84 = 1.0 dan kelas sistem = 0  
 Kelas data latih 85 = 1.0 dan kelas sistem = 1  
 Kelas data latih 86 = 0.0 dan kelas sistem = 0  
 Kelas data latih 87 = 1.0 dan kelas sistem = 1  
 Kelas data latih 88 = 0.0 dan kelas sistem = 0  
 Kelas data latih 89 = 0.0 dan kelas sistem = 0  
 Kelas data latih 90 = 0.0 dan kelas sistem = 0  
 Kelas data latih 91 = 0.0 dan kelas sistem = 0  
 Kelas data latih 92 = 1.0 dan kelas sistem = 0  
 Kelas data latih 93 = 0.0 dan kelas sistem = 0  
 Kelas data latih 94 = 1.0 dan kelas sistem = 1  
 Kelas data latih 95 = 0.0 dan kelas sistem = 0  
 Kelas data latih 96 = 1.0 dan kelas sistem = 1  
 Kelas data latih 97 = 1.0 dan kelas sistem = 0  
 Kelas data latih 98 = 1.0 dan kelas sistem = 1  
 Kelas data latih 99 = 0.0 dan kelas sistem = 0  
 Kelas data latih 100 = 1.0 dan kelas sistem = 1  
 Akurasi sistem = 85.0%

Gambar 5.2 Halaman *Training* dan *Testing*



### 5.4.3 Antarmuka Halaman Input Data

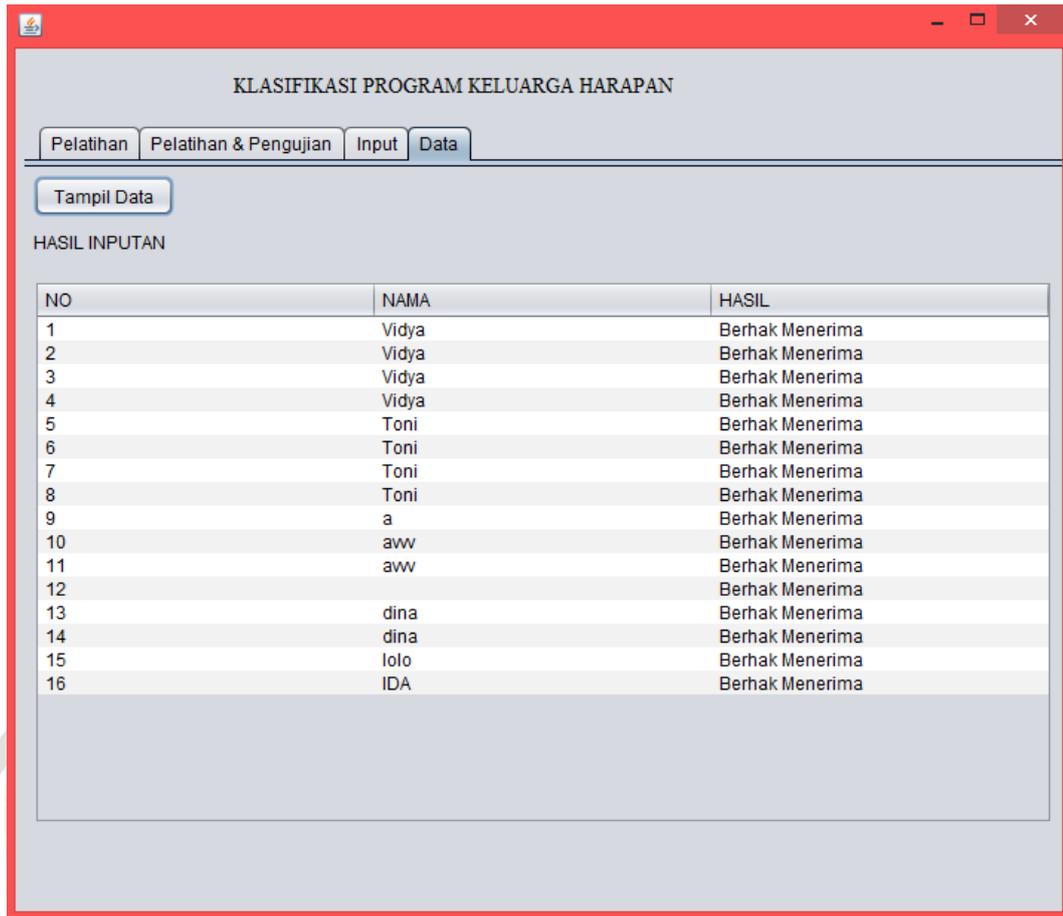
Pada halaman *input* data digunakan untuk melakukan proses klasifikasi penerima dana Program Keluarga Harapan, petugas harus mengisi semua data yang benar, ada nama, lalu mengisi 10 kriteria yang harus sesuai dengan masing-masing kepala keluarga. Setelah mengisi 10 kriteria, maka selanjutnya admin harus menekan tombol selesai untuk mengetahui hasilnya. Perancangan antarmuka halaman klasifikasi atau *input* bisa dilihat pada Gambar 5.3.

KLASIFIKASI PROGRAM KELUARGA HARAPAN	
Pelatihan   Pelatihan & Pengujian   <b>Input</b>   Data	
Inputkan kriteria-kriteria yang akan diklasifikasi	
Nama	<input type="text" value="IDA"/>
Luas Tanah Bangunan	<input type="text" value="SEMPIT"/>
Jenis Lantai Bangunan Terbuat dari Tanah atau Bambu	<input type="text" value="TANAH"/>
Jenis Dinding Bangunan Terbuat dari Bambu, Kayu atau Semen	<input type="text" value="BAMBU"/>
Jamban	<input type="text" value="TIDAK MEMILIKI"/>
Sumber Air Untuk Keseharian Berasal dari Sumur, Sanyo atau PDAM	<input type="text" value="SUMUR"/>
Sumber Penerangan Dirumah Bukan Listrik, Gabung Tetangga atau PLN	<input type="text" value="BUKAN LISTRIK"/>
Bahan Bakar Yang Digunakan untuk masak berasal dari minyak tanah, kayu, atau LPG	<input type="text" value="MINYAK TANAH"/>
Tidak Mampu Membayar Biaya Anggota Keluarga Berobat ke Puskesmas	<input type="text" value="IYA"/>
Memiliki Ibu Hamil/nifas/anak balita	<input type="text" value="IYA"/>
Memiliki anak yang sedang sekolah	<input type="text" value="IYA"/>
<input type="checkbox"/> Berhak Menerima	<input type="button" value="SELESAI"/>

Gambar 5.3 Halaman input data

### 5.4.4 Antarmuka Halaman Tampilkan Data

Halaman ini menampilkan nama calon penerima dana Program Keluarga Harapan yang sudah di isikan pada halaman *input* sebelumnya. Menampilkan nama dan hasil dari sistem apakah orang tersebut menerima atau tidak Program Keluarga Harapan. Perancangan antarmuka halaman hasil klasifikasi dapat dilihat pada Gambar 5.4.



Gambar 5.4 Halaman tampilan data

## BAB 6 PENGUJIAN DAN ANALISIS

Bab pengujian dan analisis akan membahas mengenai tahap pengujian terhadap sistem “Klasifikasi Penerimaan Program Keluarga Harapan (PKH) Menggunakan Metode *Learning vector quantization*”. Pada pengujian ini, menguji 5 parameter diantaranya adalah pengujian pengaruh perubahan nilai *learning rate* terhadap akurasi, pengujian *epoch* atau *iterasi* terhadap akurasi, pengujian nilai pengali *learning rate* atau *decalpha* terhadap akurasi, pengujian jumlah data latih yang digunakan dan pengujian nilai *minimum alpha* terhadap hasil akurasi, pengujian dilakukan sebanyak 10 kali dengan menggunakan bobot yang berbeda. Menggunakan 100 data latih, dan 2 kelas. Kemudian dari hasil 5 pengujian parameter, akan diambil nilai-nilai terbaik yang akan digunakan pada pengujian data uji.

### 6.1 Pelatihan Metode *Learning vector quantization*

Untuk melakukan tahap pelatihan pada lvq memerlukan data latih, bobot, menentukan parameter awal yaitu *alpha*, *DecAlpha*, *Epoch*, dan *MinAlpha*. Pada pelatihan ini menggunakan 100 data latih.

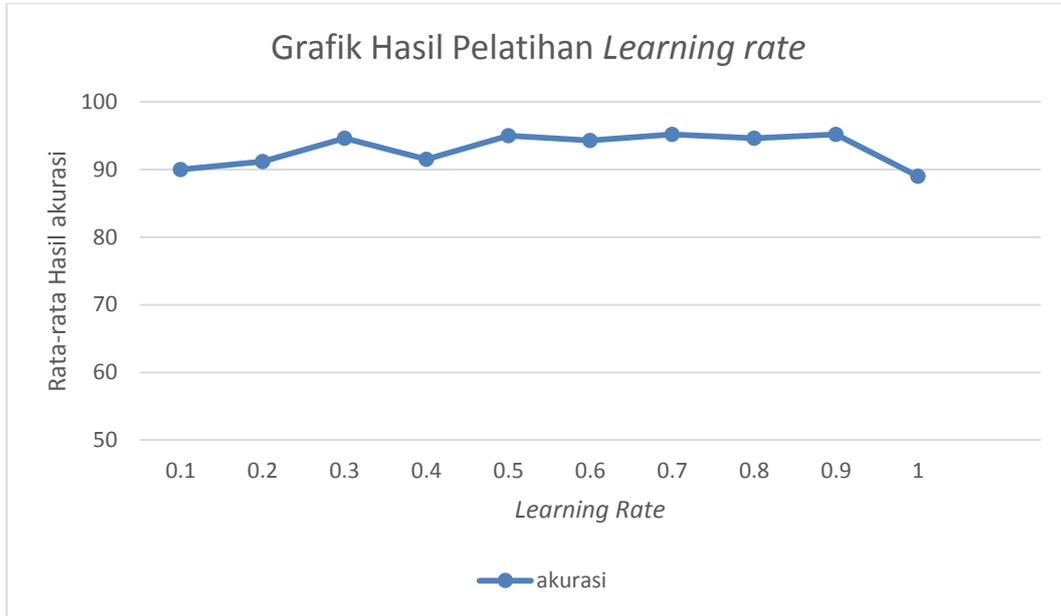
#### 6.1.1 Pengaruh Nilai *Learning rate (Alpha)* Terhadap Hasil Akurasi

Pengujian pengaruh nilai *learning rate* terhadap hasil akurasi. Dilakukan sebanyak 10 kali percobaan untuk setiap nilai *learning rate* dengan nilai 0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8, 0,9 dan 1. Parameter awal yang digunakan adalah *learning rate* 0,1, pengali alpha 0,1, epoch sebanyak 2 dan *MinAlpha* 0,01. Bobot awal yang digunakan didapatkan dari data yang ada. Berikut Tabel 6.1 menjabarkan hasil pelatihan nilai *learning rate*.

Tabel 6.1 Hasil pengujian *learning rate*

<i>learning rate</i>	Percobaan Ke-										Rata-rata(%)
	1	2	3	4	5	6	7	8	9	10	
0.1	92	93	98	86	88	99	78	79	99	93	90.5
0.2	97	98	85	98	93	54	97	98	94	98	91.2
0.3	93	97	98	98	94	98	98	87	85	98	94.6
0.4	81	98	98	98	82	94	98	98	95	73	91.5
0.5	91	98	98	98	98	83	93	98	99	94	95
0.6	97	99	95	78	95	96	96	98	99	90	94.3
0.7	89	90	99	99	98	95	99	94	92	97	95.2
0.8	78	99	98	89	96	99	98	96	99	94	94.6
0.9	84	97	98	97	98	99	97	97	91	94	95.2
1	98	97	98	98	97	98	99	85	98	84	95.2

Untuk memudahkan pembaca menganalisis hasil pelatihan dengan mengubah ubah nilai *learning rate* maka dibawah ini adalah hasil pelatihan *learning rate* pada Gambar 6.1.



**Gambar 6.1 Grafik hasil pelatihan *learning rate***

Pada Gambar 6.1, setelah dilakukan pengujian dari *learning rate* 0,1 sampai dengan 1, dengan menggunakan bobot secara acak, dilakukan 10 kali percobaan. Penentuan nilai *learning rate*. Nilai *alpha* baiknya memenuhi  $0 < \alpha < 1$ , akan tetapi disetiap penelitian memberikan hasil yang berbeda beda, karna pengaruh sebuah *alpha* adalah, semakin kecil nilainya, maka laju pembeajaran akan stabil, sebaliknya semakin besar maka akan tidak stabil. Pada penelitian ini, berdasarkan hasil pelatihan *learning rate* menghasilkan rata-rata paling maksimal yaitu terdapat pada nilai *learning rate* 0,7. Untuk pengujian selanjutnya, akan menggunakan nilai *learning rate* 0,7

**6.1.2 Pengaruh Pengali *Learning rate* (DecAlpha) Terhadap Hasil Akurasi**

Skenario kedua, pengujian pengaruh nilai pengali *learning rate*, pengujian ini dilakukan untuk mengetahui pengaruh nila pengali terhadap akurasi, karena semakin besar nilainya maka akan semakin cepat nilai *learning rate* mencapai *minimum alpha* yang digunakan untuk batas perulangan. Pada pengujian ini menggunakan nilai *Decalpha* 0,1 sampai dengan 1. Pengujian pengaruh nilai *DecAlpha* dapat dilihat pada Tabel 6.2.

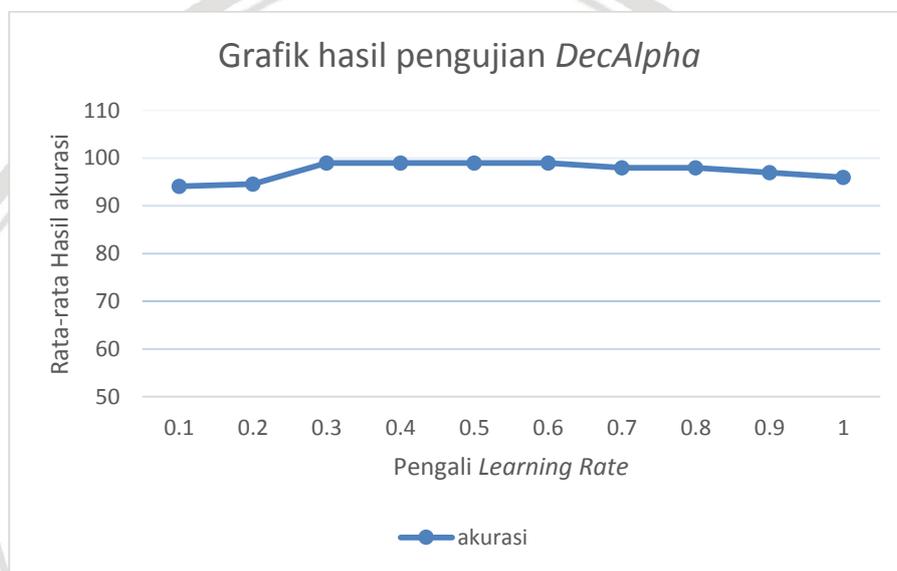
**Tabel 6.2 Hasil pengujian pengali *learning rate***

DecAlpha	Percobaan Ke										Rata-Rata(%)
	1	2	3	4	5	6	7	8	9	10	
0.1	93	89	97	83	98	96	95	98	94	98	94.1
0.2	99	97	99	99	90	99	99	75	90	99	94.6
0.3	99	99	99	99	99	99	99	99	99	99	99



0.4	99	99	99	99	99	99	99	99	99	99	99
0.5	99	99	99	99	99	99	99	99	99	99	99
0.6	99	99	99	99	99	99	99	99	99	99	99
0.7	98	98	98	98	98	98	98	98	98	98	98
0.8	98	98	98	98	98	98	98	98	98	98	98
0.9	97	97	97	97	97	97	97	97	97	97	97
1	96	96	96	96	96	96	96	96	96	96	96

Pada Tabel 6.2 menunjukan hasil rata-rata akurasi pengaruh nilai pengali *learning rate* dalam tahap pelatihan. Untuk memudahkan pembaca menganalisis hasil pelatihan dengan mengubah ubah nilai *learning rate* maka dibawah ini adalah hasil pelatihan pengali *learning rate* pada Gambar 6.2.



**Gambar 6.2 Grafik pelatihan pengali learning rate**

Berdasarkan grafik diatas, setelah dilakukan pengujian dari pengali *learning rate* atau *DecAlpha* 0,1 sampai dengan 1, dengan menggunakan bobot yang acak, dilakukan 10 kali percobaan dan hasilnya adalah pada nilai 0,1 memperoleh rata-rata 94,1, pada saat 0,2 sampai dengan 0,6 naik sampai dengan 0,3, lalu turun hingga akurasi mencapai 96% pada nilai pengali *learning rate* 1. Jadi, dari hasil pengujian pengali *learning rate*, didapatkan hasil terbaik pada saat nilainya 0,3. Untuk pengujian selanjutnya akan digunakan nilai pengali *learning rate* 0,3. Maka untuk pengujian selanjutnya *DecAlpha* yang baik digunakan adalah 0,3 dengan rata-rata akurasi 99%.

### 6.1.3 Pengaruh Maksimum iterasi (*Epoch*) Terhadap Hasil Akurasi

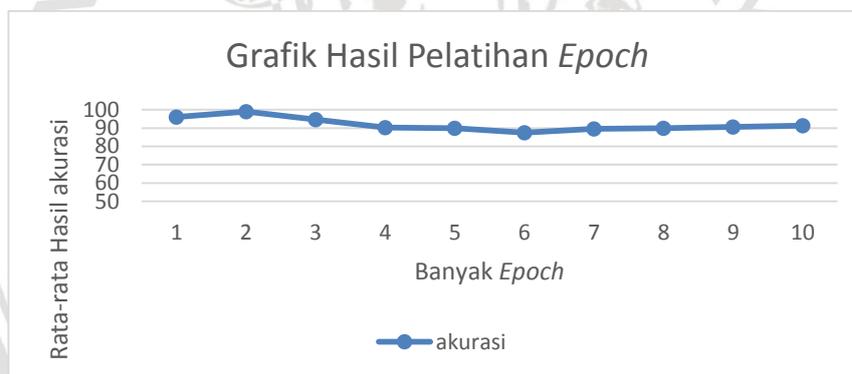
Pengujian pengaruh iterasi ini dilakukan untuk melihat bagaimana pengaruh nilai iterasi tersebut terhadap hasil dari akurasi. Pengujian dilakukan dengan menggunakan banyak *epoch* 1 sampai dengan 10, *Learning rate* 0,7 Nilai pengali *learning rate* yang digunakan 0,3, dan *MinAlpha* 0,01. Pengaruh yang diberikan oleh iterasi bahwa apabila semakin besar nilai iterasinya maka proses yang akan

dilakukan akan semakin banyak. Perancangan pengujian maksimum *epoch* dapat dilihat pada Tabel 6.3.

**Tabel 6.3 Perancangan pengujian *epoch***

<i>epoch</i>	Percobaan ke										Rata-rata(%)
	1	2	3	4	5	6	7	8	9	10	
1	96	96	96	96	96	96	96	96	96	96	96
2	99	99	99	99	99	99	99	99	99	99	99
3	97	97	95	96	91	91	97	91	96	95	94.6
4	99	99	89	94	89	85	89	85	89	85	90.3
5	94	89	85	94	85	96	94	85	93	85	90
6	85	94	85	85	90	85	85	96	85	85	87.5
7	96	89	90	94	89	89	85	90	89	85	89.6
8	89	94	85	85	94	98	94	85	90	85	89.9
9	94	96	85	94	94	94	85	85	94	85	90.6
10	85	89	89	94	94	96	85	96	89	96	91.3

Untuk memudahkan pembaca menganalisis hasil pelatihan dengan mengubah ubah nilai *learning rate* maka dibawah ini adalah hasil pelatihan pengali *learning rate* pada Gambar 6.3.



**Gambar 6.3 Grafik pengujian maksimal epoch**

Dari grafik diatas setelah dilakukan pengujian dari *epoch* 1 sampai dengan 10, *epoch* atau iterasi digunakan untuk kondisi berhentinya proses pembelajaran, apabila nilai iterasi makin banyak, maka proses yang akan dilakukan oleh sistem akan semakin banyak dan lama, sebaliknya, jika nilainya kecil maka proses pembelajaran yang dilakukan dengann cepat. Pada pengujian ini, hasil rata-rata maksimum yang didapatkan pada *epoch* ke2 dengan rata-rata hasil akurasinya adalah 99%, jadi untuk skenario selanjutnya akan menggunakan *epoch* 2.

#### 6.1.4 Pengaruh Jumlah Data Latih Terhadap Hasil Akurasi

Pada skenario keempat ini akan dilakukan pengujian pegraruh jumlah data latib terhadap hasil akurasi. Pengujian pengaruh jumlah data latih dilakukan dengan menggunakan nilai *learning rate* 0,7, pengali *alpha* 0,3 dan *epoch* 2, dan *MinAlpha* 0,01, pelatihan data latih banyak 10%,20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% dan

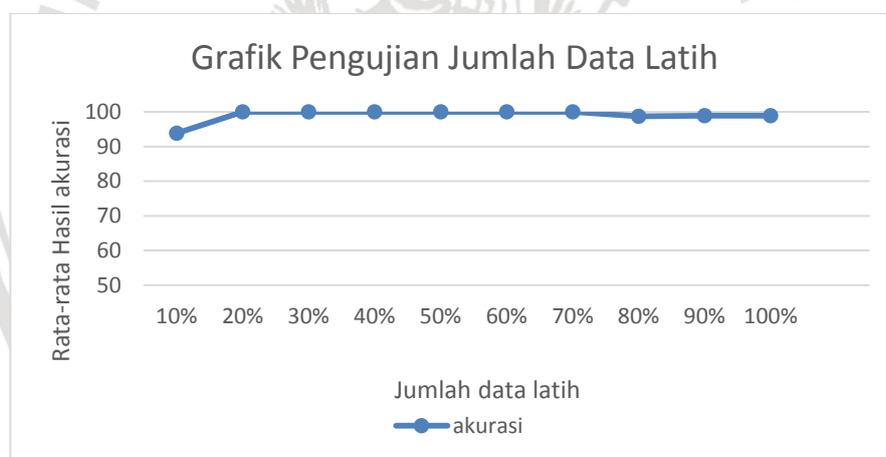


100% dari total 100 data. Pengujian pengaruh jumlah data latih dapat dilihat pada Tabel 6.4.

**Tabel 6.4 Pengujian jumlah data latih**

Jumlah data latih	Percobaan Ke-										Rata-rata(%)
	1	2	3	4	5	6	7	8	9	10	
10%	100	100	90	90	98	90	90	100	90	90	93.8
20%	100	100	100	100	100	100	100	100	100	100	100
30%	100	100	100	100	100	100	100	100	100	100	100
40%	100	100	100	100	100	100	100	100	100	100	100
50%	100	100	100	100	100	100	100	100	100	100	100
60%	100	100	100	100	100	100	100	100	100	100	100
70%	100	100	100	100	100	100	100	100	100	100	100
80%	98.75	98.75	98.75	98.75	98.75	98.75	98.75	98.75	98.75	98.75	98.75
90%	98.88	98.88	98.88	98.88	98.88	98.88	98.88	98.88	98.88	98.88	98.88
100%	98.88	98.88	98.88	98.88	98.88	98.88	98.88	98.88	98.88	98.88	98.88

Untuk memudahkan pembaca menganalisis hasil pelatihan dengan mengubah ubah nilai *learning rate* maka dibawah ini adalah hasil pelatihan data latih pada Gambar 6.2



**Gambar 6.4 Grafik hasil pengujian jumlah data latih**

Berdasarkan Gambar 6.4, dilakukan pengujian jumlah data latih 10% hingga 100% dari total data latih sebanyak 100 data, dengan menggunakan bobot yang acak. Dapat dilihat bahwa ketika jumlah data latih 10% memperoleh 93,8%, selanjutnya ketika 20% sampai dengan 70% data latih, akurasi naik hingga 100%, saat jumlah data latih 80 sampai dengan 100%, akurasi turun dengan rata-rata akurasi 98.8%. Mekan untuk scenario selanjutnya akan digunakan 70% data latih dan 30% data uji dari keseluruhan jumlah data.

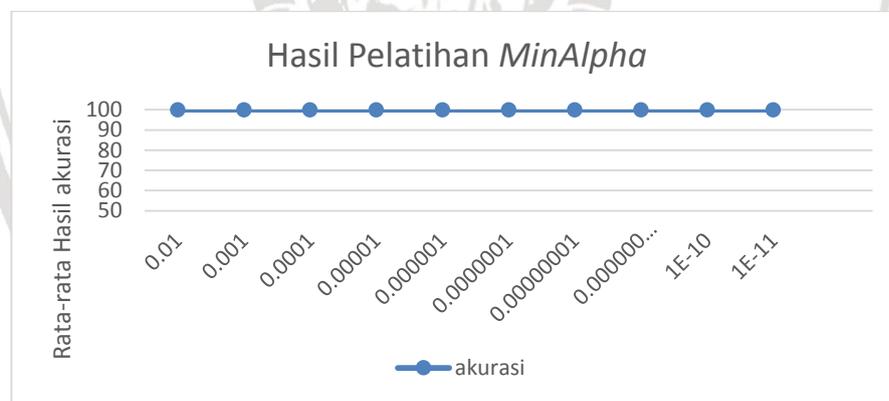
### 6.1.5 Pengaruh Minimum Learning Rate Terhadap Hasil Akurasi

Pengujian yang terakhir, yaitu pengujian nilai minimum *learning rate* untuk mengetahui pengaruhnya terhadap hasil akurasi. Pengujian dilakukan dengan *learning rate* 0,7, pengali *alpha* 0,3 dan *epoch* 2, dan pelatihan data latih sebanyak 70%, pengujian minimum *learning rate* menggunakan nilai 0,1 sampai dengan 0,000000001. Hasil dari pengujian dapat dilihat pada Tabel 6.5.

**Tabel 6.5 Pelatihan minimum *alpha***

MinAlpha	Percobaan Ke										Rata-rata (%)
	1	2	3	4	5	6	7	8	9	10	
0.01	100	100	100	100	100	100	100	100	100	100	100
0.001	100	100	100	100	100	100	100	100	100	100	100
0.0001	100	100	100	100	100	100	100	100	100	100	100
0.00001	100	100	100	100	100	100	100	100	100	100	100
0.000001	100	100	100	100	100	100	100	100	100	100	100
1E-07	100	100	100	100	100	100	100	100	100	100	100
1E-08	100	100	100	100	100	100	100	100	100	100	100
1E-09	100	100	100	100	100	100	100	100	100	100	100
1E-10	100	100	100	100	100	100	100	100	100	100	100
1E-11	100	100	100	100	100	100	100	100	100	100	100

Untuk memudahkan pembaca menganalisis hasil pelatihan dengan mengubah nilai minimum *alpha* maka dibawah ini adalah hasil pelatihan minimum *alpha* pada Gambar 6.5.



**Gambar 6.5 Grafik hasil pelatihan *MinAlpha***

*MinAlpha* atau minimum *alpha* merupakan satu tahap dimana berhentinya suatu proses pelatihan, proses perulangan atau iterasi semakin banyak apabila nilai dari *MinAlpha* ini semakin kecil, dan sebaliknya jika nilainya besar maka proses akan cepat selesai. Pada grafik diatas, menunjukkan hasil pengujian dari nilai 0,01 samapai dengan 1E-11, hasil yang didapatkan adalah 100%, hal ini bias terjadi karena proses sebelumnya telah diambil *learning rate* 0,7 dan pengali *learning rate* terbaik yaitu 0,3 yang mengakitbatnya *konvergen* sejak minimum *learning rate* 0,01. Sehingga dapat diambil kesimpulan dari pengujian kelima ini nilai minimum *learning rate* yang diambil adalah 0.01 karena nilai tersebut merupakan nilai parameter

minimum *learning rate* terbesar yang diuji namun memiliki hasil akurasi terbesar dengan 100%.

## 6.2 Pengujian Metode *Learning vector quantization*

Dari hasil pengujian yang sudah dilakukan sebelumnya, didapatkan beberapa parameter terbaik, yaitu *Alpha* 0,7, pengali *alpha* 0,3, minimum *alpha* 0.01 dan jumlah data uji sebanyak 45 data.

Dengan menggunakan parameter diatas, pengujian dapat dilihat pada Tabel 6.6.

**Tabel 6.6 Hasil pengujian *learning vector quantization***

Percobaan ke-	Akurasi
1	100%
2	100%
3	100%
4	100%
5	100%
6	100%
7	100%
8	100%
9	100%
10	100%
Rata-rata	100%

## BAB 7 PENUTUP

Bab ini menjelaskan kesimpulan dan saran dari hasil penelitian klasifikasi penerimaan Program Keluarga Harapan menggunakan metode *learnin vector quantization*. Penjelasan kesimpulan dan saran adalah sebagai berikut.

### 1.1 Kesimpulan

Berdasarkan hasil dari perancangan, implementasi dan pengujian yang telah dilakukan sebelumnya didapatkan sebuah kesimpulan yang menjawab rumusan masalah sebagai berikut:

1. Proses yang harus dilakukan dalam mengimplementasikan *Learning vector quantization* (LVQ) untuk klasifikasi penerimaan Program Keluarga Harapan adalah dengan mengumpulkan data kriteria-kriteria syarat calon penerima bantuan Program Keluarga Harapan. Proses pertama adalah pelatihan, yang harus disiapkan adalah bobot awal yang bisa didapatkan dari 100 data latih yang akan diujikan, lalu menentukan parameter awal seperti *learning rate*, pengali *alpha*, *epoch*, dan juga *MinAlpha*. Selanjutnya dapat dilakukan pelatihan untuk mencari parameter terbaik dan hasilnya akan digunakan untuk melakukan pengujian atau klasifikasi.
2. Berdasarkan 5 skenario pengujian yang sudah dilakukan, sistem merekomendasikan *learning rate* dengan nilai terbaik yaitu 0,7, pengali *learning rate* dengan nilai paling baik yaitu 0,3, *epoch* adalah 2, minimal *alpha* 0,01 jumlah data latih sebanyak 100%, dari parameter tersebut dihasilkan nilai akurasi sebesar 100%.

### 1.2 Saran

Peneliti mengaggap bahwa penelitian ini masih jauh dari sempurna, karena itu berikut adalah beberapa saran yang dapat peneliti berikan untuk pengembangan skripsi selanjutnya.

1. Menggunakan data yang lebih bervariasi atau menambah kriteria pada data agar hasil akurasi yang didapatkan dapat mencapai akurasi terbaik dan beraturan, yang artinya hasil akurasi dari pengujian cenderung naik atau turun.

## DAFTAR PUSTAKA

- Afriandi, E., 2016. *Identifikasi Telapak Tangan Menggunakan Jaringan Syaraf Tiruan Learning Vector Quantization (LVQ)*. S1. Fakultas Ilmu Komputer Universitas Brawijaya Malang.
- Agustinus, I., 2017. *Klasifikasi Risiko Hipertensi Menggunakan Metode Learning Vector Quantization*. S1. Fakultas Ilmu Komputer Universitas Brawijaya Malang.
- Arifando, R., 2016. *Klasifikasi Calon Penerima Bantuan Keluarga Miskin Menggunakan Metode Learning vector quantization*. S1. Fakultas Ilmu Komputer Universitas Brwajijaya Malang.
- BPS/Badan Pusat Statistik dan Depsos/Departemen Sosial (2002), *Penduduk Fakir Miskin Indonesia 2002*, Jakarta: BPS
- BPS/Badan Pusat Statistik Kabupaten Jombang “Angka Kemiskinan Kabupaten Jombang 2017” 9 Oktober 2018.  
<https://jombangkab.bps.go.id/statictable/2018/10/16/390/angka-kemiskinan-kabupaten-jombang-2012---2017.html>
- Fausett, L., 1994. *Fundamentals of Neural Networks : Architectures, Algorithms, and Applications*. Prentice Hall Inc. New Jersey.
- Hamidi, R., 2017. *Implementasi Learning Vector Quantization (LVQ) untuk Klasifikasi Kualitas Air Sungai*. S1. Fakultas Ilmu Komputer Universitas Brawijaya.
- Kusumadewi, S., 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Graha Ilmu. Yogyakarta."
- Nugroho, H., 1995. *Kemiskinan, Ketimpangan dan Kesenjangan*. Yogyakarta: Aditya Media.
- Suryawati. 2004. *Teori Ekonomi Mikro*. UPP. AMP YKPN. Yogyakarta: Jarnasy.
- Umam, M. S., 2015. *Implementasi Metode Learning Vector Qauntization(LVQ) Untuk Identifikasi Penyakit Pada Citra Daun Tanaman Kedelai* . Fakultas Ilmu Komputer. Universitas Brawijaya.
- Wijayanti, D. M. P., 2016. *Identifikasi Diagnosis Perubahan Hasil Perawatan Kulit Menggunakan Learning Vector Qauntization(LVQ)*. Malang. Fakultas Ilmu Komputer. Universitas Brawijaya.
- Umam, M. S., 2015. *Implementasi Metode Learning Vector Qauntization(LVQ) Untuk Identifikasi Penyakit Pada Citra Daun Tanaman Kedelai* . Fakultas Ilmu Komputer. Universitas Brawijaya.