

**SISTEM PENDETEKSI DIABETES MELITUS MENGGUNAKAN
JARINGAN SYARAF TIRUAN LVQ (*LEARNING VECTOR
QUANTIZATION*)**



SKRIPSI

**Disusun Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Komputer
pada Departemen Ilmu Komputer/ Informatika**

Disusun Oleh :

REYHAN SYARIF

24010313130104

**DEPARTEMEN ILMU KOMPUTER/ INFORMATIKA
FAKULTAS SAINS DAN MATEMATIKA
UNIVERSITAS DIPONEGORO**

2017

PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya menyatakan bahwa dalam tugas akhir/ skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan di dalam daftar pustaka.

Semarang,

Reyhan Syarif
24010313130104

HALAMAN PENGESAHAN

Judul : Sistem Pendekripsi Diabetes Melitus Menggunakan Jaringan Syaraf Tiruan LVQ
(Learning Vector Quantization)

Nama : Reyhan Syarif

NIM : 24010313130104

Telah diujikan pada sidang tugas akhir tanggal

Mengetahui,

Ketua Departemen Ilmu Komputer/Informatika

Panitia Pengujii Tugas Akhir

Ketua,

Dr. Retno Kusumaningrum, S.Si, M.Kom

.....

NIP 19810420 200501 2 001

NIP

ABSTRAK

Pada tahun 2012 kanker payudara menempati peringkat kedua kanker yang paling sering ditemui di dunia. Di Indonesia, kanker payudara menempati peringkat kedua sebagai kanker paling mematikan. Setidaknya, diperkirakan 10 dari 100.000 penduduk di Indonesia terkena penyakit kanker payudara. Diagnosis dini diperlukan karena pengobatan pada stadium lanjut lebih sulit dilakukan dibandingkan dengan pengobatan pada stadium dini. Salah satu cara yang dapat dilakukan untuk melakukan diagnosis dini kanker payudara adalah dengan mengidentifikasi profil *microRNA* karena *microRNA* merupakan *biomarker* dengan tingkat spesifisitas tinggi. Diagnosis dini dapat dilakukan menggunakan jaringan syaraf tiruan yaitu *Backpropagation*. Pada penelitian ini penulis mengembangkan model Jaringan Syaraf Tiruan *Backpropagation* yang dapat digunakan untuk diagnosis dini kanker payudara berdasarkan profil *microRNA* dengan Algoritma Optimasi Nguyen-Widrow dan Momentum. Hasil penelitian menunjukkan bahwa model yang dikembangkan dapat mencapai nilai rata-rata akurasi pengujian terbaik yaitu 99%.

Kata Kunci : Jaringan syaraf tiruan, *Backpropagation*, *Under-sampling*, Nguyen-Widrow, Momentum, Kanker payudara, *MicroRNA*

ABSTRACT

In 2012 breast cancer was ranked as second most common cancer in the world. In Indonesia, breast cancer was ranked second as the most deadly cancer. At least, an estimated 10 out of 100,000 people in Indonesia were affected by breast cancer. Early diagnosis is necessary because treatment at an advanced stage is more difficult than treatment at an early stage. Early diagnosis of breast cancer can be done with identifying the microRNA profile since microRNA is a biomarker with a high degree of specificity. This early-diagnosis can be done using Artificial Neural Network Backpropagation. In this study the authors developed a model of Artificial Neural Network Backpropagation that can be used for early diagnosis of breast cancer based on microRNA profile with Nguyen-Widrow and Momentum Optimization Algorithm. The results show that the developed model can achieve the best average test accuracy value of 99%.

Keywords : Artificial neural network, Backpropagation, Under-sampling, Nguyen-Widrow, Momentum, Breast cancer, MicroRNAs

DAFTAR ISI

PERNYATAAN KEASLIAN SKRIPSI	ii
HALAMAN PENGESAHAN	iii
ABSTRAK	iv
ABSTRACT	v
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
DAFTAR LAMPIRAN	xi
BAB I	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan dan Manfaat	3
1.4. Ruang Lingkup	3
1.5. Sistematika Penulisan.....	4
BAB II	5
TINJAUAN PUSTAKA	5
2.1. Perkembangan Penelitian dalam Prediksi Diabetes Melitus.....	5
2.2. Diabetes Melitus.....	5
2.2.1. Faktor Risiko Diabetes Melitus.....	6
2.2.2. Diagnosa Diabetes Melitus.....	7
2.3. Jaringan Syaraf Tiruan	7
2.3.1. Arsitektur Jaringan Syaraf Tiruan	8
2.3.2. Pelatihan Jaringan Syaraf Tiruan	8
2.4. Algoritma <i>Learning Vector Quantization</i>	9
2.4.1. Arsitektur Jaringan <i>Learning Vector Quantization</i>	10
2.4.2. Pelatihan Standar <i>Learning Vector Quantization</i>	10
2.5. Evaluasi Kinerja <i>Classifier</i>	16
2.5.1. <i>K-Fold Cross Validation</i>	16
2.5.2. <i>Confussion Matrix</i>	16
2.6. Normalisasi Min Max.....	18
2.7. Model Waterfall	18

2.8.	Pemodelan Analisis	20
2.8.1.	Pemodelan Data	21
2.8.2.	Pemodelan Fungsional	22
2.9.	Perancangan Struktur Data	23
2.10.	Bahasa Pemrograman PHP	24
2.11.	DBMS MySQL.....	25
2.12.	Pengujian Perangkat Lunak	25
BAB III		28
METODOLOGI PENELITIAN		28
3.1.	Pengumpulan Data.....	28
3.2.	Arsitektur Sistem	29
3.3.	Pemetaan Data	31
3.4.	Normalisasi Data	31
3.5.	Pembagian Data Latih dan Data Uji.....	33
3.6.	Pelatihan Menggunakan <i>Learning Vector Quantization</i>	34
3.6.1.	<i>Flowchart</i> Fungsi Pelatihan LVQ.....	34
3.6.2.	Perhitungan Manual Pelatihan LVQ	36
3.7.	Pengujian Menggunakan <i>Learning Vector Quantization</i>	38
3.7.1.	<i>Flowchart</i> Fungsi Pengujian LVQ.....	39
3.7.2.	Perhitungan Manual Pengujian LVQ	40
3.8.	Evaluasi Klasifikasi	42
3.9.	Analisis Kebutuhan	43
3.9.1.	Kebutuhan Pengguna	43
3.9.2.	Kebutuhan Fungsional dan Non Fungsional.....	44
3.9.3.	Pemodelan Data Sistem Pendekripsi Diabetes Melitus	45
3.9.4.	Pemodelan Fungsional Sistem Pendekripsi Diabetes Melitus.....	46
3.10.	Desain Sistem	50
3.10.1.	Desain Struktur Data	51
3.10.2.	Desain Antarmuka	52
3.10.3.	Desain Fungsi.....	63
BAB IV		76
IMPLEMENTASI DAN PENGUJIAN		76
4.1.	Implementasi.....	76
4.1.1.	Spesifikasi Perangkat	76
4.1.2.	Implementasi Data	76
4.1.3.	Implementasi Fungsi	79

4.1.4.	Implementasi Antarmuka.....	80
4.2.	Pengujian Fungsional.....	91
4.2.1.	Rencana Pengujian	91
4.2.2.	Pelaksanaan Pengujian.....	92
4.2.3.	Evaluasi Pengujian.....	92
4.3.	Pengujian Jaringan Syaraf Tiruan LVQ.....	92
4.3.1.	Skenario Pengujian	92
4.4.	Pembahasan Skenario Pengujian	94
4.4.1.	Pembahasan Skenario 1	94
4.4.2.	Pembahasan Skenario 2	96
4.4.3.	Pembahasan Skenario 3	101
4.5.	Evaluasi Hasil Skenario Pengujian	106
BAB V		108
PENUTUP		108
5.1.	Kesimpulan	108
5.2.	Saran.....	108
DAFTAR PUSTAKA.....		110
LAMPIRAN		112

DAFTAR GAMBAR

Gambar 2. 1. Struktur <i>Neuron</i> pada JST	8
Gambar 2. 2. Arsitektur Jaringan <i>Learning Vector Quantization (LVQ)</i>	10
Gambar 2. 3. Arsitektur Jaringan <i>Learning Vector Quantization (LVQ)</i>	12
Gambar 2. 4. Model <i>Waterfall</i>	19
Gambar 2. 5. Contoh <i>Conceptual Data Model (CDM)</i>	24
Gambar 2. 6. Contoh <i>Physical Data Model (PDM)</i>	24
Gambar 3. 1. Arsitektur Sistem Pendekripsi Diabetes Melitus Menggunakan Jaringan Syaraf Tiruan LVQ.....	30
Gambar 3. 2. Tahapan Alur Pengolahan Data Deteksi Diabetes Melitus	30
Gambar 3. 3. <i>Flow Garis Besar Penyelesaian Masalah</i>	Error! Bookmark not defined.
Gambar 3. 4 K-Fold Cross Validation pada dataset.....	33

DAFTAR TABEL

Tabel 2. 1. Perkembangan Penelitian dalam Prediksi Diabetes Melitus	5
Tabel 2. 2. Data Pelatihan Kasus <i>Learning Vector Quantization</i>	12
Tabel 2. 3. Bobot Awal Data Pelatihan	12
Tabel 2. 4. Data Pelatihan	12
Tabel 2. 5 Tabel <i>Confussion Matrix</i> dengan 2 Kelas	17
Tabel 2. 6. Tabel SRS.....	19
Tabel 2. 7. Tabel Notasi Pemodelan Data	22
Tabel 2. 8. Tabel Notasi Pemodelan Fungsional.....	23
Tabel 2. 9. Format Pendefinisian Rencana Pengujian.....	26
Tabel 2. 10. Format Pendefinisian Hasil Pengujian	27
Tabel 3. 1 Rangkuman Data Penelitian.....	29
Tabel 3. 2 Hasil Pengumpulan Data.....	31
Tabel 3. 3. Hasil Normalisasi	32
Tabel 3. 4 Pembagian data latih dan data uji.....	33
Tabel 3. 5 Tabel Data Pelatihan (telah dinormalisasi)	37
Tabel 3. 6 Tabel Data Pengujian (telah dinormalisasi)	40
Tabel 3. 7. Tabel <i>Confussion Matrix</i> Dengan Dua Kelas.....	42
Tabel 3. 8 Tabel <i>Confusion Matrix</i> Menghitung Sensitivitas dan Spesifisitas	42

DAFTAR LAMPIRAN

Lampiran 1. Data Hasil Rekam Medis	113
Lampiran 2. Implementasi Fungsi.....	117
Lampiran 3. Tabel Rencana Pengujian	132
Lampiran 4. Deskripsi dan Hasil Uji Pengujian Fungsional Sistem	134

BAB I

PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, tujuan dan manfaat, ruang lingkup, dan sistematika penulisan tugas akhir mengenai pendekripsi diabetes melitus dengan menggunakan algoritma *Learning Vector Quantization* (LVQ).

1.1. Latar Belakang

Degenerasi merupakan penurunan fungsi jaringan sebagai akibat dari perubahan-perubahan jaringan itu sendiri (degenerasi murni), ataupun akibat dari endapan-endapan bahan lain pada jaringan tersebut (infiltrasi). Salah satu penyakit degeneratif dengan proporsi tertinggi di Indonesia dan merupakan penyebab kematian tertinggi keenam di Indonesia adalah diabetes melitus (DM). DM merupakan kumpulan gejala yang timbul pada seseorang akibat kadar gula yang tinggi (Waspadji, 2007).

Diabetes Melitus dapat menimbulkan berbagai dampak negatif. Diabetes merupakan faktor resiko dari penyakit kardiovaskuler (Hill, 2011). Penderita diabetes beresiko mengalami *coronary artery disease* sebanyak 3,2 kali lebih besar dibandingkan non-penderita, resiko mengalami stroke sebanyak 2,9 kali lebih besar, dan resiko mengalami penyakit terkait jantung sebanyak 1,9 kali lebih besar dari non-penderita. Resiko ini meningkat pada penderita diabetes melitus yang berusia 35-64 tahun (Ariza, 2010). Selain penyakit kardiovaskuler, DM juga merupakan salah satu penyebab utama penyakit ginjal dan kebutaan pada usia di bawah 65 tahun, dan juga amputasi (Hill, 2011). Selain itu, diabetes juga menjadi penyebab terjadinya amputasi (bukan disebabkan oleh trauma), disabilitas, hingga kematian (Praet, 2009).

Diabetes melitus mempunyai beberapa indikator sebagai penilaian untuk menentukan apakah seseorang terkena diabetes melitus atau tidak, yaitu umur, jenis kelamin, kadar glukosa darah, kadar glukosa urin, kadar aseton urin, kadar kolesterol, dan kadar trigliserida. Indikator utamanya adalah kadar glukosa darah yaitu seseorang dapat dikatakan terkena diabetes melitus jika kadar glukosa darah ≥ 7 mmol/L pada saat puasa atau $\geq 11,1$ mmol/L saat 2 jam setelah puasa.

Saat ini penanganan atau diagnosis penyakit diabetes melitus hanya dapat dilakukan secara manual langsung oleh dokter atau tenaga medis, yang pada kenyataannya pasien harus mengeluarkan biaya dan usaha untuk datang ke pusat pelayanan kesehatan seperti puskesmas atau rumah sakit dengan prosedur pemeriksaan yang rumit sehingga banyak orang yang malas

untuk memeriksakan dirinya sebagai tindakan pencegahan dini dari penyakit diabetes melitus. Selain itu, karena jumlah tenaga medis yang tersedia juga terbatas, menyebabkan proses pemeriksaan akan memakan waktu yang cukup lama. Padahal pada zaman yang serba modern ini masyarakat menuntut untuk memiliki kehidupan yang serba praktis.

Teknologi semakin berkembang pesat di era ini, memudahkan manusia dalam mengambil keputusan ataupun mengidentifikasi suatu hal. Komputer dapat bekerja meniru cara kerja otak manusia sehingga dapat menimbang dan mengambil keputusan dengan menggunakan metode jaringan syaraf tiruan. Jaringan syaraf tiruan adalah sistem komputasi dimana arsitektur dan operasi diilhami dari pengetahuan tentang sel syaraf biologis dalam otak, salah satu representasi buatan dari otak manusia yang selalu mencoba menstimulasi proses pembelajaran pada otak manusia tersebut. Jaringan syaraf tiruan memiliki kemampuan untuk belajar, selain itu juga mampu menghasilkan aturan atau operasi dari beberapa contoh atau *input* yang dimasukkan, dan membuat prediksi tentang kemungkinan output yang akan muncul (Hermawan, 2006). Dalam perkembangannya, telah ada beberapa penelitian mengenai deteksi dini diabetes melitus. Dari beberapa penelitian tersebut menunjukkan bahwa menggunakan metode pada jaringan syaraf tiruan menghasilkan tingkat akurasi yang lebih baik dibandingkan dengan metode lainnya. Penelitian yang dilakukan oleh Fajar Rohman Hariri (2013) menunjukkan penggunaan metode jaringan syaraf tiruan menghasilkan tingkat akurasi yang lebih baik dengan akurasi sebesar 86% dibandingkan dengan metode lain, salah satunya adalah data mining menggunakan *decision tree C4.5* (Andriani, 2013) dengan akurasi sebesar 73,33%

Salah satu metode dalam jaringan syaraf tiruan adalah metode LVQ. Jenis ini telah banyak diterapkan dalam sistem pembelajaran untuk identifikasi. Hal ini disebabkan karena konsepnya yang sederhana dan mudah dipahami, komputasinya yang efisien, tingkat *error* yang rendah, dan memiliki kemampuan identifikasi dengan tingkat keakuratan yang lebih baik dari metode-metode lainnya. Sudah banyak penelitian yang menggunakan LVQ terhadap beberapa kasus mengenai klasifikasi dibidang medis diantaranya penerapan LVQ untuk klasifikasi status gizi anak (Budianita & Widodo, 2013) dengan tingkat akurasi sebesar 88%, klasifikasi stroke berdasarkan kelainan patologis (Arifianto, Sarosa, & Setyawati, 2014) dengan tingkat akurasi sebesar 96%, dan pengenalan jenis penyakit THT (Sela & Hartati, 2015) dengan tingkat akurasi sebesar 94%. Selain itu, berdasarkan suatu penelitian yang telah dilakukan dalam membandingkan metode jaringan syaraf tiruan LVQ dan backpropagation, didapatkan bahwa LVQ memberikan kinerja dan tingkat akurasi yang lebih baik (Nurkhozin, Irawan, & Mukhlis, 2011).

Dari beberapa hasil penelitian sebelumnya menunjukkan bahwa LVQ mampu memberikan kinerja serta tingkat akurasi yang baik sehingga penulis tertarik untuk melakukan deteksi diabetes melitus menggunakan metode LVQ. Dari permasalahan tersebut, maka tugas akhir yang dilakukan adalah menerapkan metode LVQ dalam mendeteksi penyakit diabetes melitus.

1.2. Rumusan Masalah

Berdasarkan permasalahan yang telah disampaikan dalam latar belakang sebelumnya, perumusan masalah tugas akhir ini bagaimana membangun sistem yang dapat digunakan untuk mendeteksi diabetes melitus dengan menggunakan algoritma LVQ.

1.3. Tujuan dan Manfaat

Tujuan dari tugas akhir ini adalah untuk menghasilkan sistem yang dapat digunakan untuk deteksi penyakit diabetes melitus dengan menggunakan algoritma LVQ. Manfaat yang diharapkan dari penelitian ini adalah:

1. Bagi Mahasiswa
 - a. Memenuhi persyaratan menyelesaikan pendidikan di Universitas Diponegoro Semarang
 - b. Mengimplementasikan ilmu yang diperoleh selama perkuliahan dalam merancang implementasi algoritma LVQ pada sistem pendekripsi penyakit diabetes melitus.
2. Bagi Instansi
 - a. Hasil penelitian dapat digunakan untuk mempermudah serta mempercepat proses dalam mendiagnosa penyakit diabetes melitus berdasarkan kriteria yang telah ditentukan.
3. Bagi Masyarakat
 - a. Hasil penelitian dapat digunakan dalam mempermudah serta mempercepat proses pemeriksaan, sehingga masyarakat mau untuk melakukan pemeriksaan dini.

1.4. Ruang Lingkup

Dalam penyusunan penelitian ini diberikan ruang lingkup yang jelas agar pembahasan lebih terarah dan tidak menyimpang dari tujuan penulisan. Ruang lingkup pada tugas akhir ini adalah sebagai berikut:

1. Data untuk studi kasus diperoleh dari RS Pusat Pertamina, Jakarta.
2. Pengembangan perangkat lunak menggunakan model *waterfall*.
3. Perangkat lunak diimplementasikan berbasis web dengan menggunakan bahasa pemrograman PHP dan DBMS MySQL.
4. Pengujian menggunakan metode *Black Box*.
5. Menggunakan metode LVQ untuk deteksi penyakit diabetes melitus.
6. *Input* sistem berupa sebelas data hasil laboratorium melalui *form* isian.
7. *Output* yang dihasilkan merupakan klasifikasi berupa dua kelas yaitu ya atau tidak diabetes melitus.

1.5. Sistematika Penulisan

Sistematika yang digunakan dalam tugas akhir ini terbagi menjadi beberapa pokok bahasan, yaitu:

BAB I	PENDAHULUAN
	Bab ini memberikan gambaran mengenai latar belakang, rumusan masalah, tujuan dan manfaat, ruang lingkup dan sistematika penulisan dalam pembuatan tugas akhir.
BAB II	TINJAUAN PUSTAKA
	Bab ini menyajikan dasar teori yang digunakan untuk membangun sistem serta teori-teori lain yang mendukung pengembangannya.
BAB III	METODOLOGI PENELITIAN
	Bab ini menyajikan garis besar penyelesaian masalah tugas akhir, beserta analisis dan perancangan sistem.
BAB IV	IMPLEMENTASI DAN HASIL
	Bab ini membahas tentang implementasi sistem, hasil pengujian dan hasil analisa metode yang digunakan..
BAB V	PENUTUP
	Bab ini berisi kesimpulan dari penelitian dan uraian yang telah dibahas pada bab sebelumnya dan saran untuk pengembangan sistem lebih lanjut.

BAB II

TINJAUAN PUSTAKA

Bab ini menjelaskan teori-teori yang digunakan untuk merancang sistem yang akan dibangun serta teori-teori lain yang mendukung pengembangannya.

2.1. Perkembangan Penelitian dalam Deteksi Diabetes Melitus

Penelitian terhadap diabetes melitus telah dilakukan oleh beberapa peneliti, seperti terlihat pada Tabel 2.1.

Tabel 2. 1. Perkembangan Penelitian dalam Deteksi Diabetes Melitus

No	Penelitian	Metode	Kelas	Bahasa
				Pemrograman
1	(Andriani, 2013)	<i>Decision Tree</i> <i>C4.5.</i>	- Diabetes melitus - Bukan diabetes melitus	Visual Basic
2	(Nugraha & Nurcahyanti, 2013)	<i>Naïve Bayes</i>	- Diabetes melitus - Bukan diabetes melitus	<i>Page Hypertext Preprocessors</i> (PHP)
3	(Suwarno & Abdillah, 2016)	<i>Bayesian Regularization</i> <i>Backpropagation</i>	- Diabetes melitus - Bukan diabetes melitus	Matlab
4	(Pangaribuan, 2016)	<i>Extreme Learning Machine</i>	- Diabetes melitus - Bukan diabetes melitus	Matlab
5	(Usulan, 2017)	<i>Learning Vector Quantization</i>	- Diabetes melitus - Bukan diabetes melitus	<i>Page Hypertext Preprocessors</i> (PHP)

2.2. Diabetes Melitus

Diabetes melitus merupakan penyakit kronis dengan metabolisme yang tidak teratur (Hospital Authority, 2016).

Ketika kita mengkonsumsi karbohidrat (termasuk gula dan pati, dll), bahan-bahan tersebut akan dihancurkan dan menjadi glukosa lalu diserap oleh usus kecil ke dalam sistem peredaran darah. Kemudian pankreas mengeluarkan insulin, yang membantu glukosa masuk ke dalam sel untuk digunakan oleh tubuh. Kadar glukosa akan meningkat bila sekresi insulin tidak mencukupi atau tubuh tidak bisa menggunakan insulin yang dihasilkan. Keadaan ini mengakibatkan gangguan metabolisme lemak dan protein, serta penghancuran berbagai macam sistem tubuh dan organ dalam jangka waktu lama. (Hospital Authority, 2016).

Penderita diabetes melitus memiliki gejala-gejala awal. Gejala awalnya adalah sebagai berikut (Hospital Authority, 2016).

1. Sering merasa haus
2. Sering buang air kecil
3. Sering merasa lapar
4. Penurunan berat badan
5. Kelelahan
6. Penglihatan yang kabur
7. Tingkat penyembuhan luka yang lambat

Berkembangnya penyakit diabetes melitus di Indonesia berkaitan dengan gaya hidup yang kurang sehat, kondisi sosial ekonomi yang buruk, serta kurang memadainya fasilitas kesehatan masyarakat yang ada sehingga memudahkan berkembangnya penyakit DM.

2.2.1. Faktor Risiko Diabetes Melitus

Beberapa faktor yang memiliki tingkat risiko yang lebih tinggi untuk menderita diabetes melitus antara lain (Hospital Authority, 2016):

1. Riwayat diabetes melitus pada anggota keluarga terdekat.
2. Penderita hipertensi atau hiperlipidemia (kadar lemak dalam darah yang sangat tinggi).
3. Wanita yang memiliki riwayat diabetes melitus gestasional (jenis diabetes melitus yang terjadi hanya selama kehamilan) atau melahirkan bayi yang mengalami kelebihan berat badan (bobot 4 kg ke atas).
4. Obesitas (dengan IMT lebih dari 23).
5. Berada di usia paruh baya (usia 45 tahun ke atas), dll.

2.2.2. Diagnosa Diabetes Melitus

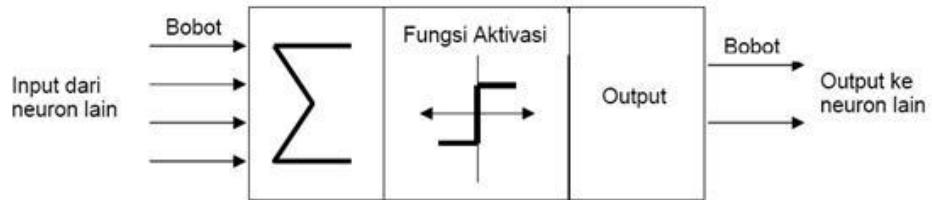
Ada beberapa tahapan tes yang dilakukan dalam mendiagnosa penyakit diabetes melitus, yaitu (Hospital Authority, 2016):

1. Tes glukosa darah secara acak, yaitu dengan mengambil darah untuk memeriksa kadar glukosa tanpa puasa. Kadar toleransi untuk glukosa darah normal sebesar 11.1 mmol/L atau sebesar 110 Mg/dl.
2. Tes glukosa darah puasa, melakukan puasa setelah tengah malam dan pengambilan darah keesokan harinya untuk memeriksa kadar glukosa. Kadar toleransi untuk glukosa darah puasa adalah sebesar 7 mmol/L atau sebesar 70 Mg/dl.
3. Tes toleransi glukosa oral, setelah tes glukosa darah puasa dilakukan, pasien diberikan glukosa sebanyak 75 gram dan dilakukan pengambilan darah setelah jangka waktu 2 jam, demi keperluan pengamatan perubahan kadar glukosa darah.

2.3. Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) adalah sistem terkomputasi dimana arsitektur dan operasinya diilhami dari pengetahuan tentang sel saraf biologis di dalam otak. JST merupakan salah satu pembelajaran pada otak manusia yang selalu mencoba menstimulasi proses pembelajaran pada otak manusia tersebut. Model jaringan syaraf ditunjukkan dengan kemampuannya dalam emulasi, analisis, prediksi dan asosiasi. Kemampuan yang dimiliki jaringan syaraf tiruan dapat digunakan untuk belajar dan menghasilkan aturan atau operasi dari beberapa contoh atau *input* yang dimasukkan dan membuat prediksi tentang kemungkinan *output* yang akan muncul atau menyimpan karakteristik input yang diberikan kepada jaringan syaraf tiruan. (Hermawan, 2006).

Setiap pengolahan elemen membuat perhitungan berdasarkan pada jumlah masukkan (*input*). Sebuah kelompok pengolahan elemen disebut layer atau lapisan dalam jaringan. Lapisan pertama adalah *input* dan yang terakhir adalah *output*. Lapisan di antara lapisan *input* dan *output* disebut dengan *hidden layer*. JST merupakan suatu bentuk arsitektur yang terdistribusi paralel dengan sejumlah besar titik dan hubungan antar titik tersebut. Tiap titik yang berhubungan mempunyai harga yang diasosiasikan sebagai bobot. Setiap titik mempunyai nilai yang diasosiasikan sebagai nilai aktivasi titik. Struktur neuron pada jaringan syaraf tiruan diilustrasikan pada Gambar 2.1.



Gambar 2. 1. Struktur *Neuron* pada JST

2.3.1. Arsitektur Jaringan Syaraf Tiruan

Ada beberapa arsitektur pada jaringan syaraf tiruan yang sering digunakan dalam berbagai aplikasi (Puspitaningrum, 2006), antara lain:

1. Jaringan Layar Tunggal

Merupakan jaringan dengan lapisan tunggal yang terdiri dari 1 *layer input* dan 1 *layer output*. Setiap *neuron/unit* yang terdapat di dalam lapisan/*layer input* selalu terhubung dengan setiap *neural* yang terdapat pada *layer output*. Jaringan ini hanya menerima *input* kemudian secara langsung akan mengolahnya menjadi *output* tanpa harus melalui lapisan tersembunyi. Contoh algoritma jaringan syaraf tiruan yang menggunakan metode ini yaitu : ADALINE, Hopfield, Perceptron.

2. Jaringan Layar Jamak

Merupakan jaringan yang memiliki ciri khas tertentu yaitu memiliki 3 jenis *layer* yakni *layer input*, *layer output* dan *layer* tersembunyi (*hidden layer*). Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang kompleks dibandingkan jaringan dengan lapisan tunggal. Namun proses pelatihan sering membutuhkan waktu yang cenderung lama. Contoh algoritma jaringan syaraf tiruan yang menggunakan metode ini yaitu : MADALINE, backpropagation, neocognitron.

3. Jaringan dengan lapisan kompetitif

Pada jaringan ini sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif. Contoh algoritma yang menggunakan metode ini adalah : LVQ.

2.3.2. Pelatihan Jaringan Syaraf Tiruan

Berdasarkan cara pembelajaran atau pelatihannya, jaringan syaraf tiruan dikelompokkan menjadi 3 (Puspitaningrum, D., 2006), yaitu:

1. *Supervised Learning* (Pembelajaran Terawasi)

Pada jenis pembelajaran ini, setiap pola yang diberikan kedalam jaringan syaraf tiruan telah diketahui *outputnya*. Selisih antara pola *output aktual* (*output* yang dihasilkan) dengan pola *output* yang dikehendaki (*output target*) yang disebut *error* digunakan untuk mengoreksi bobot jaringan syaraf tiruan sehingga jaringan syaraf tiruan mampu menghasilkan output sedekat mungkin dengan pola target yang telah diketahui oleh jaringan syaraf tiruan. Contoh algoritma jaringan syaraf tiruan yang menggunakan metode ini adalah : Hebbian, Perceptron, ADALINE,Boltzman, Hopfield, *Backpropagation*.

2. *Unsupervised Learning* (Pembelajaran Tidak-Terawasi)

Pada jenis pembelajaran ini tidak memerlukan target *output*. Pada metode ini tidak dapat ditentukan hasil seperti apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu range tertentu tergantung pada nilai input yang diberikan. Tujuan pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pembelajaran ini biasanya sangat cocok untuk klasifikasi pola. Contoh algoritma jaringan syaraf tiruan yang menggunakan metode ini adalah : Competitive, Hebbian, Kohonen, LVQ, Neocognitron.

3. *Hybrid Learning* (Pembelajaran Hibrida)

Merupakan kombinasi dari metode pembelajaran *supervised learning* dan *unsupervised learning*, sebagian dari bobot-bobotnya ditentukan melalui pembelajaran terawasi dan sebagian lainnya melalui pembelajaran tak terawasi. Contoh algoritma jaringan syaraf tiruan yang menggunakan metode ini adalah: algoritma RBF. Metode algoritma yang baik dan sesuai dalam melakukan pengenalan pola-pola gambar adalah algoritma *Backpropagation* dan Perceptron. Untuk mengenali teks berdasarkan tipe font akan digunakan algoritma *Backpropagation*.

2.4. Algoritma *Learning Vector Quantization*

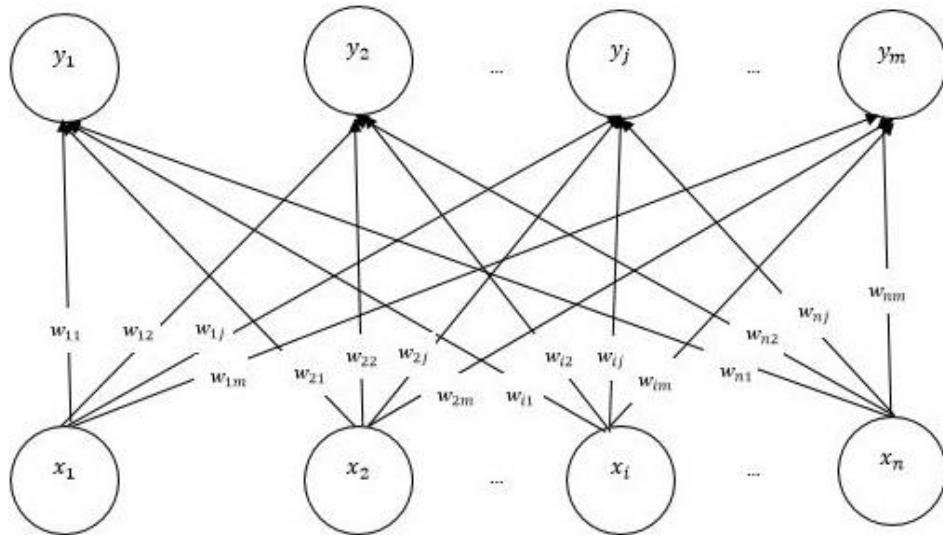
Learning Vector Quantization merupakan suatu metode jaringan syaraf tiruan untuk melakukan pembelajaran pada lapisan kompetitif. Lapisan kompetitif akan secara otomatis mempelajarinya untuk mengklasifikasikan vektor-vektor input. Kelas-kelas yang diperoleh sebagai hasil dari lapisan kompetitif ini hanya bergantung pada jarak antara vektor-vektor

input. Jika dua vektor input mendekati sama, maka lapisan kompetitif akan meletakan vektor input tersebut ke dalam kelas yang sama (Ranadhi, Indarto, dan Hidayat, 2006).

2.4.1. Arsitektur Jaringan *Learning Vector Quantization*

Arsitektur jaringan LVQ hanya terdapat dua *layer* saja yaitu *layer input* dan *layer output*. Setiap unit dari *layer input* pada jaringan LVQ selalu terhubung dengan setiap unit yang berada pada *layer output*. Jaringan LVQ terdiri dari banyak lapisan/*layer* (multilayer network), yaitu *layer input* yang terdiri dari 1 hingga n unit *input* dan *layer output* yang terdiri dari 1 hingga m unit *output*.

Arsitektur jaringan LVQ secara umum dapat dilihat pada Gambar 2.2.



Gambar 2. 2. Arsitektur Jaringan *Learning Vector Quantization (LVQ)* (Fausset, 1994)

Penjelasan :

x : vektor input, $x = [x_1, x_2, \dots, x_i, \dots, x_n]$

y : vektor output, $y = [y_1, y_2, \dots, y_j, \dots, y_m]$

$$w : \text{bobot}, w = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \dots & w_{nm} \end{bmatrix}$$

2.4.2. Pelatihan Standar *Learning Vector Quantization*

Secara rinci algoritma pelatihan jaringan *LVQ* dapat diuraikan sebagai berikut (T.Sutoyo, 2011) :

1. Menetapkan bobot awal (w) berukuran $t \times n$, maksimum *epoch* (*MaxEpoch*), *error minimum* (*eps*) dan *learning rate* (α) yang bernilai $0 < \alpha < 1$.
2. Memasukkan input x dan target (T)

3. Menetapkan kondisi awal $epoch = 0$
 4. Dikerjakan jika ($epoch \leq MaxEpoch$) atau ($\alpha < eps$)
 - a. $Epoch = epoch + 1$
 - b. Untuk $i = 1$ sampai n
- Untuk $k = 1$ sampai t

$$\text{Menghitung nilai } S_{ik} \sqrt{\sum_{j=1}^m (x_{ij} - w_{kj})^2} \dots \dots \dots \quad (2.1)$$

- i. Menentukan nilai S_{ik} terkecil dan simpan k
- ii. Membandingkan nilai k dengan T_{1k}

a. Jika $k = T_{1k}$ maka,

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \alpha (x_i - w_{kj}(\text{lama})) \dots \dots \dots \quad (2.2)$$

b. Jika $k \neq T_{1k}$ maka,

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) - \alpha (x_i - w_{kj}(\text{lama})) \dots \dots \dots \quad (2.3)$$

5. Mengurangi nilai α

$$\alpha = \alpha - (0.1 * \alpha) \dots \dots \dots \quad (2.4)$$

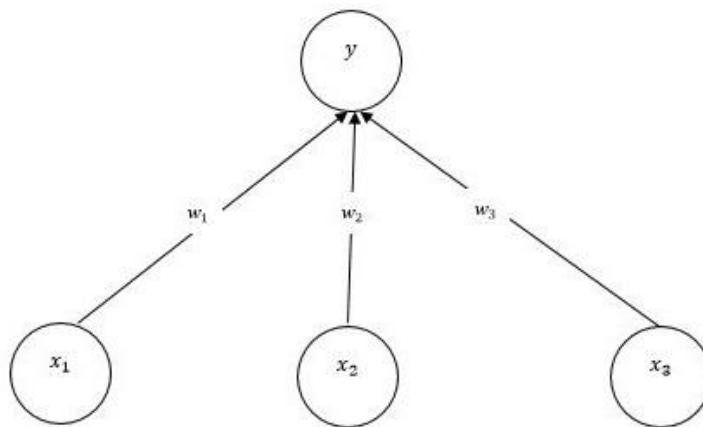
Penjelasan :

- i : indeks untuk jumlah data pelatihan
- j : indeks untuk variabel penentu kelas output
- k : indeks untuk kelas output
- m : jumlah variabel penentu kelas output
- n : jumlah data pelatihan
- t : jumlah kelas output
- x : vektor pelatihan dalam hal ini berukuran $m \times n$
- T : Target pelatihan dalam hal ini berukuran $m \times 1$
- w_k : Vektor bobot kelas ke- k
- S_{ik} : Nilai Euclidean Distance vektor pelatihan ke- i dan kelas ke- k

Perhitungan kasus metode *Learning Vector Quantization* dapat dilihat pada contoh berikut. Diketahui 6 buah data yang terbagi menjadi dua buah kelompok, yaitu kelompok data yang menjadi bobot awal data pelatihan serta kelompok data pelatihan pada Tabel 2.2 dengan arsitektur jaringannya dapat dilihat pada Gambar 2.3.

Tabel 2. 2. Data Pelatihan Kasus *Learning Vector Quantization*

No	Input Vektor	Kelas (T)
1	001	1
2	111	2
3	011	1
4	010	2
5	110	2
6	101	1



Gambar 2. 3. Arsitektur Jaringan *Learning Vector Quantization (LVQ)*

- Menetapkan *learning rate* (α) = 0.05, maksimum *epoch* = 2, penurunan α = 0,1. α (lama) dan dua input pertama dijadikan bobot awal (w) sebagai penentu kelas output (k), dapat dilihat pada tabel 2.3.

Tabel 2. 3. Bobot Awal Data Pelatihan

No	Input Vektor	Target (T)
1	001	1
2	111	2

- Memasukkan input x dan target (T), dapat dilihat pada tabel 2.4.

Tabel 2. 4. Data Pelatihan

No	Input Vektor	Target (T)
1	011	1
2	010	2
3	110	2
4	101	1

- Menetapkan kondisi awal *epoch* = 0
- Dikerjakan jika ($epoch \leq MaxEpoch$)

- a. $Epoch = epoch + 1$
- b. Untuk $i = 1$ sampai 4

Data ke $i = 1$, $x_1 = 011$.

- i. Untuk $k = 1$ sampai 2

$$\text{Menghitung nilai } S_{ik} = \sqrt{\sum_{j=1}^m (x_{ij} - w_{kj})^2}$$

Kelas $k = 1$

$$S_{11} = \sqrt{(0-0)^2 + (1-0)^2 + (1-1)^2} = 1$$

Kelas $k = 2$

$$S_{12} = \sqrt{(0-1)^2 + (1-1)^2 + (1-1)^2} = 1$$

- ii. Menentukan nilai S_{ik} terkecil dan simpan k

Nilai terkecil pada $k = 1$

- iii. Membandingkan nilai k dengan T_{Ik}

$k = 1$ dan $T_{1k} = 1$, jadi $k = T_{1k}$ maka :

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \alpha (x_i - w_{kj}(\text{lama}))$$

$$w_{11}(\text{baru}) = 0 + 0.05 * (0-0) = 0$$

$$w_{12}(\text{baru}) = 0 + 0.05 * (1-0) = 0.05$$

$$w_{13}(\text{baru}) = 1 + 0.05 * (1-1) = 1$$

Jadi $w_1 = (0 \ 0.05 \ 1)$; $w_2 = (1 \ 1 \ 1)$

Data ke $i = 2$, $x_2 = 010$.

- i. Untuk $k = 1$ sampai 2

$$\text{Menghitung nilai } S_{ik} = \sqrt{\sum_{j=1}^m (x_{ij} - w_{kj})^2}$$

Kelas $k = 1$

$$S_{21} = \sqrt{(0-0)^2 + (1-0.05)^2 + (0-1)^2} = 1.3793$$

Kelas $k = 2$

$$S_{22} = \sqrt{(0-1)^2 + (1-1)^2 + (0-1)^2} = 1.4142$$

i. Menentukan nilai S_{ik} terkecil dan simpan k

Nilai terkecil pada $k = 1$

ii. Membandingkan nilai k dengan T_{Ik}

$k = 1$ dan $T_{Ik} = 2$, jadi $k \neq T_{Ik}$ maka :

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) - \alpha (x_i - w_{kj}(\text{lama}))$$

$$w_{11}(\text{baru}) = 0 - 0.05 * (0-0) = 0$$

$$w_{12}(\text{baru}) = 0 - 0.05 * (1-0.05) = 0.0025$$

$$w_{13}(\text{baru}) = 1 - 0.05 * (0-1) = 1.05$$

Jadi $w_1 = (0 \ 0.0025 \ 1.05)$; $w_2 = (1 \ 1 \ 1)$

Data ke $i = 3$, $x_3 = 110$.

i. Untuk $k = 1$ sampai 2

$$\text{Menghitung nilai } S_{ik} = \sqrt{\sum_{j=1}^m (x_{ij} - w_{kj})^2}$$

Kelas $k = 1$

$$S_{31} = \sqrt{(1-0)^2 + (1-0.0025)^2 + (0-1.05)^2} = 1.7599$$

Kelas $k = 2$

$$S_{32} = \sqrt{(1-1)^2 + (1-1)^2 + (0-1)^2} = 1$$

ii. Menentukan nilai S_{ik} terkecil dan simpan k

Nilai terkecil pada $k = 2$

iii. Membandingkan nilai k dengan T_{Ik}

$k = 2$ dan $T_{Ik} = 2$, jadi $k = T_{Ik}$ maka :

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \alpha (x_i - w_{kj}(\text{lama}))$$

$$w_{11}(\text{baru}) = 1 + 0.05 * (1-1) = 1$$

$$w_{12}(\text{baru}) = 1 + 0.05 * (1-1) = 1$$

$$w_{13}(\text{baru}) = 1 + 0.05 * (0-1) = 0.95$$

Jadi $w_1 = (0 \ 0.0025 \ 1.05)$; $w_2 = (1 \ 1 \ 0.95)$

Data ke $i = 4$, $x_4 = 101$.

i. Untuk $k = 1$ sampai 2

$$\text{Menghitung nilai } S_{ik} = \sqrt{\sum_{j=1}^m (x_{ij} - w_{kj})^2}$$

Kelas $k = 1$

$$S_{41} = \sqrt{(1-0)^2 + (0-0.0025)^2 + (1-1.05)^2} = 1.0012$$

Kelas $k = 2$

$$S_{42} = \sqrt{(1-1)^2 + (0-1)^2 + (1-0.95)^2} = 1.0010$$

ii. Menentukan nilai S_{ik} terkecil dan simpan k

Nilai terkecil pada $k = 2$

iii. Membandingkan nilai k dengan T_{Ik}

$k = 2$ dan $T_{Ik} = 1$, jadi $k \neq T_{Ik}$ maka :

$$w_{kj} (\text{baru}) = w_{kj} (\text{lama}) + \alpha (x_i - w_{kj} (\text{lama}))$$

$$w_{11}(\text{baru}) = 1 - 0.05 * (1-1) = 1$$

$$w_{12}(\text{baru}) = 1 - 0.05 * (0-1) = 0.05$$

$$w_{13}(\text{baru}) = 0.95 - 0.05 * (1-0.95) = 0.9475$$

Jadi $w_1 = (0 \ 0.0025 \ 1.05)$; $w_2 = (1 \ 0.05 \ 0.9475)$

Proses diteruskan sampai *epoch* ke-2, dengan menggunakan cara yang sama. Berdasarkan aturan berhenti $\text{epoch} \leq \text{MaxEpoch}$, karena jumlah *epoch* sudah sama dengan *max epoch*, yaitu 2, maka proses perhitungan berhenti hanya sampai pada *epoch* ke-2. Setelah mencapai *epoch* yang ke-2 diperoleh bobot akhir berikut.

$$w_1 = (0.0000 \ 0.0030 \ 1.0540)$$

$$w_2 = (1.0000 \ 0.0550 \ 0.9420)$$

Misal data yang akan diuji (010)

Kelas $k = 1$

$$S_1 = \sqrt{(0-0)^2 + (1-0.0030)^2 + (0-1.0540)^2} = 1.4508$$

Kelas $k = 2$

$$S_2 = \sqrt{(0-1)^2 + (1-0.0550)^2 + (0-0.9420)^2} = 1.6674$$

Diketahui nilai minimum ada pada $k = 1$ maka data (010) masuk ke kelas 1

2.5. Evaluasi Kinerja *Classifier*

Evaluasi kinerja *classifier* digunakan untuk mengevaluasi tingkat akurasi pada proses klasifikasi. Evaluasi kinerja *classifier* yang digunakan pada penelitian ini adalah *K-Fold Cross Validation* dan *Confusion Matrix*.

2.5.1. *K-Fold Cross Validation*

K-Fold Cross Validation merupakan salah satu metode yang umum digunakan untuk mengevaluasi kinerja *classifier*. Metode *K-Fold Cross Validation* dilakukan dengan membagi dataset secara acak menjadi k himpunan bagian (subset) (Kohavi, 1995). *K-Fold Cross Validation* melakukan iterasi sebanyak k kali untuk data pelatihan dan data pengujian. Metode ini berguna untuk memvalidasi keakuratan sebuah prediksi atau klasifikasi terhadap suatu data yang belum muncul dalam dataset. Dataset dibagi menjadi k subset secara acak yang masing-masing subset memiliki jumlah *instance* dan perbandingan jumlah kelas yang sama. Pembagian data ini digunakan pada proses iterasi klasifikasi. Iterasi dilakukan sesuai nilai k . Setiap iterasi satu subset digunakan untuk menguji sedangkan subset-subset lainnya digunakan untuk pelatihan.

Kelebihan dari metode ini adalah tidak adanya masalah dalam pembagian data. Setiap data akan menjadi *test set* sebanyak satu kali dan akan menjadi *training set* sebanyak $K-1$ kali. Kekurangan dari metode ini adalah algoritma pembelajaran harus dilakukan sebanyak K kali. Yang berarti menggunakan K kali waktu komputasi. Metode *K-Fold Cross Validation* pada umumnya menggunakan $k = 10$ (Fitri, 2007). Hasil uji coba metode ini dengan $k = 10$ menghasilkan rata-rata akurasi yang cukup tinggi yaitu 97% lebih baik dibandingkan metode Naïve Bayesian yang menghasilkan rata-rata akurasi 96,24% (Zamani A M, 2012).

2.5.2. *Confussion Matrix*

Evaluasi dari kinerja klasifikasi didasarkan pada banyak *test record* yang diprediksi secara benar dan secara tidak benar oleh sebuah model. *Test record* ini ditabulasikan dalam sebuah tabel yang dikenal sebagai *confussion matrix*. *Confussion matrix* berisi informasi tentang klasifikasi aktual dan prediksi yang dilakukan oleh sistem klasifikasi (Provost, 1998). Kinerja sistem tersebut umumnya dievaluasi dengan menggunakan data dalam matriks. Tabel *confussion matrix* dengan dua kelas dapat dilihat pada tabel 2.5.

Tabel 2. 5 Tabel *Confussion Matrix* dengan 2 Kelas

Actual Class	Predicted Class	
	Class = 1	Class = 0
Class = 1	M ₁₁	M ₁₀
Class = 0	M ₀₁	M ₀₀

Nilai akurasi dan *error rate* dapat dihitung dengan menggunakan persamaan 2.5 dan 2.6.

$$\text{akurasi} = \frac{\text{banyaknya prediksi yang benar}}{\text{total banyaknya prediksi}} = \frac{M_{11} + M_{00}}{M_{11} + M_{10} + M_{01} + M_{00}} \dots\dots\dots(2.5)$$

$$\text{error rate} = \frac{\text{banyaknya prediksi yang salah}}{\text{total banyaknya prediksi}} = \frac{M_{10} + M_{01}}{M_{11} + M_{10} + M_{01} + M_{00}} \dots\dots\dots(2.6)$$

Dalam kasus medis, proses evaluasi dapat menggunakan sensitivitas dan spesifisitas. Sensitivitas digunakan untuk mengukur seberapa sering tes menjadi positif pada orang-orang yang memiliki penyakit pada kenyataannya. Sedangkan spesifisitas digunakan untuk mengukur seberapa baik sebuah tes dalam mengklasifikasikan orang yang tidak sakit sebagai orang yang benar-benar tidak memiliki penyakit pada kenyataannya. Tabel *confussion matrix* untuk mengukur sensitivitas dan spesifisitas dapat dilihat pada tabel 2.6.

Tabel 2. 6. Tabel *Confussion Matrix* Pengukuran Sensitivitas dan Spesifisitas

Target	Hasil Klasifikasi	
	Kelas ke-1	Bukan kelas ke- 1
Kelas ke-1	TP	FN
Bukan kelas ke-1	FP	TN

Keterangan:

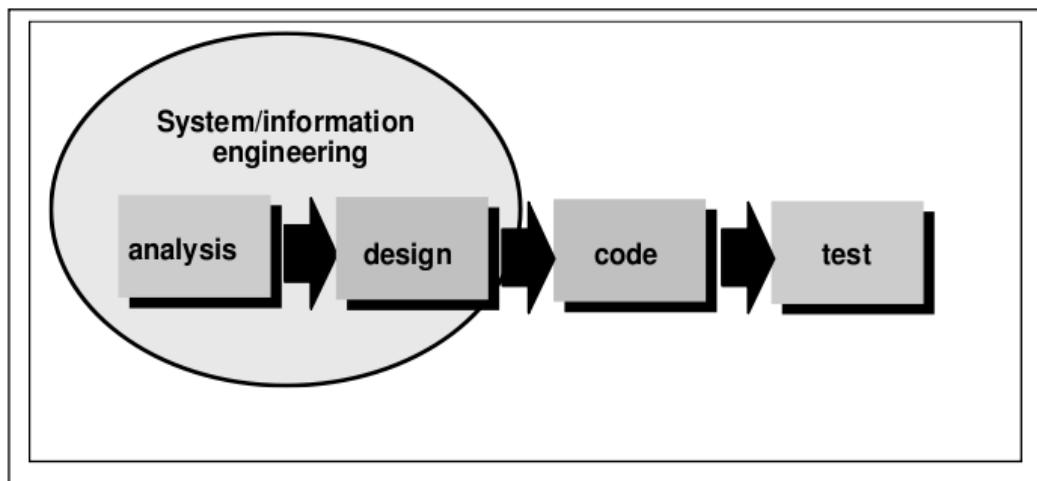
TP : *True Positive*

FP : *False Positive*

FN : *False Negative*

TN : *True Negative*

True positive menyatakan pada target masuk kelas 1 dan terdeteksi kelas 1. *False positive* menyatakan pada target bukan termasuk kelas 1, namun terdeteksi kelas 1. *False negative* menyatakan pada target termasuk kelas 1 dan terdeteksi bukan kelas 1. *True*



Gambar 2. 4. Model *Waterfall*

Tahap-tahap dalam model waterfall adalah sebagai berikut (Pressman, 2001).

a. Analisa Kebutuhan Software

Seluruh kebutuhan *software* harus bisa didapatkan dalam tahap ini, kemudian dianalisis dan didefinisikan termasuk di dalamnya kegunaan *software* yang diharapkan pengguna dan batasan *software* seperti fungsi yang dibutuhkan, perilaku, performa, dan *interface*. Tahap ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap. Tahap ini menghasilkan SRS (*Software Requirement Specification*), ERD (*Entity Relationship Diagram*) dan DFD (*Data Flow Diagram*). SRS adalah dokumen yang berisi deskripsi lengkap mengenai kemampuan *software*. SRS diperlukan karena banyak kesalahan yang timbul pada tahap *requirement* dan tidak terdeteksi sejak dini, serta untuk menghemat biaya perbaikan. ERD menggambarkan hubungan antara obyek data. Sedangkan DFD menyediakan informasi yang digunakan untuk menganalisa ruang lingkup dan sebagai dasar dari pemodelan fungsional.

Standar penulisan SRS yang digunakan adalah IEEE/ANSI 830-1998. Setiap *requirement* dalam SRS diberikan *unique identifier label*. Berikut format pendefinisan kebutuhan fungsional dalam SRS pada Tabel 2.7.

Tabel 2. 7. Tabel SRS

SRS ID	Deskripsi
SRS-XXXX-F-YY

Keterangan:

SRS : *Software Requirement Specification*

XXXX : *Nickname* sistem yang dibangun

FXX : F adalah fungsional dapat *optional* berupa NF (Non Fungsional)

YY : Nomor urut SRS

b. Desain Sistem

Desain sistem merupakan proses desain dengan menggunakan analisis kebutuhan sebagai acuannya. Proses desain akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat pengkodean. Proses ini berfokus pada perancangan struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail algoritma prosedural. Tahapan ini akan menghasilkan dokumen desain. Dokumen inilah yang akan digunakan untuk melakukan aktivitas pembuatan sistem.

c. Desain Program

Desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Tahap ini merupakan implementasi dari tahap desain yang digunakan untuk membangun sistem. Jika desain dilakukan secara detail, maka pengkodean dapat dilakukan secara mekanis. Setelah pengkodean selesai, maka akan dilakukan *testing* terhadap sistem yang telah dibuat dengan tujuan untuk mencari dan menemukan kesalahan-kesalahan pada sistem sehingga dapat diperbaiki.

d. Pengujian

Ketika proses pengkodean selesai, maka akan dilakukan pengujian (*testing*). Proses pengujian berfokus pada logika di dalam *software*, memastikan semua *statement* dan fungsi eksternal telah dites, sehingga dapat menemukan kesalahan untuk dapat diperbaiki selanjutnya sehingga didapatkan hasil yang sesuai dengan yang diinginkan.

Kelebihan model waterfall adalah lebih disiplin. Selain itu, dokumentasi selalu tersedia dalam setiap tahapan (*documentation driven*). Masalah dengan model waterfall adalah akan menyebabkan “*blocking states*”, yaitu keadaan dimana suatu tugas hanya dapat dilakukan ketika tugas sebelumnya diselesaikan. Sehingga terdapat waktu tunggu pada tugas selanjutnya untuk dapat dilakukan.

2.8. Pemodelan Analisis

Pemodelan analisis merupakan gabungan dari beberapa model yang pertama kali merepresentasikan sistem secara teknis. Terdapat dua pemodelan, yaitu pemodelan data dan pemodelan fungsional. Pemodelan analisis harus memenuhi tiga tujuan yaitu, mendeskripsikan

keinginan *customer*, sebagai dasar dalam membuat desain *software*, dan mendefinisikan kebutuhan yang dapat divalidasi ketika membangun *software*.

2.8.1. Pemodelan Data

Pemodelan data menjelaskan semua hal tentang obyek utama yang akan diproses, apa komposisi dari tiap obyek serta atributnya, dan hubungan antar obyek. Semua hal tersebut digambarkan dalam bentuk ERD (*Entity Relationship Diagram*). Keluaran yang dihasilkan adalah struktur basis data (Pressman, 2001).

ERD merupakan suatu *graph* yang menyajikan obyek data, atribut dan hubungannya yang bertujuan untuk mengetahui hubungan antar obyek data. Pada pemodelan data ada tiga bagian informasi yang digunakan, yaitu: (Pressman, 2001)

a. Obyek Data

Obyek data merupakan obyek yang mewakili sesuatu informasi yang nyata dan dapat dibedakan dari sesuatu yang lain. Obyek data harus berhubungan satu sama lainnya. Properti dari obyek data antara lain: nama, daftar atribut, dan *primary key*.

b. Relasi

Hubungan antara sejumlah obyek data yang berasal dari himpunan obyek yang data berbeda. Relasi yang terjadi di antara dua himpunan obyek data (misalnya A dan B) dalam satu basis data yaitu:

i. *One to one* (1:1).

Hubungan satu ke satu yaitu setiap *instance* pada obyek data A berhubungan paling banyak dengan satu *instance* pada obyek data B.

ii. *One to many* (1:N).

Setiap *instance* pada obyek data A dapat berhubungan dengan banyak *instance* pada obyek data B, tetapi setiap *instance* pada obyek data B dapat berhubungan dengan satu *instance* pada obyek data A.

iii. *Many to many* (M:N).

Setiap *instance* pada obyek data A dapat berhubungan dengan banyak *instance* pada obyek data B.

c. Atribut

Atribut mendefinisikan properti dari obyek data. Satu atau lebih atribut dari suatu obyek data harus menjadi penanda sebagai *primary key* yang membedakan dengan obyek data lain. Berikut ini adalah tabel notasi pemodelan data dapat dilihat pada Tabel 2.8.

Tabel 2. 8. Tabel Notasi Pemodelan Data

Komponen	Deskripsi
	Obyek data adalah sebuah obyek yang dapat dibedakan dengan obyek lain.
	Relasi menunjukkan adanya hubungan di antara sejumlah obyek data yang berbeda.
	Garis sebagai penghubung antara relasi dengan obyek data, dan obyek data dengan atribut
1 : 1 1 : N M : N	Kardinalitas merupakan angka yang menunjukkan banyaknya kemunculan suatu obyek terkait dengan kemunculan obyek lain pada relasi.

2.8.2. Pemodelan Fungsional

Seluruh fungsi yang tercakup dalam sistem ini digambarkan dalam *Data Context Diagram* (DCD) dan *Data Flow Diagram* (DFD). DCD dapat juga dikatakan sebagai DFD Level-0 atau *Context Diagram* (Pressman, 2001). Untuk DFD merupakan penjabaran lebih lanjut dari DCD. DFD berguna untuk menggambarkan fungsi-fungsi yang mentransformasikan data, serta berguna untuk menggambarkan bagaimana data ditransformasikan pada perangkat lunak.

DCD merupakan tingkatan tertinggi dalam diagram aliran data yang hanya memuat satu proses, dan menunjukkan sistem secara keseluruhan. Proses tersebut diberi nomor nol. Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran-aliran data utama menuju dan dari sistem. Diagram tersebut tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan, entitas-entitas eksternal serta aliran-aliran data menuju dan dari sistem diketahui oleh seorang analis dari wawancara dengan pengguna sebagai hasil analisis dokumen.

DFD atau dapat juga disebut dengan model sistem fundamental dan *bubble chart* adalah representasi grafik dari aliran informasi serta transformasi atau proses yang dilakukan mulai dari data dimasukkan sampai mendapatkan *output* (Pressman, 2001). DFD menggambarkan komponen-komponen sebuah sistem, aliran-aliran data di mana

komponen-komponen tersebut, asal, tujuan, dan penyimpanan dari data tersebut. DFD digunakan untuk dua hal utama, yaitu untuk membuat dokumentasi dari sistem informasi yang ada serta untuk menyusun dokumentasi untuk sistem informasi yang baru. DFD mempunyai empat komponen utama, yaitu *external entity*, *data flow*, proses, dan *data store*. Notasi yang digunakan untuk pemodelan fungsional dapat dilihat pada Tabel 2.9.

Tabel 2. 9. Tabel Notasi Pemodelan Fungsional

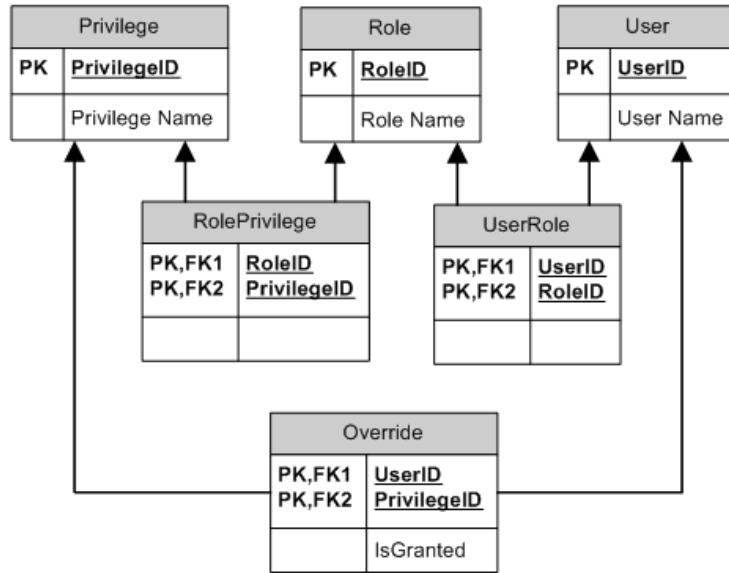
Notasi	Keterangan
	<i>External Entity</i> (Entitas Eksternal)
	<i>Data Flow</i> (Aliran Data)
	<i>Process</i> (Proses)
	<i>Data Store</i>

2.9. Perancangan Struktur Data

Struktur data merupakan sebuah representasi dari hubungan logis antara elemen individual dari data. Karena struktur informasi akan selalu mempengaruhi perancangan prosedural akhir, struktur data sama pentingnya dengan struktur program. Perancangan struktur data mentransformasikan pemodelan data yang telah dibuat pada fase analisis menjadi struktur data yang akan digunakan dalam implementasi perangkat lunak. Desain atau perancangan struktur data meliputi level pemodelan yaitu *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM).

1. *Conceptual Data Model*

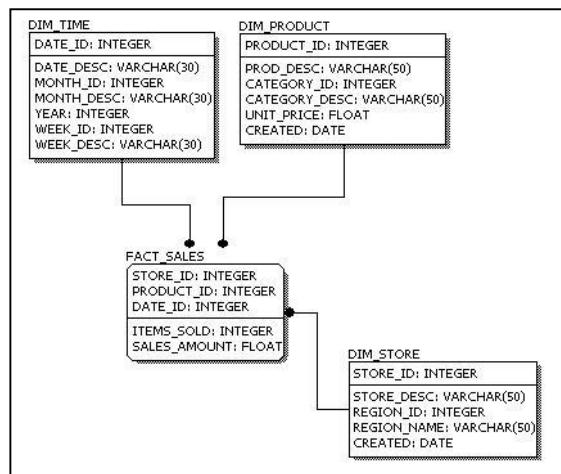
Conceptual Data Model atau CDM merupakan model data dalam bentuk konseptual yang berisi hubungan antar obye data dalam sistem yang dibangun. CDM dibuat sudah dalam bentuk tabel-tabel tanpa tipe data yang menggambarkan relasi antar tabel untuk keperluan omlementasi ke basis data. CDM merupakan hasil penjabaran lanjut dari ERD. (Rosa & M.Shalahuddin, 2013). Berikut ilustrasi contoh dari CDM yang dapat dilihat pada Gambar 2.5.



Gambar 2. 5. Contoh *Conceptual Data Model* (CDM)

2. Physical Data Model

Physical Data Model atau PDM merupakan hasil mapping dari CDM, merupakan model data yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antar data. Setiap tabel mempunyai sejumlah kolom dimana setiap kolom memiliki nama yang unik beserta tipe datanya. PDM merupakan bentuk fisik perancangan basis data yang sudah siap diimplementasikan ke dalam DBMS (Rosa & M.Shalahuddin, 2013). Berikut ilustrasi contoh PDM yang dapat dilihat pada Gambar 2.6.



Gambar 2. 6. Contoh *Physical Data Model* (PDM)

2.10. Bahasa Pemrograman PHP

PHP (PHP Hypertext Preprocessor) adalah bahasa pemrograman *scripting* sisi *server* artinya sintaks-sintaks dan perintah yang diberikan akan sepenuhnya dijalankan oleh *server*

tetapi disertakan pada halaman HTML biasa. Sistem yang dibangun oleh PHP pada umumnya akan memberikan hasil pada web browser, tetapi prosesnya secara keseluruhan dijalankan di *server*. Pada prinsipnya *server* akan bekerja apabila ada permintaan dari *client*. Dalam hal ini *client* menggunakan kode-kode PHP untuk mengirimkan permintaan ke *server*.

Kelebihan PHP dari bahasa pemrograman lain (Peranginangin, 2006)

- a. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- b. *Web Server* yang mendukung PHP dapat ditemukan di mana-mana dari mulai Apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
- c. Lebih mudah dalam pengembangan.
- d. Dalam sisi pemahamanan, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
- e. PHP adalah *open source* yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

2.11. DBMS MySQL

MySQL adalah sistem manajemen *database* SQL yang bersifat *open source* dan paling populer saat ini. Sistem database MySQL mendukung beberapa fitur seperti *multi-threaded*, *multi-user*, dan *SQL database management system* (DBMS). Database ini dibuat untuk keperluan sistem database yang cepat, handal dan mudah digunakan (Fathansyah, 2007).

Berikut ini beberapa kelebihan MySQL sebagai *database server* antara lain (Fathansyah, 2007):

- a. *Source MySQL* dapat diperoleh dengan mudah dan gratis.
- b. Sintaksnya lebih mudah dipahami dan tidak rumit.
- c. Pengaksesan basis data dapat dilakukan dengan mudah.
- d. MySQL merupakan program yang *multi-threaded*, sehingga dapat dipasang pada *server* yang memiliki *multi CPU*.
- e. Didukung program-program umum seperti C, C++, Java, Perl, PHP, Python, dan lain sebagainya.
- f. Bekerja pada berbagai platform.

2.12. Pengujian Perangkat Lunak

Menurut IEEE, pengujian perangkat lunak adalah proses sistem operasi atau komponen menurut kondisi tertentu, pengamatan atau pencatatan hasil dan mengevaluasi beberapa aspek sistem atau komponen, proses analisis item perangkat lunak untuk mendeteksi perbedaan antara

kondisi yang ada dengan yang diinginkan dan mengevaluasi fitur perangkat lunak. Tujuan pengujian perangkat lunak (Peranganingin, 2006):

- a. Tujuan langsung
 - i. Identifikasi dan menemukan beberapa kesalahan yang mungkin ada dalam perangkat lunak yang diuji.
 - ii. Setelah perangkat lunak dibetulkan, diidentifikasi lagi kesalahan dan dites ulang untuk menjamin kualitas level penerimaan.
 - iii. Membentuk tes yang efisien dan efektif dengan anggaran dan jadwal yang terbatas.
- b. Tujuan tidak langsung

Mengumpulkan daftar kesalahan untuk digunakan dalam daftar pencegahan kesalahan.

Terdapat dua strategi pengujian perangkat lunak berdasarkan konsep pengujian sebagai berikut.

- i. *Black box (functionality) testing*. Mengidentifikasi kesalahan yang berhubungan dengan kesalahan fungsionalitas perangkat lunak yang tampak dalam kesalahan *output*. Definisi menurut IEEE, adalah pengujian yang mengabaikan mekanisme internal sistem atau komponen dan fokus semata-mata pada *output* yang dihasilkan yang merespon *input* yang dipilih dan kondisi eksekusi. Pengujian
- ii. yang dilakukan untuk mengevaluasi pemenuhan sistem atau komponen dengan kebutuhan fungsional tertentu.
- iii. *White box (structural) testing*. Memeriksa kalkulasi internal *path* untuk mengidentifikasi kesalahan. Definisi menurut IEEE, adalah pengujian yang memegang perhitungan mekanisme internal sistem atau komponen

Format pendefinisian rencana pengujian dengan standar penulisan IEEE dapat dilihat pada pada Tabel 2.10.

Tabel 2. 10. Format Pendefinisian Rencana Pengujian

KELAS UJI	BUTIR UJI	IDENTIFIKASI		TINGKAT PENGUJIAN	JENIS PENGUJIAN	PENGUJI
		SKPL	DUPL			

Keterangan:

Kelas Uji : Kelompok pengujian, misalnya pengujian *login*

Butir Uji: Sub kelompok pengujian

SKPL: Spesifikasi Kebutuhan Perangkat Lunak (SRS)

DUPL: Dokumen Uji Perangkat Lunak. Mengacu pada butir uji.

Tingkat Pengujian: Level dimana pengujian dilakukan (pengujian sistem, regresi, integrasi, dsb)

Jenis Pengujian: *Black-box*, *white-box*, dsb

Penguji: Subjek pengujian

Format pendefinisian hasil pengujian dengan standar penulisan IEEE dapat dilihat pada pada Tabel 2.11.

Tabel 2. 11. Format Pendefinisian Hasil Pengujian

IDENTIFIKASI	DESKRIPSI	PROSEDUR PENGUJIAN	MASUKAN	KELUARAN YANG DIHARAPKAN	KRITERIA EVALUASI HASIL	HASIL YANG DIDAPAT	KESIMPULAN

Keterangan:

Identifikasi: Mengacu pada DUPL.

Deskripsi: Mengacu pada kelas uji.

Prosedur Pengujian: Cara bagaimana pengujian dilakukan.

Masukan: nilai yang menjadi inputan.

Keluaran yang diharapkan: *Output* yang diharapkan.

Kriteria Evaluasi Hasil: Aksi yang dilakukan setelah mengerjakan sesuatu.

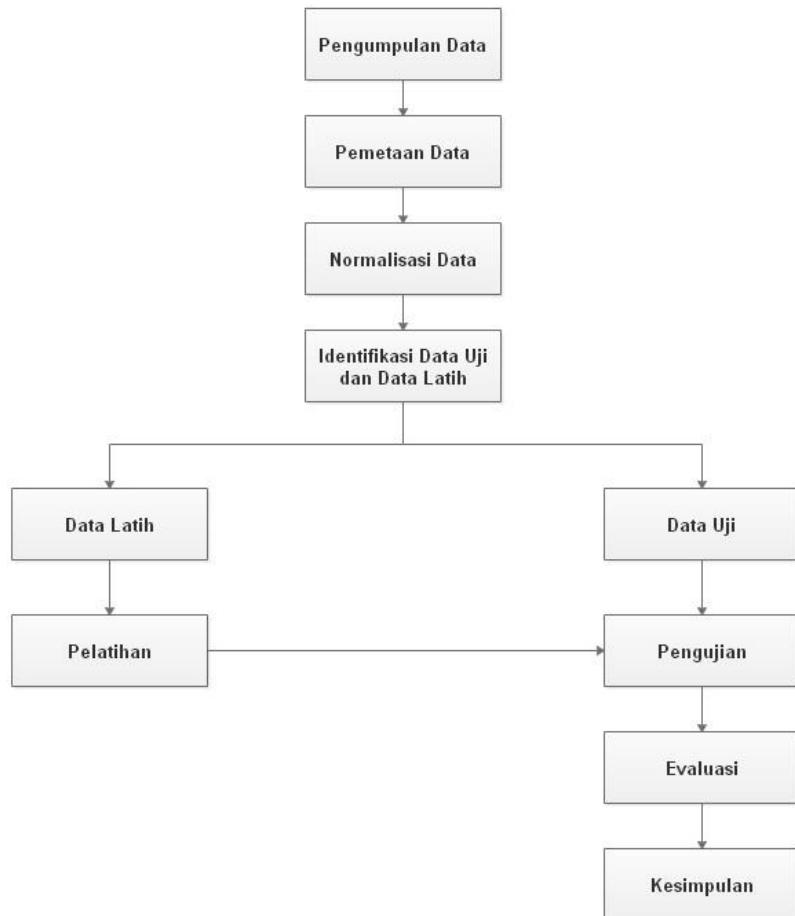
Hasil Yang Didapat: Kejadian yang muncul.

Kesimpulan: Diterima atau ditolak.

BAB III

METODOLOGI PENELITIAN

Bab ini membahas tentang langkah-langkah penyelesaian masalah, tahapan penelitian, dan rancangan sistem. Garis besar penyelesaian masalah disajikan pada gambar 3.1.



Gambar 3. 1. Langkah-Langkah Penyelesaian Masalah

3.1. Pengumpulan Data

Data yang digunakan dalam penelitian ini didapatkan dari data rekam medis pasien yang ada di Rumah Sakit Pusat Pertamina, Jakarta, dalam kurun waktu 3 tahun mulai dari Januari 2013 hingga Desember 2015 sebanyak 100 data. Data yang diperlukan yaitu data pasien yang menderita diabetes melitus, serta data pasien yang tidak diabetes melitus. Data tersebut akan digunakan sebagai data pelatihan dan pengujian dari penelitian ini. Data penelitian yang digunakan tiap kelasnya memiliki jumlah yang sama antara lain diabetes melitus berjumlah 50 data, tidak diabetes melitus berjumlah 50 data. Rangkuman data penelitian dapat dilihat pada tabel 3.1.

Tabel 3. 1 Rangkuman Data Penelitian

Status	Total
Diabetes Melitus	50
Bukan Diabetes Melitus	50
Jumlah	100

Perancangan data Jaringan Syaraf Tiruan LVQ merupakan desain data yang digunakan dalam proses pelatihan algoritma LVQ. Data yang digunakan berisi atribut utama yang diperlukan dalam diagnosa diabetes melitus menurut Hospital Authority yaitu kadar glukosa darah, serta atribut tambahan yang digunakan oleh RS Pusat Pertamina untuk mengoptimalkan dalam mendiagnosa penyakit diabetes melitus. Atribut data rekam medis penyakit diabetes melitus dapat dilihat pada tabel 3.2.

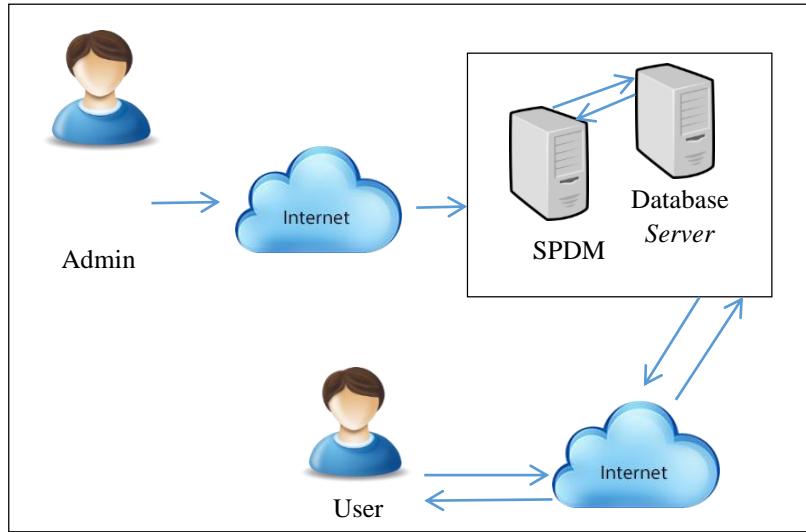
Tabel 3. 2. Atribut Data Rekam Medis

No.	Atribut	Keterangan
1.	umur	Umur pasien
2.	glun	Glukosa darah puasa
3.	gpost	Glukosa darah 2 jam setelah makan
4.	upost	Glukosa urin 2 jam setelah makan
5.	actn	Aseton urin puasa
6.	urn	Glukosa urin puasa
7.	actpp	Aseton urin 2 jam setelah makan
8.	ldl	Kolesterol ldl
9.	hdl	Kolesterol hdl
10.	chol	Kolesterol total
11.	tg	Triglicerida
12.	class	Kelas penyakit diabetes melitus

3.2. Arsitektur Sistem

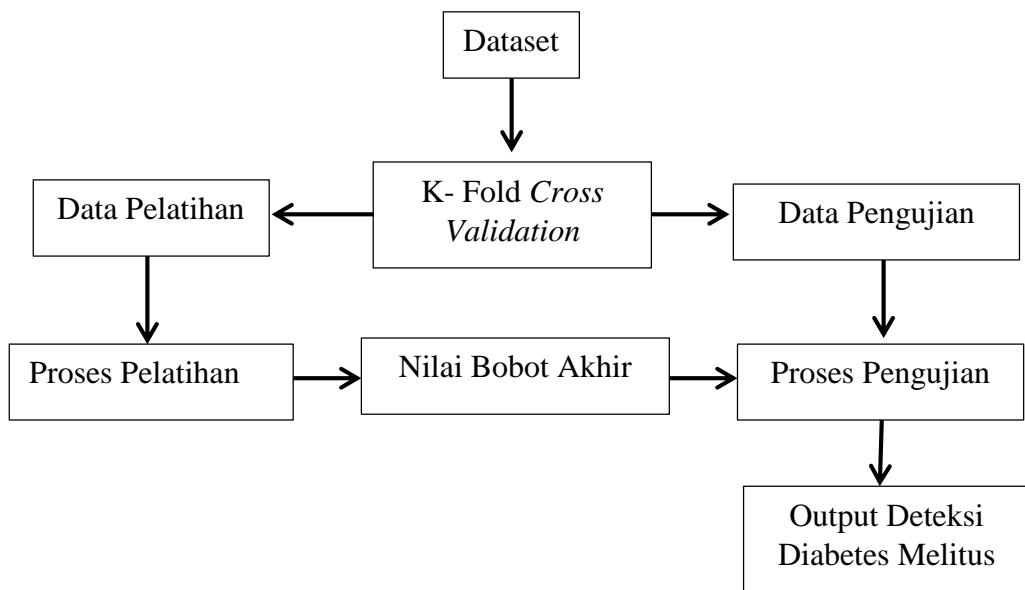
Pada saat ini untuk menganalisa hasil tes di laboratorium hanya dapat dilakukan secara manual oleh dokter, sehingga membutuhkan waktu dalam proses analisa. Sistem pendekripsi diabetes melitus ini dibangun untuk dapat memudahkan dalam menganalisa hasil tes yang telah dilakukan di laboratorium. Sistem ini berbasis web sehingga membutuhkan internet untuk dapat terhubung serta memiliki *database server* untuk penyimpanan data. Terdapat dua jenis

pengguna dalam sistem ini yaitu admin dan *guest*. Admin pada sistem ini akan dipegang oleh kepala bagian laboratorium, sedangkan *guest* merupakan semua orang yang bekerja di bagian laboratorium rumah sakit. Gambaran mengenai arsitektur sistem ini dapat dilihat pada gambar 3.2.



Gambar 3. 2. Arsitektur Sistem Pendekripsi Diabetes Melitus Menggunakan Jaringan Syaraf Tiruan LVQ

Sedangkan untuk tahapan alur pengolahan data deteksi penyakit diabetes melitus di dalam sistem dapat dilihat pada gambar 3.3.



Gambar 3. 3. Tahapan Alur Pengolahan Data Deteksi Diabetes Melitus

3.3. Pemetaan Data

Tahapan pemetaan data dilakukan untuk mengidentifikasi data yang telah diperoleh ke dalam arsitektur jaringan LVQ. Data diidentifikasi menjadi data *input neuron* dan data *output neuron*. Tabel 3.3 merupakan contoh hasil pengumpulan data. Keseluruhan data dapat dilihat pada Lampiran 1.

Tabel 3. 3 Hasil Pengumpulan Data

No	<i>Input Neuron</i>											<i>Output Neuron</i>
	umur	glun	gpost	upost	actn	urn	actpp	ldl	hdl	chol	tg	
1	53	77	161	0	0	0	0	26	77	208	176	Yes
2	61	114	152	0	0	0	0	87	87	114	159	Yes
3	57	88	83	0	0	0	0	31	28	87	139	Yes
4	61	127	193	0	0	0	0	33	92	209	161	Yes
5	61	119	238	1000	0	0	0	34	68	100	165	Yes
6	56	132	172	0	0	0	0	35	89	143	300	Yes
7	55	234	364	1000	0	1000	0	35	77	120	262	Yes
...
51	86	107	119	0	0	0	0	58	56	129	117	No
52	60	107	117	0	0	0	0	58	52	147	125	No
53	56	105	111	0	0	0	0	58	31	123	126	No
54	71	82	120	0	0	0	0	8	51	89	151	No
55	63	87	131	0	0	0	0	21	20	115	370	No
56	84	75	107	0	0	0	0	25	20	83	169	No
57	71	130	135	0	0	0	0	29	17	89	37	No
...

Berdasarkan tabel 3.3, dapat dilihat bahwa atribut yang menjadi *input neuron* adalah sebelas gejala diabetes melitus yaitu umur, glun (glukosa darah puasa), gpost (glukosa darah 2 jam setelah makan), upost (glukosa urin 2 jam setelah makan), actn (aseton urin puasa), urn (glukosa urin puasa), actpp (aseton urin 2 jam setelah makan), ldl (kolesterol ldl), hdl (kolesterol hdl), chol (kolesterol total), dan tg (trigliserida). Sedangkan yang menjadi *output neuron* adalah class, yaitu klasifikasi positif diabetes melitus atau negatif diabetes melitus.

3.4. Normalisasi Data

Pada penelitian ini setiap gejala perlu dilakukan normalisasi karena data input belum dalam rentang [0,1]. Normalisasi data bertujuan untuk menghindari terjadinya kesalahan, anomali data dan tidak konsistensinya data sehingga data nantinya dapat dilatih dan memperoleh hasil pengujian yang maksimal. Normalisasi yang digunakan pada penelitian ini adalah normalisasi *min-max*

Berikut adalah contoh perhitungan normalisasi *min max*. Data pada tabel 3.3 akan dinormalisasi dengan nilai maksimum dan minimum tiap atribut sesuai dengan tabel 3.4.

Tabel 3. 4. Nilai Maksimum dan Minimum Tiap Atribut

Nilai	Atribut										
	umur	glun	gpost	upost	actn	urn	actpp	ldl	hdl	chol	tg
Max	92	500	691	1000	50	1000	5	145	92	231	375
Min	18	61	83	0	0	0	0	8	13	53	34

Normalisasi data pertama :

$$umur = \frac{53-18}{92 - 18} = 0.47$$

$$glun = \frac{77-61}{500 - 61} = 0.04$$

$$gpost = \frac{161-83}{691 - 83} = 0.13$$

...

$$tg = \frac{176-34}{375 - 34} = 0.42$$

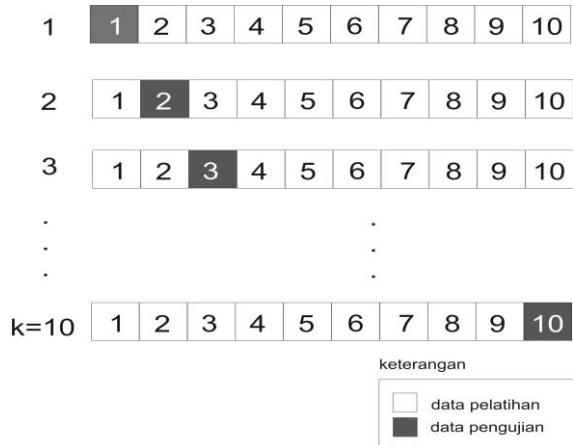
Normalisasi lalu akan dilanjutkan hingga semua data telah dinormalisasi. Sehingga hasil normalisasi dari tabel 3.3 dapat dilihat pada tabel 3.5.

Tabel 3. 5. Hasil Normalisasi Data

No	Input Neuron											Output Neuron
	umur	glun	gpost	upost	actn	urn	actpp	ldl	hdl	chol	tg	
1	0.47	0.04	0.13	0	0	0	0	0.24	0.70	0.45	0.42	1
2	0.36	0.12	0.22	0	0	0	0	0.32	0.35	0.36	0.36	1
3	0.40	0.16	0.17	0	0	0	0	0.35	0.67	0.51	0.37	1
4	0.36	0.15	0.15	0	0	0	0	0.33	0.52	0.48	0.38	1
5	0.36	0.17	0.18	1	0	0	0	0.33	0.52	0.42	0.31	1
6	0.45	0.12	0.29	0	0	0	0	0.33	0.74	0.33	0.41	1
7	0.47	0.03	0.12	1	0	1	0	0.13	0.81	0.87	0.41	1
...
51	0.45	0.12	0.29	0	0	0	0	0.33	0.74	0.33	0.41	2
52	0.47	0.03	0.12	0	0	0	0	0.13	0.81	0.87	0.41	2
53	0.23	0.18	0.15	0	0	0	0	0.33	0.28	0.39	0.46	2
54	0.50	0.15	0.12	0	0	0	0	0.23	0.35	0.47	0.30	2
55	0.54	0.12	0.13	0	0	0	0	0.32	0.21	0.63	0.87	2
56	0.45	0.10	0.07	0	0	0	0	0.35	0.39	0.32	0.04	2
57	0.48	0.06	0.02	0	0	0	0	0.22	0.35	0.23	0.11	2
...

3.5. Pembagian Data Latih dan Data Uji

Setelah dilakukan normalisasi, kemudian data akan diproses menggunakan *K-Fold Cross Validation*. Dengan menggunakan $K = 10$, dataset akan dibagi menjadi 10 partisi dimana tiap partisi akan memiliki data dengan jumlah kelas yang sama. Gambar 3.4 merupakan ilustrasi *K-Fold Cross Validation* pada dataset.



Gambar 3. 4. *K-Fold Cross Validation* Pada Dataset

Adapun gambaran hasil K-Fold dengan bernilai 10 dan kelas sebanyak 2 kelas dapat dilihat pada tabel 3.6.

Tabel 3. 6 Pembagian 10-Fold Untuk Data Latih dan Data Uji

K = 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
1,2,3, 4,5,51, 52,53,5 4,55	6,7,8,9, 10,56,5 7,58,59 .60	11,12,1 3,14,15 .61,62, 63,64,6 5	16,17,18, 19,20,66, 67,68,69, 70	21,22,23, 24,25,71, 72,73,74, 75	26,27,28, 29,30,76, 77,78,79, 80	31,32,33, 34,35,81, 82,83,84, 85	36,37,38, 39,40,86, 87,88,89, 90	41,42,43 ,44,45,9 1,92,93, 94,95	46,47,48, 49,50,96, 97,98,99, 100
1,2,3, 4,5,51, 52,53,5 4,55	6,7,8,9, 10,56,5 7,58,59 .60	11,12,1 3,14,15 .61,62, 63,64,6 5	16,17,18, 19,20,66, 67,68,69, 70	21,22,23, 24,25,71, 72,73,74, 75	26,27,28, 29,30,76, 77,78,79, 80	31,32,33, 34,35,81, 82,83,84, 85	36,37,38, 39,40,86, 87,88,89, 90	41,42,43 ,44,45,9 1,92,93, 94,95	46,47,48, 49,50,96, 97,98,99, 100
1,2,3, 4,5,51, 52,53,5 4,55	6,7,8,9, 10,56,5 7,58,59 .60	11,12,1 3,14,15 .61,62, 63,64,6 5	16,17,18, 19,20,66, 67,68,69, 70	21,22,23, 24,25,71, 72,73,74, 75	26,27,28, 29,30,76, 77,78,79, 80	31,32,33, 34,35,81, 82,83,84, 85	36,37,38, 39,40,86, 87,88,89, 90	41,42,43 ,44,45,9 1,92,93, 94,95	46,47,48, 49,50,96, 97,98,99, 100
1,2,3, 4,5,51, 52,53,5 4,55	6,7,8,9, 10,56,5 7,58,59 .60	11,12,1 3,14,15 .61,62, 63,64,6 5	16,17,18, 19,20,66, 67,68,69, 70	21,22,23, 24,25,71, 72,73,74, 75	26,27,28, 29,30,76, 77,78,79, 80	31,32,33, 34,35,81, 82,83,84, 85	36,37,38, 39,40,86, 87,88,89, 90	41,42,43 ,44,45,9 1,92,93, 94,95	46,47,48, 49,50,96, 97,98,99, 100

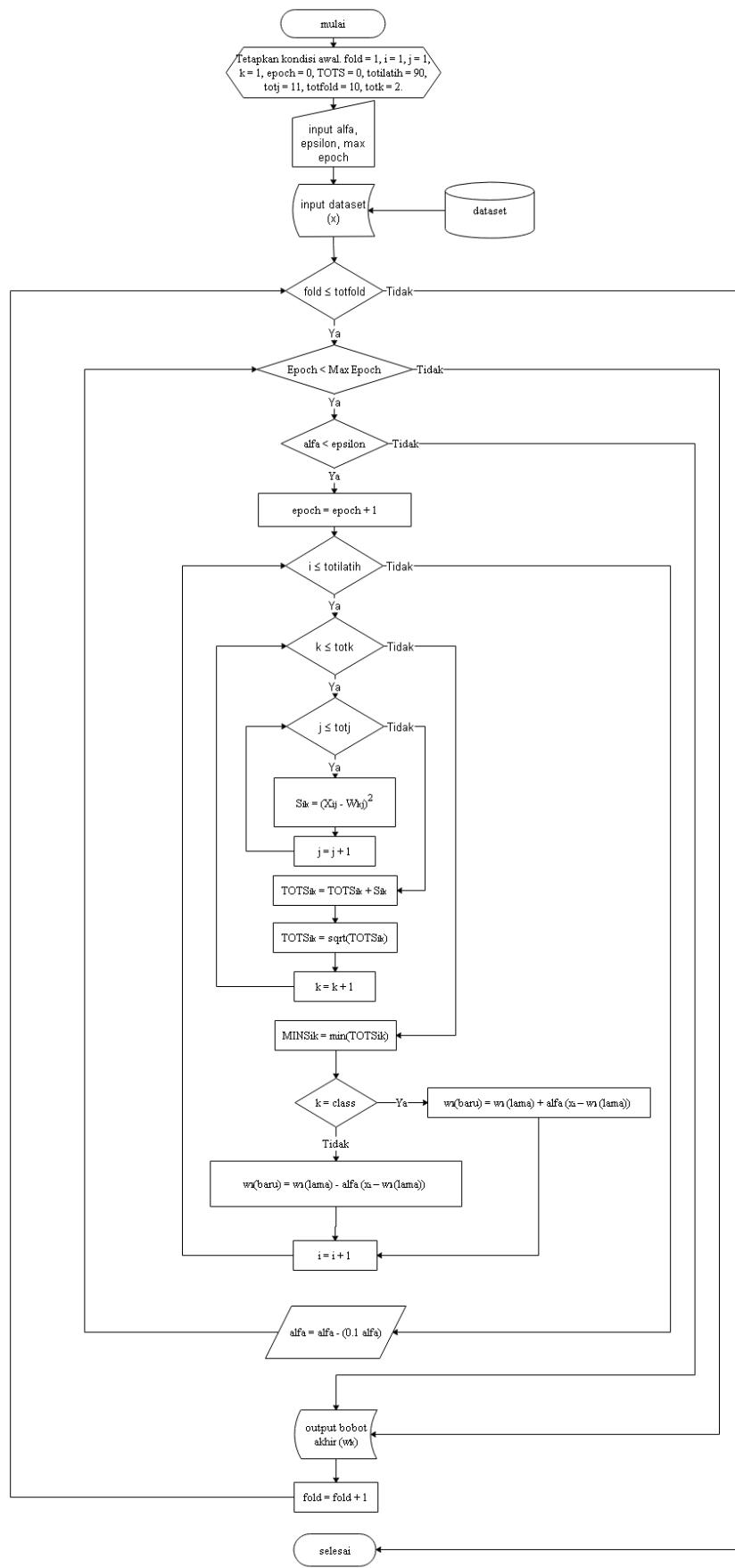
1,2,3, 4,5,51, 52,53,5 4,55	6,7,8,9, 10,56,5 7,58,59 ,60	11,12,1 3,14,15 ,61,62, 63,64,6 5	16,17,18, 19,20,66, 67,68,69, 70	21,22,23, 24,25,71, 72,73,74, 75	26,27,28, 29,30,76, 77,78,79, 80	31,32,33, 34,35,81, 82,83,84, 85	36,37,38, 39,40,86, 87,88,89, 90	41,42,43 ,44,45,9 1,92,93, 94,95	46,47,48, 49,50,96, 97,98,99, 100
1,2,3, 4,5,51, 52,53,5 4,55	6,7,8,9, 10,56,5 7,58,59 ,60	11,12,1 3,14,15 ,61,62, 63,64,6 5	16,17,18, 19,20,66, 67,68,69, 70	21,22,23, 24,25,71, 72,73,74, 75	26,27,28, 29,30,76, 77,78,79, 80	31,32,33, 34,35,81, 82,83,84, 85	36,37,38, 39,40,86, 87,88,89, 90	41,42,43 ,44,45,9 1,92,93, 94,95	46,47,48, 49,50,96, 97,98,99, 100
1,2,3, 4,5,51, 52,53,5 4,55	6,7,8,9, 10,56,5 7,58,59 ,60	11,12,1 3,14,15 ,61,62, 63,64,6 5	16,17,18, 19,20,66, 67,68,69, 70	21,22,23, 24,25,71, 72,73,74, 75	26,27,28, 29,30,76, 77,78,79, 80	31,32,33, 34,35,81, 82,83,84, 85	36,37,38, 39,40,86, 87,88,89, 90	41,42,43 ,44,45,9 1,92,93, 94,95	46,47,48, 49,50,96, 97,98,99, 100
1,2,3, 4,5,51, 52,53,5 4,55	6,7,8,9, 10,56,5 7,58,59 ,60	11,12,1 3,14,15 ,61,62, 63,64,6 5	16,17,18, 19,20,66, 67,68,69, 70	21,22,23, 24,25,71, 72,73,74, 75	26,27,28, 29,30,76, 77,78,79, 80	31,32,33, 34,35,81, 82,83,84, 85	36,37,38, 39,40,86, 87,88,89, 90	41,42,43 ,44,45,9 1,92,93, 94,95	46,47,48, 49,50,96, 97,98,99, 100
1,2,3, 4,5,51, 52,53,5 4,55	6,7,8,9, 10,56,5 7,58,59 ,60	11,12,1 3,14,15 ,61,62, 63,64,6 5	16,17,18, 19,20,66, 67,68,69, 70	21,22,23, 24,25,71, 72,73,74, 75	26,27,28, 29,30,76, 77,78,79, 80	31,32,33, 34,35,81, 82,83,84, 85	36,37,38, 39,40,86, 87,88,89, 90	41,42,43 ,44,45,9 1,92,93, 94,95	46,47,48, 49,50,96, 97,98,99, 100

3.6. Pelatihan Menggunakan *Learning Vector Quantization*

Proses pelatihan bertujuan untuk melatih metode *Learning Vector Quantization* untuk mengenali suatu pola pada data diabetes melitus. Proses ini bertujuan untuk menemukan unit *output* yang paling dekat dengan vektor *input*.

3.6.1. Flowchart Fungsi Pelatihan LVQ

Pada metode LVQ, proses pertama pada pelatihan adalah menghitung bobot dengan mencari nilai *euclidean distance*, kemudian dilakukan perubahan bobot pada kelas yang memiliki nilai *euclidean distance* terkecil. Hasil dari proses ini adalah didapatkannya nilai bobot yang akan digunakan pada proses pengujian. Alur pelatihan LVQ dapat dilihat pada *flowchart* proses pelatihan LVQ gambar 3.5.



Gambar 3. 5. Flowchart Fungsi Pelatihan LVQ

Keterangan :

fold	: indeks untuk menentukan posisi kfold
i	: indeks untuk menentukan posisi data
j	: indeks untuk menentukan posisi atribut dari data
k	: indeks untuk menentukan posisi dari kelas target
<i>epoch</i>	: indeks untuk menentukan posisi <i>epoch</i>
S	: nilai <i>euclidean distance</i> dari dataset dengan bobot
TOTS	: total dari nilai <i>euclidean distance</i> semua data pada kelas tertentu
<i>alfa</i>	: nilai <i>alfa</i> yang digunakan pada proses pelatihan
<i>epsilon</i>	: nilai <i>epsilon</i> yang digunakan untuk proses pelatihan
<i>max epoch</i>	: jumlah <i>epoch</i> maksimal yang digunakan untuk proses pelatihan
dataset (x)	: dataset penyakit diabetes melitus yang akan dimasukan
bobot awal (w)	: bobot yang digunakan pada awal proses pelatihan
totilatih	: jumlah data latih keseluruhan yang digunakan
totj	: jumlah atribut yang ada
totk	: jumlah kelas yang ada
class	: kelas target
MINS	: nilai <i>euclidean distance</i> minimum diantara dua kelas target

3.6.2. Perhitungan Manual Pelatihan LVQ

Berikut adalah contoh perhitungan manual pelatihan menggunakan LVQ. Bobot awal pada proses pelatihan pada penelitian ini akan dilakukan secara acak dari tiap kelas yang sudah ternormalisasi. Perhitungan manual LVQ menggunakan bobot awal diambil acak dari dataset, *learning rate* = 0.01, *epsilon* = 0.01, maksimum *epoch* = 1 dan kfold ke-1. Data pelatihan yang telah dinormalisasi dapat dilihat pada tabel 3.7.

Tabel 3. 7 Tabel Data Pelatihan (Telah Dinormalisasi)

Dat a Pel atih an Ke-	Id Dat aset	umur	glun	gpost	upost	actn	urn	actpp	ldl	hdl	chol	tg	class
1	6	0.45	0.12	0.29	0	0	0	0	0.33	0.74	0.33	0.41	1
2	7	0.47	0.03	0.12	1	0	1	0	0.13	0.81	0.87	0.41	1
...
25	35	0.58	0.12	0.11	0	0	0	0	0.15	0.93	0.34	0.36	1
...
46	56	0.45	0.10	0.07	0	0	0	0	0.35	0.39	0.32	0.04	2
47	57	0.48	0.06	0.02	0	0	0	0	0.22	0.35	0.23	0.11	2
...
62	72	0.56	0.02	0.06	0	0	0	0	0.25	0.16	0.26	0.36	2
...
90	100	0.48	0.16	0.17	0	0	0	0	0.28	0.43	0.38	0.29	2

Berikut ini adalah proses perhitungan pelatihan LVQ pada fold ke-1 dengan data pelatihan ke 1.

1. Inisialisasi bobot awal, $MaxEpoch = 1$, $epsilon = 0.01$, dan $learning rate (\alpha) = 0.01$. Inisialisasi bobot awal yang digunakan pada contoh perhitungan ini menggunakan data dari setiap kelas yang diambil acak dari dataset, sehingga jumlah bobot yang digunakan sesuai dengan jumlah kelas target, yaitu dua.

$$w_1 = (0.58, 0.12, 0.11, 0, 0, 0, 0, 0.15, 0.93, 0.34, 0.36) \quad (\text{data ke-35})$$

$$w_2 = (0.56, 0.02, 0.06, 0, 0, 0, 0, 0.25, 0.16, 0.26, 0.36) \quad (\text{data ke-72})$$

2. Menetapkan kondisi awal $epoch = 0$.
3. Dikerjakan jika ($epoch < MaxEpoch$) atau ($\alpha < epsilon$)
 - a. $Epoch = epoch + 1$
 - b. Dikerjakan $i = 1$ sampai m (m merupakan jumlah data pelatihan yaitu 90).
Dalam contoh hanya menggunakan data pelatihan ke-1)
Data ke $i = 1$, $x_1 = (0.45, 0.12, 0.29, 0, 0, 0, 0, 0.33, 0.74, 0.33, 0.41)$
 - i. Mengerjakan $k = 1$ sampai n (n merupakan jumlah kelas yaitu 2).

$$\text{Menghitung nilai } S_{ik} = \sqrt{\sum_{j=1}^n (x_{ij} - w_{kj})^2}$$

$$S_{II} = \sqrt{(0.45 - 0.58)^2 + (0.12 - 0.12)^2 + (0.29 - 0.11)^2 \dots + (0.41 - 0.36)^2}$$

$$= 0.3505709629$$

$$S_{I2} = \sqrt{(0.45 - 0.56)^2 + (0.12 - 0.02)^2 + (0.29 - 0.06)^2 \dots + (0.41 - 0.36)^2}$$

$$= 0.6520736155$$

Menentukan nilai S_{ik} terkecil dan simpan k .

Nilai S_{ik} terkecil, $k = 1$

- ii. Memperbaiki nilai w_k , dengan membandingkan nilai k dengan T

Karena nilai $k = 1$ dan $T_1 = 1$, $k = T$ maka

$$w_k (\text{baru}) = w_k (\text{lama}) + \alpha (x_i - w_k (\text{lama}))$$

$$w_1 (\text{baru}) = (0.58, 0.12, 0.11, 0, 0, 0, 0, 0.15, 0.93, 0.34, 0.36) + 0.01 * ((0.45, 0.12, 0.29, \dots, 0.41) - (0.58, 0.12, \dots, 0.36))$$

$$w_1 (\text{baru}) = (1.0025, 0.3333, 0.6734, 0, 0, 0, 0, 0.1518, 0.9281, 0.3399, 0.3605)$$

Sehingga

$$w_1 = (1.0025, 0.3333, 0.6734, 0, 0, 0, 0, 0.1518, 0.9281, 0.3399, 0.3605)$$

$$w_2 = (0.56, 0.02, 0.06, 0, 0, 0, 0, 0.25, 0.16, 0.26, 0.36)$$

Proses pelatihan terus berlanjut seterusnya dan akan berhenti ketika sudah mencapai *MaxEpoch* atau nilai $\alpha < epsilon$. Dari perhitungan pelatihan ini didapatkan bobot akhir dari tiap fold, namun pada pelatihan ini yang akan diambil adalah bobot akhir yang diperoleh dari fold pertama. Berikut adalah bobot akhir yang digunakan.

$$w_1 = (0.5571, 0.1219, 0.1193, 0.0262, 0.0066, 0.0299, -0.0120, 0.1586, 0.9246, 0.3590, 0.3562)$$

$$w_2 = (0.5488, 0.0180, 0.0649, 0.0120, 0, 0.0048, 0, 0.2674, 0.1674, 0.2759, 0.3678)$$

Nilai bobot akhir dari pelatihan ini akan digunakan pada proses pengujian.

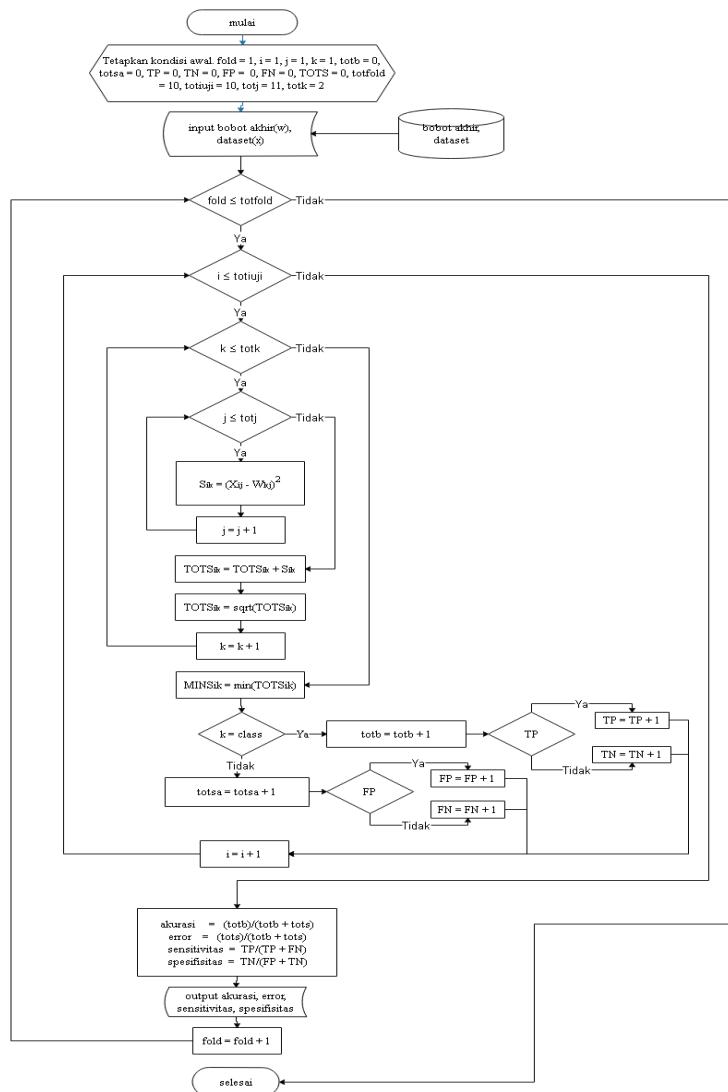
3.7. Pengujian Menggunakan *Learning Vector Quantization*

Proses pengujian merupakan proses untuk menghasilkan hasil uji. Menggunakan data pengujian dan bobot akhir yang diperoleh pada proses pelatihan. Kemudian dengan menggunakan *euclidean distance* untuk mencari jarak pada data pengujian dan bobot akhir dari

masing-masing kelas. Kelas yang memiliki nilai *euclidean distance* terkecil akan dikenali sebagai kelas dari hasil deteksi diabetes melitus.

3.7.1. Flowchart Fungsi Pengujian LVQ

Pada proses pengujian, Proses pengujian hampir sama dengan proses pelatihan yang melibatkan perhitungan *euclidean distance*, yang membedakan adalah pada proses perubahan bobot. Pada proses pengujian tidak terjadi perubahan bobot setelah mendapat nilai *euclidean distance* terkecil, melainkan melakukan perhitungan akurasi, *error*, sensitivitas, spesifisitas dari tiap kelas. Kemudian akan diproses menggunakan metode evaluasi *confussion matrix* agar mendapatkan hasil presentase akurasi, *error*, sensitivitas, dan spesifisitas. *Flowchart* fungsi pengujian ditunjukkan pada gambar 3.6.



Gambar 3. 6. Flowchart Fungsi Pengujian LVQ

Keterangan :

fold	: indeks untuk menentukan posisi kfold
i	: indeks untuk menentukan posisi data
j	: indeks untuk menentukan posisi atribut dari data
k	: indeks untuk menentukan posisi dari kelas target
epoch	: indeks untuk menentukan posisi <i>epoch</i>
S	: nilai <i>euclidean distance</i> dari dataset dengan bobot
TOTS	: total dari nilai <i>euclidean distance</i> semua data pada kelas tertentu
alfa	: nilai <i>alfa</i> yang digunakan pada proses pelatihan
epsilon	: nilai <i>epsilon</i> yang digunakan untuk proses pelatihan
max epoch	: jumlah <i>epoch</i> maksimal yang digunakan untuk proses pelatihan
dataset (x)	: dataset penyakit diabetes melitus yang akan dimasukan
bobot akhir (w)	: bobot yang digunakan pada proses pengujian
totuji	: jumlah data pengujian yang digunakan
totj	: jumlah atribut yang ada
totk	: jumlah kelas yang ada
totb	: jumlah deteksi yang teridentifikasi positif diabetes melitus
totsa	: jumlah deteksi yang teridentifikasi negatif diabetes melitus
TP	: jumlah <i>true positive</i> yang teridentifikasi
TN	: jumlah <i>true negative</i> yang teridentifikasi
FP	: jumlah <i>false positive</i> yang teridentifikasi
FN	: jumlah <i>false negative</i> yang teridentifikasi
class	: kelas target
MINS	: nilai <i>euclidean distance</i> minimum diantara dua kelas target

3.7.2. Perhitungan Manual Pengujian LVQ

Berikut adalah contoh perhitungan manual pengujian menggunakan LVQ. Data pengujian yang telah dinormalisasi dapat dilihat pada tabel 3.8.

Tabel 3. 8 Tabel Data Pengujian (Telah Dinormalisasi)

Dat a Pen guji an Ke-	Id	Dat aset	umur	glun	gpost	upost	actn	urn	actpp	ldl	hdl	chol	tg	class
1	1	0.47	0.04	0.13	0	0	0	0	0.24	0.70	0.45	0.42	1	

2	2	0.36	0.12	0.22	0	0	0	0	0.32	0.35	0.36	0.36	1
...
6	51	0.45	0.12	0.29	0	0	0	0	0.33	0.74	0.33	0.41	2
7	52	0.47	0.03	0.12	0	0	0	0	0.13	0.81	0.87	0.41	2
...	0	0.32	0.21	0.63	0.87	...
10	55	0.54	0.12	0.13	0	0	0	0	0.32	0.21	0.63	0.87	2

Berikut adalah proses perhitungan pengujian LVQ dengan data pengujian ke-1:

- Menetapkan bobot akhir (w_k) dari hasil pelatihan dan memasukkan data pengujian yang digunakan.

$$w_1 = (0.5571, 0.1219, 0.1193, 0.0262, 0.0066, 0.0299, -0.0120, 0.1586, 0.9246, 0.3590, 0.3562)$$

$$w_2 = (0.5488, 0.0180, 0.0649, 0.0120, 0, 0.0048, 0, 0.2674, 0.1674, 0.2759, 0.3678)$$

- Mengerjakan $i = 1$ sampai m (m merupakan jumlah data pengujian yaitu 10)

- Mengerjakan untuk $k=1$ sampai n . Menghitung nilai $S_{ik} = \sqrt{\sum_{j=1}^n (x_{ij} - w_{kj})^2}$

$$\text{Data uji ke-1, } x_1 = (0.47, 0.04, 0.13, 0, 0, 0, 0, 0.24, 0.70, 0.45, 0.42)$$

$$\begin{aligned} S_{11} &= \sqrt{(0.47 - 0.5571)^2 + (0.04 - 0.1219)^2 + \dots + (0.42 - 0.3562)^2} \\ &= 0.29161618 \end{aligned}$$

$$\begin{aligned} S_{12} &= \sqrt{(0.47 - 0.5488)^2 + (0.04 - 0.0180)^2 + \dots + (0.42 - 0.3678)^2} \\ &= 0.58242596 \end{aligned}$$

- Tentukan S_{ik} terkecil dan simpan k

Karena nilai S_{11} memiliki nilai terkecil, maka $k = 1$. Sehingga data pengujian ke-1 diklasifikasikan menjadi kelas ke-1. Karena nilai $k = 1$ dan $T = 1$, maka $k = T$. Sehingga nilai klasifikasi benar.

Proses pengujian LVQ tersebut akan dilakukan seterusnya sampai dengan data pengujian ke-10. Berikut tabel uji data kfold ke-1 yang ditunjukkan pada tabel 3.9.

Tabel 3. 9. Tabel Uji Data K-Fold ke-1

data pengujian ke-	id dataset	umur	glun	gpost	...	tg	Target	hasil klasifikasi
1	1	0.47	0.04	0.13	...	0.42	1	1
2	2	0.36	0.12	0.22	...	0.36	1	1
3	3	0.40	0.16	0.17	...	0.37	1	1
4	4	0.36	0.15	0.15	...	0.38	1	1
5	5	0.45	0.17	0.18	...	0.31	1	1
6	51	0.45	0.12	0.29	...	0.41	2	2
7	52	0.47	0.03	0.12	...	0.41	2	2

data pengujian ke-	id dataset	umur	glun	gpost	...	tg	Target	hasil klasifikasi
8	53	0.23	0.18	0.15	...	0.46	2	1
9	54	0.50	0.15	0.12	...	0.30	2	2
10	55	0.54	0.12	0.13	...	0.87	2	2

3.8. Evaluasi Klasifikasi

Evaluasi akurasi dari klasifikasi pada penelitian ini dilakukan dengan menggunakan *confussion matrix* dengan keluaran akurasi, *error rate*, sensitivitas, dan spesifisitas untuk menganalisa keberhasilan algoritma dalam melakukan deteksi penyakit diabetes melitus. Akurasi merupakan perbandingan dari kasus yang diidentifikasi benar dengan jumlah keseluruhan kasus, sedangkan *error rate* merupakan perbandingan dari kasus yang diidentifikasi salah dengan jumlah keseluruhan kasus.

Setelah melakukan perhitungan pengujian, maka dapat diperoleh *confussion matrix* untuk memperoleh akurasi dan *error*. Tabel uji tersebut ditabulasikan dalam *confussion matrix* dengan dua kelas dapat dilihat pada tabel 3.10.

Tabel 3. 10. Tabel *Confussion Matrix* Dengan Dua Kelas

Target	Hasil Klasifikasi	
	Kelas ke-1	Kelas ke-2
Kelas ke-1	5	0
Kelas ke-2	1	4

Sedangkan untuk menghitung sensitivitas dan spesifisitas, menggunakan matrix 2x2 untuk menghitung sensitivitas dan spesifisitas tiap kelas. Sebagai contoh untuk menghitung sensitivitas dan spesifisitas, dapat dilihat pada tabel 3.11.

Tabel 3. 11 Tabel *Confussion Matrix* Menghitung Sensitivitas dan Spesifisitas

Target	Hasil Klasifikasi	
	Kelas ke-1	Kelas ke- 2
Kelas ke-1	5 (TP)	0 (FN)
Kelas ke-2	1 (FP)	4 (TN)

Keterangan:

TP : *True Positive*

FP : *False Positive*

FN : *False Negative*

TN : *True Negative*

True positive menyatakan pada target masuk kelas 1 dan teridentifikasi sebagai kelas 1. *False positive* menyatakan pada target bukan termasuk kelas 1, namun teridentifikasi sebagai kelas 1. *False negative* menyatakan pada target termasuk kelas 1 dan teridentifikasi sebagai bukan kelas 1. *True negative* menyatakan pada target menyatakan bukan kelas 1 dan teridentifikasi sebagai bukan kelas 1. Berikut perhitungan sensitivitas dan spesifisitas. Berikut perhitungan tingkat akurasi, *error*, sensitivitas, dan spesifisitas.

$$\text{akurasi} = \frac{\text{banyaknya prediksi yang benar}}{\text{total banyaknya prediksi}} = \frac{9}{10} = 0.9 = 90\%$$

$$\text{error rate} = \frac{\text{banyaknya prediksi yang salah}}{\text{total banyaknya prediksi}} = \frac{1}{10} = 0.1 = 10\%$$

$$\text{sensitivitas} = \frac{TP}{TP + FN} = \frac{5}{5 + 0} = 1 = 100\%$$

$$\text{spesifisitas} = \frac{TN}{FP + TN} = \frac{4}{4 + 1} = 0.8 = 80\%$$

Dari hasil perhitungan tersebut didapatkan tingkat akurasi pada fold ke-1 sebesar 90%, tingkat *error* sebesar 10%, tingkat sensitivitas sebesar 100% dan tingkat spesifisitas sebesar 80%. Yang memiliki arti pada tingkat sensitivitas 100% maka sistem dapat mengidentifikasi positif terklasifikasi benar tiap kelas sebesar 100%. Tingkat spesifisitas 80%, maka sistem dapat mengidentifikasi negatif terklasifikasi benar tiap kelas sebesar 80%.

3.9. Analisis Kebutuhan

Analisis kebutuhan sistem meliputi analisis kebutuhan pengguna, kebutuhan fungsional dan non fungsional, pemodelan data, serta pemodelan fungsional.

3.9.1. Kebutuhan Pengguna

Pengguna dalam sistem ini dibagi menjadi dua yaitu admin dan *guest*. Detail tanggung jawab pengguna dapat dilihat pada tabel 3.12.

Tabel 3. 12. Tabel Pengguna Sistem

Pengguna	Deskripsi
Admin	<ul style="list-style-type: none">- Mengelola seluruh data yaitu dataset, data latih, data uji, dan data penanganan.- Melakukan pelatihan dan pengujian sistem terkait metode yang digunakan.

Pengguna	Deskripsi
Guest	<ul style="list-style-type: none"> - Mengisi atribut yang diperlukan melalui <i>form</i> pada sistem - Melihat hasil deteksi - Melihat penanganan terkait hasil deteksi

3.9.2. Kebutuhan Fungsional dan Non Fungsional

Analisis kebutuhan perangkat lunak ini menghasilkan *software requirement specification* (SRS) yang terdiri dari kebutuhan fungsional dan non fungsional. Kebutuhan fungsional disajikan pada tabel 3.13 dan kebutuhan non fungsional pada tabel 3.14.

Tabel 3. 13. Kebutuhan Fungsional Sistem

No.	SRS – ID	Deskripsi
1	SRS – SPDM – F – 001	Sistem dapat melakukan otentikasi <i>login</i> untuk admin
2	SRS – SPDM – F – 002	Sistem dapat mengakomodasi admin untuk mengubah <i>password</i>
3	SRS – SPDM – F – 003	Sistem dapat mengakomodasi admin untuk <i>import</i> dataset
4	SRS – SPDM – F – 004	Sistem dapat menampilkan dataset
5	SRS – SPDM – F – 005	Sistem dapat menampilkan data penanganan
6	SRS – SPDM – F – 006	Sistem dapat mengakomodasi admin untuk mengubah data penanganan
7	SRS – SPDM – F – 007	Sistem dapat mengakomodasi admin untuk melakukan pelatihan
8	SRS – SPDM – F – 008	Sistem dapat menampilkan hasil pelatihan
9	SRS – SPDM – F – 009	Sistem dapat mengakomodasi admin untuk menghapus data pelatihan
10	SRS – SPDM – F – 010	Sistem dapat mengakomodasi admin untuk melakukan pengujian
11	SRS – SPDM – F – 011	Sistem dapat menampilkan hasil pengujian
12	SRS – SPDM – F – 012	Sistem dapat mengakomodasi admin untuk menghapus data pengujian
13	SRS – SPDM – F – 013	Sistem dapat mengakomodasi admin untuk memilih bobot final untuk deteksi
14	SRS – SPDM – F – 014	Sistem dapat menampilkan bobot final yang digunakan untuk deteksi
15	SRS – SPDM – F – 015	Sistem dapat melakukan <i>Logout</i> untuk admin
16	SRS – SPDM – F – 016	Sistem dapat mengakomodasi <i>guest</i> untuk melakukan deteksi diabetes melitus

Tabel 3. 14. Kebutuhan Non Fungsional Sistem

No	SRS – ID	Deskripsi
1	SRS – SPDM – NF – 001	Sistem dijalankan menggunakan jaringan internet dan <i>web browser</i> yang mendukung HTML 5 dan JQuery

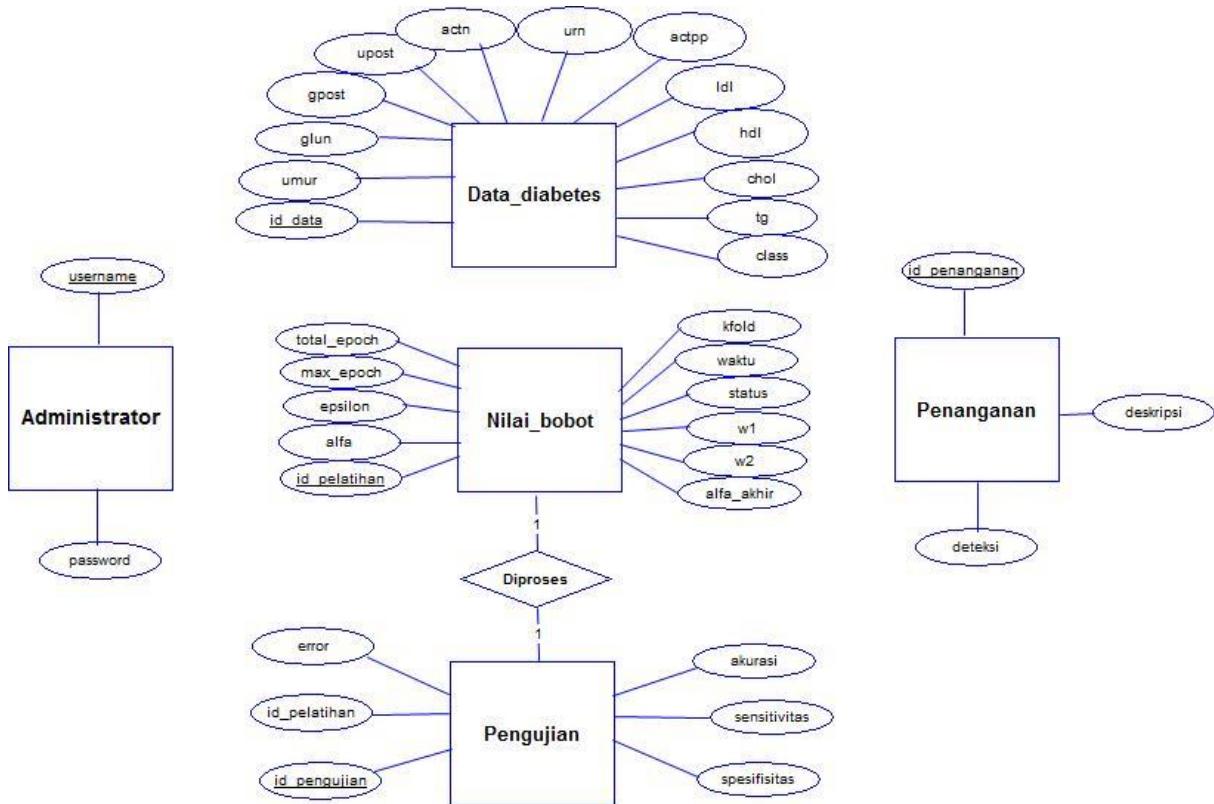
3.9.3. Pemodelan Data Sistem Pendeksi Diabetes Melitus

Sistem ini menggunakan pemodelan data *Entity Relationship Diagram* (ERD).

Pemodelan data ERD diperlukan untuk mengakomodasi kebutuhan data pada sistem.

Terdapat lima obyek data yang terlibat dalam sistem ini yaitu data_diabetes, nilai_bobot, pengujian, penanganan, dan administrator. Masing-masing obyek data tersebut memiliki beberapa atribut.

- a. Obyek data Data_diabetes memiliki atribut yang digunakan untuk menyimpan data gejala.
- b. Obyek data Nilai_bobot memiliki atribut yang digunakan untuk proses pelatihan LVQ, serta atribut yang dihasilkan dari proses pelatihan.
- c. Obyek data Pengujian memiliki atribut yang dihasilkan dari proses pengujian menggunakan LVQ.
- d. Obyek data Penanganan memiliki atribut yang digunakan untuk memberikan penanganan berdasarkan hasil deteksi.
- e. Obyek data Administrator berisi atribut yang digunakan untuk proses *login* admin. Kelima obyek data tersebut digambarkan dalam ERD pada gambar 3.7.



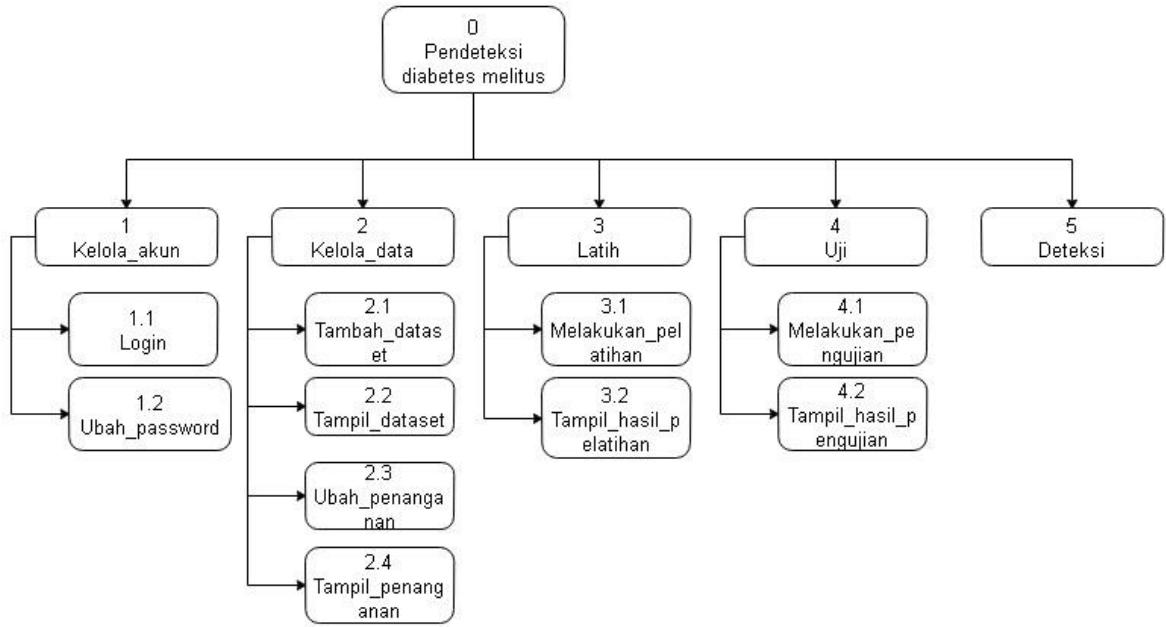
Gambar 3. 7. ERD Sistem Pendeksi Diabetes Melitus

3.9.4. Pemodelan Fungsional Sistem Pendeksi Diabetes Melitus

Model analisis yang digunakan untuk mendeskripsikan pemodelan fungsional sistem disajikan menggunakan Diagram Dekomposisi, *Data Context Diagram* dan *Data Flow Diagram*.

3.9.4.1. Diagram Dekomposisi Sistem Pendeksi Diabetes Melitus

Sistem Pendeksi Diabetes Melitus terbagi dalam 2 level proses. Pada level 1 didefinisikan 5 proses utama yaitu Kelola_akun, Kelola_data, Latih, Uji, Deteksi. Pada level 2, masing-masing proses didekomposisi menjadi sub-proses yang lebih detail. Penjabaran dekomposisi proses-proses dalam sistem dapat dilihat pada diagram dekomposisi sistem pada gambar 3.8.



Gambar 3. 8. Diagram Dekomposisi Sistem Pendekripsi Diabetes Melitus

3.9.4.2. Data Context Diagram Sistem Pendekripsi Diabetes Melitus

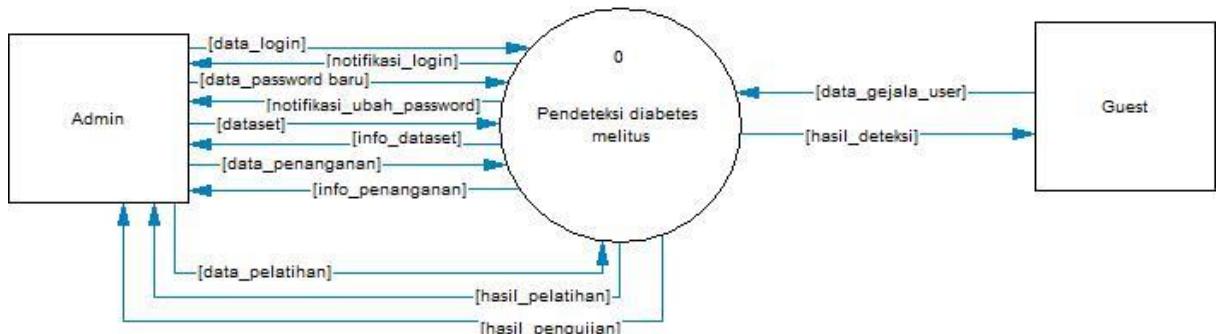
Data Context Diagram (DCD) menunjukkan proses aliran data dari entitas eksternal ke sistem ataupun sebaliknya. Terdapat dua entitas eksternal yaitu admin dan *guest*.

a. Admin

Entitas admin merupakan pengguna sistem yang memiliki wewenang untuk mengelola data-data yang dibutuhkan maupun data-data yang dihasilkan oleh sistem,

b. Guest

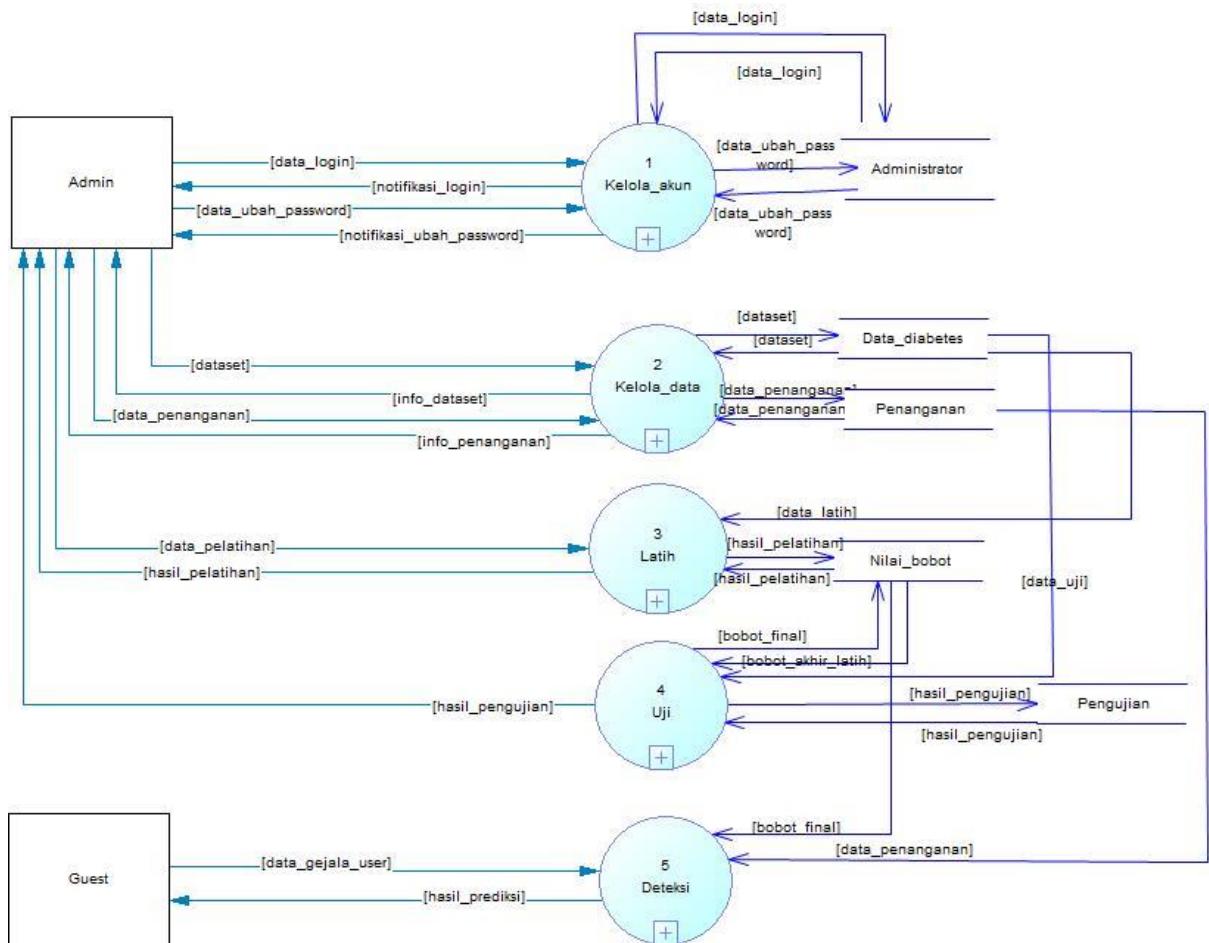
Entitas *guest* merupakan pengguna sistem yang hanya dapat melakukan fungsi deteksi pada sistem serta mendapatkan hasil dan penanganan terkait hasil deteksi. Detail pada DCD dapat dilihat pada gambar 3.9.



Gambar 3. 9. Data Context Diagram Sistem Pendekripsi Diabetes Melitus

3.9.4.3. Data Flow Diagram Level 1 Sistem Pendeksi Diabetes Melitus

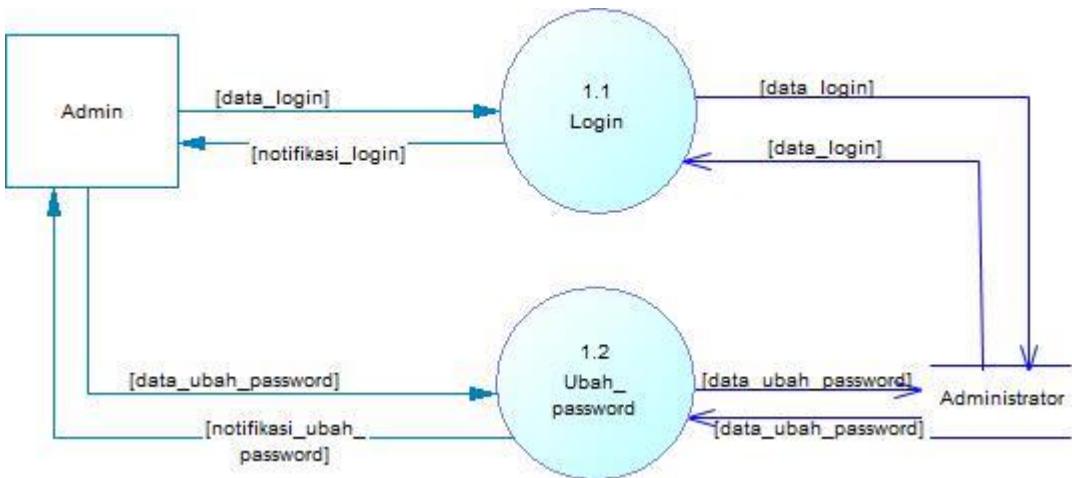
Data Flow Diagram (DFD) level 1 ini dikembangkan berdasarkan DCD yang telah dijelaskan sebelumnya. DFD level 1 ini menjelaskan aliran data dalam sistem secara lebih rinci. Terdapat 5 proses pada DFD level 1 ini, yaitu proses Kelola_akun, Kelola_data, Latih, Uji, dan Deteksi. DFD level 1 Sistem Pendeksi Diabetes Melitus dapat dilihat pada gambar 3.10.



Gambar 3. 10. DFD level 1 Sistem Pendeksi Diabetes Melitus

3.9.4.4. Data Flow Diagram Level 2 Proses Kelola_akun

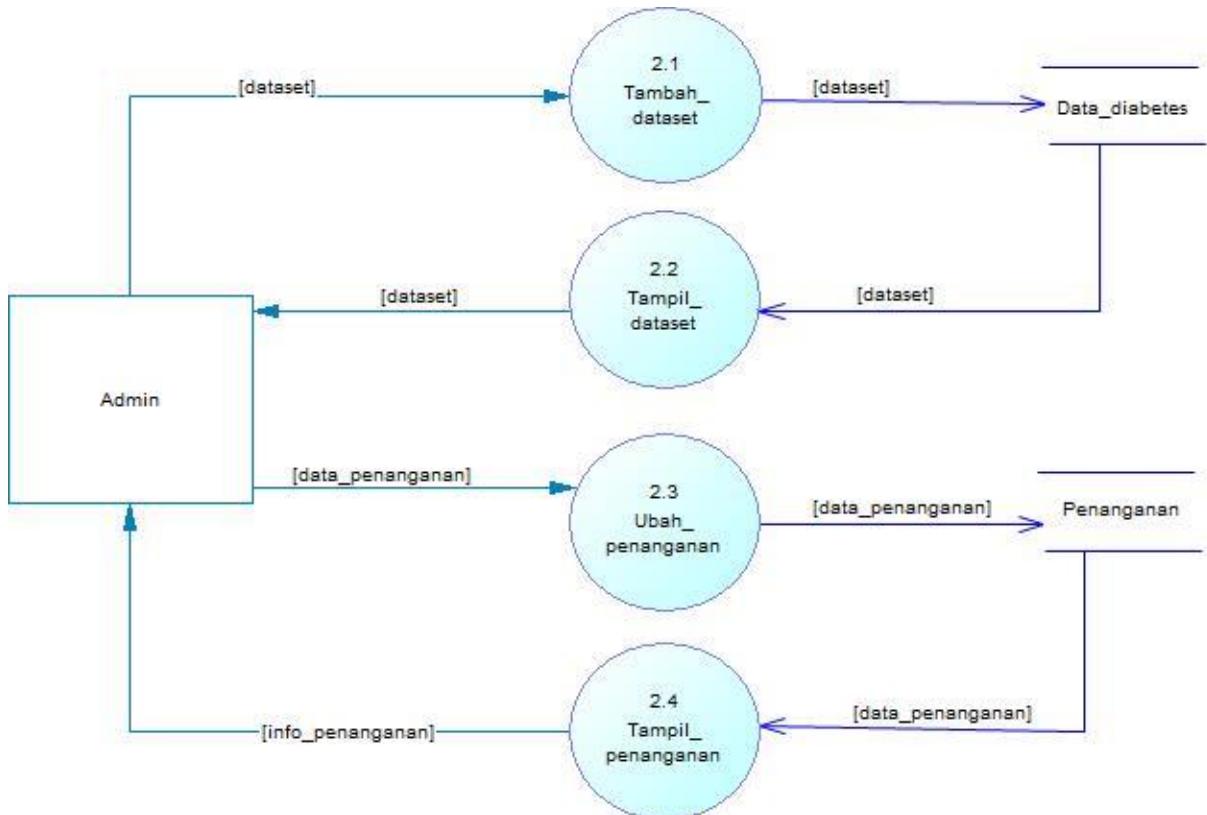
DFD level 2 proses Kelola_akun mempunyai dua sub-proses, yaitu *Login* dan *Ubah_password*. Proses ini dilakukan oleh admin. DFD level 2 proses Kelola_akun dapat dilihat pada gambar 3.11.



Gambar 3. 11. DFD Level 2 Proses Kelola_akun Sistem Pendekripsi Diabetes Melitus

3.9.4.5. *Data Flow Diagram Level 2 Proses Kelola_data*

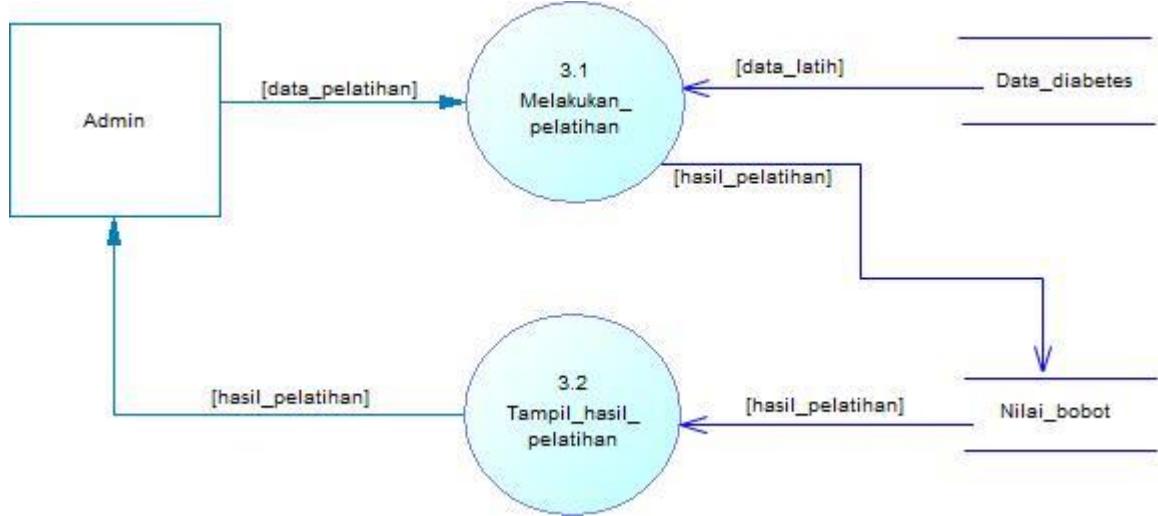
DFD level 2 proses Kelola_data mempunyai empat sub-proses, yaitu proses Tambah_dataset, Tampil_dataset, Ubah_penanganan, dan Tampil_penanganan. Proses ini dilakukan oleh admin. DFD level 2 proses Kelola_data dapat dilihat pada gambar 3.12.



Gambar 3. 12. DFD Level 2 Proses Kelola_data Sistem Pendekripsi Diabetes Melitus

3.9.4.6. Data Flow Diagram Level 2 Proses Latih

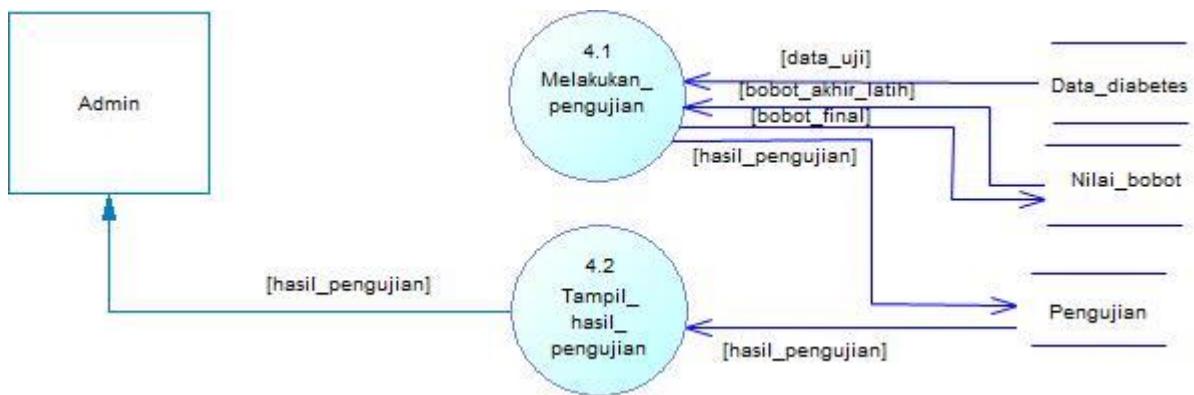
DFD level 2 proses Latih mempunyai dua sub-proses, yaitu proses Melakukan_pelatihan dan Tampil_hasil_pelatihan. Proses ini dilakukan oleh admin. DFD level 2 proses Latih dapat dilihat pada gambar 3.13.



Gambar 3. 13. DFD Level 2 Proses Latih Sistem Pendekripsi Diabetes Melitus

3.9.4.7. Data Flow Diagram Level 2 Proses Uji

DFD level 2 proses Uji mempunyai dua sub-proses, yaitu proses Melakukan_pengujian dan Tampil_hasil_pengujian. Proses ini dilakukan oleh admin. DFD level 2 proses Uji dapat dilihat pada gambar 3.14.



Gambar 3. 14. DFD Level 2 Proses Uji Sistem Pendekripsi Diabetes Melitus

3.10. Desain Sistem

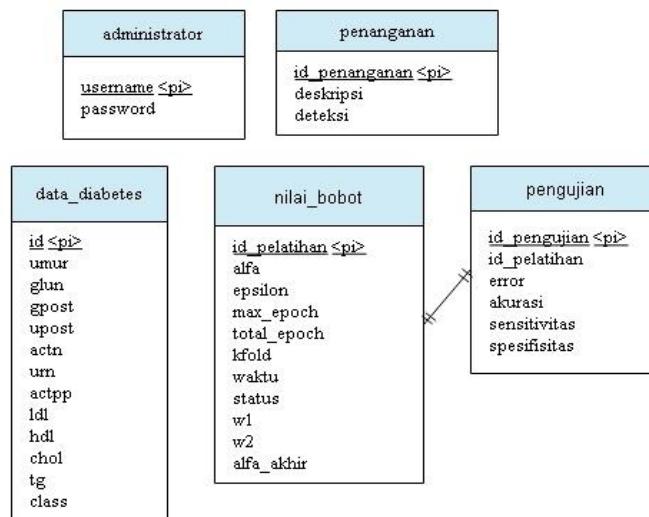
Tahap desain/perancangan ini dilakukan berdasarkan informasi yang diperoleh dari tahap analisis. Aktivitas desain sistem yang dilakukan meliputi struktur data, desain antarmuka dan desain fungsi.

3.10.1. Desain Struktur Data

Desain atau perancangan struktur data meliputi level pemodelan yaitu *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM).

1. Conceptual Data Model

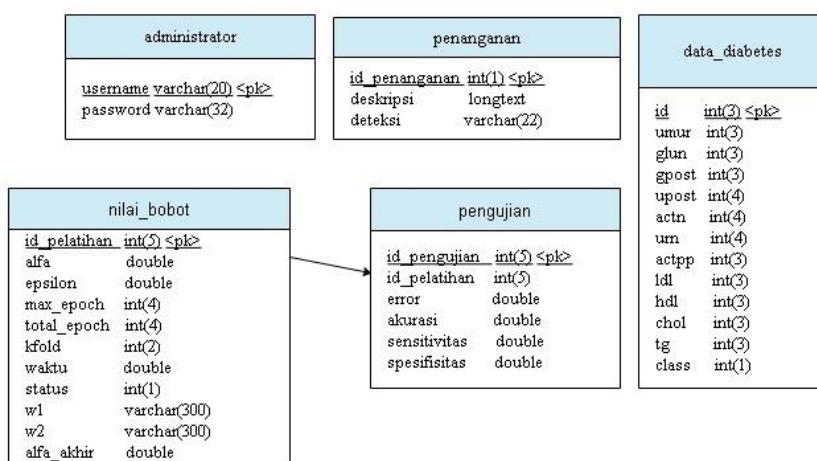
CDM merupakan model data dalam bentuk konseptual yang berisi hubungan antar obyek data dalam sistem yang dibangun. CDM dari Sistem Pendeksi Diabetes Melitus ditunjukkan pada gambar 3.15.



Gambar 3. 15. Perancangan CDM Sistem Pendeksi Diabetes Melitus

2. Physical Data Model

PDM merupakan hasil pemetaan dari CDM, merupakan model data yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antar data. PDM dari Sistem Pendeksi Diabetes Melitus ditunjukkan pada gambar 3.16.



Gambar 3. 16. Perancangan PDM Sistem Pendeksi Diabetes Melitus

3.10.2. Desain Antarmuka

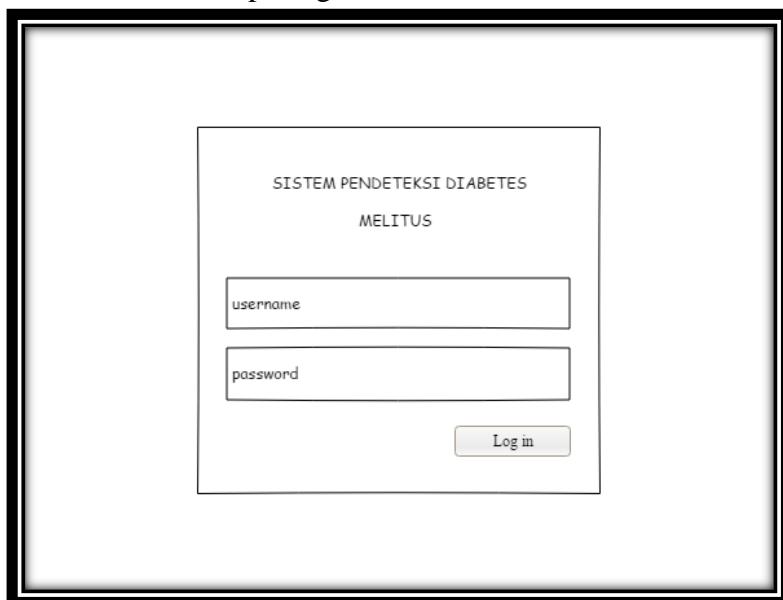
Perancangan antarmuka meliputi perancangan bentuk atau tampilan dari sistem. Perancangan antarmuka ditampilkan dalam bentuk gambar sketsa yang dapat dilihat berturut-turut pada gambar 3.17 sampai 3.38.

1. Antarmuka Login Admin

ID Antarmuka : AM – SPDM – 01

Antarmuka : *Login Admin*

Deskripsi : Admin dapat mengakses sistem dengan cara melakukan *login* terlebih dahulu. Pada proses ini admin harus menginputkan *username* dan *password*. Desain antarmuka proses ini dapat dilihat pada gambar 3.17.



Gambar 3. 17. Desain Antarmuka *Login Admin*

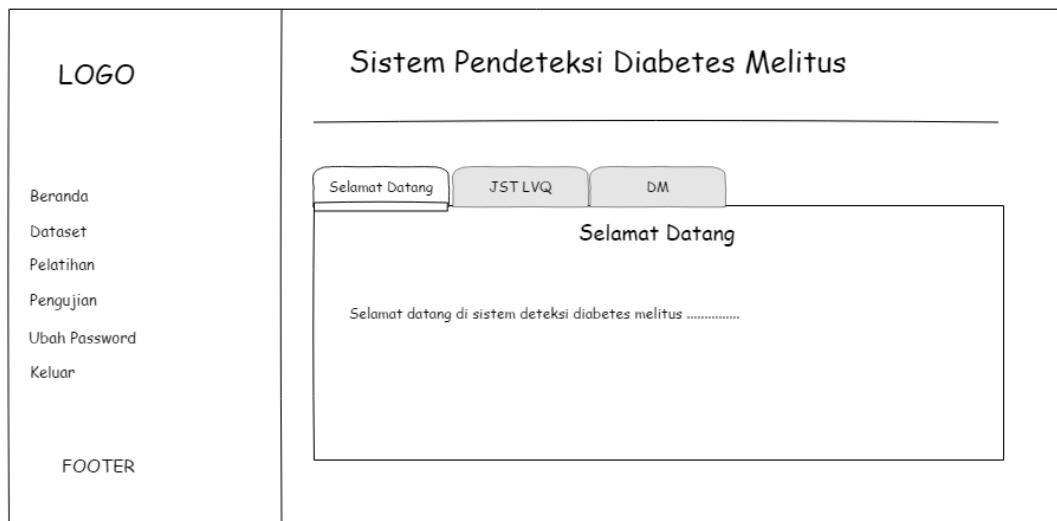
2. Antarmuka Beranda Admin

ID Antarmuka : AM – SPDM – 02

Antarmuka : Beranda Admin

Deskripsi : Halaman Beranda Admin dapat diakses admin setelah melakukan *login*. Halaman ini memiliki beberapa menu yaitu Beranda, Dataset, Pelatihan, Pengujian, Ubah *Password* dan Keluar. Pada halaman ini menampilkan informasi singkat mengenai sistem, JST LVQ dan juga penyakit diabetes

melitus. Desain antarmuka Beranda Admin dapat dilihat pada gambar 3.18.



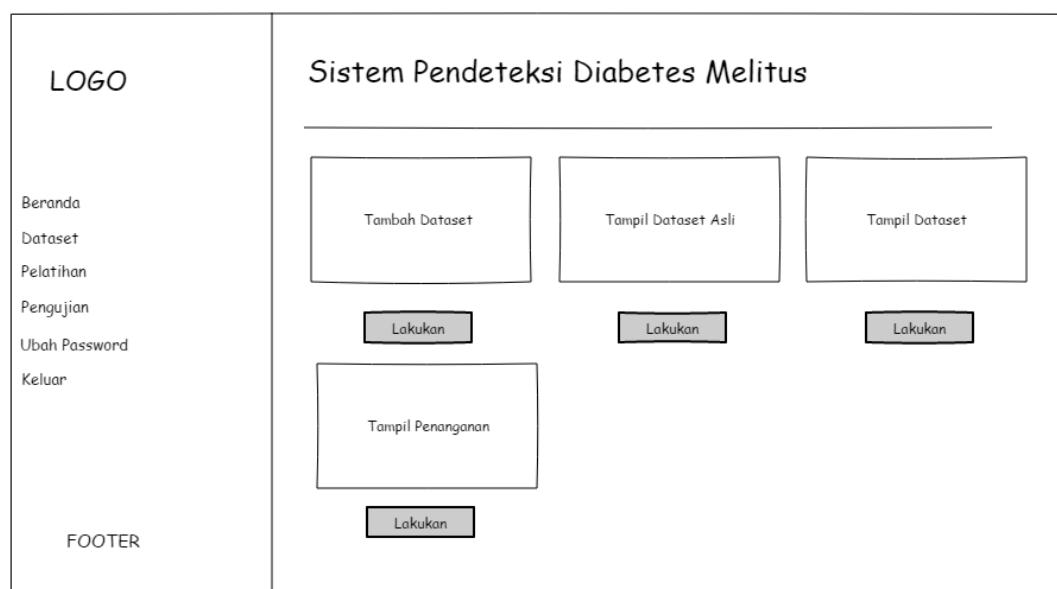
Gambar 3. 18. Desain Antarmuka Beranda Admin

3. Antarmuka Pilih Dataset

ID Antarmuka : AM – SPDM – 03

Antarmuka : Pilih Dataset

Deskripsi : Halaman ini memiliki beberapa menu yaitu Tambah Dataset, Tampil Dataset Asli, Tampil Dataset, dan Tampil Penanganan. Desain antarmuka Pilih Dataset dapat dilihat pada gambar 3.19.



Gambar 3. 19. Desain Antarmuka Pilih Dataset

4. Antarmuka Tambah Dataset

ID Antarmuka : AM – SPDM – 04

Antarmuka : Tambah Dataset

Deskripsi : Halaman ini digunakan untuk melakukan penambahan dataset.

Desain antarmuka Tambah Dataset dapat dilihat pada gambar 3.20.

LOGO Beranda Dataset Pelatihan Pengujian Ubah Password Keluar FOOTER	<h3>Sistem Pendekripsi Diabetes Melitus</h3> <hr/> <p>Tambah Dataset</p> <p>Upload File</p> <p><input type="button" value="Ubah File"/> <input type="file"/></p> <p><input type="button" value="Upload"/> <input type="button" value="Batal"/></p>
---	--

Gambar 3. 20. Desain Antarmuka Tambah Dataset

5. Antarmuka Tampil Dataset Asli

ID Antarmuka : AM – SPDM – 05

Antarmuka : Tampil Dataset Asli

Deskripsi : Halaman ini digunakan untuk melihat dataset asli yang belum dilakukan normalisasi. Desain antarmuka Tampil Dataset Asli dapat dilihat pada gambar 3.21.

Umur	Glun	Gpost	Upost	Actn	Urn	Actpp	Ldl	Hdl	Chol	Tg	Class

Gambar 3. 21. Desain Antarmuka Tampil Dataset Asli

6. Antarmuka Tampil Dataset

ID Antarmuka : AM – SPDM – 06

Antarmuka : Tampil Dataset

Deskripsi : Halaman ini digunakan untuk melihat dataset yang telah dilakukan normalisasi. Desain antarmuka Tampil Dataset dapat dilihat pada gambar 3.22.

<p>LOGO</p> <p>Beranda Dataset Pelatihan Pengujian Ubah Password Keluar</p> <p>FOOTER</p>	<p>Sistem Pendekripsi Diabetes Melitus</p> <hr/> <p>Tampil Dataset</p> <p>Show : <input type="button" value="10"/> Entries Search : <input type="text"/></p> <table border="1"><thead><tr><th>Umur</th><th>Glu</th><th>Gpost</th><th>Upost</th><th>Actn</th><th>Urn</th><th>Actpp</th><th>Ldl</th><th>Hdl</th><th>Chol</th><th>Tg</th><th>Class</th></tr></thead><tbody><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></tbody></table> <p><input type="button" value="Export"/></p>	Umur	Glu	Gpost	Upost	Actn	Urn	Actpp	Ldl	Hdl	Chol	Tg	Class												
Umur	Glu	Gpost	Upost	Actn	Urn	Actpp	Ldl	Hdl	Chol	Tg	Class														

Gambar 3. 22. Desain Antarmuka Tampil Dataset

7. Antarmuka Tampil Penanganan

ID Antarmuka : AM – SPDM – 07

Antarmuka : Tampil Penanganan

Deskripsi : Halaman ini berguna untuk melihat daftar penanganan yang akan ditampilkan setelah melakukan proses deteksi. Desain antarmuka Tampil Penanganan dapat dilihat pada gambar 3.23.

<p>LOGO</p> <p>Beranda Dataset Pelatihan Pengujian Ubah Password Keluar</p> <p>FOOTER</p>	<p>Sistem Pendekripsi Diabetes Melitus</p> <hr/> <p>Tampil Penanganan</p> <p>Show : <input type="button" value="10"/> Entries Search : <input type="text"/></p> <table border="1"><thead><tr><th>No.</th><th>Deteksi</th><th>Deskripsi</th><th>Aksi</th></tr></thead><tbody><tr><td>1</td><td>XXX</td><td>XXX</td><td>Ubah Penanganan</td></tr><tr><td>2</td><td>YYY</td><td>YYY</td><td>Ubah Penanganan</td></tr></tbody></table>	No.	Deteksi	Deskripsi	Aksi	1	XXX	XXX	Ubah Penanganan	2	YYY	YYY	Ubah Penanganan
No.	Deteksi	Deskripsi	Aksi										
1	XXX	XXX	Ubah Penanganan										
2	YYY	YYY	Ubah Penanganan										

Gambar 3. 23. Desain Antarmuka Tampil Penanganan

8. Antarmuka Ubah Penanganan

ID Antarmuka : AM – SPDM – 08
Antarmuka : Ubah Penanganan
Deskripsi : Halaman ini digunakan untuk mengubah penanganan yang akan ditampilkan setelah melakukan proses deteksi. Desain antarmuka Ubah Penanganan dapat dilihat pada gambar 3.24.

<p>LOGO</p> <p>Beranda Dataset Pelatihan Pengujian Ubah Password Keluar</p> <p>FOOTER</p>	<p>Sistem Pendekripsi Diabetes Melitus</p> <hr/> <p>Ubah Penanganan</p> <p>Deteksi :</p> <input type="text"/> <p>Deskripsi :</p> <input type="text"/> <p>[Simpan] [Batal]</p>
---	---

Gambar 3. 24. Desain Antarmuka Ubah Penanganan

9. Antarmuka Pilih Pelatihan

ID Antarmuka : AM – SPDM – 09
Antarmuka : Pilih Pelatihan
Deskripsi : Halaman ini memiliki beberapa menu terkait proses pelatihan, yaitu Mulai Pelatihan dan Tampil Data Pelatihan. Desain antarmuka Pilih Pelatihan dapat dilihat pada gambar 3.25.

<p>LOGO</p> <p>Beranda Dataset Pelatihan Pengujian Ubah Password Keluar</p> <p>FOOTER</p>	<p>Sistem Pendekripsi Diabetes Melitus</p> <hr/> <p>Mulai Pelatihan Tampil Data Pelatihan</p> <p>[Lakukan] [Lakukan]</p>
---	---

Gambar 3. 25. Desain Antarmuka Pilih Pelatihan

10. Antarmuka Mulai Pelatihan

- ID Antarmuka : AM – SPDM – 10
Antarmuka : Mulai Pelatihan
Deskripsi : Halaman ini digunakan admin untuk melakukan proses pelatihan dengan menginputkan beberapa parameter pelatihan seperti *alfa*, *epsilon*, dan *max epoch*. Desain antarmuka Mulai Pelatihan dapat dilihat pada gambar 3.26.

<p>LOGO</p> <p>Beranda Dataset Pelatihan Pengujian Ubah Password Keluar</p> <p>FOOTER</p>	<p>Sistem Pendekripsi Diabetes Melitus</p> <hr/> <p>Mulai Pelatihan</p> <p>Nilai Alfa :</p> <input type="text"/> <p>Nilai Epsilon :</p> <input type="text"/> <p>Maksimal Epoch :</p> <input type="text"/> <p>Simpan Batal</p>
---	--

Gambar 3. 26. Desain Antarmuka Mulai Pelatihan

11. Antarmuka Tampil Data Pelatihan

- ID Antarmuka : AM – SPDM – 11
Antarmuka : Tampil Data Pelatihan
Deskripsi : Halaman ini digunakan untuk menampilkan hasil pelatihan yang telah dilakukan secara umum. Desain antarmuka Tampil Data Pelatihan dapat dilihat pada gambar 3.27.

<p>LOGO</p> <p>Beranda Dataset Pelatihan Pengujian Ubah Password Keluar</p> <p>FOOTER</p>	<p>Sistem Pendekripsi Diabetes Melitus</p> <hr/> <p>Tampil Data Pelatihan</p> <p>Show : <input type="text" value="10"/> Entries Search : <input type="text"/></p> <table border="1" style="width: 100%; border-collapse: collapse;"><thead><tr><th>No.</th><th>Alfa</th><th>Epsilon</th><th>Epoch</th><th>Aksi</th></tr></thead><tbody><tr><td>1</td><td>XXX</td><td>XXX</td><td>XXX</td><td>Lihat Detail Lakukan Pengujian Hapus</td></tr><tr><td>2</td><td>YYY</td><td>YYY</td><td>YYY</td><td>Lihat Detail Lakukan Pengujian Hapus</td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td></tr></tbody></table> <p>Hapus semua pelatihan</p>	No.	Alfa	Epsilon	Epoch	Aksi	1	XXX	XXX	XXX	Lihat Detail Lakukan Pengujian Hapus	2	YYY	YYY	YYY	Lihat Detail Lakukan Pengujian Hapus																														
No.	Alfa	Epsilon	Epoch	Aksi																																										
1	XXX	XXX	XXX	Lihat Detail Lakukan Pengujian Hapus																																										
2	YYY	YYY	YYY	Lihat Detail Lakukan Pengujian Hapus																																										

Gambar 3. 27. Desain Antarmuka Tampil Data Pelatihan

12. Antarmuka Tampil Detail Data Pelatihan

ID Antarmuka : AM – SPDM – 12

Antarmuka : Tampil Detail Data Pelatihan

Deskripsi : Halaman ini berfungsi untuk menampilkan hasil pelatihan secara lengkap dan detail. Desain antarmuka Tampil Detail Data Pelatihan dapat dilihat pada gambar 3.28.

No.	KFold	Alfa	Epsilon	Epoch	Bobot Akhir
1	XXX	XXX	XXX	XXX	w1:.... , w2:.....
2	YYY	YYY	YYY	YYY	w1:.... , w2:.....

Gambar 3. 28. Desain Antarmuka Tampil Detail Data Pelatihan

13. Antarmuka Pilih Pengujian

ID Antarmuka : AM – SPDM – 13

Antarmuka : Pilih Pengujian

Deskripsi : Halaman ini berisi beberapa menu terkait dengan proses pengujian yaitu Mulai Pengujian, Tampil Data Pengujian, dan Tampil Bobot Final. Desain antarmuka Pilih Pengujian dapat dilihat pada gambar 3.29.

Mulai Pengujian

Tampil Data Pengujian

Tampil Bobot Final

Lakukan

Lakukan

Lakukan

Gambar 3. 29. Desain Antarmuka Pilih Pengujian

14. Antarmuka Mulai Pengujian

- ID Antarmuka : AM – SPDM – 14
 Antarmuka : Mulai Pengujian
 Deskripsi : Halaman ini berfungsi untuk melakukan pengujian dari data pelatihan yang telah dilakukan sebelumnya. Melakukan pengujian juga dapat dilakukan melalui antarmuka Tampil Data Pelatihan. Desain antarmuka Mulai Pengujian dapat dilihat pada gambar 3.30.

LOGO Beranda Dataset Pelatihan Pengujian Ubah Password Keluar FOOTER	<p align="center">Sistem Pendekripsi Diabetes Melitus</p> <hr/> <p align="center">Mulai Pengujian</p> <p align="center">Show : 10 Entries Search : <input type="text"/></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #cccccc;"> <th>No.</th> <th>Alfa</th> <th>Epsilon</th> <th>Epoch</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>XXX</td> <td>XXX</td> <td>XXX</td> <td>Lihat Detail Lakukan Pengujian</td> </tr> <tr> <td>2</td> <td>YYY</td> <td>YYY</td> <td>YYY</td> <td>Lihat Detail Lakukan Pengujian</td> </tr> <tr><td colspan="5" style="height: 50px;"></td></tr> </tbody> </table> <p align="center">Hapus semua pelatihan</p>	No.	Alfa	Epsilon	Epoch	Aksi	1	XXX	XXX	XXX	Lihat Detail Lakukan Pengujian	2	YYY	YYY	YYY	Lihat Detail Lakukan Pengujian																																			
No.	Alfa	Epsilon	Epoch	Aksi																																															
1	XXX	XXX	XXX	Lihat Detail Lakukan Pengujian																																															
2	YYY	YYY	YYY	Lihat Detail Lakukan Pengujian																																															

Gambar 3. 30. Desain Antarmuka Mulai Pengujian

15. Antarmuka Tampil Data Pengujian

- ID Antarmuka : AM – SPDM – 15
 Antarmuka : Tampil Data Pengujian
 Deskripsi : Halaman ini berguna untuk menampilkan hasil dari pengujian yang telah dilakukan secara umum. Desain antarmuka Tampil Data Pengujian dapat dilihat pada gambar 3.31.

LOGO Beranda Dataset Pelatihan Pengujian Ubah Password Keluar FOOTER	<p align="center">Sistem Pendekripsi Diabetes Melitus</p> <hr/> <p align="center">Tampil Data Pengujian</p> <p align="center">Show : 10 Entries Search : <input type="text"/></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #cccccc;"> <th>No.</th> <th>Alfa</th> <th>Epsilon</th> <th>Epoch</th> <th>Akurasni</th> <th>Error</th> <th>Sensitivitas</th> <th>Spesifisitas</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>XXX</td> <td>XXX</td> <td>XXX</td> <td>XXX</td> <td>XXX</td> <td>XXX</td> <td>XXX</td> <td>Lihat Detail Hapus</td> </tr> <tr> <td>2</td> <td>YYY</td> <td>YYY</td> <td>YYY</td> <td>YYY</td> <td>YYY</td> <td>YYY</td> <td>YYY</td> <td>Lihat Detail Hapus</td> </tr> <tr><td colspan="9" style="height: 50px;"></td></tr> </tbody> </table> <p align="center">First Prev 1 Next Last</p> <p align="center">Hapus semua pengujian</p>	No.	Alfa	Epsilon	Epoch	Akurasni	Error	Sensitivitas	Spesifisitas	Aksi	1	XXX	Lihat Detail Hapus	2	YYY	Lihat Detail Hapus																																																																											
No.	Alfa	Epsilon	Epoch	Akurasni	Error	Sensitivitas	Spesifisitas	Aksi																																																																																			
1	XXX	XXX	XXX	XXX	XXX	XXX	XXX	Lihat Detail Hapus																																																																																			
2	YYY	YYY	YYY	YYY	YYY	YYY	YYY	Lihat Detail Hapus																																																																																			

Gambar 3. 31. Desain Antarmuka Tampil Data Pengujian

16. Antarmuka Tampil Detail Data Pengujian

ID Antarmuka : AM – SPDM – 16
Antarmuka : Tampil Detail Data Pengujian
Deskripsi : Halaman ini berguna untuk menampilkan hasil dari pengujian yang telah dilakukan secara lengkap dan detail. Desain antarmuka Tampil Detail Data Pengujian dapat dilihat pada gambar 3.32.

LOGO Beranda Dataset Pelatihan Pengujian Ubah Password Keluar FOOTER	<p style="text-align: center;">Sistem Pendekripsi Diabetes Melitus</p> <hr/> <p style="text-align: center;">Tampil Detail Data Pengujian</p> <table style="width: 100%; border-collapse: collapse;"><tr><td style="width: 10%;">Show :</td><td style="width: 10%; text-align: center;"><input type="button" value="10"/>▼</td><td style="width: 10%;">Entries</td><td style="width: 10%; text-align: right;">Search : <input type="text"/></td><td style="width: 10%;"></td></tr><tr><th>No.</th><th>KFold</th><th>Alfa</th><th>Epsilon</th><th>Bobot Akhir</th><th>Akurasi</th><th>Error</th><th>Sensitifitas</th><th>Spesifisitas</th><th>Aksi</th></tr><tr><td>1</td><td>XXX</td><td>XXX</td><td>XXX</td><td>w1:.... , w2:.....</td><td></td><td></td><td></td><td></td><td>Simpan Bobot</td></tr><tr><td>2</td><td>YYY</td><td>YYY</td><td>YYY</td><td>w1:.... , w2:.....</td><td></td><td></td><td></td><td></td><td>Simpan Bobot</td></tr><tr><td colspan="10" style="height: 100px;"></td></tr></table> <p style="text-align: right; margin-top: -10px;">First Prev 1 Next Last</p>	Show :	<input type="button" value="10"/> ▼	Entries	Search : <input type="text"/>		No.	KFold	Alfa	Epsilon	Bobot Akhir	Akurasi	Error	Sensitifitas	Spesifisitas	Aksi	1	XXX	XXX	XXX	w1:.... , w2:.....					Simpan Bobot	2	YYY	YYY	YYY	w1:.... , w2:.....					Simpan Bobot										
Show :	<input type="button" value="10"/> ▼	Entries	Search : <input type="text"/>																																											
No.	KFold	Alfa	Epsilon	Bobot Akhir	Akurasi	Error	Sensitifitas	Spesifisitas	Aksi																																					
1	XXX	XXX	XXX	w1:.... , w2:.....					Simpan Bobot																																					
2	YYY	YYY	YYY	w1:.... , w2:.....					Simpan Bobot																																					

Gambar 3. 32. Desain Antarmuka Tampil Detail Data Pengujian

17. Antarmuka Tampil Bobot Final

ID Antarmuka : AM – SPDM – 17
Antarmuka : Tampil Bobot Final
Deskripsi : Halaman ini berguna untuk menampilkan bobot final yang telah dipilih untuk digunakan pada proses deteksi. Desain antarmuka Tampil Bobot Final dapat dilihat pada gambar 3.33.

LOGO Beranda Dataset Pelatihan Pengujian Ubah Password Keluar FOOTER	<p style="text-align: center;">Sistem Pendekripsi Diabetes Melitus</p> <hr/> <p style="text-align: center;">Tampil Bobot Final</p> <table style="width: 100%; border-collapse: collapse;"><tr><td style="width: 10%;">Show :</td><td style="width: 10%; text-align: center;"><input type="button" value="10"/>▼</td><td style="width: 10%;">Entries</td><td style="width: 10%; text-align: right;">Search : <input type="text"/></td><td style="width: 10%;"></td></tr><tr><th>No.</th><th>KFold</th><th>Alfa</th><th>Epsilon</th><th>Epoch</th><th>Bobot Akhir</th><th>Akurasi</th><th>Error</th><th>Sensitifitas</th><th>Spesifisitas</th></tr><tr><td>1</td><td>XXX</td><td>XXX</td><td>XXX</td><td></td><td>w1:.... , w2:.....</td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td>YYY</td><td>YYY</td><td>YYY</td><td></td><td>w1:.... , w2:.....</td><td></td><td></td><td></td><td></td></tr><tr><td colspan="10" style="height: 100px;"></td></tr></table> <p style="text-align: right; margin-top: -10px;">First Prev 1 Next Last</p>	Show :	<input type="button" value="10"/> ▼	Entries	Search : <input type="text"/>		No.	KFold	Alfa	Epsilon	Epoch	Bobot Akhir	Akurasi	Error	Sensitifitas	Spesifisitas	1	XXX	XXX	XXX		w1:.... , w2:.....					2	YYY	YYY	YYY		w1:.... , w2:.....														
Show :	<input type="button" value="10"/> ▼	Entries	Search : <input type="text"/>																																											
No.	KFold	Alfa	Epsilon	Epoch	Bobot Akhir	Akurasi	Error	Sensitifitas	Spesifisitas																																					
1	XXX	XXX	XXX		w1:.... , w2:.....																																									
2	YYY	YYY	YYY		w1:.... , w2:.....																																									

Gambar 3. 33. Desain Antarmuka Tampil Bobot Final

18. Antarmuka Ubah Password

ID Antarmuka : AM – SPDM – 18
Antarmuka : Ubah Password
Deskripsi : Halaman ini berguna bagi admin untuk dapat mengubah password mereka. Desain antarmuka Ubah Password dapat dilihat pada gambar 3.34.

<p>LOGO</p> <p>Beranda Dataset Pelatihan Pengujian Ubah Password Keluar</p> <p>FOOTER</p>	<p>Sistem Pendekripsi Diabetes Melitus</p> <hr/> <p>Ubah Password</p> <p>Password Lama :</p> <input type="text"/> <p>Password Baru :</p> <input type="text"/> <p>Konfirmasi Password :</p> <input type="text"/> <p style="text-align: center;"><input type="button" value="Simpan"/> <input type="button" value="Batal"/></p>
---	---

Gambar 3. 34. Desain Antarmuka Ubah Password

19. Antarmuka Beranda Guest

ID Antarmuka : AM – SPDM – 19
Antarmuka : Beranda Guest
Deskripsi : Halaman Beranda Guest merupakan halaman utama untuk guest. Halaman ini memiliki beberapa menu yaitu Beranda, Mulai Deteksi dan Pengembang. Pada halaman ini menampilkan informasi singkat mengenai sistem, JST LVQ dan juga penyakit diabetes melitus. Desain antarmuka Beranda Guest dapat dilihat pada gambar 3.35.

<p>LOGO</p> <p>Beranda Mulai Deteksi Pengembang</p> <p>FOOTER</p>	<p>Sistem Pendekripsi Diabetes Melitus</p> <hr/> <p style="text-align: center;"><input type="button" value="Selamat Datang"/> <input type="button" value="JST LVQ"/> <input type="button" value="DM"/></p> <p style="text-align: center;">Selamat Datang</p> <p style="text-align: center;">Selamat datang di sistem deteksi diabetes melitus</p>
---	--

Gambar 3. 35. Desain Antarmuka Beranda Guest

20. Antarmuka Mulai Deteksi

ID Antarmuka : AM – SPDM – 20
Antarmuka : Mulai Deteksi
Deskripsi : Halaman ini digunakan *guest* untuk melakukan deteksi diabetes melitus. Pada halaman ini *guest* harus menginputkan beberapa atribut yang dibutuhkan untuk melakukan deteksi diabetes melitus. Desain antarmuka Mulai Deteksi dapat dilihat pada gambar 3.36.

LOGO Beranda Mulai Deteksi Pengembang FOOTER	<p>Sistem Pendeksi Diabetes Melitus</p> <hr/> <p>Mulai Deteksi</p> <p>Umur : <input type="text"/></p> <p>Glu : <input type="text"/></p> <p>Gpost : <input type="text"/></p> <p>⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋮</p> <p>Tg : <input type="text"/></p> <p style="text-align: center;"><input type="button" value="Simpan"/> <input type="button" value="Batal"/></p>
--	---

Gambar 3. 36. Desain Antarmuka Mulai Deteksi

21. Antarmuka Hasil Deteksi

ID Antarmuka : AM – SPDM – 21
Antarmuka : Hasil Deteksi
Deskripsi : Halaman ini menampilkan hasil deteksi apakah pengguna positif diabetes melitus atau tidak. Desain antarmuka Hasil Deteksi dapat dilihat pada gambar 3.37.

LOGO Beranda Mulai Deteksi Keluar FOOTER	<p>Sistem Pendeksi Diabetes Melitus</p> <hr/> <p>Hasil Deteksi</p> <p style="text-align: center;">Tidak Diabetes Melitus</p> <p>Gejala yang anda rasakan tidak mengarah ke diabetes melitus. Tetap jaga pola hidup sehat, olahraga dan makan secara baik dan teratur.....</p>
--	--

Gambar 3. 37. Desain Antarmuka Hasil Deteksi

22. Antarmuka Pengembang

- ID Antarmuka : AM – SPDM – 22
Antarmuka : Pengembang
Deskripsi : Halaman ini berisi informasi mengenai latar belakang pengembang dan juga sistem. Desain antarmuka Pengembang dapat dilihat pada gambar 3.38.



Gambar 3. 38. Desain Antarmuka Pengembang

3.10.3. Desain Fungsi

Perancangan fungsi meliputi spesifikasi proses yang digunakan untuk membuat sistem. Perancangan fungsi dibuat berdasarkan pemodelan fungsional yang telah dihasilkan pada tahap analisis sistem. Perancangan fungsi disajikan dalam bentuk *pseudocode*.

1. Fungsi Log In

- ID Fungsi : FS – SPDM – 1.1
Nama Fungsi : Log In
Deskripsi : Melakukan validasi admin untuk mendapatkan hak akses memasuki sistem

Initial State (IS) :
Database administrator ready Admin pada AM-SPDM-01
Final State (FS) :
Otorisasi akses diberikan
Spesifikasi Proses/ Algoritma :

```

Input(username, password) at AM-SPDM-01
Q1 : SELECT * FROM `administrator`
      WHERE administrator.username=username and
            administrator.password=password

      IF (Q1 == 1) THEN
          Otorisasi akses diberikan
      ELSE
          Print 'username or password invalid' at AM-SPDM-01
      END IF

```

2. Fungsi Mengubah Password

ID Fungsi : FS – SPDM – 1.2

Nama Fungsi : Mengubah Password

Deskripsi : Mengubah *password* admin

Initial State (IS) :

Database administrator ready
Admin pada AM-SPDM-18

Final State (FS) :

Password berubah
Database administrator terupdate

Spesifikasi Proses/ Algoritma :

```

Get(username)
Input(password_lama, password_baru,
       konfirmasi_password_baru) at AM-SPDM-18
IF (password_baru = password_lama) THEN
    Q1 : SELECT *
          FROM `administrator`
          WHERE administrator.username=username and
                password=password_lama
    IF (Q1 >= 1) THEN
        Q2 : UPDATE `administrator`
              SET administrator.password=password_baru
              WHERE administrator.username=username
    ELSE
        Print 'password lama salah' at AM-SPDM-18
    END IF
ELSE
    Print 'password baru yang dimasukkan tidak cocok' at
          AM-SPDM-18
END IF

```

3. Fungsi Tambah Dataset

ID Fungsi : FS – SPDM – 2.1
Nama Fungsi : Tambah Dataset
Deskripsi : Menambah dataset ke sistem

Initial State (IS) :

Database data_diabetes ready
Admin pada AM-SPDM-04

Final State (FS) :

Dataset hasil tes berhasil dimasukkan
Database data_diabetes terupdate

Spesifikasi Proses/ Algoritma :

```
Input(dataset) at AM-SKK-04
INSERT INTO `data_diabetes`
    (data_diabetes.umur, data_diabetes.glun,
     data_diabetes.gpost, data_diabetes.upost,
     data_diabetes.actn, data_diabetes.urn,
     data_diabetes.actpp, data_diabetes.ldl, data_diabetes.hdl,
     data_diabetes.chol, data_diabetes.tg, data_diabetes.class)
VALUES (umur, glun, gpost, upost, actn, urn, actpp, ldl,
        hdl, chol, tg, class)
```

4. Fungsi Tampil Dataset

ID Fungsi : FS – SPDM – 2.2
Nama Fungsi : Tampil Dataset
Deskripsi : Menampilkan dataset yang telah diinputkan admin ke sistem

Initial State (IS) :

Database data_diabetes ready
Admin pada AM-SPDM-06

Final State (FS) :

Dataset ditampilkan

Spesifikasi Proses/ Algoritma :

```
Q1 : SELECT * FROM `data_diabetes`  
Print(Q1) at AM-SPDM-06
```

5. Fungsi Ubah Penanganan

ID Fungsi : FS – SPDM – 2.3

Nama Fungsi : Ubah Penanganan

Deskripsi : Mengubah data penanganan yang telah diinputkan admin ke sistem.

Initial State (IS) :

Database penanganan ready
Admin pada AM-SPDM-08

Final State (FS) :

Database penanganan terupdate

Spesifikasi Proses/ Algoritma :

```
Get(id)
Input(deteksi, deskripsi) at AM-SPDM-08
Q1 : SELECT *
      FROM `penanganan`
      WHERE penanganan.id_penanganan=id
IF (Q1 >= 1) THEN
    Q2 : UPDATE `penanganan`
          SET penanganan.deteksi=deteksi,
              penanganan.deskripsi=deskripsi
          WHERE penanganan.id_penanganan=id
ELSE
    Print 'data gagal disimpan' at AM-SPDM-08
END IF
```

6. Fungsi Tampil Penanganan

ID Fungsi : FS – SPDM – 2.4

Nama Fungsi : Tampil Penanganan

Deskripsi : Menampilkan data penanganan yang telah diinputkan admin ke sistem

Initial State (IS) :

Database penanganan ready
Admin pada AM-SPDM-07

Final State (FS) :

Data penanganan ditampilkan

Spesifikasi Proses/ Algoritma :

```
Q1 : SELECT * FROM `penanganan`
```

Print(Q1) at AM-SPDM-07

7. Fungsi Melakukan Pelatihan

ID Fungsi : FS – SPDM – 3.1

Nama Fungsi : Melakukan Pelatihan

Deskripsi : Melakukan proses pelatihan dengan algoritma jaringan syaraf tiruan LVQ

Initial State (IS) :

```
Database data_diabetes & nilai_bobot ready  
Admin pada AM-SPDM-10
```

Final State (FS) :

```
Pelatihan berhasil  
Database nilai_bobot terupdate
```

Spesifikasi Proses/ Algoritma :

```
Input(alfa, epsilon, maxepoch) at AM-SPDM-10
```

```
b1 ← random(1,50)  
b2 ← random(51,100)
```

```
Q1 : SELECT umur-tg AS bobot1 FROM `data_diabetes`  
      WHERE data_diabetes.id = b1  
Q2 : SELECT umur-tg AS bobot2 FROM `data_diabetes`  
      WHERE data_diabetes.id = b2
```

```
w[1] ← bobot1(Q1)  
w[2] ← bobot2(Q2)
```

```
Q5 : SELECT umur-tg FROM `data_diabetes`  
i ← 1  
FOREACH Q5 DO  
    data[i] ← array(n1,n2,n3,...,n15)  
END FOREACH
```

```
Q6 : SELECT class FROM `data_diabetes`  
FOREACH Q6 DO  
    target[] ← class  
END FOREACH
```

```
kfold[1] ← array(6,7,8,9,10,56,57,58,59,60,  
                  11,12,13,14,15,61,62,63,64,65  
                  16,17,18,19,20,66,67,68,69,70  
                  21,22,23,24,25,71,72,73,74,75  
                  26,27,28,29,30,76,77,78,79,80  
                  31,32,33,34,35,81,82,83,84,85)
```

```

36,37,38,39,40,86,87,88,89,90
41,42,43,44,45,91,92,93,94,95
46,47,48,49,50,96,97,98,99,100)
kfold[2] ← array(1,2,3,4,5,51,52,53,54,55,
11,12,13,14,15,61,62,63,64,65
16,17,18,19,20,66,67,68,69,70
21,22,23,24,25,71,72,73,74,75
26,27,28,29,30,76,77,78,79,80
31,32,33,34,35,81,82,83,84,85
36,37,38,39,40,86,87,88,89,90
41,42,43,44,45,91,92,93,94,95
46,47,48,49,50,96,97,98,99,100)
kfold[3] ← array(1,2,3,4,5,51,52,53,54,55,
6,7,8,9,10,56,57,58,59,60,
16,17,18,19,20,66,67,68,69,70
21,22,23,24,25,71,72,73,74,75
26,27,28,29,30,76,77,78,79,80
31,32,33,34,35,81,82,83,84,85
36,37,38,39,40,86,87,88,89,90
41,42,43,44,45,91,92,93,94,95
46,47,48,49,50,96,97,98,99,100)
kfold[4] ← array(1,2,3,4,5,51,52,53,54,55,
6,7,8,9,10,56,57,58,59,60,
11,12,13,14,15,61,62,63,64,65
21,22,23,24,25,71,72,73,74,75
26,27,28,29,30,76,77,78,79,80
31,32,33,34,35,81,82,83,84,85
36,37,38,39,40,86,87,88,89,90
41,42,43,44,45,91,92,93,94,95
46,47,48,49,50,96,97,98,99,100)
kfold[5] ← array(1,2,3,4,5,51,52,53,54,55,
6,7,8,9,10,56,57,58,59,60,
11,12,13,14,15,61,62,63,64,65
16,17,18,19,20,66,67,68,69,70
26,27,28,29,30,76,77,78,79,80
31,32,33,34,35,81,82,83,84,85
36,37,38,39,40,86,87,88,89,90
41,42,43,44,45,91,92,93,94,95
46,47,48,49,50,96,97,98,99,100)
kfold[6] ← array(1,2,3,4,5,51,52,53,54,55,
6,7,8,9,10,56,57,58,59,60,
11,12,13,14,15,61,62,63,64,65
16,17,18,19,20,66,67,68,69,70
21,22,23,24,25,71,72,73,74,75
31,32,33,34,35,81,82,83,84,85
36,37,38,39,40,86,87,88,89,90
41,42,43,44,45,91,92,93,94,95
46,47,48,49,50,96,97,98,99,100)
kfold[7] ← array(1,2,3,4,5,51,52,53,54,55,
6,7,8,9,10,56,57,58,59,60,
11,12,13,14,15,61,62,63,64,65
16,17,18,19,20,66,67,68,69,70
21,22,23,24,25,71,72,73,74,75
26,27,28,29,30,76,77,78,79,80

```

```

            36,37,38,39,40,86,87,88,89,90
            41,42,43,44,45,91,92,93,94,95
            46,47,48,49,50,96,97,98,99,100)
kfold[8] ← array(1,2,3,4,5,51,52,53,54,55,
                  6,7,8,9,10,56,57,58,59,60,
                  11,12,13,14,15,61,62,63,64,65
                  16,17,18,19,20,66,67,68,69,70
                  21,22,23,24,25,71,72,73,74,75
                  26,27,28,29,30,76,77,78,79,80
                  31,32,33,34,35,81,82,83,84,85
                  41,42,43,44,45,91,92,93,94,95
                  46,47,48,49,50,96,97,98,99,100)
kfold[9] ← array(1,2,3,4,5,51,52,53,54,55,
                  6,7,8,9,10,56,57,58,59,60,
                  11,12,13,14,15,61,62,63,64,65
                  16,17,18,19,20,66,67,68,69,70
                  21,22,23,24,25,71,72,73,74,75
                  26,27,28,29,30,76,77,78,79,80
                  31,32,33,34,35,81,82,83,84,85
                  36,37,38,39,40,86,87,88,89,90
                  46,47,48,49,50,96,97,98,99,100)
kfold[10] ← array(1,2,3,4,5,51,52,53,54,55,
                   6,7,8,9,10,56,57,58,59,60,
                   11,12,13,14,15,61,62,63,64,65
                   16,17,18,19,20,66,67,68,69,70
                   21,22,23,24,25,71,72,73,74,75
                   26,27,28,29,30,76,77,78,79,80
                   31,32,33,34,35,81,82,83,84,85
                   36,37,38,39,40,86,87,88,89,90
                   41,42,43,44,45,91,92,93,94,95)

w_lama ← w
MaxEpoch ← maxepoch
alfa ← alfa
eps ← epsilon

//iterasi kfold
FOR (k_fold=1 to 10) DO
    w ← w_lama
    epoch ← 0

    //algoritma LVQ
    WHILE (epoch < MaxEpoch) DO
        epoch ← epoch + 1
        FOR (i=1 to 90) DO
            FOR (k=1 to 2) DO
                jarak[i][k] ← e_distance(data[kfold
                    [k_fold][i-1]],w[k])
            END FOR

            minim ← min(jarak[i])

```

```

        FOR (k=1 to 2) DO
            IF jarak[i][k] = minim THEN
                k_min[i] ← k
            END IF
        END FOR

        IF target[i-1] = k_min[i] THEN
            k ← k_min[i]
            FOR (j=1 to 11) DO
                w[k][j] ← w[k][j]+alpa*
                (data[kfold[k_fold][i-1]][j-1] - w[k][j])
            END FOR
        ELSE
            k ← k_min[i]
            FOR (j=1 to 11) DO
                w[k][j] ← w[k][j]-alpa*
                (data[kfold[k_fold][i-1]][j-1] - w[k][j])
            END FOR
        END IF
    END FOR

    alpa ← min_alpa(alpa)

    IF alpa < eps(a-1) THEN
        break
    END IF
END WHILE

w1 ← w[1]
w2 ← w[2]
alfa_akhir ← alfa[b-1]
eps ← eps[a-1]
time = (time_end - time_start)

INSERT INTO `nilai_bobot`
(nilai_bobot.alfa, nilai_bobot.epsilon,
nilai_bobot.max_epoch, nilai_bobot.total_epoch,
nilai_bobot.kfold, nilai_bobot.waktu,
nilai_bobot.w1, nilai_bobot.w2,
nilai_bobot.alfa_akhir)
VALUES
(alfa_awal,epsilon,MaxEpoch,epoch,kfold_ke,time,w1,
w2,alfa_akhir)

END FOR
END FOR
END FOR

FUNCTION e_distance(data,init)
kunci = array_keys(data)
seqma ← 0
FOR (i=1 to 11) DO

```

```

        seqma ← seqma + pow (data[kunci[i-1]]-init[i],2)
    END FOR
    hasil ← sqrt(seqma)
    return hasil
END FUNCTION

FUNCTION min_alpa(alpa)
    hasil ← alpa-(0.01*alpa)
    return hasil
END FUNCTION

```

8. Fungsi Tampil Data Pelatihan

ID Fungsi : FS – SPDM – 3.2

Nama Fungsi : Tampil Data Pelatihan

Deskripsi : Menampilkan data hasil pelatihan dari proses pelatihan dengan jaringan syaraf tiruan LVQ yang telah dilakukan sebelumnya

Initial State (IS) :

Database nilai_bobot ready
Admin pada AM-SPDM-11

Final State (FS) :

Data hasil pelatihan ditampilkan

Spesifikasi Proses/ Algoritma :

Q1 : SELECT * FROM `nilai_bobot` GROUP BY `alfa`, `epsilon`,
`max_epoch` ORDER BY `id_pelatihan`
Print(Q1) at AM-SPDM-11

9. Fungsi Melakukan Pengujian

ID Fungsi : FS – SPDM – 4.1

Nama Fungsi : Melakukan Pengujian

Deskripsi : Melakukan proses pengujian dengan algoritma jaringan syaraf tiruan LVQ

Initial State (IS) :

Database pengujian ready
Admin pada AM-SPDM-14

Final State (FS) :

Pengujian berhasil

Database pengujian terupdate

Spesifikasi Proses/ Algoritma :

```
Get(id)
id_pelatihan ← id
Q1 : SELECT * FROM nilai_bobot WHERE nilai_bobot.id_pelatihan =
id_pelatihan

alfa ← alfa
epsilon ← epsilon
epoch ← max_epoch

Q2 : SELECT umur-tg FROM data_diabetes
i ← 1
FOREACH Q2 DO
    data[i] ← array(umur,glun,gpost,..,tg)
END FOREACH

kfold[1] ← array(1,2,3,4,5,51,52,53,54,55)
kfold[2] ← array(6,7,8,9,10,56,57,58,59,60)
kfold[3] ← array(11,12,13,14,15,61,62,63,64,65)
kfold[4] ← array(16,17,18,19,20,66,67,68,69,70)
kfold[5] ← array(21,22,23,24,25,71,72,73,74,75)
kfold[6] ← array(26,27,28,29,30,76,77,78,79,80)
kfold[7] ← array(31,32,33,34,35,81,82,83,84,85)
kfold[8] ← array(36,37,38,39,40,86,87,88,89,90)
kfold[9] ← array(41,42,43,44,45,91,92,93,94,95)
kfold[10] ← array(46,47,48,49,50,96,97,98,99,100)

//iterasi kfold
FOR (k_fold=1 to 10) DO
    Q1 : SELECT * FROM nilai_bobot
        WHERE nilai_bobot.alfa=alfa and
        nilai_bobot.epsilon=epsilon and
        nilai_bobot.max_epoch=epoch

    w[1] ← Q1[w1]
    w[2] ← Q1[w2]

    benar ← 0
    salah ← 0
    aa ← 0
    bb ← 0
    ab ← 0
    ba ← 0
    bb ← 0
    x ← 0

    FOR (i=1 to 2)
        FOR (j=1 to 5) DO
            IF (i=1 & j=1)
```

```

x=1
IF (i=1 & j=2)
x=2
IF (i=1 & j=3)
x=3
IF (i=1 & j=4)
x=4
IF (i=1 & j=5)
x=5
IF (i=1 & j=1)
x=6
IF (i=1 & j=2)
x=7
IF (i=1 & j=3)
x=8
IF (i=1 & j=4)
x=9
IF (i=1 & j=5)
x=10

uji ← data[kfold[k_fold][i-1]]
FOR (k=1 to 2) DO
    jarak[i][k] ← e_distance(uji,w[k])
END FOR

min_jarak[i] ← min(jarak[i])

FOR (k=1 to 2) DO
    IF min_jarak = jarak[i][k] THEN
        hasil ← k
    END IF

    IF (i=1)
        IF (hasil=1)
            aa ← aa + 1
        END IF
        IF (hasil=2)
            ab ← ab + 1
        END IF
    ELSE
        IF (i=1)
            IF (hasil=1)
                ba ← ba + 1
            END IF
            IF (hasil=2)
                bb ← bb + 1
            END IF
        END IF
    END FOR

END FOR
akurasi ← (aa+bb) / (aa+ab+ba+bb)

```

```

        error ← (ab+ba) / (aa+ab+ba+bb)
        sensitivitas ← (aa) / (aa+ab)
        spesifisitas ← (bb) / (bb+ba)

        INSERT INTO
        pengujian(id_pelatihan,akurasi,error,sensitivitas,spesifisitas
        ) VALUES
        (id_pelatihan,akurasi,error,sensitivitas,spesifisitas)

    END FOR
END FOR
END FOR

FUNCTION e_distance(data,init)
    seqma ← 0
    k = array_keys(data)
    FOR (i=1 to 11) DO
        seqma ← seqma + pow (data[k[i-1]]-init[i],2)
    END FOR
    hasil ← sqrt(seqma)
    return hasil
END FUNCTION

```

10. Fungsi Tampil Hasil Pengujian

ID Fungsi : FS – SPDM – 4.2

Nama Fungsi : Tampil Hasil Pengujian

Deskripsi : Menampilkan data hasil pengujian dari proses pengujian dengan jaringan syaraf tiruan LVQ yang telah dilakukan sebelumnya

Initial State (IS) :

Database nilai_bobot & pengujian ready
Admin pada AM-SPDM-15

Final State (FS) :

Data hasil pengujian ditampilkan

Spesifikasi Proses/ Algoritma :

```

Q1 : SELECT *, AVG(akurasi) as rata, AVG(error) as errors,
AVG(sensitivitas) as sensitivitass, AVG(spesifisitas) as
spesifisitass FROM nilai_bobot, pengujian WHERE
nilai_bobot.id_pelatihan=pengujian.id_pelatihan GROUP BY
nilai_bobot.alfa,nilai_bobot.epsilon,nilai_bobot.max_epoch
ORDER BY pengujian.id_pengujian Print(Q1) at AM-SPDM-15

```

11. Fungsi Melakukan Deteksi

ID Fungsi : FS – SPDM – 5.1

Nama Fungsi : Melakukan Deteksi

Deskripsi : Melakukan deteksi diabetes melitus berdasarkan hasil data yang dimasukkan *guest*.

Initial State (IS) :

Database nilai_bobot ready
Admin pada AM-SPDM-20

Final State (FS) :

Deteksi diabetes melitus berhasil

Spesifikasi Proses/ Algoritma :

```
input(hasil tes) AS x[] at AM-SPDM-20

Q1 : SELECT * FROM nilai_bobot
      WHERE nilai_bobot.status=1

FOR (k=1 to 2) DO
    jarak[k] ← e_distance(x,w[k])
END FOR

min_jarak ← min(jarak)

FOR (k=1 to 2) DO
    IF (jarak[k] = min_jarak) THEN
        hasil ← k
    END IF
END FOR

IF hasil = 1 THEN
    hasil ← 'diabetes'
Print(hasil) at AM-SPDM-21
ELSE IF hasil = 2 THEN
    hasil ← 'tidak diabetes'
Print(hasil) at AM-SPDM-22
END IF

FUNCTION e_distance(x,w)
    seqma ← 0
    FOR (i=1 to 11) DO
        seqma ← seqma + pow (x[i]-w[i],2)
    END FOR
    hasil ← sqrt(seqma)
    return hasil
END FUNCTION
```

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi

Sistem Pendekripsi Diabetes Melitus ini dibuat berdasarkan analisis dan perancangan yang telah dibuat sebelumnya. Implementasi dari sistem ini meliputi spesifikasi perangkat, implementasi data, implementasi fungsi, dan implementasi antarmuka.

4.1.1. Spesifikasi Perangkat

Spesifikasi perangkat keras yang digunakan dalam membangun Sistem Pendekripsi Diabetes Melitus ini adalah sebagai berikut:

1. *Processor* : AMD A8-7410 APU with AMD Radeon R5 Graphics 2.20GHz.
2. RAM : 4.00 GB
3. Harddisk : 1 TB

Sedangkan perangkat lunak yang digunakan dalam membangun Sistem Pendekripsi Diabetes Melitus ini adalah sebagai berikut:

1. OS Windows 8
2. Web Server Apache 2.4
3. PHP versi 5.2
4. DBMS MySQL
5. Browser Google Chrome
6. Tools Pencil, E Draw, Power Designer
7. Notepad++.

4.1.2. Implementasi Data

Implementasi data merupakan hasil transformasi dari perancangan data menjadi *database* pada DBMS MySQL. *Database* terdiri dari 5 tabel, yaitu tabel administrator, data_diabetes, nilai_bobot, pengujian, dan penanganan. *Database* dibuat dalam DBMS MySQL dengan nama db_spdm menggunakan query:

```
CREATE DATABASE `db_spdm` /*!40100 DEFAULT CHARACTER SET latin1*/
```

Implementasi tabel dapat dilihat pada gambar 4.1 sampai gambar 4.5.

1. Tabel administrator

Tabel administrator merupakan tabel yang berisi data admin untuk dapat masuk ke dalam sistem. Pembuatan tabel administrator menggunakan *query*:

```
CREATE TABLE IF NOT EXISTS `administrator` (`username` varchar(20) NOT NULL,
`password` varchar(32) NOT NULL, PRIMARY KEY (`username`)) ENGINE=InnoDB
DEFAULT CHARSET=latin1;
```

Tampilan struktur tabel administrator dapat dilihat pada gambar 4.1.

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra
1	username	varchar(20)	latin1_swedish_ci		Tidak		Tidak ada	
2	password	varchar(32)	latin1_swedish_ci		Tidak		Tidak ada	

Gambar 4. 1. Struktur Tabel administrator pada MySQL

2. Tabel data_diabetes

Tabel data_diabetes merupakan tabel yang berisi gejala penyakit yang digunakan untuk pemrosesan data dalam sistem. Pembuatan tabel data_diabetes menggunakan *query*:

```
CREATE TABLE IF NOT EXISTS `data_diabetes` (`id` int(3) NOT NULL, `umur` int(3)
NOT NULL, `glun` int(3) NOT NULL, `gpost` int(3) NOT NULL, `upost` int(4) NOT
NULL, `actn` int(4) NOT NULL, `urn` int(4) NOT NULL, `actpp` int(3) NOT NULL,
`ldl` int(3) NOT NULL, `hdl` int(3) NOT NULL, `chol` int(3) NOT NULL, `tg`
int(3) NOT NULL, `class` int(1) NOT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB
DEFAULT CHARSET=latin1;
```

Tampilan struktur tabel data_diabetes dapat dilihat pada gambar 4.2.

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra
1	id	int(3)			Tidak		Tidak ada	AUTO_INCREMENT
2	umur	int(3)			Tidak		Tidak ada	
3	glun	int(3)			Tidak		Tidak ada	
4	gpost	int(3)			Tidak		Tidak ada	
5	upost	int(4)			Tidak		Tidak ada	
6	actn	int(4)			Tidak		Tidak ada	
7	urn	int(4)			Tidak		Tidak ada	
8	actpp	int(3)			Tidak		Tidak ada	
9	ldl	int(3)			Tidak		Tidak ada	
10	hdl	int(3)			Tidak		Tidak ada	
11	chol	int(3)			Tidak		Tidak ada	
12	tg	int(3)			Tidak		Tidak ada	
13	class	int(1)			Tidak		Tidak ada	

Gambar 4. 2. Struktur Tabel data_diabetes pada MySQL

3. Tabel nilai_bobot

Tabel nilai_bobot merupakan tabel yang berisi data nilai variabel-variabel yang digunakan untuk proses pelatihan jaringan dalam sistem. Pembuatan tabel nilai_bobot menggunakan *query*:

```
CREATE TABLE IF NOT EXISTS `nilai_bobot` (`id_pelatihan` int(5) NOT NULL,
`alfa` double NOT NULL, `epsilon` double NOT NULL, `max_epoch` int(4) NOT NULL,
`total_epoch` int(4) NOT NULL, `kfold` int(2) NOT NULL, `waktu` double NOT
NULL, `status` int(1) NOT NULL, `w1` varchar(300) NOT NULL, `w2` varchar(300)
NOT NULL, `alfa_akhir` double NOT NULL, PRIMARY KEY (`id_pelatihan`))
ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tampilan struktur tabel nilai_bobot dapat dilihat pada gambar 4.3.

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra
1	id_pelatihan 	int(5)			Tidak	Tidak ada		AUTO_INCREMENT
2	alfa	double			Tidak	Tidak ada		
3	epsilon	double			Tidak	Tidak ada		
4	max_epoch	int(4)			Tidak	Tidak ada		
5	total_epoch	int(4)			Tidak	Tidak ada		
6	kfold	int(2)			Tidak	Tidak ada		
7	waktu	double			Tidak	Tidak ada		
8	status	int(1)			Tidak	Tidak ada		
9	w1	varchar(300) latin1_swedish_ci			Tidak	Tidak ada		
10	w2	varchar(300) latin1_swedish_ci			Tidak	Tidak ada		
11	alfa_akhir	double			Tidak	Tidak ada		

Gambar 4. 3. Struktur Tabel nilai_bobot pada MySQL

4. Tabel pengujian

Tabel pengujian merupakan tabel yang berisi hasil dari proses pengujian yang telah dilakukan. Pembuatan tabel pengujian menggunakan *query*:

```
CREATE TABLE IF NOT EXISTS `pengujian` (`id_pengujian` int(5) NOT NULL,
`id_pelatihan` int(5) NOT NULL, `error` double NOT NULL, `akurasi` double NOT
NULL, `sensitivitas` double NOT NULL, `spesifisitas` double NOT NULL, PRIMARY
KEY (`id_pengujian`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tampilan struktur tabel pengujian dapat dilihat pada gambar 4.4.

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra
1	id_pengujian	int(5)			Tidak	Tidak ada		AUTO_INCREMENT
2	id_pelatihan	int(5)			Tidak	Tidak ada		
3	error	double			Tidak	Tidak ada		
4	akurasi	double			Tidak	Tidak ada		
5	sensitivitas	double			Tidak	Tidak ada		
6	spesifisitas	double			Tidak	Tidak ada		

Gambar 4. 4. Struktur Tabel pengujian pada MySQL

5. Tabel penanganan

Tabel penanganan merupakan tabel yang berisi data penanganan penyakit yang digunakan untuk rekomendasi. Pembuatan tabel penanganan menggunakan *query*:

```
CREATE TABLE IF NOT EXISTS `penanganan` (`id_penanganan` int(1) NOT NULL,
`deskripsi` longtext NOT NULL, `deteksi` varchar(22) NOT NULL, PRIMARY KEY
(`id_penanganan`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tampilan struktur tabel penanganan dapat dilihat pada gambar 4.5.

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra
1	id_penanganan	int(1)				Tidak	Tidak ada	
2	deskripsi	longtext	latin1_swedish_ci			Tidak	Tidak ada	
3	deteksi	varchar(22)	latin1_swedish_ci			Tidak	Tidak ada	

Gambar 4. 5. Struktur Tabel Penanganan pada MySQL

4.1.3. Implementasi Fungsi

Implementasi fungsi adalah hasil implementasi dari desain fungsi yang direalisasikan dalam sistem. Fungsi diimplementasikan dengan menggunakan bahasa pemrograman PHP dan sistem manajemen basis data MySQL. Berikut ini merupakan hasil implementasi fungsi pada Sistem Pendekripsi Diabetes Melitus menggunakan LVQ. Penjelasan yang menjadi fokus dan contoh implementasi diambil 3 fungsi utama sebagai berikut.

1. Fungsi Pelatihan

Fungsi Pelatihan berfungsi untuk melakukan proses pelatihan algoritma jaringan saraf tiruan *Learning Vector Quantization*. Fungsi Pelatihan ini menghasilkan bobot akhir yang nantinya digunakan dalam proses pengujian. *Source code* dari Fungsi Pelatihan dapat dilihat pada Lampiran 2 nomor 1.

2. Fungsi Pengujian

Fungsi Pengujian berfungsi untuk melakukan proses pengujian algoritma jaringan syaraf tiruan *Learning Vector Quantization*. Fungsi Pengujian ini menghasilkan nilai tingkat akurasi, *error*, sensitivitas, dan spesifisitas. *Source code* dari Fungsi Pengujian dapat dilihat pada Lampiran 2 nomor 2.

3. Fungsi Deteksi

Fungsi Deteksi berfungsi untuk melakukan proses klasifikasi kelas menggunakan algoritma jaringan syaraf tiruan LVQ. Fungsi Deteksi ini menghasilkan hasil deteksi di halaman *guest*. *Source code* dari Fungsi Deteksi dapat dilihat pada Lampiran 2 nomor 3.

4.1.4. Implementasi Antarmuka

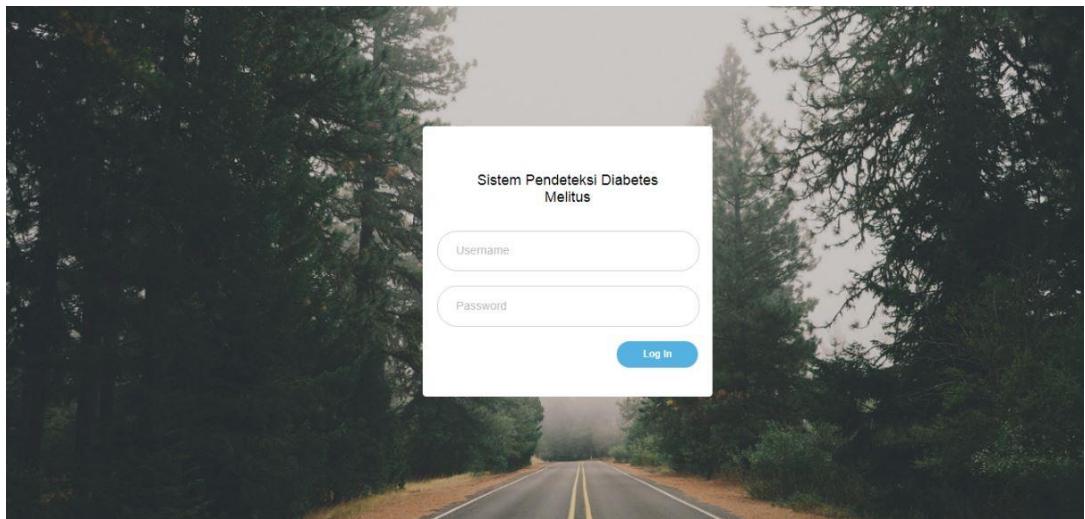
Implementasi antarmuka merupakan realisasi antarmuka sistem yang dihasilkan dari desain antarmuka sebelumnya. Tampilan antarmuka pada sistem ini terdiri dari dua jenis tampilan yaitu tampilan untuk admin dan tampilan untuk *guest*. Tampilan untuk admin terdiri dari menu-menu yang dapat digunakan admin untuk mengelola jaringan syaraf tiruan LVQ dan data yang digunakan sistem. Sedangkan tampilan untuk *guest* terdiri dari tampilan untuk melakukan deteksi diabetes melitus. Implementasi antarmuka dapat dilihat berturut-turut pada gambar 4.6 sampai gambar 4.27.

1. Halaman *Login* Admin

Halaman *Login* Admin merupakan halaman yang digunakan admin untuk masuk ke sistem. Admin harus menginputkan *username* dan *password* untuk dapat masuk ke sistem. Implementasi tampilan Halaman *Login* Admin dapat dilihat pada gambar 4.6.

2. Halaman Beranda Admin

Halaman Beranda Admin merupakan halaman yang diakses admin setelah melakukan *login*. Halaman Beranda Admin terdiri dari beberapa menu yaitu Beranda, Dataset, Pengujian, Pelatihan, Ubah *Password*, dan Keluar. Implementasi tampilan Halaman Beranda Admin dapat dilihat pada gambar 4.7.



Gambar 4. 6. Tampilan Halaman *Login* Admin

A screenshot of the Admin Home Page. At the top left is the logo (a stylized building with a cross). To its right is the title "Sistem Pendekripsi Diabetes Melitus". Below the title is a horizontal menu bar with three items: "Selamat Datang", "Jaringan Syaraf Tiruan LVQ", and "Diabetes Melitus". The "Selamat Datang" item is highlighted with a grey background. The main content area has a light grey background and contains the text "Selamat Datang" in bold. Below it is a smaller paragraph explaining the system's purpose: "Selamat datang di Sistem Prediksi Diabetes Melitus. Sistem Prediksi Diabetes Melitus merupakan sistem yang digunakan untuk memprediksi penyakit diabetes melitus berdasarkan dari hasil tes laboratorium. Sistem ini hanya dapat memprediksi, tidak dengan pasti memberikan vonis penyakit pada Anda, untuk itu Anda perlu pergi ke dokter untuk kebenaran prediksi dan penanganan lebih lanjut." At the bottom left of the page are social media icons for Instagram, Facebook, and Twitter, followed by the copyright notice: "© Reyhan Syarif Universitas Diponegoro".

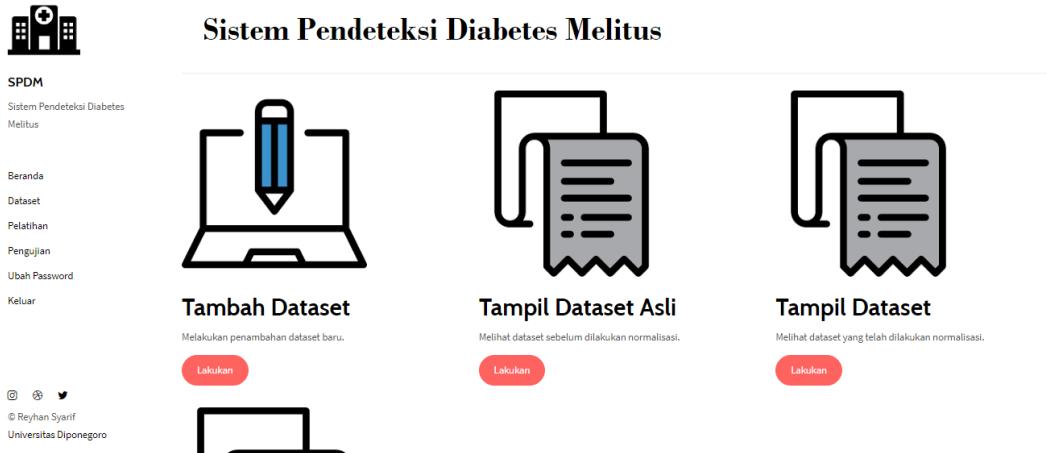
Gambar 4. 7. Tampilan Halaman Beranda Admin

3. Halaman Pilih Dataset

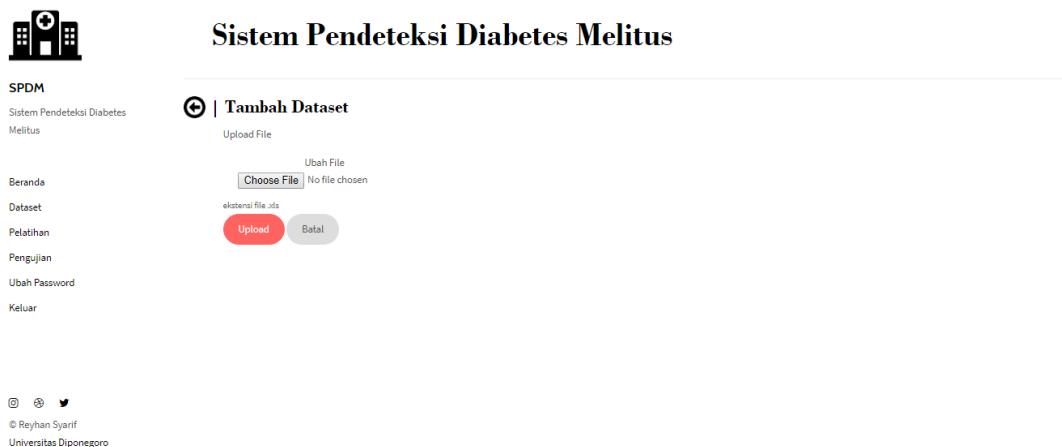
Halaman Pilih Dataset merupakan halaman yang digunakan admin untuk mengelola data yang ada dalam sistem seperti dataset serta data penanganan. Terdapat beberapa menu pada halaman ini antara lain Tambah Dataset, Tampil Dataset Asli, Tampil Dataset, serta Tampil Penanganan. Implementasi tampilan Halaman Pilih Dataset dapat dilihat pada gambar 4.8.

4. Halaman Tambah Dataset

Halaman Tambah Dataset digunakan untuk melakukan *upload* dataset yang akan digunakan pada sistem. Implementasi tampilan Halaman Tambah Dataset dapat dilihat pada gambar 4.9.



Gambar 4. 8. Tampilan Halaman Pilih Dataset



Gambar 4. 9. Tampilan Halaman Tambah Dataset

5. Halaman Tampil Dataset Asli

Halaman Tampil Dataset Asli digunakan untuk menampilkan dataset yang telah di *upload* sebelumnya ke dalam sistem. Halaman ini menampilkan dataset sebelum di normalisasi. Implementasi tampilan Halaman Tampil Dataset Asli dapat dilihat pada gambar 4.10.

6. Halaman Tampil Dataset

Halaman Tampil Dataset hampir sama dengan Halaman Tampil Dataset Asli. Yang membedakan adalah dataset yang ditampilkan disini merupakan dataset setelah dilakukan normalisasi. Implementasi tampilan Halaman Tampil Dataset dapat dilihat pada gambar 4.11.



Sistem Pendeksi Diabetes Melitus

SPDM
Sistem Pendeksi Diabetes Melitus

Beranda

Dataset

Pelatihan

Pengujian

Ubah Password

Keluar

[@](#) [%](#) <#>
© Reyhan Syarif
Universitas Diponegoro

⌚ | Tampil Dataset

Show 10 entries Search:

Umur	Glin	Gpost	Upost	Actn	Urn	Actpp	Ldl	Hdl	Chol	Tg	Class
18	118	125	0	0	0	0	42	35	125	239	2
34	81	95	0	0	0	0	58	45	119	79	2
39	101	106	0	0	0	0	54	32	131	225	2
40	90	139	0	0	0	0	48	60	115	35	2
42	339	480	1000	0	1000	0	42	69	134	277	1
45	117	219	300	0	50	0	52	41	118	157	1
48	132	189	1000	0	0	0	57	66	145	162	1
49	136	144	100	0	0	0	55	35	102	59	1
51	134	257	0	0	0	0	47	70	207	203	1

Gambar 4. 10. Tampilan Halaman Tampil Dataset Asli



Sistem Pendeksi Diabetes Melitus

SPDM
Sistem Pendeksi Diabetes Melitus

Beranda

Dataset

Pelatihan

Pengujian

Ubah Password

Keluar

[@](#) [%](#) <#>
© Reyhan Syarif
Universitas Diponegoro

⌚ | Tampil Dataset

Show 10 entries Search:

Umur	Glin	Gpost	Upost	Actn	Urn	Actpp	Ldl	Hdl	Chol	Tg
0	0.12984054689704	0.069078947368421	0	0	0	0	0.24817518248175	0.27848101265823	0.40449438202247	0.60117302052786
0.21621621621622	0.045558086560364	0.019736842105263	0	0	0	0	0.64696350564964	0.40506329113924	0.37078651688393	0.13196480938416
0.28378378378378	0.091116173120729	0.037828947368421	0	0	0	0	0.33576642335768	0.24050632911392	0.43820224719101	0.56011730205279
0.2872972972973	0.066059225512528	0.092105263157895	0	0	0	0	0.29197080291971	0.59493670886076	0.34831460674157	0.0029325513196481
0.32423423423432	0.63325740318907	0.65296052631579	1	0	1	0	0.24817518248175	0.70886075949367	0.45505617977528	0.7126097067449
0.36486488486486	0.12756264236902	0.22368421052632	0.3	0	0.05	0	0.32116788321188	0.35443037974684	0.36516853932584	0.36070381231672
0.40540540540541	0.16173120728929	0.17434210526316	1	0	0	0	0.35766423357664	0.67088607594937	0.51685393258427	0.37536656891496
0.41891891891892	0.17084282460137	0.10032894736842	0.1	0	0	0	0.34306569343066	0.27848101265823	0.2752808988764	0.073313782991202
0.44594594594595	0.16628701594533	0.28618421052632	0	0	0	0	0.28467153284672	0.72151898734177	0.86516853932584	0.49560117302053

Gambar 4. 11. Tampilan Halaman Tampil Dataset

7. Halaman Tampil Penanganan

Halaman Tampil Penanganan digunakan untuk menampilkan penanganan terkait hasil deteksi positif diabetes melitus dan negatif diabetes melitus. Implementasi tampilan Halaman Tampil Penanganan dapat dilihat pada gambar 4.12.

8. Halaman Ubah Penanganan

Halaman Ubah Penanganan digunakan untuk mengubah isi penanganan yang telah dibuat sebelumnya. Implementasi tampilan Halaman Ubah Penanganan dapat dilihat pada gambar 4.13.

No.	Deteksi	Deskripsi	Aksi
1	Diabetes Melitus	Gejala yang Anda rasakan tidak mengarah ke Diabetes Melitus. Tetap jaga pola hidup sehat, olahraga dan makan secara baik dan teratur. Dan rutinlah untuk cek ke dokter agar tidak terindikasi berbagai penyakit lainnya.	Ubah Penanganan
2	Tidak Diabetes Melitus		Ubah Penanganan

Gambar 4. 12. Tampilan Halaman Tampil Penanganan

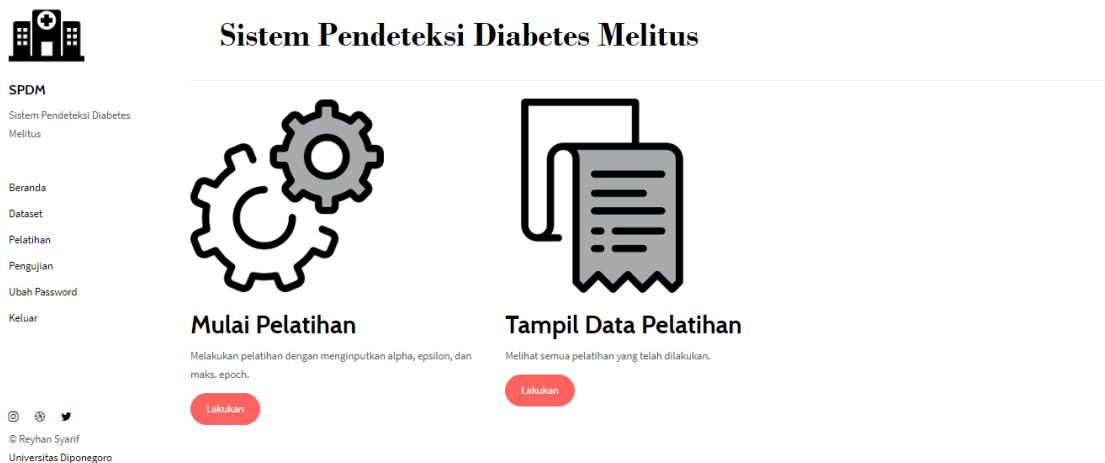
Gambar 4. 13. Tampilan Halaman Ubah Penanganan

9. Halaman Pilih Pelatihan

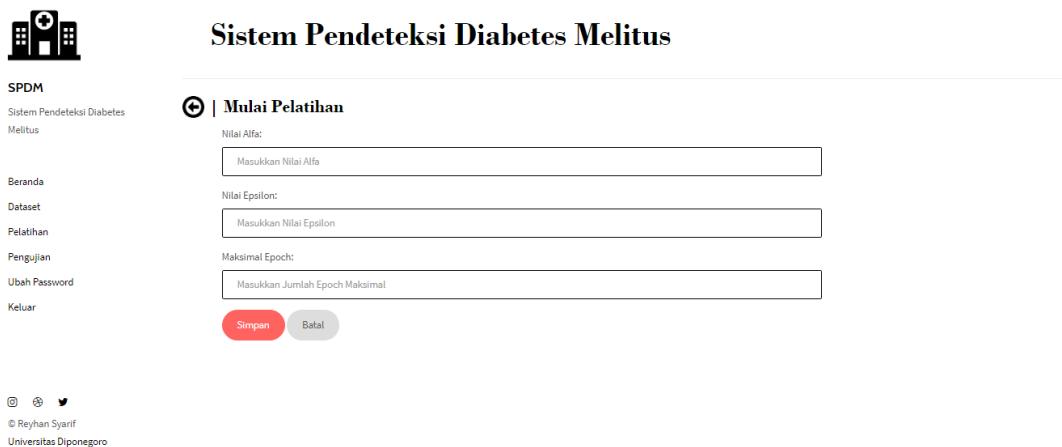
Halaman Pilih Pelatihan merupakan halaman yang berkaitan dengan proses pelatihan jaringan syaraf tiruan LVQ. Halaman ini terdiri dari dua menu yaitu Mulai Pelatihan dan Tampil Data Pelatihan. Implementasi tampilan Halaman Pilih Pelatihan dapat dilihat pada gambar 4.14.

10. Halaman Mulai Pelatihan

Halaman Mulai Pelatihan digunakan admin untuk melakukan proses pelatihan dengan menginputkan beberapa parameter pelatihan yaitu *alfa*, *epsilon* dan maksimal *epoch*. Implementasi tampilan Halaman Mulai Pelatihan dapat dilihat pada gambar 4.15.



Gambar 4. 14. Tampilan Halaman Pilih Pelatihan



Gambar 4. 15. Tampilan Halaman Mulai Pelatihan

11. Halaman Tampil Data Pelatihan

Halaman Tampil Data Pelatihan berfungsi untuk menampilkan data hasil pelatihan yang telah dilakukan di Halaman Mulai Pelatihan. Implementasi tampilan Halaman Tampil Data Pelatihan dapat dilihat pada gambar 4.16.

12. Halaman Tampil Detail Data Pelatihan

Halaman Tampil Detail Data Pelatihan digunakan admin untuk menampilkan detail dari pelatihan yang telah dilakukan. Halaman ini menampilkan detail data dari tiap fold beserta bobot akhirnya. Implementasi tampilan Halaman Tampil Detail Data Pelatihan dapat dilihat pada gambar 4.17.



Sistem Pendekripsi Diabetes Melitus

⊕ | Tampil Data Pelatihan

Show 10 entries Search:

No.	Alfa	Epsilon	Epoch	Aksi
1	0.1	0.01	100	Lihat Detail Lakukan Pengujian Hapus
2	0.1	0.001	100	Lihat Detail Lakukan Pengujian Hapus
3	0.1	0.0001	100	Lihat Detail Lakukan Pengujian Hapus
4	0.1	0.00001	100	Lihat Detail Lakukan Pengujian Hapus
5	0.1	0.000001	100	Lihat Detail Lakukan Pengujian Hapus
6	0.2	0.01	100	Lihat Detail Lakukan Pengujian Hapus
7	0.2	0.001	100	Lihat Detail Lakukan Pengujian Hapus
8	0.2	0.0001	100	Lihat Detail Lakukan Pengujian Hapus
9	0.2	0.00001	100	Lihat Detail Lakukan Pengujian Hapus

© Reyhan Syarif
Universitas Diponegoro

Gambar 4. 16. Tampilan Halaman Tampil Data Pelatihan



Sistem Pendekripsi Diabetes Melitus

⊕ | Tampil Detail Data Pelatihan

Show 10 entries Search:

No.	K-Fold	Alfa	Epsilon	Epoch	Bobot Akhir
1	1	0.1	0.01	100	w1 = 0.45339867908342, 0.16210271249264, 0.21488279603898, 0.11306241972067, 0.057791379246525, 0.1600214789863, -0.14091791860707, 0.076709684821252, 0.4662563431872, 0.40224829912618, 0.4011496381448 w2 = 0.59759532476331, 0.26190623826186, 0.32307582702355, 0.004532812944747, -0.059222185456298, 0.004324953221732, 0.32687265239927, 0.17318981671962, 0.56911624352472, 0.745083886026298
2	2	0.1	0.01	100	w1 = 0.47627534450141, 0.13056401135721, 0.17560384990616, 0.1254384554717, 0.064861442780641, 0.062199678943792, -0.15904807983803, 0.099023882602047, 0.6031830429485, 0.48177028190814, 0.3447788323311 w2 = 0.56154559765857, 0.54789910349677, 0.6494197689264, 0.2467077056114, 0.080031848497114, -0.14162710976049, 0.0800031848497114, 0.58866147362379, 0.25244229850087, 0.65176102363088, 0.86468788424285
3	3	0.1	0.01	100	w1 = 0.476119701416, -0.00271840644625223, 0.075251484004182, -0.114539969117, 0.067197750517598, -0.04664185044261, -0.15766028362199, 0.073275484390598, 0.5317053320439, 0.4658731238492, 0.33219634323651 w2 = 0.68285634649927, 0.49122314115137, 0.62828056429871, 0.2785086496661, 0.017493834021819, -0.19786898600981, 0.016614857423264, 0.50270007759636, 0.19251084092105, 0.66660219123214, 0.86942888062097

© Reyhan Syarif
Universitas Diponegoro

Gambar 4. 17. Tampilan Halaman Tampil Detail Data Pelatihan

13. Halaman Pilih Pengujian

Halaman Pilih Pengujian merupakan halaman yang digunakan admin berkaitan dengan proses pengujian. Halaman ini terdiri dari beberapa menu antara lain Mulai Pengujian, Tampil Data Pengujian dan Tampil Bobot Final. Implementasi tampilan Halaman Pilih Pengujian dapat dilihat pada gambar 4.18.

14. Halaman Mulai Pengujian

Halaman Mulai Pengujian merupakan halaman yang digunakan admin untuk melakukan proses pengujian. Proses pengujian juga dapat dilakukan melalui Halaman Tampil Data Pelatihan. Implementasi tampilan Halaman Mulai Pengujian dapat dilihat pada gambar 4.19.



Gambar 4. 18. Tampilan Halaman Pilih Pengujian

This screenshot shows the 'Mulai Pengujian' (Start Testing) page. At the top, there's a header with the system title and a plus sign icon. Below it is a search bar and a table with 10 entries. The table columns are labeled 'No.', 'Alfa', 'Epsilon', 'Epoch', and 'Aksi'. Each row contains four numerical values and two buttons: 'Lihat Detail' and 'Lakukan Pengujian'. The rows are numbered 1 through 9.

No.	Alfa	Epsilon	Epoch	Aksi
1	0.1	0.01	100	Lihat Detail Lakukan Pengujian
2	0.1	0.001	100	Lihat Detail Lakukan Pengujian
3	0.1	0.0001	100	Lihat Detail Lakukan Pengujian
4	0.1	0.00001	100	Lihat Detail Lakukan Pengujian
5	0.1	0.000001	100	Lihat Detail Lakukan Pengujian
6	0.2	0.01	100	Lihat Detail Lakukan Pengujian
7	0.2	0.001	100	Lihat Detail Lakukan Pengujian
8	0.2	0.0001	100	Lihat Detail Lakukan Pengujian
9	0.2	0.00001	100	Lihat Detail Lakukan Pengujian

Gambar 4. 19. Tampilan Halaman Mulai Pengujian

15. Halaman Tampil Data Pengujian

Halaman Tampil Data Pengujian merupakan halaman yang digunakan admin untuk menampilkan hasil pengujian yang sebelumnya telah dilakukan. Implementasi tampilan Halaman Tampil Data Pengujian dapat dilihat pada gambar 4.20.

16. Halaman Tampil Detail Data Pengujian

Halaman Tampil Detail Data Pelatihan merupakan halaman yang digunakan untuk menampilkan detail dari data pengujian yang telah dilakukan. Detail yang ditampilkan adalah detail data tiap fold, bobot akhir serta akurasi, *error*, sensitivitas, dan spesifisitas. Implementasi tampilan Halaman Tampil Detail Data Pengujian dapat dilihat pada gambar 4.21.



Sistem Pendekripsi Diabetes Melitus

⊕ Tampil Data Pengujian										
Show 10 entries										
No.	Alfa	Epsilon	Epoch	Akurasi	Error	Sensitivitas	Spesifitas	Aksi		Search:
1	0.1	0.01	100	42%	58%	72%	12%	Lihat Detail	Hapus	
2	0.1	0.001	100	45%	55%	78%	12%	Lihat Detail	Hapus	
3	0.1	0.0001	100	45%	55%	78%	12%	Lihat Detail	Hapus	
4	0.1	0.00001	100	45%	55%	78%	12%	Lihat Detail	Hapus	
5	0.1	0.000001	100	45%	55%	78%	12%	Lihat Detail	Hapus	
6	0.2	0.01	100	52%	48%	100%	4%	Lihat Detail	Hapus	
7	0.2	0.001	100	52%	48%	100%	4%	Lihat Detail	Hapus	
8	0.2	0.0001	100	52%	48%	100%	4%	Lihat Detail	Hapus	
9	0.2	0.00001	100	52%	48%	100%	4%	Lihat Detail	Hapus	

Gambar 4. 20. Tampilan Halaman Tampil Data Pengujian



Sistem Pendekripsi Diabetes Melitus

⊕ Tampil Detail Data Pengujian										
Show 10 entries										
No.	K-Fold	Alfa	Epsilon	Bobot Akhir	Akurasi(%)	Error(%)	Sensitivitas(%)	Spesifitas(%)	Aksi	Search:
1	7	0.1	0.01	w1 = 0.674557486704 , 0.11533800807982 , 0.235950503437273 , 0.18987374765399 , 0.12563113440173 , 0.17691240516765 , 6.5333275913314E-7 , -0.095949388588833 , 0.40372764008403 , 0.3554235687948 , 0.4398046504755 w2 = 0.4829563654343 , 0.36257132826449 , 0.4169723461193 , 0.36697000225845 , 0.017641115869906 , -0.11367983641012 , 0.017060411761659 , 0.52442324097194 , 0.20893803125348 , 0.67555526877687 , 0.76025290191166	60	40	100	20	Simpan Bobot	
2	1	0.1	0.01	w1 = 0.453359867908342 , 0.16210271248264 , 0.21488279603898 , 0.11306241972067 .	60	40	100	20		Simpan Bobot

Gambar 4. 21. Tampilan Halaman Tampil Detail Data Pengujian

17. Halaman Tampil Bobot Final

Halaman Tampil Bobot Final merupakan halaman yang digunakan untuk menampilkan bobot final yang telah dipilih pada Halaman Tampil Detail Data Pengujian. Bobot inilah yang akan digunakan pada proses deteksi. Implementasi tampilan Halaman Tampil Bobot Final dapat dilihat pada gambar 4.22.

18. Halaman Ubah Password

Halaman Ubah Password merupakan halaman yang berguna bagi admin untuk dapat mengganti password akunnya. Implementasi tampilan Halaman Ubah Password dapat dilihat pada gambar 4.23.

Sistem Pendekripsi Diabetes Melitus

Tampil Bobot Final

No	K-Fold	Alfa	Epsilon	Epoch	Bobot Akhir	Akurasi(%)	Error(%)	Sensitivitas(%)	Spesifitas(%)
----	--------	------	---------	-------	-------------	------------	----------	-----------------	---------------

Showing 0 to 0 of 0 entries

First Previous Next Last

SPDM
Sistem Pendekripsi Diabetes Melitus

Beranda
Dataset
Pelatihan
Pengujian
Ubah Password
Keluar

© Reyhan Syarif
Universitas Diponegoro

Gambar 4. 22. Tampilan Halaman Tampil Bobot Final

Sistem Pendekripsi Diabetes Melitus

Ubah Password

>Password Lama:

Password Baru:

Konfirmasi Password Baru:

Simpan **Batal**

SPDM
Sistem Pendekripsi Diabetes Melitus

Beranda
Dataset
Pelatihan
Pengujian
Ubah Password
Keluar

© Reyhan Syarif
Universitas Diponegoro

Gambar 4. 23. Tampilan Halaman Ubah Password

19. Halaman Beranda *Guest*

Halaman Beranda *Guest* merupakan halaman pertama kali dapat diakses oleh *guest*. Halaman ini memiliki beberapa menu yaitu Beranda, Mulai Deteksi dan Pengembang. Implementasi tampilan Halaman Beranda *Guest* dapat dilihat pada gambar 4.24.

20. Halaman Mulai Deteksi

Halaman Mulai Deteksi merupakan halaman yang digunakan *guest* untuk melakukan deteksi diabetes melitus. Pada halaman ini *guest* harus menginputkan 11 parameter yang dibutuhkan sistem untuk dapat melakukan proses deteksi. Implementasi tampilan Halaman Mulai Deteksi dapat dilihat pada gambar 4.25.



Gambar 4. 24. Tampilan Halaman Beranda Guest

Gambar 4. 25. Tampilan Halaman Mulai Deteksi

21. Halaman Hasil Deteksi

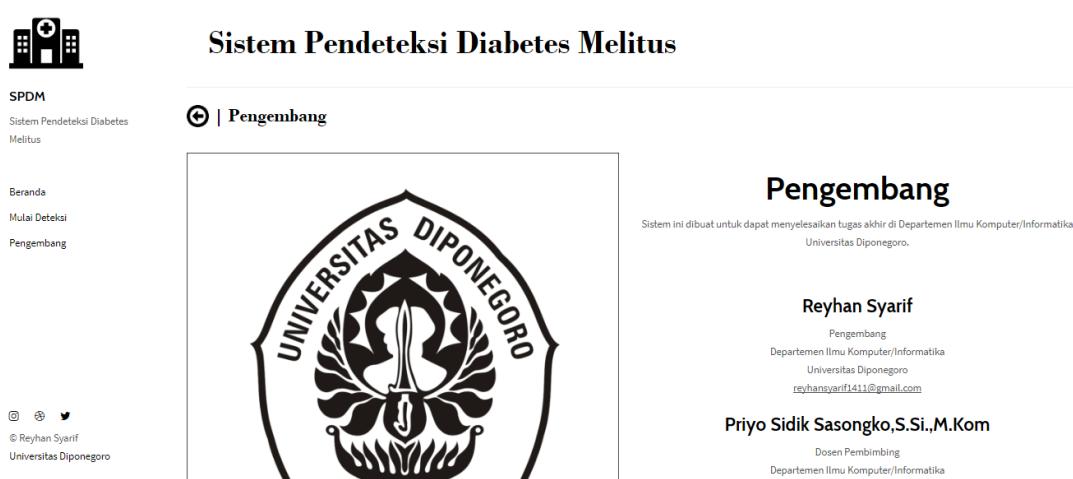
Halaman Hasil Deteksi merupakan halaman yang berfungsi untuk menampilkan hasil deteksi yang mengacu dari parameter yang telah dimasukkan sebelumnya, apakah hasil deteksinya berupa positif diabetes atau negatif, dan juga menampilkan detail penanganannya. Implementasi tampilan Halaman Hasil Deteksi dapat dilihat pada gambar 4.26.

22. Halaman Pengembang

Halaman Pengembang merupakan halaman yang menampilkan infomasi mengenai sistem dan juga pengembang sistem. Implementasi tampilan Halaman Pengembang dapat dilihat pada gambar 4.27.



Gambar 4. 26. Tampilan Halaman Hasil Deteksi



Gambar 4. 27. Tampilan Halaman Pengembang

4.2. Pengujian Fungsional

Pengujian Sistem Pendekripsi Diabetes Melitus dilakukan dengan menggunakan metode *blackbox*, yaitu melakukan pengujian fungsionalitas dari perangkat lunak untuk menemukan kesalahan pada *requirement* dengan mengabaikan mekanisme internal atau komponen dari program tersebut. Pengujian *blackbox* diterima jika fitur-fitur dari perangkat lunak telah memenuhi kebutuhan sistem.

4.2.1. Rencana Pengujian

Rencana pengujian Sistem Pendekripsi Diabetes Melitus dapat dilihat pada Lampiran 3. Pengujian dilakukan terhadap semua kebutuhan fungsional sistem yang ada pada tabel 3.13 dengan metode *blackbox*.

4.2.2. Pelaksanaan Pengujian

Pelaksanaan pengujian dilakukan dengan cara merealisasikan rencana pengujian yang telah disusun. Pengujian dilakukan pada alamat www.spdm.000webhost.com dengan menggunakan *browser* google chrome. Hasil pengujian fungsional dapat dilihat pada Lampiran 4.

4.2.3. Evaluasi Pengujian

Deskripsi dari hasil pengujian fungsional disajikan pada Lampiran 4. Berdasarkan hasil yang didapat pada Lampiran 4, dapat dilihat bahwa seluruh pengujian telah diterima. Hal ini menunjukkan bahwa Sistem Pendekripsi Diabetes Melitus telah memenuhi kebutuhan fungsional sistem yang telah diidentifikasi sebelumnya.

4.3. Pengujian Jaringan Syaraf Tiruan LVQ

Jaringan Syaraf Tiruan LVQ pada Sistem Pendekripsi Diabetes Melitus akan diuji menggunakan beberapa skenario untuk memperoleh arsitektur terbaik yang membantu dalam mengoptimalkan akurasi dari proses deteksi.

4.3.1. Skenario Pengujian

Skenario pengujian ini bertujuan untuk melakukan pengujian terhadap tingkat akurasi, *error*, sensitifitas, dan spesivisitas dari metode jaringan syaraf tiruan LVQ yang diterapkan pada Sistem Pendekripsi Diabetes Melitus. Terdapat tiga skenario yang dilakukan pada pengujian ini. Skenario pengujian dapat dilihat pada tabel 4.1.

Tabel 4. 1. Skenario Pengujian

Eksperimen	Inisialisasi Bobot Awal	Learning Rate (α)	Error Minimum (epsilon)	Maksimal Epoch
1	random	0.01,0.02,0.03,0.04 ,0.05,0.06,0.07,0.08, 0.09	0	100, 500, 1000
2	0.5	0.01,0.02,0.03,0.04 ,0.05,0.06,0.07,0.08, 0.09	0.01,0.001,0.0001, 0.00001,0.000001	Maksimum epoch hasil eksperimen 1
3	random	0.01,0.02,0.03,0.04 ,0.05,0.06,0.07,0.08, 0.09	0.01,0.001,0.0001, 0.00001,0.000001	Maksimum epoch hasil eksperimen 1

Proses pelatihan dilakukan berkali-kali dengan melakukan uji coba berdasarkan kombinasi *learning rate* (α) dan *error* minimum (*epsilon*). Maksimum *epoch* yang digunakan adalah maksimum *epoch* yang memiliki akurasi tertinggi. Kemudian setelah proses pelatihan selesai, dilakukan pengujian untuk mengetahui nilai akurasi dan *error*. Lalu akan dicari penggunaan kombinasi yang tepat antara *learning rate*, *epsilon* dan bobot awal sehingga menghasilkan arsitektur jaringan LVQ yang mempunyai keakuratan paling tinggi setiap bagiannya.

Keakuratan LVQ dihitung menggunakan *confusion matrix*. Selain itu akan dihitung juga nilai sensitivitas dan spesifisitas. Hasil tingkat akurasi, *error*, sensitivitas, dan spesifisitas untuk data kfold ke-1 sampai kfold ke-10 diambil rata-rata untuk setiap kombinasi *learning rate* (α) dan *epsilon*.

4.3.1.1.Skenario 1

Eksperimen 1 pada penelitian ini dilakukan untuk mengetahui pengaruh penggunaan maksimum *epoch* terhadap tingkat akurasi LVQ. Maksimum *epoch* yang menghasilkan akurasi terbaik akan diambil pada eksperimen 2 dan 3. Eksperimen 1 dilakukan dengan cara meneliti penggunaan maksimum *epoch* 100, 500, dan 1000, bobot awal random, *learning rate* bernilai 0.01-0.09, dan *epsilon* yang digunakan adalah 0 agar pelatihan berjalan hingga maksimum *epoch* tercapai.

4.3.1.2.Skenario 2

Eksperimen 2 pada penelitian dilakukan dengan cara meneliti pengaruh penggunaan bobot awal 0.5 terhadap tingkat akurasi LVQ. Selain itu, nilai *learning rate* yang digunakan bernilai 0.01-0.09, dan *epsilon* bernilai 0.01 sampai 0.000001. Maksimum *epoch* diambil dari hasil eksperimen 1. Setelah didapatkan akurasi, *error*, sensitivitas, dan spesifisitas yang paling baik, akan dilakukan analisa pengaruh *learning rate* terhadap akurasi.

4.3.1.3.Skenario 3

Eksperimen 3 pada penelitian dilakukan dengan cara meneliti pengaruh penggunaan bobot awal *random* terhadap tingkat akurasi LVQ. Nilai *epsilon*, *learning rate* dan *epoch* sama seperti pada eksperimen 2 yaitu *epsilon* bernilai 0.01 sampai 0.000001. Maksimum *epoch* diambil dari hasil eksperimen 1. Nilai *learning rate* yang digunakan bernilai 0.01-0.09. Setelah didapatkan akurasi, *error*, sensitivitas, dan spesifisitas yang paling baik, akan dilakukan analisa pengaruh *learning rate* terhadap akurasi.

4.4. Pembahasan Skenario Pengujian

Pembahasan dari skenario yang telah disusun untuk menganalisis hasil penelitian tugas akhir mengenai deteksi diabetes melitus akan disajikan di bawah ini.

4.4.1. Pembahasan Skenario 1

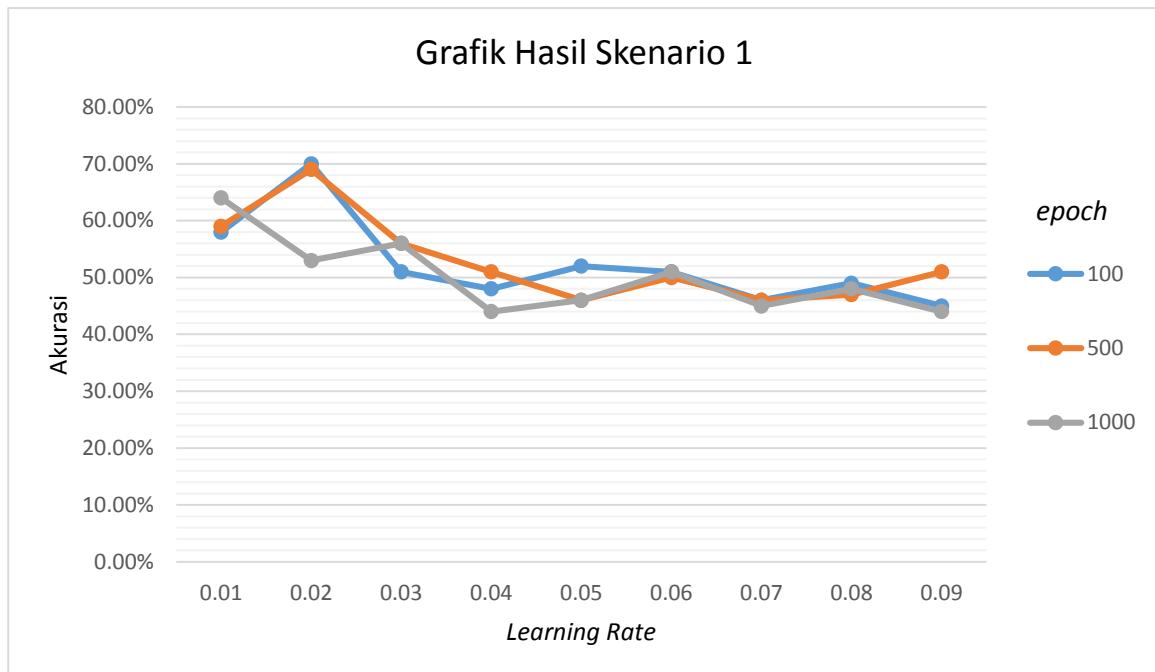
Skenario 1 ini bertujuan untuk mengetahui pengaruh dari variasi jumlah maksimum *epoch* terhadap akurasi yang dihasilkan serta mendapatkan jumlah maksimum *epoch* yang menghasilkan akurasi paling tinggi. Maksimum *epoch* yang diteliti adalah 100, 500, dan 1000. Penentuan bobot dilakukan secara *random*, *learning rate* antara 0.01 hingga 0.09, dan *epsilon* 0. Hasil eksperimen 1 ditunjukkan pada tabel 4.2.

Tabel 4. 2. Tabel Hasil Skenario 1

No	Learning rate	Epsilon	Maksimum Epoch	Akurasi	Error
1	0.01	0	100	58%	42%
2	0.02	0	100	70%	30%
3	0.03	0	100	51%	49%
4	0.04	0	100	48%	52%
5	0.05	0	100	52%	48%
6	0.06	0	100	51%	49%
7	0.07	0	100	46%	54%
8	0.08	0	100	49%	51%
9	0.09	0	100	45%	55%
10	0.01	0	500	59%	41%
11	0.02	0	500	69%	31%
12	0.03	0	500	56%	44%
13	0.04	0	500	51%	49%
14	0.05	0	500	46%	54%
15	0.06	0	500	50%	50%
16	0.07	0	500	46%	54%
17	0.08	0	500	47%	53%
18	0.09	0	500	51%	49%
19	0.01	0	1000	64%	36%
20	0.02	0	1000	53%	47%
21	0.03	0	1000	56%	44%
22	0.04	0	1000	44%	56%
23	0.05	0	1000	46%	54%
24	0.06	0	1000	51%	49%
25	0.07	0	1000	45%	55%

No	Learning rate	Epsilon	Maksimum Epoch	Akurasi	Error
26	0.08	0	1000	48%	52%
27	0.09	0	1000	44%	56%

Nilai akurasi yang ditampilkan adalah nilai akurasi rata-rata dari akurasi pada fold 1 sampai fold 10 untuk masing-masing kombinasi parameter. Berdasarkan hasil skenario 1, nilai akurasi tertinggi didapatkan ketika maksimum *epoch* berada pada nilai 100. Grafik hubungan *epoch* dengan akurasi dapat dilihat pada gambar 4.28.



Gambar 4. 28. Grafik Hasil Skenario 1

Dari hasil skenario 1 dapat dilihat bahwa nilai akurasi tertinggi didapatkan saat nilai maksimum *epoch* bernilai 100 dengan tingkat akurasi tertinggi yaitu 70%. Tingkat akurasi pada tiap maksimum *epoch* memiliki tingkat akurasi yang cukup beragam, namun pada tingkat *learning rate* tertentu menunjukkan tingkat akurasi yang hampir sama. Hal ini menunjukkan bahwa maksimum *epoch* tidak terlalu berpengaruh terhadap akurasi. Hal tersebut dimungkinkan karena dalam metode LVQ kondisi berhenti tidak hanya dilihat dari maksimum *epoch* saja, tetapi juga dilihat dari nilai *epsilon*. Berikut hasil tingkat akurasi, *error*, sensitivitas dan spesifisitas secara detil tiap data kfold nilai $\alpha = 0.02$, nilai $\text{eps} = 0$ dan maksimum *epoch* = 100 yang ditunjukkan pada Tabel 4.3.

Tabel 4. 3. Detail Hasil Pengujian Dengan Akurasi Tertinggi Skenario 1

No.	Kfold	Learni ng Rate (α)	Error Minimum (eps)	Maksi mum <i>Epoch</i>	Akuras i	Error	Sensiti vitas	Spesifi sitas
-----	-------	-----------------------------------	--------------------------------------	------------------------------	-------------	-------	------------------	------------------

1	1	0.02	0	100	60%	40%	80%	40%
2	2	0.02	0	100	90%	10%	100%	80%
3	3	0.02	0	100	40%	60%	40%	40%
4	4	0.02	0	100	70%	30%	80%	60%
5	5	0.02	0	100	70%	30%	100%	40%
6	6	0.02	0	100	70%	30%	80%	60%
7	7	0.02	0	100	60%	40%	80%	40%
8	8	0.02	0	100	70%	30%	100%	40%
9	9	0.02	0	100	70%	30%	80%	60%
10	10	0.02	0	100	100%	0%	100%	100%

Maksimum *epoch* 100 ini akan digunakan pada skenario 2 dan 3 untuk meneliti pengaruh penentuan bobot awal, *learning rate*, dan *epsilon*.

4.4.2. Pembahasan Skenario 2

Dari skenario 1, diperoleh maksimum *epoch* yang akan digunakan pada skenario 2 dan 3, yaitu bernilai 100. Pada skenario 2 dan 3 ini bertujuan untuk mengetahui pengaruh penentuan bobot awal serta pengaruh *learning rate* dan *epsilon* terhadap akurasi, sensitivitas, dan spesifisitas. Skenario 2 memiliki parameter *learning rate* antara 0.01 sampai dengan 0.09, *epsilon* 0.01 sampai 0.000001, maksimum *epoch* 100, dan bobot awal 0.5. Maka hasil skenario 2 ditunjukkan pada tabel 4.4.

Tabel 4. 4. Tabel Hasil Skenario 2

No	Learning rate	Epsilon	Akurasi	Error	Sensitivitas	Spesifisitas
1	0.01	0.01	49%	51%	98%	0%
2	0.02	0.01	45%	55%	92%	0%
3	0.03	0.01	42%	58%	84%	0%
4	0.04	0.01	42%	58%	82%	2%
5	0.05	0.01	42%	58%	82%	2%
6	0.06	0.01	40%	60%	78%	2%
7	0.07	0.01	41%	59%	76%	6%
8	0.08	0.01	43%	57%	76%	10%
9	0.09	0.01	41%	59%	72%	10%
10	0.01	0.001	46%	54%	92%	0%
11	0.02	0.001	43%	57%	86%	0%
12	0.03	0.001	43%	57%	86%	0%
13	0.04	0.001	43%	57%	84%	2%
14	0.05	0.001	43%	57%	84%	2%
15	0.06	0.001	44%	56%	86%	2%

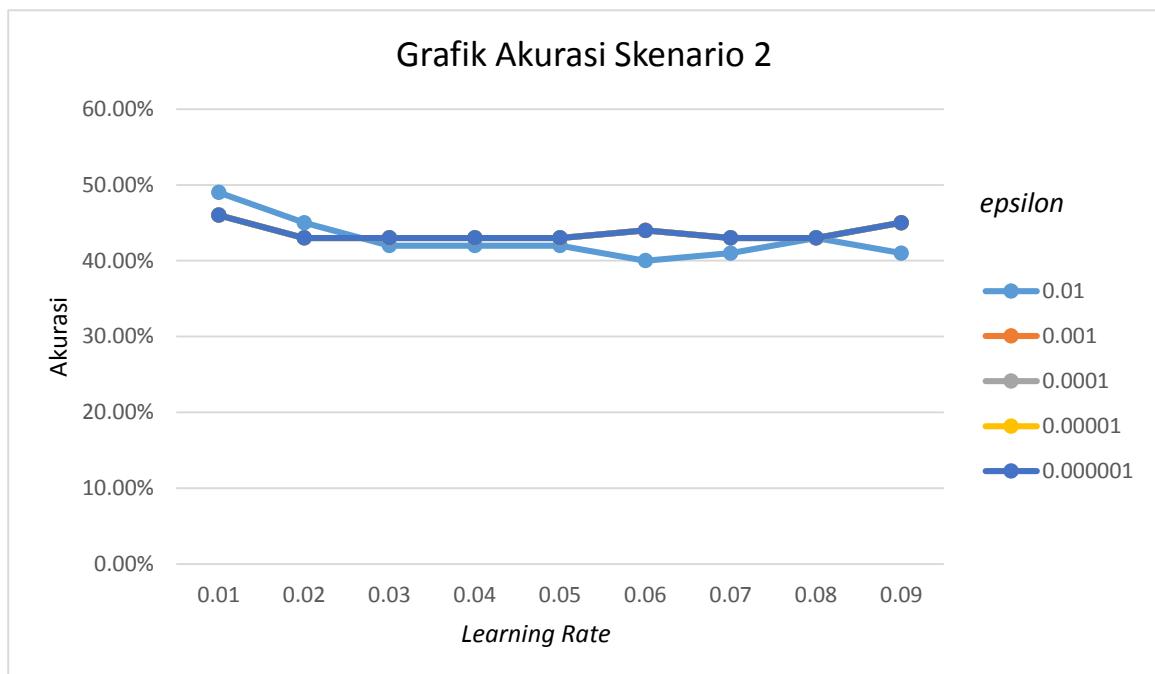
No	Learning rate	Epsilon	Akurasi	Error	Sensitivitas	Spesifisitas
16	0.07	0.001	43%	57%	80%	6%
17	0.08	0.001	43%	57%	78%	8%
18	0.09	0.001	45%	55%	80%	10%
19	0.01	0.0001	46%	54%	92%	0%
20	0.02	0.0001	43%	57%	86%	0%
21	0.03	0.0001	43%	57%	86%	0%
22	0.04	0.0001	43%	57%	84%	2%
23	0.05	0.0001	43%	57%	84%	2%
24	0.06	0.0001	44%	56%	86%	2%
25	0.07	0.0001	43%	57%	80%	6%
26	0.08	0.0001	43%	57%	78%	8%
27	0.09	0.0001	45%	55%	80%	10%
28	0.01	0.00001	46%	54%	92%	0%
29	0.02	0.00001	43%	57%	86%	0%
30	0.03	0.00001	43%	57%	86%	0%
31	0.04	0.00001	43%	57%	84%	2%
32	0.05	0.00001	43%	57%	84%	2%
33	0.06	0.00001	44%	56%	86%	2%
34	0.07	0.00001	43%	57%	80%	6%
35	0.08	0.00001	43%	57%	78%	8%
36	0.09	0.00001	45%	55%	80%	10%
37	0.01	0.000001	46%	54%	92%	0%
38	0.02	0.000001	43%	57%	86%	0%
39	0.03	0.000001	43%	57%	86%	0%
40	0.04	0.000001	43%	57%	84%	2%
41	0.05	0.000001	43%	57%	84%	2%
42	0.06	0.000001	44%	56%	86%	2%
43	0.07	0.000001	43%	57%	80%	6%
44	0.08	0.000001	43%	57%	78%	8%
45	0.09	0.000001	45%	55%	80%	10%

Pada tabel 4.4 dapat dilihat bahwa nilai akurasi tertinggi yang didapat pada nilai $\alpha = 0.01$, nilai ϵ = 0.01 dan maksimum epoch = 100 yaitu sebesar 49% dengan tingkat sensitivitas 98% dan spesifisitas sebesar 0%. Berikut hasil tingkat akurasi, error, sensitivitas dan spesifisitas secara detil tiap data kfold nilai $\alpha = 0.01$, nilai ϵ = 0.01 dan maksimum epoch = 100 yang ditunjukkan pada Tabel 4.5.

Tabel 4. 5. Detail Hasil Pengujian Dengan Akurasi Tertinggi Skenario 2

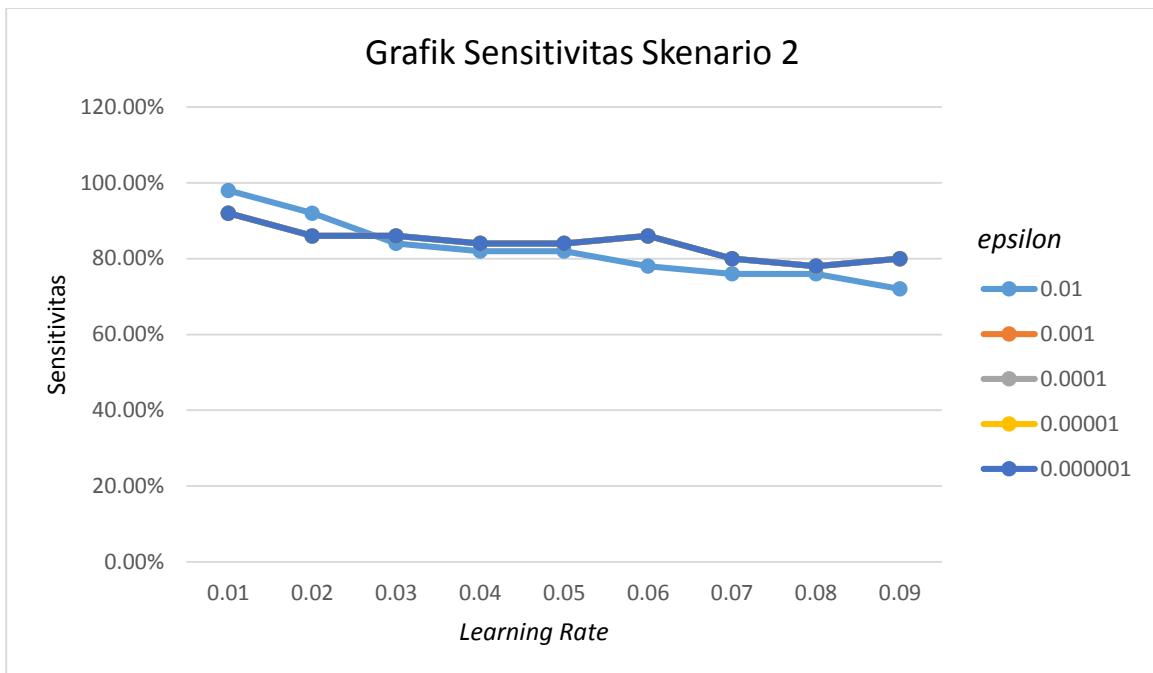
No.	Kfold	Learning Rate (α)	Error Minimum (eps)	Maksimum Epoch	Akurasi	Error	Sensitivitas	Spesifitas
1	1	0.01	0.01	100	50%	50%	100%	0%
2	2	0.01	0.01	100	50%	50%	100%	0%
3	3	0.01	0.01	100	40%	60%	80%	0%
4	4	0.01	0.01	100	50%	50%	100%	0%
5	5	0.01	0.01	100	50%	50%	100%	0%
6	6	0.01	0.01	100	50%	50%	100%	0%
7	7	0.01	0.01	100	50%	50%	100%	0%
8	8	0.01	0.01	100	50%	50%	100%	0%
9	9	0.01	0.01	100	50%	50%	100%	0%
10	10	0.01	0.01	100	50%	50%	100%	0%

Grafik akurasi yang didapat pada skenario 2 dapat dilihat pada gambar 4.29.



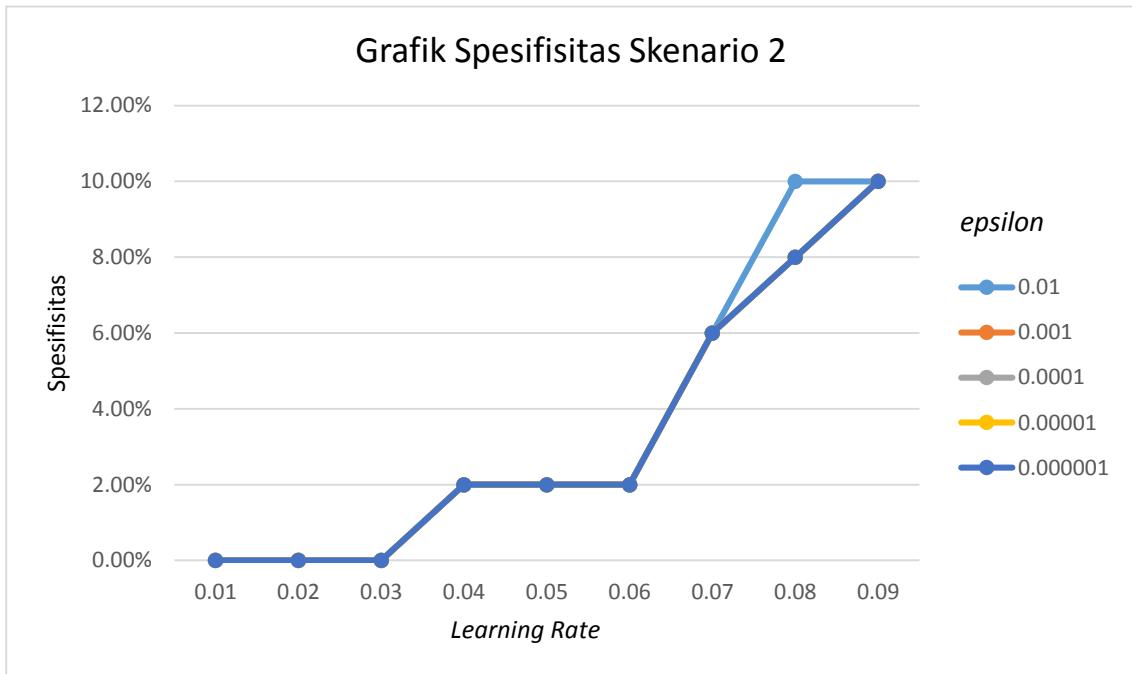
Gambar 4. 29. Grafik Akurasi Skenario 2

Sedangkan untuk sensitivitas pada akurasi tertinggi mencapai 98%. Hal ini menunjukkan bahwa kemampuan sistem untuk memberikan hasil positif terklasifikasi benar sebesar 98%. Grafik sensitivitas skenario 2 dapat dilihat pada gambar 4.30.



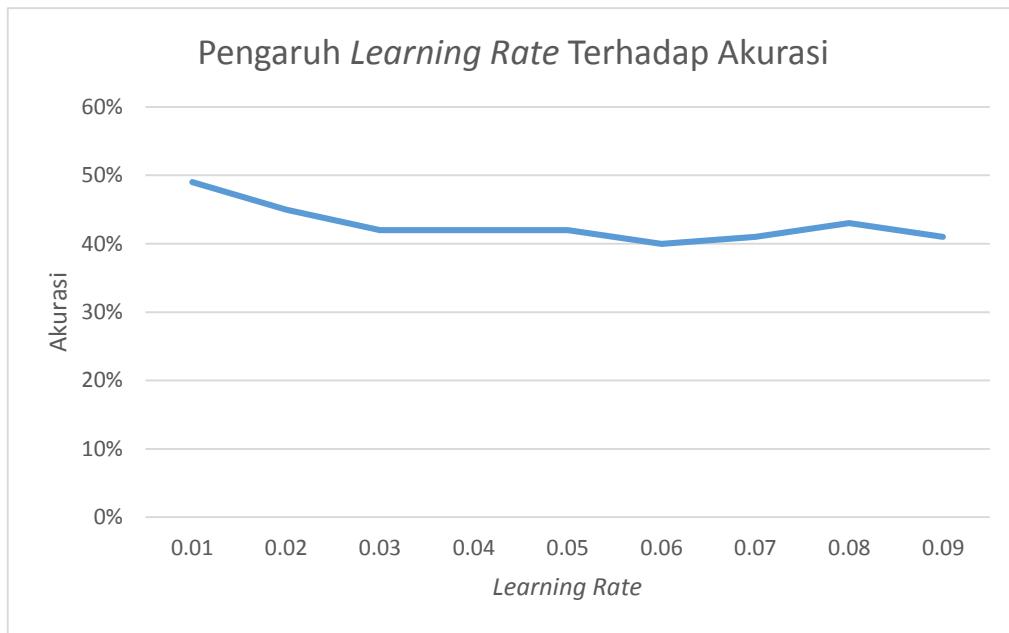
Gambar 4. 30. Grafik Sensitivitas Skenario 2

Tingkat spesifisitas pada akurasi tertinggi mencapai 0%. Hal ini menunjukkan bahwa sistem tidak mampu untuk memberikan hasil negatif terklasifikasi benar. Grafik spesifisitas skenario 2 dapat dilihat pada gambar 4.31.



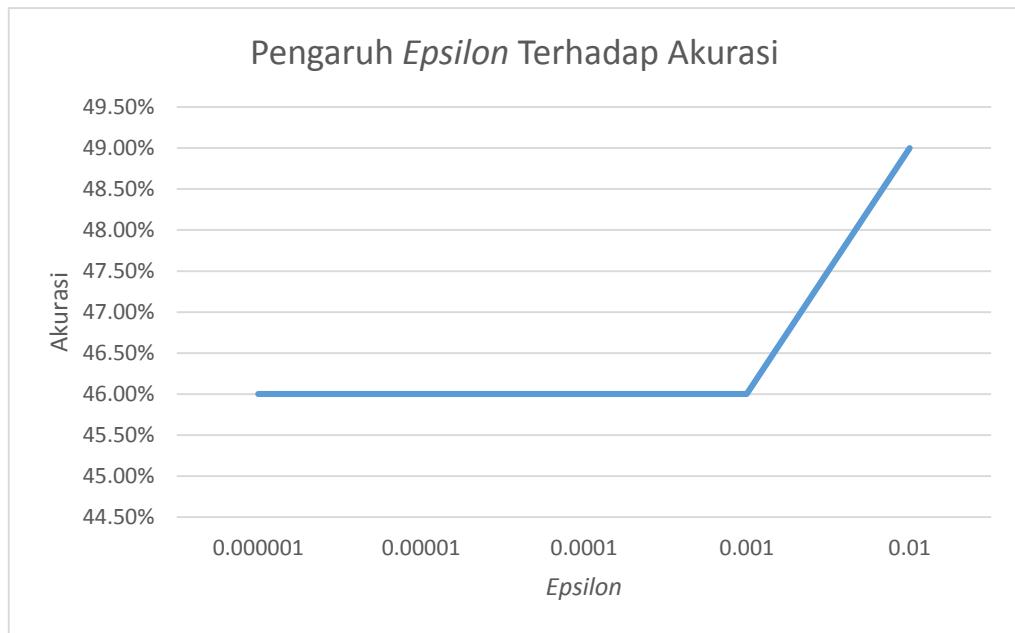
Gambar 4. 31. Grafik Spesifisitas Skenario 2

Berdasarkan hasil skenario 2, diambil salah satu kombinasi parameter yang memiliki akurasi tertinggi yaitu *learning rate* 0.01, *epsilon* 0.01, dan maksimum *epoch* 100, dengan akurasi mencapai angka 49%, sensitivitas 98%, dan spesifisitas 0%. Dari hasil tersebut, selanjutnya dilakukan analisa pengaruh *learning rate* dan *epsilon* terhadap akurasi. Pengaruh *learning rate* terhadap akurasi ditunjukkan pada gambar 4.32 dan pengaruh *epsilon* terhadap akurasi ditunjukkan pada gambar 4.33.



Gambar 4. 32. Grafik Pengaruh *Learning Rate* Terhadap Akurasi Skenario 2

Dari gambar 4.32, kesimpulan yang dapat diambil adalah semakin besar *learning rate*, maka tingkat akurasi yang dihasilkan cenderung semakin rendah. Tingkat akurasi tertinggi didapat pada *learning rate* 0.01. Hal ini terjadi karena pada saat *learning rate* 0.01, perubahan bobot yang terjadi memiliki nilai yang lebih kecil daripada *learning rate* yang lebih besar, sehingga perubahan bobot yang semakin kecil akan semakin mendekati vektor bobot awal.



Gambar 4. 33. Grafik Pengaruh *Epsilon* Terhadap Akurasi Skenario 2

Berdasarkan grafik pada gambar 4.33, kesimpulan yang dapat diambil adalah semakin besar *epsilon*, maka semakin tinggi akurasi yang dihasilkan. Akurasi tertinggi didapat pada *epsilon* 0.01. Hal ini dapat terjadi karena saat *epsilon* tinggi, proses iterasi yang terjadi sudah mendekati vektor bobot awal, sehingga akurasi yang dihasilkan tinggi.

4.4.3. Pembahasan Skenario 3

Skenario 3 ini bertujuan sebagai pembanding untuk skenario 2 dalam menentukan pengaruh antara *learning rate*, *epsilon*, serta bobot awal yang digunakan terhadap tingkat akurasi yang dihasilkan. Parameter yang digunakan pada skenario 3 ini sama seperti skenario 2, namun yang membedakan hanyalah bobot awal yang digunakan. Pada skenario 3 ini bobot awal yang digunakan diinisialisasi secara *random*. Hasil yang diperoleh dari skenario 3 dapat dilihat pada tabel 4.6.

Tabel 4. 6. Tabel Hasil Skenario 3

No	<i>Learning rate</i>	<i>Epsilon</i>	Akurasi	Error	Sensitivitas	Spesifisitas
1	0.01	0.01	86%	14%	94%	78%
2	0.02	0.01	82%	18%	74%	90%
3	0.03	0.01	44%	56%	62%	26%
4	0.04	0.01	52%	48%	98%	6%
5	0.05	0.01	41%	59%	38%	44%
6	0.06	0.01	48%	52%	50%	46%
7	0.07	0.01	52%	48%	88%	16%

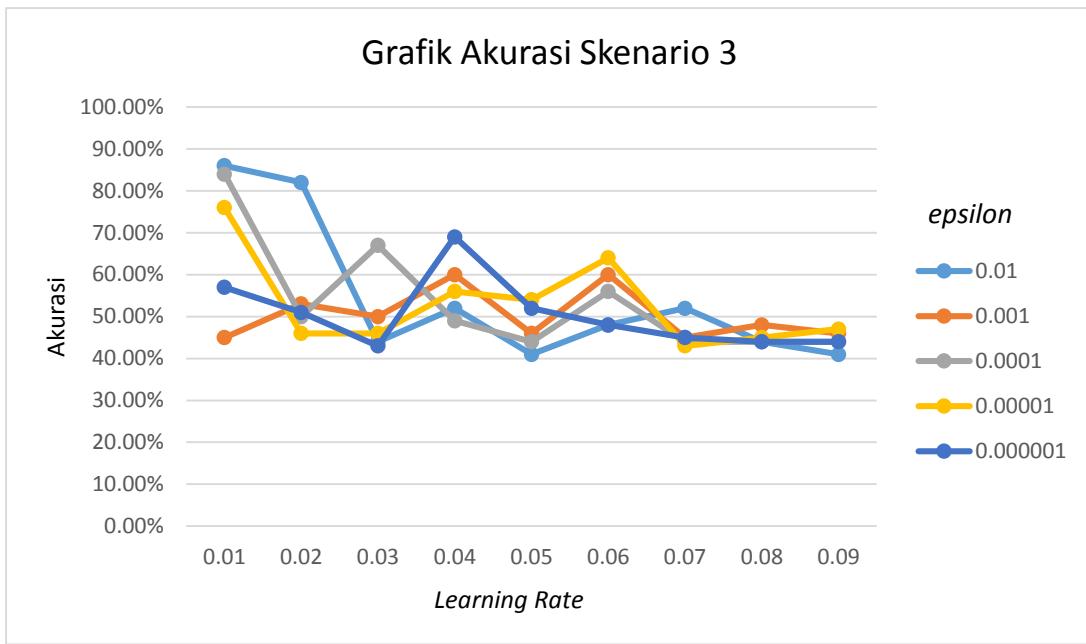
No	Learning rate	Epsilon	Akurasi	Error	Sensitivitas	Spesifisitas
8	0.08	0.01	44%	56%	62%	26%
9	0.09	0.01	41%	59%	66%	16%
10	0.01	0.001	45%	55%	82%	8%
11	0.02	0.001	53%	47%	78%	28%
12	0.03	0.001	50%	50%	100%	0%
13	0.04	0.001	60%	40%	92%	28%
14	0.05	0.001	46%	54%	80%	12%
15	0.06	0.001	60%	40%	74%	46%
16	0.07	0.001	45%	55%	80%	10%
17	0.08	0.001	48%	52%	74%	22%
18	0.09	0.001	46%	54%	82%	10%
19	0.01	0.0001	84%	16%	90%	78%
20	0.02	0.0001	50%	50%	100%	0%
21	0.03	0.0001	67%	33%	72%	62%
22	0.04	0.0001	49%	51%	90%	8%
23	0.05	0.0001	44%	56%	78%	10%
24	0.06	0.0001	56%	44%	96%	16%
25	0.07	0.0001	44%	56%	74%	14%
26	0.08	0.0001	44%	56%	78%	10%
27	0.09	0.0001	44%	56%	76%	12%
28	0.01	0.00001	76%	24%	94%	58%
29	0.02	0.00001	46%	54%	82%	10%
30	0.03	0.00001	46%	54%	90%	2%
31	0.04	0.00001	56%	44%	78%	34%
32	0.05	0.00001	54%	46%	98%	10%
33	0.06	0.00001	64%	36%	74%	54%
34	0.07	0.00001	43%	57%	86%	0%
35	0.08	0.00001	45%	55%	68%	20%
36	0.09	0.00001	47%	53%	74%	20%
37	0.01	0.000001	57%	43%	80%	34%
38	0.02	0.000001	51%	49%	58%	44%
39	0.03	0.000001	43%	57%	38%	48%
40	0.04	0.000001	69%	31%	90%	48%
41	0.05	0.000001	52%	48%	90%	14%
42	0.06	0.000001	48%	52%	70%	26%
43	0.07	0.000001	45%	55%	60%	30%
44	0.08	0.000001	44%	56%	68%	20%
45	0.09	0.000001	44%	56%	72%	16%

Pada tabel 4.4 dapat dilihat bahwa nilai akurasi tertinggi yang didapat ada pada nilai $\alpha = 0.01$, nilai $epsilon = 0.01$ dan maksimum $epoch = 100$ yaitu sebesar 86% dengan tingkat sensitivitas 94% dan spesifisitas sebesar 78%. Berikut hasil tingkat akurasi, $error$, sensitivitas dan spesifisitas secara detil tiap data kfold nilai $\alpha = 0.01$, nilai $epsilon = 0.01$ dan maksimum $epoch = 100$ yang ditunjukkan pada Tabel 4.7.

Tabel 4. 7. Detail Hasil Pengujian Dengan Akurasi Tertinggi Skenario 3

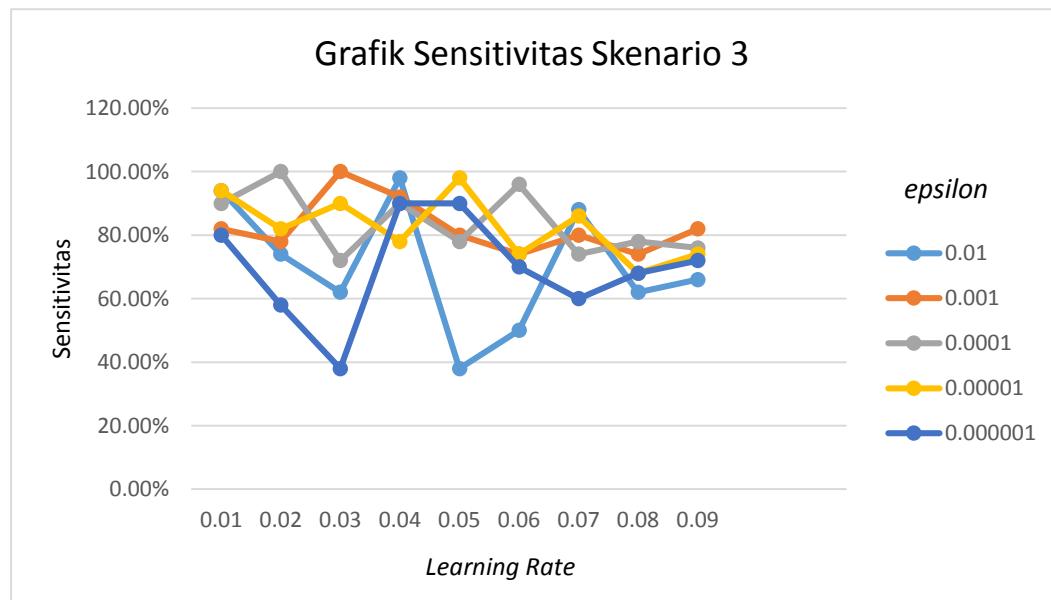
No.	Kfold	Learning Rate (α)	Error Minimum (eps)	Maksimum Epoch	Akurasi	Error	Sensitivitas	Spesifisitas
1	1	0.01	0.01	100	70%	30%	80%	60%
2	2	0.01	0.01	100	90%	10%	100%	80%
3	3	0.01	0.01	100	90%	10%	100%	80%
4	4	0.01	0.01	100	90%	10%	100%	80%
5	5	0.01	0.01	100	90%	10%	100%	80%
6	6	0.01	0.01	100	100%	0%	100%	100%
7	7	0.01	0.01	100	70%	30%	80%	60%
8	8	0.01	0.01	100	90%	10%	100%	80%
9	9	0.01	0.01	100	80%	20%	80%	80%
10	10	0.01	0.01	100	90%	10%	100%	80%

Jika dibandingkan dengan tingkat akurasi yang didapat dari skenario 2 sebelumnya, pada skenario 3 ini tingkat akurasi yang didapat lebih tinggi daripada skenario 2. Hal ini menunjukkan bahwa penggunaan bobot awal secara *random* memiliki tingkat akurasi yang lebih tinggi. Hal tersebut dikarenakan pada metode LVQ hasil klasifikasi didapat dari perhitungan nilai *euclidean distance* yang melibatkan bobot awal, sehingga jika bobot awal yang digunakan semakin merepresentasikan tiap kelas target maka hasil klasifikasinya akan semakin akurat. Grafik akurasi skenario 3 dapat dilihat pada gambar 4.34.



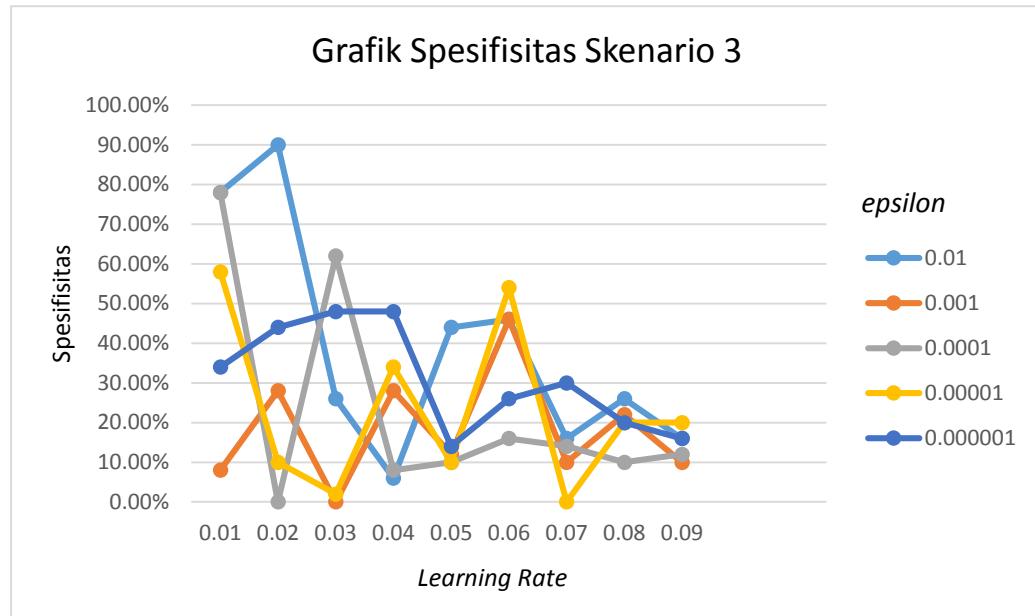
Gambar 4. 34. Grafik Akurasi Skenario 3

Sedangkan untuk sensitivitas pada akurasi tertinggi mencapai 94%. Hal ini menunjukkan bahwa kemampuan sistem untuk memberikan hasil positif terkласifikasi benar sebesar 94%. Grafik sensitivitas skenario 3 dapat dilihat pada gambar 4.35.



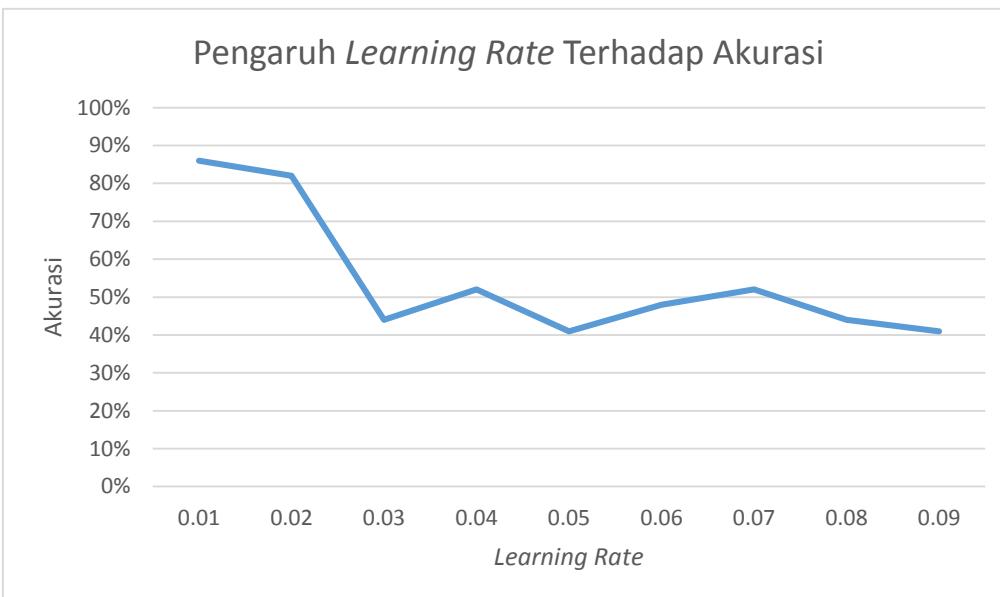
Gambar 4. 35. Grafik Sensitivitas Skenario 3

Dan untuk spesifisitas pada akurasi tertinggi mencapai angka 78%. Hal ini menunjukkan bahwa kemampuan sistem untuk memberikan hasil negatif terklasifikasi benar sebesar 78%. Grafik spesifisitas skenario 3 dapat dilihat pada gambar 4.36.



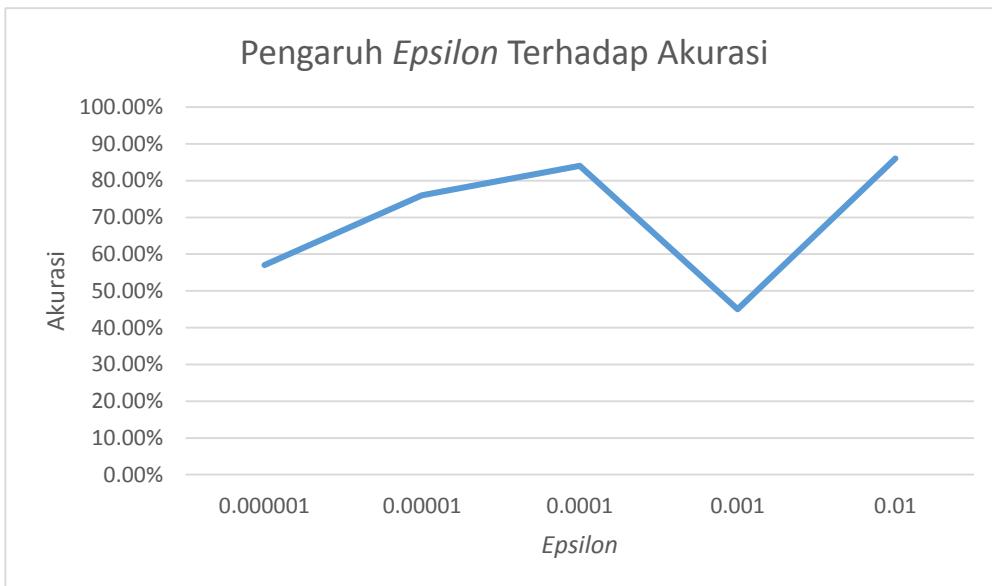
Gambar 4. 36. Grafik Spesifisitas Skenario 3

Berdasarkan hasil skenario 3, diambil salah satu kombinasi parameter yang memiliki akurasi tertinggi yaitu *learning rate* 0.01, ϵ 0.01, dan maksimum epoch 100, dengan akurasi mencapai angka 86%, sensitivitas 94%, dan spesifisitas 78%. Dari hasil tersebut, selanjutnya dilakukan analisa pengaruh *learning rate* dan ϵ terhadap akurasi. Pengaruh *learning rate* terhadap akurasi ditunjukkan pada gambar 4.37 dan pengaruh ϵ terhadap akurasi ditunjukkan pada gambar 4.38.



Gambar 4. 37. Grafik Pengaruh *Learning Rate* Terhadap Akurasi Skenario 3

Dari gambar 4.37, kesimpulan yang dapat diambil adalah semakin besar *learning rate*, maka tingkat akurasi yang dihasilkan cenderung semakin rendah. Tingkat akurasi tertinggi didapat pada *learning rate* 0.01. Hal ini terjadi karena pada saat *learning rate* 0.01, perubahan bobot yang terjadi memiliki nilai yang lebih kecil daripada *learning rate* yang lebih besar, sehingga perubahan bobot yang semakin kecil akan semakin mendekati vektor bobot awal.



Gambar 4. 38. Grafik Pengaruh *Epsilon* Terhadap Akurasi Skenario 3

Berdasarkan grafik pada gambar 4.38, kesimpulan yang dapat diambil adalah semakin besar *epsilon*, maka semakin tinggi akurasi yang dihasilkan. Akurasi tertinggi didapat pada *epsilon* 0.01. Hal ini dapat terjadi karena saat *epsilon* tinggi, proses iterasi yang terjadi sudah mendekati vektor bobot awal, sehingga akurasi yang dihasilkan tinggi.

4.5. Pengujian Fungsi Deteksi JST LVQ

Pengujian fungsi deteksi JST LVQ dilakukan dengan menggunakan arsitektur terbaik yang telah didapatkan sebelumnya. Pengujian ini menggunakan 20 data yang diambil dengan gejala yang berbeda, yang berjumlah masing-masing 10 data dari tiap kelas. Hasil pengujian fungsi deteksi dapat dilihat pada Lampiran 5.

Pada Lampiran 5, diperoleh hasil deteksi dengan jumlah deteksi yang sesuai dengan kelas target adalah 17 dan jumlah deteksi yang tidak sesuai dengan kelas target adalah 3 dari 20 deteksi. Hasil deteksi menunjukkan bahwa kombinasi nilai gejala yang berbeda memiliki pengaruh terhadap hasil deteksi. Namun, ada beberapa atribut gejala yang tidak memiliki

pengaruh besar terhadap hasil deteksi, karena nilainya yang cenderung stabil atau tidak memiliki perbedaan yang cukup besar pada dua kelas target, yaitu atribut umur dan actpp.

4.6. Evaluasi Hasil Skenario Pengujian

Berdasarkan keseluruhan skenario pengujian, didapatkan bahwa penggunaan maksimum *epoch* bernilai 100 menghasilkan tingkat akurasi yang lebih baik. Penggunaan bobot awal secara *random* memberikan hasil akurasi yang lebih baik dibandingkan inisialisasi bobot awal sebesar 0.5. Begitu juga dengan *learning rate* dan *epsilon*, semakin kecil *learning rate* yang digunakan dan juga semakin besar *epsilon* yang digunakan, maka akan menghasilkan tingkat akurasi yang lebih baik. Dari hasil tersebut dapat diketahui bahwa penggunaan bobot awal, *learning rate*, dan *epsilon* sangat berpengaruh terhadap tingkat akurasi yang dihasilkan.

Dari hasil skenario pengujian yang telah dilakukan maka dipilih parameter terbaik pada penelitian ini. Variasi parameter-parameter tersebut yaitu

- a. Inisialisasi bobot awal : *random*
- b. *Learning Rate* (α) : 0.01
- c. *Error minimum (epsilon)* : 0.01
- d. Maksimum *epoch* : 100

Dengan tingkat akurasi sebesar 86%, *error* 14%, sensitivitas 94%, dan spesifisitas 78%. Nilai sensitivitas deteksi diabetes melitus sebesar 94%, hal ini menunjukkan bahwa kemampuan sistem untuk memberikan hasil positif terklasifikasi benar pada kelas target sebesar 94%. Nilai spesifisitas 78% menunjukkan bahwa kemampuan sistem untuk memberikan hasil negatif terklasifikasi benar pada kelas target sebesar 78%. Nilai *error* 14% menunjukkan bahwa kemampuan sistem untuk mengidentifikasi salah seluruh subjek yang diuji sebesar 14%. Dan nilai akurasi 86% menunjukkan bahwa kemampuan sistem untuk mengidentifikasi benar seluruh subjek yang diuji sebesar 86%.

BAB V

PENUTUP

Bab ini membahas mengenai kesimpulan dari pengerajan uraian yang telah dijabarkan pada bab-bab sebelumnya dan saran untuk pengembangan penelitian lebih lanjut.

5.1. Kesimpulan

Kesimpulan yang dapat diambil dari penelitian mengenai Sistem Pendeksi Diabetes Melitus Menggunakan Jaringan Syaraf Tiruan *Learning Vector Quantization* (LVQ) adalah sebagai berikut :

1. Sistem Pendeksi Diabetes Melitus Menggunakan Jaringan Syaraf Tiruan *Learning Vector Quantization* (LVQ) telah berhasil dibangun dan dapat berjalan sesuai kebutuhan fungsional berdasarkan pengujian sistem.
2. Sistem Pendeksi Diabetes Melitus ini menggunakan metode jaringan syaraf tiruan LVQ dengan parameter eksperimen antara lain penentuan bobot awal, maksimum *epoch* dengan nilai 100, 500, dan 1000, *learning rate* (α) dengan parameter nilai antara 0.01 sampai 0.09 dan *error minimum* (*epsilon*) dengan parameter nilai antara 0.000001 sampai 0.01 untuk mengetahui pengaruh terhadap tingkat akurasi klasifikasi.
3. Dari hasil skenario pengujian yang dilakukan, Sistem Pendeksi Diabetes Melitus mampu melakukan klasifikasi dengan tingkat akurasi terbaik yaitu sebesar 86% dan tingkat *error* 14%, sensitivitas 94%, dan spesifisitas 78%. Hasil ini diperoleh dari arsitektur terbaik yaitu inisialisasi bobot awal yang diambil secara *random* dari data yang mewakili tiap kelas, maksimum *epoch* = 100, nilai *learning rate* (α) = 0.01 dan *epsilon* = 0.01.

5.2. Saran

Saran-saran yang dapat dilaksanakan untuk penelitian selanjutnya adalah sebagai berikut :

1. Inisialisasi bobot awal yang diambil dari data masing-masing kelas secara *random* terkadang menghasilkan akurasi yang tidak cukup baik sehingga perlu dilakukan uji coba berkali-kali agar dapat menghasilkan tingkat akurasi yang lebih optimal.
2. Penelitian ini disarankan menggunakan data yang memiliki pola yang benar-benar representatif terhadap kelasnya karena data dengan pola yang kurang representatif terhadap kelasnya menghasilkan tingkat akurasi yang kurang cukup baik.

3. Melakukan eksperimen algoritma LVQ dengan metode yang berkembang saat ini dan mengkombinasikan dengan parameter lain agar dapat meningkatkan akurasi yang lebih baik.
4. Menambah data yang digunakan, supaya meningkatkan kemampuan deteksi.
5. Melakukan penambahan seleksi fitur pada penelitian selanjutnya untuk menghilangkan atribut yang tidak memiliki pengaruh besar terhadap proses deteksi.

DAFTAR PUSTAKA

- Ariza, M. A. (2010). *The economic consequences of diabetes and cardiovascular disease in the United States*. Boston: University School of Medicine.
- Budianita, E. (2013). Penerapan Learning Vector Quantization untuk klasifikasi status gizi anak. *Indonesian Journal of Computing and Cybernetics Systems*.
- Fagustina, A., Syaifuddin, K., Ardita, V. R., & Hakim, F. A. (2011). *Memprediksi penyakit jantung koroner dengan menggunakan algoritma LVQ*. Surakarta: Universitas Sebelas Maret.
- Fathansyah. (2007). *Buku Teks Komputer Basis Data*. Bandung: Informatika.
- Fitri, R. (2007). Retrieved November 09, 2015, from digilib.itb.ac.id/files/.../jbptitbpp-gdl-radenfitri-28974-3-2007ta-2.pdf
- Gelinas, U. J., Oram, A. E., & Wiggins, W. P. (1990). Accounting Information System. PWS-KENT: Publishing Company.
- Hermawan, A. (2006). *Jaringan Syaraf Tiruan Teori dan Aplikasi*. Yogyakarta: Penerbit Andi.
- Hill, J. (2011). *Diabetes monitoring: risk factors, complications and management*. Birmingham: Birmingham Community Healthcare NHS Trust.
- Hospital Authority. (2016). *Diabetes Melitus (Kencing Manis)*. Jakarta: Departemen Kesehatan RI.
- Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation. *International Joint Conference on Article Intelligence (IJCAI)*.
- Leman. (1998). *Metodologi Pengembangan Sistem Informasi*. Jakarta: PT Elex Media Komputindo.
- Nurkhozin, A., Irawan, M., & Mukhlis, L. (2011). *Klasifikasi Penyakit Diabetes Melitus Menggunakan Jaringan Syaraf Tiruan Backpropagation dan Learning Vector Quantization*. Surabaya: Institut Teknologi Surabaya.
- Peranginangin, K. (2006). *Aplikasi Web dengan PHP dan MySQL*. Yogyakarta: Penerbit Andi.
- Praet, S. F. (2009). *Exercise therapy in type 2 diabetes*. Rotterdam: Erasmus University Medical Center.
- Pressman, R. S. (2001). *Software Engineering : A Practitioner's Approach Fifth Edition*. McGraw - Hill Companies, Inc.
- Pusptaningrum, D. (2006). *Pengantar jaringan Syaraf Tiruan*. Yogyakarta: Penerbit Andi.
- Ramachandran, A. d. (2004). Temporal changes in prevalence of diabetes and impaired glucose tolerance associated with lifestyle transition occurring in the rural population in India. *Diabetologia*, 860-865.

- Ranadhi, D., Indarto, W., & Hidayat, T. (2006). *Implementasi Learning Vector Quantization Untuk Pengenalan Pola Sidik Jari Pada Sistem Informasi Narapidana LP Wirogunan*. Media Informatika.
- Rosa, A., & M.Shalahuddin. (2013). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek* (1st ed.). Bandung: Informatika.
- T.Sutojo, E. M. (2011). *Kecerdasan Buatan*. Yogyakarta: ANDI Yogyakarta.
- Waspadji, S. (2007). *Diabetes Melitus : Apakah itu. Dalam Hidup Sehat dengan Diabetes*. Jakarta: Balai Penerbit FKUI.
- Zamani A M, A. B. (2012). Implementasi Algoritma Genetika pada Struktur Backpropagation Neural Network untuk Klasifikasi Kanker Payudara. *Jurnal Teknik ITS*, 222.

LAMPIRAN

Lampiran 1. Data Hasil Rekam Medis

Tabel L.1 Data Hasil Rekam Medis

id	umur	glun	gpost	upost	actn	urn	actpp	ldl	hdl	chol	tg	class
1.	53	77	161	0	0	0	0	26	77	208	176	Yes
2.	61	114	152	0	0	0	0	29	87	114	159	Yes
3.	57	88	83	0	0	0	0	31	28	87	139	Yes
4.	61	127	193	0	0	0	0	33	92	209	161	Yes
5.	61	119	238	1000	0	0	0	34	68	100	165	Yes
6.	56	132	172	0	0	0	0	35	89	143	300	Yes
7.	55	234	364	1000	0	1000	0	35	77	120	262	Yes
8.	68	117	155	3	0	1	0	37	75	109	167	Yes
9.	66	162	321	0	0	0	0	38	72	75	165	Yes
10.	62	128	161	0	0	0	0	39	75	137	164	Yes
11.	70	161	286	1000	0	1000	0	40	74	118	169	Yes
12.	54	116	190	0	0	0	0	41	74	87	178	Yes
13.	65	214	301	50	0	0	0	42	70	209	196	Yes
14.	70	284	407	1000	0	100	0	42	73	131	178	Yes
15.	42	339	480	1000	0	1000	0	42	69	134	277	Yes
16.	58	187	437	3	0	0	0	43	69	223	165	Yes
17.	78	137	146	0	0	0	0	44	73	231	368	Yes
18.	67	119	151	0	0	0	0	44	85	212	276	Yes
19.	65	111	176	0	0	0	0	46	65	123	165	Yes
20.	66	137	198	0	0	0	0	145	69	92	187	Yes
21.	70	125	205	100	0	0	0	46	66	116	170	Yes
22.	51	134	257	0	0	0	0	47	70	207	203	Yes
23.	78	133	188	0	0	0	0	48	70	131	165	Yes
24.	51	124	125	0	50	0	0	49	72	92	163	Yes
25.	73	114	123	0	0	0	0	50	65	124	164	Yes
26.	53	155	150	0	0	0	0	145	69	220	240	Yes
27.	63	124	141	0	0	0	0	51	67	131	163	Yes
28.	52	113	155	0	0	0	0	51	68	125	171	Yes
29.	77	123	158	0	0	0	0	51	76	106	194	Yes

id	umur	glun	gpost	upost	actn	urn	actpp	ldl	hdl	chol	tg	class
30.	59	242	308	1000	0	50	0	51	67	125	183	Yes
31.	84	127	138	0	15	0	0	52	71	113	160	Yes
32.	70	148	213	50	0	0	5	52	65	102	198	Yes
33.	76	160	166	300	0	100	0	52	60	107	177	Yes
34.	45	117	219	300	0	50	0	52	41	118	157	Yes
35.	66	130	216	0	0	0	0	53	72	103	189	Yes
36.	60	115	167	0	0	0	0	54	67	106	183	Yes
37.	77	118	212	0	0	0	0	54	85	158	194	Yes
38.	51	114	259	0	0	0	0	54	69	146	167	Yes
39.	52	115	264	0	0	0	0	54	72	113	174	Yes
40.	69	333	126	3	0	3	0	54	66	119	225	Yes
41.	76	180	303	300	0	0	0	54	71	139	172	Yes
42.	49	136	144	100	0	0	0	55	35	102	59	Yes
43.	66	123	154	0	5	0	0	55	72	138	207	Yes
44.	61	500	691	3	0	3	0	55	50	175	349	Yes
45.	66	186	365	1000	0	300	0	55	42	135	192	Yes
46.	71	211	224	50	0	0	0	56	69	158	198	Yes
47.	61	132	405	1000	0	0	0	56	63	177	338	Yes
48.	82	132	220	100	0	0	0	57	66	173	250	Yes
49.	48	132	189	1000	0	0	0	57	66	145	162	Yes
50.	65	126	185	0	0	0	0	58	67	108	163	Yes
51.	86	107	119	0	0	0	0	58	56	129	117	No
52.	60	107	117	0	0	0	0	58	52	147	125	No
53.	56	105	111	0	0	0	0	58	31	123	126	No
54.	71	82	120	0	0	0	0	8	51	89	151	No
55.	63	87	131	0	0	0	0	21	20	115	370	No
56.	84	75	107	0	0	0	0	25	20	83	169	No
57.	71	130	135	0	0	0	0	29	17	89	37	No
58.	52	84	93	0	0	0	0	31	65	115	137	No
59.	69	103	111	0	0	0	0	32	21	89	169	No
60.	51	61	105	50	0	100	0	35	21	124	138	No
61.	56	91	90	0	0	0	0	37	37	132	290	No
62.	69	80	129	0	0	0	0	37	13	64	70	No

id	umur	glun	gpost	upost	actn	urn	actpp	ldl	hdl	chol	tg	class
63.	73	94	93	0	0	0	0	39	44	108	125	No
64.	54	90	98	0	0	0	0	39	41	94	72	No
65.	90	254	260	0	0	0	0	41	37	98	102	No
66.	52	109	127	3	0	2	0	41	23	109	125	No
67.	18	66	93	0	0	0	0	42	35	125	239	No
68.	62	157	394	300	0	0	0	42	62	111	34	No
69.	84	107	104	0	0	0	0	43	28	104	164	No
70.	60	73	125	0	0	0	0	43	26	100	157	No
71.	51	92	111	0	0	0	0	45	47	113	107	No
72.	63	90	133	0	0	0	0	47	65	127	76	No
73.	54	134	188	0	0	0	0	47	47	121	136	No
74.	72	253	373	2	0	0	0	47	25	96	120	No
75.	55	93	96	0	0	0	0	48	41	129	200	No
76.	40	90	139	0	0	0	0	48	60	115	35	No
77.	61	68	106	0	0	0	0	50	50	111	53	No
78.	60	102	120	0	0	0	0	50	48	128	152	No
79.	65	119	150	0	0	0	0	50	30	112	161	No
80.	56	166	161	50	0	0	0	50	41	131	189	No
81.	62	90	91	0	0	0	0	51	31	126	219	No
82.	55	83	121	0	0	0	0	51	42	106	66	No
83.	81	88	161	0	0	0	0	51	67	130	60	No
84.	76	139	278	100	0	0	0	51	46	172	375	No
85.	66	91	151	0	0	0	0	52	45	122	125	No
86.	62	112	155	0	0	0	0	52	38	108	92	No
87.	58	116	168	0	0	0	0	53	30	149	331	No
88.	39	101	106	0	0	0	0	54	32	131	225	No
89.	62	102	119	0	0	0	0	54	60	132	89	No
90.	72	191	224	50	0	300	0	54	31	126	207	No
91.	67	91	143	0	0	0	0	55	20	96	106	No
92.	57	104	148	0	0	0	0	55	48	150	236	No
93.	80	82	87	0	0	0	0	56	40	105	43	No
94.	57	74	88	0	0	0	0	56	30	118	159	No
95.	92	90	104	0	0	0	0	56	79	152	85	No

id	umur	glun	gpost	upost	actn	urn	actpp	ldl	hdl	chol	tg	class
96.	67	163	205	0	0	0	0	56	22	91	64	No
97.	58	216	268	3	0	0	0	56	28	166	361	No
98.	52	105	130	0	0	0	0	57	44	111	50	No
99.	34	81	95	0	0	0	0	58	45	119	79	No
100	55	66	119	0	0	0	0	58	42	121	103	No

Lampiran 2. Implementasi Fungsi

1. Fungsi Pelatihan

```
<?php
include '../config/db_login.php';
session_start();
if(!isset($_SESSION['login_user'])){
    header('location:../index.php');
}

//ini_set('max_execution_time', 3600);
$time_start = 0;
pelatihan();

function pelatihan() {

    $query3 = mysql_query("SELECT MIN(umur), MIN(glun), MIN(gpost),
MIN(upost), MIN(actn), MIN(urn), MIN(actpp), MIN(ldl), MIN(hdl),
MIN(chol), MIN(tg) FROM data_diabetes");
    while ($row = mysql_fetch_array($query3)) {
        $min['umur'] = $row['MIN(umur)'];
        $min['glun'] = $row['MIN(glun)'];
        $min['gpost'] = $row['MIN(gpost)'];
        $min['upost'] = $row['MIN(upost)'];
        $min['actn'] = $row['MIN(actn)'];
        $min['urn'] = $row['MIN(urn)'];
        $min['actpp'] = $row['MIN(actpp)'];
        $min['ldl'] = $row['MIN(ldl)'];
        $min['hdl'] = $row['MIN(hdl)'];
        $min['chol'] = $row['MIN(chol)'];
        $min['tg'] = $row['MIN(tg)'];
    }
    $query4 = mysql_query("SELECT MAX(umur), MAX(glun), MAX(gpost),
MAX(upost), MAX(actn), MAX(urn), MAX(actpp), MAX(ldl), MAX(hdl),
MAX(chol), MAX(tg) FROM data_diabetes");
    while ($row = mysql_fetch_array($query4)) {
        $max['umur'] = $row['MAX(umur)'];
        $max['glun'] = $row['MAX(glun)'];
        $max['gpost'] = $row['MAX(gpost)'];
        $max['upost'] = $row['MAX(upost)'];
        $max['actn'] = $row['MAX(actn)'];
        $max['urn'] = $row['MAX(urn)'];
        $max['actpp'] = $row['MAX(actpp)'];
        $max['ldl'] = $row['MAX(ldl)'];
        $max['hdl'] = $row['MAX(hdl)'];
        $max['chol'] = $row['MAX(chol)'];
        $max['tg'] = $row['MAX(tg)'];
    }
    //inisiasi bobot random
    $a = rand(1,50);
    $b = rand(51,100);

    $w = array();
}
```

```

$random1 = mysql_query("SELECT
umur,glun,gpost,upost,actn,urn,actpp,ldl,hdl,chol,tg FROM
data_diabetes WHERE id='\$a'");
while ($row1 = mysql_fetch_array($random1)) {
    $w[1][1] = $row1['umur'];
    $w[1][2] = $row1['glun'];
    $w[1][3] = $row1['gpost'];
    $w[1][4] = $row1['upost'];
    $w[1][5] = $row1['actn'];
    $w[1][6] = $row1['urn'];
    $w[1][7] = $row1['actpp'];
    $w[1][8] = $row1['ldl'];
    $w[1][9] = $row1['hdl'];
    $w[1][10] = $row1['chol'];
    $w[1][11] = $row1['tg'];
}
$random2 = mysql_query("SELECT
umur,glun,gpost,upost,actn,urn,actpp,ldl,hdl,chol,tg FROM
data_diabetes WHERE id='\$b'");
while ($row2 = mysql_fetch_array($random2)) {
    $w[2][1] = $row2['umur'];
    $w[2][2] = $row2['glun'];
    $w[2][3] = $row2['gpost'];
    $w[2][4] = $row2['upost'];
    $w[2][5] = $row2['actn'];
    $w[2][6] = $row2['urn'];
    $w[2][7] = $row2['actpp'];
    $w[2][8] = $row2['ldl'];
    $w[2][9] = $row2['hdl'];
    $w[2][10] = $row2['chol'];
    $w[2][11] = $row2['tg'];
}

$w[1][1] = normalize($w[1][1], $min['umur'], $max['umur']);
$w[1][2] = normalize($w[1][2], $min['glun'], $max['glun']);
$w[1][3] = normalize($w[1][3], $min['gpost'], $max['gpost']);
$w[1][4] = normalize($w[1][4], $min['upost'], $max['upost']);
$w[1][5] = normalize($w[1][5], $min['actn'], $max['actn']);
$w[1][6] = normalize($w[1][6], $min['urn'], $max['urn']);
$w[1][7] = normalize($w[1][7], $min['actpp'], $max['actpp']);
$w[1][8] = normalize($w[1][8], $min['ldl'], $max['ldl']);
$w[1][9] = normalize($w[1][9], $min['hdl'], $max['hdl']);
$w[1][10] = normalize($w[1][10], $min['chol'], $max['chol']);
$w[1][11] = normalize($w[1][11], $min['tg'], $max['tg']);

$w[2][1] = normalize($w[2][1], $min['umur'], $max['umur']);
$w[2][2] = normalize($w[2][2], $min['glun'], $max['glun']);
$w[2][3] = normalize($w[2][3], $min['gpost'], $max['gpost']);
$w[2][4] = normalize($w[2][4], $min['upost'], $max['upost']);
$w[2][5] = normalize($w[2][5], $min['actn'], $max['actn']);
$w[2][6] = normalize($w[2][6], $min['urn'], $max['urn']);
$w[2][7] = normalize($w[2][7], $min['actpp'], $max['actpp']);
$w[2][8] = normalize($w[2][8], $min['ldl'], $max['ldl']);
$w[2][9] = normalize($w[2][9], $min['hdl'], $max['hdl']);
$w[2][10] = normalize($w[2][10], $min['chol'], $max['chol']);

```

```

$w[2][11] = normalize($w[2][11], $min['tg'], $max['tg']);
lvq($w);
}
//proses lvq
function lvq($w)
{
set_time_limit (0);
$time_start = microtime(true);
$w_lama = $w;

//mengambil data gejala
$sql = mysql_query("SELECT
umur,glun,gpost,upost,actn,urn,actpp,ldl,hdl,chol,tg FROM
data_diabetes");
$i = 0;
while ($row = mysql_fetch_array($sql)) {
    $data[$i]['umur'] = $row['umur'];
    $data[$i]['glun'] = $row['glun'];
    $data[$i]['gpost'] = $row['gpost'];
    $data[$i]['upost'] = $row['upost'];
    $data[$i]['actn'] = $row['actn'];
    $data[$i]['urn'] = $row['urn'];
    $data[$i]['actpp'] = $row['actpp'];
    $data[$i]['ldl'] = $row['ldl'];
    $data[$i]['hdl'] = $row['hdl'];
    $data[$i]['chol'] = $row['chol'];
    $data[$i]['tg'] = $row['tg'];
    //$data[$i]['class'] = $row['class'];
    $i++;
}
foreach($data[0] as $key => $value) {
    if ($key == 'class')
        continue;
    $array = to_array($data, $key);
    //min
    $min = cari_min($array);
    //maks
    $maks = cari_max($array);
    //nomralin
    for($i = 0; $i < count($data); $i++) {
        $data[$i][$key] = normalize($data[$i][$key], $min,
$maks);
    }
}

//data pelatihan tiap fold, yang tidak terpakai adalah data
pengujian
$kfold[1] = array(      //0,1,2,3,4,50,51,52,53,54,55,
5,6,7,8,9,55,56,57,58,59,
10,11,12,13,14,60,61,62,63,64,
15,16,17,18,19,65,66,67,68,69,
20,21,22,23,24,70,71,72,73,74,
25,26,27,28,29,75,76,77,78,79,

```

```

30,31,32,33,34,80,81,82,83,84,
35,36,37,38,39,85,86,87,88,89,
40,41,42,43,44,90,91,92,93,94,
45,46,47,48,49,95,96,97,98,99);
$kfold[2] = array(
    0,1,2,3,4,50,51,52,53,54,55,
    //5,6,7,8,9,55,56,57,58,59,
    10,11,12,13,14,60,61,62,63,64,
    15,16,17,18,19,65,66,67,68,69,
    20,21,22,23,24,70,71,72,73,74,
    25,26,27,28,29,75,76,77,78,79,
    30,31,32,33,34,80,81,82,83,84,
    35,36,37,38,39,85,86,87,88,89,
    40,41,42,43,44,90,91,92,93,94,
    45,46,47,48,49,95,96,97,98,99);
$kfold[3] = array(
    0,1,2,3,4,50,51,52,53,54,55,
    5,6,7,8,9,55,56,57,58,59,
    //10,11,12,13,14,60,61,62,63,64,
    15,16,17,18,19,65,66,67,68,69,
    20,21,22,23,24,70,71,72,73,74,
    25,26,27,28,29,75,76,77,78,79,
    30,31,32,33,34,80,81,82,83,84,
    35,36,37,38,39,85,86,87,88,89,
    40,41,42,43,44,90,91,92,93,94,
    45,46,47,48,49,95,96,97,98,99);
$kfold[4] = array(
    0,1,2,3,4,50,51,52,53,54,55,
    5,6,7,8,9,55,56,57,58,59,
    10,11,12,13,14,60,61,62,63,64,
    //15,16,17,18,19,65,66,67,68,69,
    20,21,22,23,24,70,71,72,73,74,
    25,26,27,28,29,75,76,77,78,79,
    30,31,32,33,34,80,81,82,83,84,
    35,36,37,38,39,85,86,87,88,89,
    40,41,42,43,44,90,91,92,93,94,
    45,46,47,48,49,95,96,97,98,99);
$kfold[5] = array(
    0,1,2,3,4,50,51,52,53,54,55,
    5,6,7,8,9,55,56,57,58,59,
    10,11,12,13,14,60,61,62,63,64,
    15,16,17,18,19,65,66,67,68,69,
    //20,21,22,23,24,70,71,72,73,74,
    25,26,27,28,29,75,76,77,78,79,
    30,31,32,33,34,80,81,82,83,84,
    35,36,37,38,39,85,86,87,88,89,
    40,41,42,43,44,90,91,92,93,94,
    45,46,47,48,49,95,96,97,98,99);
$kfold[6] = array(
    0,1,2,3,4,50,51,52,53,54,55,
    5,6,7,8,9,55,56,57,58,59,
    10,11,12,13,14,60,61,62,63,64,
    15,16,17,18,19,65,66,67,68,69,
    20,21,22,23,24,70,71,72,73,74,
    //25,26,27,28,29,75,76,77,78,79,
    30,31,32,33,34,80,81,82,83,84,
    35,36,37,38,39,85,86,87,88,89,
    40,41,42,43,44,90,91,92,93,94,
    45,46,47,48,49,95,96,97,98,99);

```

```

$kfold[7] = array(
    0,1,2,3,4,50,51,52,53,54,55,
    5,6,7,8,9,55,56,57,58,59,
    10,11,12,13,14,60,61,62,63,64,
    15,16,17,18,19,65,66,67,68,69,
    20,21,22,23,24,70,71,72,73,74,
    25,26,27,28,29,75,76,77,78,79,
    //30,31,32,33,34,80,81,82,83,84,
    35,36,37,38,39,85,86,87,88,89,
    40,41,42,43,44,90,91,92,93,94,
    45,46,47,48,49,95,96,97,98,99);
$kfold[8] = array(
    0,1,2,3,4,50,51,52,53,54,55,
    5,6,7,8,9,55,56,57,58,59,
    10,11,12,13,14,60,61,62,63,64,
    15,16,17,18,19,65,66,67,68,69,
    20,21,22,23,24,70,71,72,73,74,
    25,26,27,28,29,75,76,77,78,79,
    30,31,32,33,34,80,81,82,83,84,
    //35,36,37,38,39,85,86,87,88,89,
    40,41,42,43,44,90,91,92,93,94,
    45,46,47,48,49,95,96,97,98,99);
$kfold[9] = array(
    0,1,2,3,4,50,51,52,53,54,55,
    5,6,7,8,9,55,56,57,58,59,
    10,11,12,13,14,60,61,62,63,64,
    15,16,17,18,19,65,66,67,68,69,
    20,21,22,23,24,70,71,72,73,74,
    25,26,27,28,29,75,76,77,78,79,
    30,31,32,33,34,80,81,82,83,84,
    35,36,37,38,39,85,86,87,88,89,
    //40,41,42,43,44,90,91,92,93,94,
    45,46,47,48,49,95,96,97,98,99);
$kfold[10] = array(
    0,1,2,3,4,50,51,52,53,54,55,
    5,6,7,8,9,55,56,57,58,59,
    10,11,12,13,14,60,61,62,63,64,
    15,16,17,18,19,65,66,67,68,69,
    20,21,22,23,24,70,71,72,73,74,
    25,26,27,28,29,75,76,77,78,79,
    30,31,32,33,34,80,81,82,83,84,
    35,36,37,38,39,85,86,87,88,89,
    40,41,42,43,44,90,91,92,93,94,
    //45,46,47,48,49,95,96,97,98,99);

//mengambil target
$select_target      = "SELECT class FROM data_diabetes";
$hasil_target        = mysql_query($select_target);
while ($baris_target = mysql_fetch_array($hasil_target)){
    $target[]          = $baris_target['class'];
}
//Input maxepoch, alfa, epsilon
$alfa            = $_POST['alfa'];
$epsilon          = $_POST['epsilon'];
$MaxEpoch        = $_POST['max_epoch'];
$id              = 1;
//hapus data yang sama
$delete = "DELETE from nilai_bobot where nilai_bobot.alfa =
'$alfa' and nilai_bobot.epsilon = '$epsilon' and
nilai_bobot.max_epoch = '$MaxEpoch':";

```

```

$hasil_delete = mysql_query($delete) or die (mysql_error());
//perulangan tiap k_fold (k = 10)
for($kfold_ke = 1; $kfold_ke <= 10; $kfold_ke++) {
    $w
        = $w_lama;
    //inisialisasi epoch = 0. Perulangan epoch
    $epoch
        = 0;
    while($epoch < $MaxEpoch ) {
        if($alfa < $epsilon) {
            break;
        }
        $epoch++;
        //lvq perulangan semua data latih
        for($i = 1; $i <= 90; $i++) {
            //perulangan kelas
            for($k = 1; $k <= 2; $k++) {
                $f = ($kfold[$kfold_ke][$i-1]);
                $jarak[$i][$k] =
euclidean_distance($data[$f],$w[$k]);
            }
            $minimum = min($jarak[$i]);
            //mencari nilai minimum jarak di kelas, simpan k
            for($k = 1; $k <= 2; $k++) {
                if($jarak[$i][$k] == $minimum) {
                    $k_minimum[$i] = $k;
                }
            }
            $kunci = array_keys($data[0]);
            //perubahan bobot
            if($target[$i-1] == $k_minimum[$i]) {
                $k = $k_minimum[$i];
                for($j = 1; $j <= 11; $j++) {
                    $f = ($kfold[$kfold_ke][$i-1]);
                    $w[$k][$j] = $w[$k][$j] +
$alfa*($data[$f][$kunci[$j-1]] - $w[$k][$j]);
                }
            }
            else {
                $k = $k_minimum[$i];
                for($j = 1; $j <= 11; $j++) {
                    $f = ($kfold[$kfold_ke][$i-1]);
                    $w[$k][$j] = $w[$k][$j] -
$alfa*($data[$f][$kunci[$j-1]] - $w[$k][$j]);
                }
            }
            //perubahan alfa
            $alfa = min_alpa($alfa);
        }
        $w1 = $w[1][1].", ".$w[1][2].", ".$w[1][3].",
".$w[1][4].", ".$w[1][5].", ".$w[1][6].", ".$w[1][7].",
".$w[1][8].", ".$w[1][9].", ".$w[1][10].", ".$w[1][11];
        $w2 = $w[2][1].", ".$w[2][2].", ".$w[2][3].",
".$w[2][4].", ".$w[2][5].", ".$w[2][6].", ".$w[2][7].",
".$w[2][8].", ".$w[2][9].", ".$w[2][10].", ".$w[2][11];

```

```

$alfa_akhir = $alfa;
$alfa_awal = $_POST['alfa'];
$time_end = microtime(true);
$time = ($time_end - $time_start);
$insert_pelatihan = "INSERT INTO
nilai_bobot(alfa,epsilon,max_epoch,total_epoch,kfold,waktu,w1,w2,alfa_
akhir) VALUES
('$alfa_awal','$epsilon','$MaxEpoch','$epoch','$kfold_ke','$time','$w1
','$w2','$alfa_akhir')";
$hasil_pelatihan = mysql_query($insert_pelatihan) or die
(mysql_error());
header("location:tampil_data_pelatihan.php");
$alfa = $_POST['alfa'];
}
}

function euclidean_distance($data,$init)
{
$kunci = array_keys($data);
$seqma = 0;
for($i=1; $i<=11; $i++)
{
    $seqma = $seqma + pow($data[$kunci[$i-1]]-$init[$i],2);
}
$hasil = sqrt($seqma);

if(is_nan($hasil)){
    $hasil = 0;
}
return $hasil;
}

function min_alpa($alfa)
{
$hasil = $alfa - (0.1 * $alfa);
return $hasil;
}

function to_array($a, $nama_att){
$array = array(); $i = 0;
foreach ($a as $b){
    $array[$i] = $b[$nama_att];
    $i++;
}
return $array;
}

function cari_min($a){
$min = $a[0];
foreach ($a as $b){
    if ($min > $b){
        $min = $b;
    }
}
return $min;
}

function cari_max($a){
$max = $a[0];
foreach ($a as $b){

```

```

$max = $a[0];
foreach ($a as $b) {
    if ($max < $b) {
        $max = $b;
    }
}
return $max;
}

function normalize($data, $min, $maks) {
    $hasil = ($data - $min) / ($maks - $min);
    return $hasil;
}

?>

```

2. Fungsi Pengujian

```

<?php
include '../config/db_login.php';
session_start();
if(!isset($_SESSION['login_user'])){
    header('location:../index.php');
}

ini_set('max_execution_time', 3600);
uji();
function uji()
{
    //mengambil data di nilai bobot
    if(isset($_GET['id'])){
        $id_pelatihan      = $_GET['id'];

        //ambil alfa dan epsilon
        $hasil = mysql_query("SELECT alfa, epsilon, max_epoch FROM
nilai_bobot WHERE id_pelatihan='".$id_pelatihan"");
        while ($row = mysql_fetch_array($hasil)) {
            $alfa           = $row['alfa'];
            $epsilon         = $row['epsilon'];
            $epoch          = $row['max_epoch'];
        }
        $id = 1;
        //query menampilkan data
        $query = mysql_query("SELECT
umur,glun,gpost,upost,actn,urn,actpp,ldl,hdl,chol,tg FROM
data_diabetes ");
        $i      = 0;
        while ($row = mysql_fetch_array($query)) {
            $data[$i]['umur'] = $row['umur'];
            $data[$i]['glun'] = $row['glun'];
            $data[$i]['gpost'] = $row['gpost'];
            $data[$i]['upost'] = $row['upost'];
            $data[$i]['actn'] = $row['actn'];
            $data[$i]['urn'] = $row['urn'];
            $data[$i]['actpp'] = $row['actpp'];
        }
    }
}

```

```

    $data[$i]['ldl'] = $row['ldl'];
    $data[$i]['hdl'] = $row['hdl'];
    $data[$i]['chol'] = $row['chol'];
    $data[$i]['tg'] = $row['tg'];
    // $data[$i]['class'] = $row['class'];

    $i++;
}
foreach($data[0] as $key => $value){
    if ($key == 'class')
        continue;
    $array = to_array($data, $key);
    //min
    $min = cari_min($array);
    //maks
    $maks = cari_max($array);
    for($i = 0; $i < count($data); $i++){
        $data[$i][$key] = normalize($data[$i][$key], $min,
$maks);
    }
}
//hapus data yang sama
$delete = "DELETE from pengujian where id_pelatihan
='$id_pelatihan'";
$hasil_delete = mysql_query($delete) or die (mysql_error());
//data pengujian tiap kfold
$kfold[1] = array( 0,1,2,3,4,50,51,52,53,54);
$kfold[2] = array( 5,6,7,8,9,55,56,57,58,59);
$kfold[3] = array( 10,11,12,13,14,60,61,62,63,64);
$kfold[4] = array( 15,16,17,18,19,65,66,67,68,69);
$kfold[5] = array( 20,21,22,23,24,70,71,72,73,74);
$kfold[6] = array( 25,26,27,28,29,75,76,77,78,79);
$kfold[7] = array( 30,31,32,33,34,80,81,82,83,84);
$kfold[8] = array( 35,36,37,38,39,85,86,87,88,89);
$kfold[9] = array( 40,41,42,43,44,90,91,92,93,94);
$kfold[10] = array( 45,46,47,48,49,95,96,97,98,99);
for($kfold_ke=1; $kfold_ke<=10; $kfold_ke++){
    $query_bobot_akhir = "SELECT * FROM nilai_bobot WHERE
alfa='$alfa' and epsilon='$epsilon' and max_epoch='$epoch' and
kfold='$kfold_ke'";
    $hasil_query = mysql_query($query_bobot_akhir);
    while($q = mysql_fetch_array($hasil_query)){
        $w1 = $q['w1'];
        $w2 = $q['w2'];
        $id = $q['id_pelatihan'];
    }
    // Dipisah dibuat array dengan pemisah koma
    $w1ar = explode(",", $w1);
    $w2ar = explode(",", $w2);
    $w[1][1] = $w1ar[0];
    $w[1][2] = $w1ar[1];
    $w[1][3] = $w1ar[2];
    $w[1][4] = $w1ar[3];
    $w[1][5] = $w1ar[4];
    $w[1][6] = $w1ar[5];
    $w[1][7] = $w1ar[6];
    $w[1][8] = $w1ar[7];
}

```

```

$w[1][9] = $w1ar[8];
$w[1][10] = $w1ar[9];
$w[1][11] = $w1ar[10];
$w[2][1] = $w2ar[0];
$w[2][2] = $w2ar[1];
$w[2][3] = $w2ar[2];
$w[2][4] = $w2ar[3];
$w[2][5] = $w2ar[4];
$w[2][6] = $w2ar[5];
$w[2][7] = $w2ar[6];
$w[2][8] = $w2ar[7];
$w[2][9] = $w2ar[8];
$w[2][10] = $w2ar[9];
$w[2][11] = $w2ar[10];
//inisialisasi benar dan salah untuk menghitung akurasi dan
error
$benar = 0;
$salah = 0;
$aa = 0;
$ab = 0;
$ba = 0;
$bb = 0;
$x = 0;
//perulangan data uji
for($i=1;$i<=2; $i++) {
    for($j=1;$j<=5; $j++) {
        if ($i == 1 && $j == 1)
            $x = 1;
        if ($i == 1 && $j == 2)
            $x = 2;
        if ($i == 1 && $j == 3)
            $x = 3;
        if ($i == 1 && $j == 4)
            $x = 4;
        if ($i == 1 && $j == 5)
            $x = 5;
        if ($i == 2 && $j == 1)
            $x = 6;
        if ($i == 2 && $j == 2)
            $x = 7;
        if ($i == 2 && $j == 3)
            $x = 8;
        if ($i == 2 && $j == 4)
            $x = 9;
        if ($i == 2 && $j == 5)
            $x = 10;
    $f = ($kfold[$kfold_ke][$x-1]);
    $uji = $data[$f];
    for($k=1; $k<=2; $k++) {
        $jarak[$x][$k] =
euclidean_distance($uji, $w[$k]);
    }
    //mencari jarak minimal
    $min_jarak[$x] = min($jarak[$x]);
    //proses pencarian jarak paling sesuai
    for($k=1; $k<=2; $k++) {

```

```

        if($min_jarak[$x] == $jarak[$x][$k]) {
            $hasil = $k;
            echo "Hasil = ".$hasil."<br>";
            if ($i == 1){
                if($hasil == 1){
                    $aa = $aa + 1;
                }
                if($hasil == 2){
                    $ab = $ab + 1;
                }
            }
            if ($i == 2){
                if($hasil == 1){
                    $ba = $ba + 1;
                }
                if($hasil == 2){
                    $bb = $bb + 1;
                }
            }
        }
    }
}
//evaluasi menggunakan confusion matrix

//menghitung akurasi
$akurasi1 = ($aa+$bb) / ($aa+$ab+$ba+$bb);

//menghitung error
$error1 =($ab+$ba) / ($aa+$ab+$ba+$bb);

//menghitung sensitivitas
$sensitivitas1 = $aa / ($aa+$ab);

//menghitung spesifisitas
$spesifisitas1 = ($bb) / ($bb+$ba);

$insert_uji         = "INSERT INTO
pengujian(id_pelatihan,akurasi,error,sensitivitas,spesifisitas) VALUES
('{$id_pelatihan}', '{$akurasi1}', '{$error1}', '{$sensitivitas1}', '{$spesifisitas1}');
";
$hasil_uji         = mysql_query($insert_uji) or die (mysql_error());
$id_pelatihan++;
$id++;

header("location:tampil_data_pengujian.php");
echo '<tr>';
echo '<td>'.$w1.'</td>';
echo '<td>'.$w2.'</td>';
echo '<td>'.$epsilon.'</td>';
echo '<td>'.$alfa.'</td>';
echo '<td>'.$akurasi1.'</td>';
echo '<td>'.$error1.'</td>';
echo '<td>'.$sensitivitas1.'</td>';
echo '<td>'.$spesifisitas1.'</td>';
echo '</tr>';
}
}
}

```

```

//ambil dataset
function to_array($a, $nama_att) {
    $array = array(); $i = 0;
    foreach ($a as $b) {
        $array[$i] = $b[$nama_att];
        $i++;
    }
    return $array;
}
function cari_min($a) {
    $min = $a[0];
    foreach ($a as $b) {
        if ($min > $b) {
            $min = $b;
        }
    }
    return $min;
}
function cari_max($a) {
    $max = $a[0];
    foreach ($a as $b) {
        if ($max < $b) {
            $max = $b;
        }
    }
    return $max;
}
function normalize($data, $min, $maks) {
    $hasil = ($data - $min) / ($maks - $min);
    return $hasil;
}
function euclidean_distance($data,$init)
{
$seqma = 0;
$k = array_keys($data);
for($i=1; $i<=11; $i++)
{
    $seqma = $seqma + pow($data[$k[$i-1]]-$init[$i],2);
}
$hasil = sqrt($seqma);
return $hasil;
}
?>

```

3. Fungsi Deteksi

```
<?php
include '../config/db_login.php';
session_start();
if(!isset($_SESSION['login_user'])){
    header('location:../index.php');
}
klasifikasi();
function klasifikasi(){
$query3 = mysql_query("SELECT MIN(umur), MIN(glun), MIN(gpost),
MIN(upost), MIN(actn), MIN(urn), MIN(actpp), MIN(ldl), MIN(hdl),
MIN(chol), MIN(tg) FROM data_diabetes");
while ($row = mysql_fetch_array($query3)) {
    $min['umur'] = $row['MIN(umur)'];
    $min['glun'] = $row['MIN(glun)'];
    $min['gpost'] = $row['MIN(gpost)'];
    $min['upost'] = $row['MIN(upost)'];
    $min['actn'] = $row['MIN(actn)'];
    $min['urn'] = $row['MIN(urn)'];
    $min['actpp'] = $row['MIN(actpp)'];
    $min['ldl'] = $row['MIN(ldl)'];
    $min['hdl'] = $row['MIN(hdl)'];
    $min['chol'] = $row['MIN(chol)'];
    $min['tg'] = $row['MIN(tg)'];
}
$query4 = mysql_query("SELECT MAX(umur), MAX(glun), MAX(gpost),
MAX(upost), MAX(actn), MAX(urn), MAX(actpp), MAX(ldl), MAX(hdl),
MAX(chol), MAX(tg) FROM data_diabetes");
while ($row = mysql_fetch_array($query4)) {
    $max['umur'] = $row['MAX(umur)'];
    $max['glun'] = $row['MAX(glun)'];
    $max['gpost'] = $row['MAX(gpost)'];
    $max['upost'] = $row['MAX(upost)'];
    $max['actn'] = $row['MAX(actn)'];
    $max['urn'] = $row['MAX(urn)'];
    $max['actpp'] = $row['MAX(actpp)'];
    $max['ldl'] = $row['MAX(ldl)'];
    $max['hdl'] = $row['MAX(hdl)'];
    $max['chol'] = $row['MAX(chol)'];
    $max['tg'] = $row['MAX(tg)'];
}
//variabel masukan
$x = array();

$x[1] = $_POST['umur'];
$x[2] = $_POST['glun'];
$x[3] = $_POST['gpost'];
$x[4] = $_POST['upost'];
$x[5] = $_POST['actn'];
$x[6] = $_POST['urn'];
$x[7] = $_POST['actpp'];
$x[8] = $_POST['ldl'];
$x[9] = $_POST['hdl'];
$x[10] = $_POST['chol'];
```

```

$x[11] = $_POST['tg'];
$x[1] = normalize($x[1], $min['umur'], $max['umur']);
$x[2] = normalize($x[2], $min['glun'], $max['glun']);
$x[3] = normalize($x[3], $min['gpost'], $max['gpost']);
$x[4] = normalize($x[4], $min['upost'], $max['upost']);
$x[5] = normalize($x[5], $min['actn'], $max['actn']);
$x[6] = normalize($x[6], $min['urn'], $max['urn']);
$x[7] = normalize($x[7], $min['actpp'], $max['actpp']);
$x[8] = normalize($x[8], $min['ldl'], $max['ldl']);
$x[9] = normalize($x[9], $min['hdl'], $max['hdl']);
$x[10] = normalize($x[10], $min['chol'], $max['chol']);
$x[11] = normalize($x[11], $min['tg'], $max['tg']);
$query = "SELECT * FROM nilai_bobot WHERE status = '1'";
$hasil = mysql_query($query);
while($baris = mysql_fetch_array($hasil)) {
    $alfa = $baris['alfa'];
    $epsilon = $baris['epsilon'];
    $w1 = $baris['w1'];
    $w2 = $baris['w2'];
}
//memisahkan w1,w2,w3
$w1 = explode(", ", $w1);
$w2 = explode(", ", $w2);
    $w[1][1] = $w1[0];
    $w[1][2] = $w1[1];
    $w[1][3] = $w1[2];
    $w[1][4] = $w1[3];
    $w[1][5] = $w1[4];
    $w[1][6] = $w1[5];
    $w[1][7] = $w1[6];
    $w[1][8] = $w1[7];
    $w[1][9] = $w1[8];
    $w[1][10] = $w1[9];
    $w[1][11] = $w1[10];
    $w[2][1] = $w2[0];
    $w[2][2] = $w2[1];
    $w[2][3] = $w2[2];
    $w[2][4] = $w2[3];
    $w[2][5] = $w2[4];
    $w[2][6] = $w2[5];
    $w[2][7] = $w2[6];
    $w[2][8] = $w2[7];
    $w[2][9] = $w2[8];
    $w[2][10] = $w2[9];
    $w[2][11] = $w2[10];
//lvq
for($k=1; $k<=2; $k++) {
    $jarak[$k] = euclidean_distance($x, $w[$k]);
}
$min_jarak = min($jarak);
for($k=1; $k<=2; $k++) {
if($min_jarak == $jarak[$k]) {
    $hasil = $k;
}
}
}

```

```

if ($hasil == '1') {
    header('location:diabetes.php');
}
else if ($hasil == '2') {
    header('location:bukandiabetes.php');
}
}

function normalize($data, $min, $maks) {
    $hasil = ($data - $min) / ($maks - $min);
    return $hasil;
}

function euclidean_distance($x,$w)
{
    $seqma = 0;
    for($i=1; $i<=11; $i++)
    {
        $seqma = $seqma + pow($x[$i]-$w[$i],2);
    }
    $hasil = sqrt($seqma);
    return $hasil;
}
?>

```

Lampiran 3. Tabel Rencana Pengujian

Tabel L.3 Tabel Rencana Pengujian

SRS ID	Kebutuhan Fungsional	Butir Pengujian	Jenis Pengujian	Identifikasi
SRS-SPDM-F-001	<i>Otentikasi Login</i> untuk admin	1. Mengisi <i>username</i> dan <i>password</i> dengan benar 2. Memasukkan <i>username</i> yang salah 3. Memasukkan <i>password</i> yang salah 4. Memasukkan <i>username</i> dan <i>password</i> yang salah 5. Mengosongkan <i>username</i> dan <i>password</i> 6. Mengosongkan <i>username</i> 7. Mengosongkan <i>password</i>	<i>Black box</i>	U-001-001 U-001-002 U-001-003 U-001-004 U-001-005 U-001-006 U-001-007
SRS-SPDM-F-002	Mengubah <i>password</i>	1. <i>Password</i> lama benar, konfirmasi <i>password</i> baru benar, <i>password</i> baru benar. 2. <i>Password</i> lama salah 3. Konfirmasi <i>password</i> baru salah 4. Mengosongkan salah satu <i>field</i>	<i>Black box</i>	U-002-001 U-002-002 U-002-003 U-002-004
SRS-SPDM- F-003	<i>Import Dataset</i>	1. <i>Import file</i> dengan ekstensi yang benar 2. <i>Import file</i> dengan ekstensi yang tidak sesuai	<i>Black box</i>	U-003-001 U-003-002
SRS-SPDM- F-004	Menampilkan dataset	1. Menampilkan dataset yang telah di <i>import</i>	<i>Black box</i>	U-004-001
SRS-SPDM- F-005	Menampilkan data penanganan	1. Menampilkan data penanganan yang telah dibuat	<i>Black box</i>	U-005-001
SRS-SPDM- F-006	Mengubah data penanganan	1. Mengubah isi data penanganan	<i>Black box</i>	U-006-001
SRS- SPDM - F-007	Melakukan pelatihan	1. Semua <i>field</i> terisi dengan benar 2. Terdapat <i>field</i> yang tidak terisi dengan format yang benar 3. Mengosongkan <i>field</i>	<i>Black box</i>	U-007-001 U-007-002 U-007-003

SRS- SPDM- F-008	Menampilkan hasil pelatihan	1. Menampilkan semua data pelatihan	<i>Black box</i>	U-008-001
SRS- SPDM- F-009	Menghapus data pelatihan	1. Menghapus satu data pelatihan 2. Tidak jadi menghapus satu data pelatihan 3. Menghapus semua data pelatihan 4. Tidak jadi menghapus semua data pelatihan	<i>Black box</i>	U-009-001 U-009-002 U-009-003 U-009-004
SRS- SPDM- F-010	Melakukan pengujian	1. Melakukan pengujian dari Halaman Tampil Data Pelatihan 2. Melakukan pengujian dari Halaman Mulai Pengujian	<i>Black box</i>	U-010-001 U-010-002
SRS- SPDM- F-011	Menampilkan hasil pengujian	1. Menampilkan hasil pengujian	<i>Black box</i>	U-011-001
SRS- SPDM- F-012	Menghapus data pengujian	1. Menghapus satu data pengujian 2. Tidak jadi menghapus satu data pengujian 3. Menghapus semua data pengujian 4. Tidak jadi menghapus semua data pengujian	<i>Black box</i>	U-012-001 U-012-002 U-012-003 U-012-004
SRS- SPDM- F-013	Memilih bobot final untuk deteksi	1. Menampilkan bobot final yang akan dipilih	<i>Black box</i>	U-013-001
SRS- SPDM- F-014	Menampilkan bobot final yang digunakan untuk proses deteksi	1. Menampilkan detail bobot final yang telah dipilih untuk proses deteksi	<i>Black box</i>	U-014-001
SRS- SPDM- F-015	<i>Logout</i>	1. Melakukan <i>logout</i>	<i>Black box</i>	U-015-001
SRS- SPDM- F-016	Melakukan deteksi diabetes melitus	1. Melakukan deteksi diabetes melitus 2. Semua <i>field</i> terisi dengan format yang benar 3. Mengosongkan <i>field</i> 4. Mengisi dengan format yang tidak sesuai	<i>Black box</i>	U-016-001 U-016-002 U-016-003 U-016-004

Lampiran 4. Deskripsi dan Hasil Uji Pengujian Fungsional Sistem

1. Pengujian *Otentikasi Login* untuk Admin

Kode SRS : SRS – SPDM – F – 001

Deskripsi : *Login* bagi Admin

Tabel L4.1 Deskripsi dan Hasil Uji *Login* bagi Admin

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
U-001-001	Memasukkan <i>username</i> dan <i>password</i> dengan benar	1. Mengisi <i>field username</i> dan <i>password</i> . 2. Menekan tombol “Masuk”	<i>Username, password</i>	Ditampilkan halaman utama <i>Admin</i>	Berhasil <i>login</i> dan menampilkan halaman utama <i>admin</i>	Berhasil <i>login</i> dan menampilkan halaman utama <i>admin</i>	Diterima
U-001-002	Memasukkan <i>username</i> yang salah		<i>Username, password</i>	Sistem menampilkan peringatan kesalahan	Dapat menampilkan peringatan kesalahan	Dapat menampilkan peringatan kesalahan	Diterima
U-001-003	Memasukkan <i>password</i> yang salah		<i>Username, password</i>	Sistem menampilkan peringatan kesalahan	Dapat menampilkan peringatan kesalahan	Dapat menampilkan peringatan kesalahan	Diterima
U-001-004	Memasukkan <i>username</i> dan <i>password</i> yang salah		<i>Username, password</i>	Sistem menampilkan peringatan kesalahan	Dapat menampilkan peringatan kesalahan	Dapat menampilkan peringatan kesalahan	Diterima
U-001-005	Mengosongkan <i>field username</i> dan <i>password</i>		-	Sistem menampilkan peringatan kesalahan	Dapat menampilkan peringatan kesalahan	Dapat menampilkan peringatan kesalahan	Diterima
U-001-006	Hanya mengosongkan <i>field username</i>		<i>password</i>	Sistem menampilkan peringatan kesalahan	Dapat menampilkan peringatan kesalahan	Dapat menampilkan peringatan kesalahan	Diterima
U-001-007	Hanya mengosongkan <i>field password</i>		<i>Username</i>	Sistem menampilkan peringatan kesalahan	Dapat menampilkan peringatan kesalahan	Dapat menampilkan peringatan kesalahan	Diterima

2. Pengujian Mengubah Password

Kode SRS : SRS – SPDM – F – 002

Deskripsi : Mengubah *password* Admin

Tabel L4.2 Deskripsi dan Hasil Uji Mengubah *Password*

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
U-002-001	<i>Password</i> lama benar, <i>password</i> baru benar, konfirmasi <i>password</i> baru benar	1. Masuk ke halaman “Ubah <i>Password</i> ” pada tombol admin di pojok kanan atas halaman utama Admin 2. Mengisi <i>password</i> lama, <i>password</i> baru, dan konfirmasi <i>password</i> baru kemudian memilih tombol “Simpan”	<i>Password</i> lama, <i>password</i> baru, konfirmasi <i>password</i> baru	<i>Password</i> berhasil diubah	Muncul pesan: “ <i>Password</i> berhasil diubah”	Muncul pesan: “ <i>Password</i> berhasil diubah”	Diterima
U-002-002	<i>Password</i> lama salah		<i>Password</i> lama, <i>password</i> baru, konfirmasi <i>password</i> baru	<i>Password</i> tidak berhasil diubah	Muncul pesan: “ <i>Password</i> lama salah. <i>Password</i> tidak berhasil diubah”	Muncul pesan: “ <i>password</i> lama salah, <i>password</i> tidak berhasil diubah”	Diterima
U-002-003	<i>Password</i> lama benar, <i>password</i> baru benar, konfirmasi <i>password</i> baru salah		<i>Password</i> lama, <i>password</i> baru, konfirmasi <i>password</i> baru	<i>Password</i> tidak berhasil diubah	Muncul pesan: “ <i>Password</i> baru dan konfirmasi tidak sesuai. <i>Password</i> tidak berhasil diubah”	Muncul pesan: “ <i>password</i> baru dan konfirmasi tidak sesuai, <i>password</i> tidak berhasil diubah”	Diterima

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
U-002-004	Terdapat <i>field</i> yang belum terisi		<i>Password</i> lama, <i>password</i> baru,	<i>Password</i> tidak berhasil diubah	Muncul pesan: “Harap isi bidang ini” pada <i>field</i> yang belum terisi	Muncul pesan: “Harap isi bidang ini” pada <i>field</i> yang belum terisi	Diterima

3. Pengujian *Import* Dataset

Kode SRS : SRS – SPDM – 003

Deskripsi : Melakukan *import* dataset

Tabel L4.3 Deskripsi dan Hasil Uji Melakukan *Import* Dataset

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
U-003-001	<i>Import</i> atau unggah <i>file</i> dengan format yang benar	1. Admin masuk ke halaman “Kelola Data”	<i>File</i> dataset	Dataset berhasil di <i>import</i>	Muncul pesan: “Proses import data selesai. Dataset berhasil di-update”	Muncul pesan: “Proses import data selesai. Dataset berhasil di-update”	Diterima
U-003-002	<i>Import</i> atau unggah <i>file</i> dengan format yang salah	2. Memilih menu “ <i>Import</i> Dataset” 3. Memilih <i>file</i> yang akan di <i>import</i> dengan format yang ditentukan 4. Pilih tombol “Upload”	<i>File</i> dengan ekstensi tidak sesuai dengan format yang ditentukan	Dataset tidak berhasil di <i>import</i>	Muncul pesan penjelasan format yang benar	Muncul pesan penjelasan format yang benar	Diterima

4. Pengujian Menampilkan Dataset

Kode SRS : SRS – SPDM - F – 004

Deskripsi : Menampilkan dataset

Tabel L4.4. Deskripsi dan Hasil Uji Menampilkan Dataset

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
U-004-001	Menampilkan dataset yang telah di <i>import</i>	<ol style="list-style-type: none"> 1. Admin masuk ke Halaman “Kelola Data”. 2. Memilih menu Lihat Dataset. 	-	Dataset yang tersimpan di <i>database</i> ditampilkan	Dataset yang tersimpan di <i>database</i> ditampilkan	Dataset yang tersimpan di <i>database</i> ditampilkan sesuai dengan yang sudah di <i>import</i>	Diterima

5. Pengujian Menampilkan Data Penanganan

Kode SRS : SRS - SPDM - F – 005

Deskripsi : Menampilkan data penanganan

Tabel L4.5 Deskripsi dan Hasil Uji Menampilkan Data Penanganan

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
U-005-001	Menampilkan data penanganan yang ada	<ol style="list-style-type: none"> 1. Admin masuk ke halaman “Kelola Data” 2. Memilih menu Lihat Penanganan. 	-	Semua data penanganan ditampilkan	Semua data penanganan ditampilkan	Semua data penanganan ditampilkan	Diterima

6. Pengujian Mengubah Data Penanganan

Kode SRS : SRS - SPDM - F - 006

Deskripsi : Mengubah data penanganan

Tabel L4.6 Deskripsi dan Hasil Uji Mengubah Data Penanganan

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
U-006-001	Mengubah Isi Data Penanganan	<ol style="list-style-type: none"> 1. Admin masuk ke halaman “Kelola Data” 2. Memilih menu Lihat Penanganan. 3. Menekan tombol “Edit Penanganan” 4. Mengubah data penanganan dan menekan tombol “Simpan” 	Data Penanganan	Data Penanganan berhasil diubah dan ditampilkan	Data Penanganan berhasil diubah dan ditampilkan	Data Penanganan berhasil diubah dan ditampilkan	Diterima

7. Pengujian Melakukan Pelatihan

Kode SRS : SRS - SPDM - F - 007

Deskripsi : Melakukan pelatihan

Tabel L4.7 Deskripsi dan Hasil Uji Melakukan Pelatihan

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
U-007-001	Semua <i>field</i> terisi dengan format yang benar	1. Admin masuk ke halaman “Pelatihan & Pengujian” 2. Memilih menu “Mulai Pelatihan” 3. Mengisi <i>field</i> parameter pelatihan <i>alfa</i> , <i>epsilon</i> , dan maksimum <i>epoch</i> 4. Memilih tombol “Submit”	<i>Alfa</i> , <i>epsilon</i> , dan <i>Max epoch</i>	Pelatihan dimulai	Pelatihan dilakukan, data ter-update.	Pelatihan dilakukan, data ter-update.	Diterima
U-007-002	Tidak semua <i>field</i> terisi dengan format yang benar		<i>Alfa</i> , <i>epsilon</i> , dan <i>Max epoch</i>	Pelatihan tidak dapat dimulai	Sistem menolak inputan dengan format yang tidak sesuai	Sistem menolak inputan dengan format yang tidak sesuai	Diterima
U-007-003	Mengosongkan <i>field</i>		-	Pelatihan tidak dapat dimulai	Muncul pesan “Harap isi bidang ini”	Muncul pesan “Harap isi bidang ini”	Diterima

8. Pengujian Menampilkan Hasil Pelatihan

Kode SRS : SRS - SPDM - F - 008

Deskripsi : Menampilkan data hasil pelatihan

Tabel L4.8 Deskripsi dan Hasil Uji Menampilkan Hasil Pelatihan

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
U-008-001	Menampilkan semua data pelatihan	<ol style="list-style-type: none"> 1. Admin masuk ke Halaman “Pelatihan & Pengujian” 2. Memilih menu Data Pelatihan 	-	Semua data pelatihan ditampilkan	Semua data pelatihan ditampilkan	Semua data pelatihan ditampilkan	Diterima

9. Pengujian Menghapus Data Pelatihan

Kode SRS : SRS - SPDM - F - 009

Deskripsi : Menghapus data pelatihan

Tabel L4.9 Deskripsi dan Hasil Uji Menghapus Daftar Pelatihan

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
U-009-001	Menghapus satu data pelatihan	<ol style="list-style-type: none"> 1. Masuk ke halaman “Pelatihan & Pengujian” 2. Memilih tombol “Hapus” pada satu baris data pelatihan 3. Memilih tombol “OK” pada dialog konfirmasi 	-	Satu baris data pelatihan berhasil terhapus	Satu baris data pelatihan berhasil terhapus dan data ter-update	Satu baris data pelatihan berhasil terhapus dan data ter-update	Diterima
U-009-002	Tidak jadi menghapus satu data pelatihan	1. Masuk ke halaman “Pelatihan &	-	Data pelatihan tidak berhasil terhapus	Tidak terjadi apa-apa	Tidak terjadi apa-apa	Diterima

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
		<p>Pengujian”</p> <p>2. Memilih tombol “Hapus” pada satu baris data pelatihan</p> <p>3. Memilih tombol “Cancel” pada dialog konfirmasi</p>	-	Seluruh data pelatihan berhasil terhapus	Tabel data pelatihan kosong	Tabel data pelatihan kosong	Diterima
U-009-003	Menghapus semua data pelatihan	<p>1. Masuk ke halaman “Pelatihan & Pengujian”</p> <p>2. Memilih tombol “Hapus Semua Pelatihan” yang terletak di bawah tabel</p> <p>3. Memilih tombol “OK” pada dialog konfirmasi</p>	-	Tidak ada data pelatihan yang berhasil terhapus	Tidak terjadi apa-apa	Tidak terjadi apa-apa	
U-009-004	Tidak jadi menghapus semua data pelatihan	<p>1. Masuk ke halaman “Pelatihan & Pengujian”</p> <p>2. Memilih tombol “Hapus Semua Pelatihan” yang</p>	-				

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
		<p>terletak di bawah tabel</p> <p>3. Memilih tombol “Cancel” pada dialog konfirmasi</p>					

10. Pengujian Melakukan Pengujian

Kode SRS : SRS - SPDM - F – 010

Deskripsi : Melakukan pengujian

Tabel L4.10 Deskripsi dan Hasil Uji Melakukan Pengujian

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
U-010-001	Melakukan pengujian dari Halaman “Data Pelatihan”	<p>1. Admin masuk ke halaman “Pelatihan & Pengujian”</p> <p>2. Memilih menu “Data Pelatihan”</p> <p>3. Memilih tombol “Lakukan Pengujian”</p>	-	Proses pengujian dilakukan	Muncul hasil pengujian	Muncul hasil pengujian	Diterima
U-010-002	Melakukan pengujian dari Halaman “Mulai Pengujian”	1. Admin masuk ke halaman	-	Proses pengujian dilakukan	Muncul hasil pengujian	Muncul hasil pengujian	Diterima

“Pelatihan &
Pengujian”

2. Memilih menu
“Mulai
Pengujian”
3. Memilih tombol
“Lakukan
Pengujian”

11. Pengujian Menampilkan Hasil Pengujian

Kode SRS : SRS - SPDM- F – 011

Deskripsi : Melakukan pengujian

Tabel L4.11 Deskripsi dan Hasil Uji Menampilkan Hasil Pengujian

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
U-011-001	Menampilkan hasil pengujian	1. Admin masuk ke halaman “Pelatihan & Pengujian” 2. Memilih menu “Data Pengujian”	-	Semua data pengujian ditampilkan	Semua data pengujian ditampilkan	Semua data pengujian ditampilkan	Diterima

12. Pengujian Menghapus Data Pengujian

Kode SRS : SRS - SPDM - F – 012

Deskripsi : Menghapus data pelatihan

Tabel L4.12 Deskripsi dan Hasil Uji Menghapus Data Pengujian

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
U-012-001	Menghapus satu data pengujian	1. Masuk ke halaman “Pelatihan & Pengujian” 2. Memilih tombol “Hapus” pada satu baris data pengujian 3. Memilih tombol “OK” pada dialog konfirmasi	-	Satu baris data pengujian berhasil terhapus	Satu baris data pengujian berhasil terhapus dan data ter-update	Satu baris data pengujian berhasil terhapus dan data ter-update	Diterima
U-012-002	Tidak jadi menghapus satu data pelatihan	1. Masuk ke halaman “Pelatihan & Pengujian” 2. Memilih tombol “Hapus” pada satu baris data pengujian 3. Memilih tombol “Cancel” pada dialog konfirmasi	-	Data pengujian tidak berhasil terhapus	Tidak terjadi apa-apa	Tidak terjadi apa-apa	Diterima
U-012-003	Menghapus semua data pelatihan	1. Masuk ke halaman	-	Seluruh data pengujian berhasil terhapus	Tabel data pengujian kosong	Tabel data pengujian kosong	Diterima

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
		<p>“Pelatihan & Pengujian”</p> <p>2. Memilih tombol “Hapus Semua Pengujian” yang terletak di bawah tabel</p> <p>3. Memilih tombol “OK” pada dialog konfirmasi</p>	-	-	-	-	-
U-012-004	Tidak jadi menghapus semua data pelatihan	<p>1. Masuk ke halaman “Pelatihan & Pengujian”</p> <p>2. Memilih tombol “Hapus Semua Pengujian” yang terletak di bawah tabel</p> <p>3. Memilih tombol “Cancel” pada dialog konfirmasi</p>	-	<p>Tidak ada data pengujian yang berhasil terhapus</p>	<p>Tidak terjadi apa-apa</p>	<p>Tidak terjadi apa-apa</p>	<p>Tidak terjadi apa-apa</p>

13. Pengujian Memilih Bobot Final Untuk Deteksi

Kode SRS : SRS - SPDM - F - 013

Deskripsi : Memilih data pelatihan yang akan menjadi bobot final dan digunakan untuk proses deteksi

Tabel L4.13 Deskripsi dan Hasil Uji Memilih Bobot Final Untuk Deteksi

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
U-013-001	Memilih data pelatihan yang bobot akhirnya akan digunakan untuk proses deteksi	<ol style="list-style-type: none"> 1. Masuk ke halaman “Pelatihan & Pengujian” 2. Memilih menu “Data Pengujian” 3. Memilih tombol “Lihat Detail” pada data pengujian yang memiliki akurasi rata-rata tertinggi 4. Memilih tombol “Simpan Bobot” pada data fold yang 		Bobot yang digunakan untuk proses deteksiberubah	Masuk ke halaman “Bobot Final”, data yang dipilih ditampilkan	Masuk ke halaman “Bobot Final”, data yang dipilih ditampilkan	Diterima

memiliki
akurasi
tertinggi

14. Pengujian Menampilkan Bobot Final Yang Digunakan Untuk Deteksi

Kode SRS : SRS - SPDM - F - 014

Deskripsi : Menampilkan data pelatihan yang sedang digunakan untuk deteksi

Tabel L4.14 Deskripsi dan Hasil Uji Menampilkan Bobot final yang digunakan

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
U-014-001	Menampilkan detail data pelatihan yang bobot akhirnya sedang digunakan untuk proses deteksi	1. Masuk ke halaman “Bobot Final”	-	Data pelatihan yang bobot finalnya sedang digunakan untuk proses deteksi ditampilkan beserta akurasi dan <i>error</i>	Data pelatihan yang bobot finalnya sedang digunakan untuk proses deteksi ditampilkan beserta akurasi dan <i>error</i>	Data pelatihan yang bobot finalnya sedang digunakan untuk proses deteksi ditampilkan beserta akurasi dan <i>error</i>	Diterima

15. Pengujian *Log out*

Kode SRS : SRS - SPDM - F - 015

Deskripsi : *Log out* untuk admin

Tabel L4.15 Deskripsi dan Hasil Uji *Log out* dari sistem

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
U-015-001	Melakukan <i>Logout</i>	1. Memilih tombol “Keluar” pada menu admin	-	Admin keluar dari halaman dan ditampilkan halaman awal <i>login</i>	Masuk ke halaman <i>login</i>	Masuk ke halaman <i>login</i>	Diterima

16. Pengujian Melakukan Deteksi Diabetes Melitus

Kode SRS : SRS - SPDM - F - 016

Deskripsi : Melakukan deteksi penyakit apakah pengguna berisiko menderita diabetes melitus atau tidak

Tabel L4.16 Deskripsi dan Hasil Uji Melakukan Deteksi Penyakit

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Hasil	Hasil yang Diperoleh	Kesimpulan
U-016-001	Melakukan deteksi penyakit diabetes melitus	1. Masuk ke halaman utama/ <i>home</i>	Data gejala diabetes meitus (gejala pengguna)	Mendeteksi apakah pengguna berisiko menderita diabetes melitus atau tidak	Hasil deteksi ditampilkan	Hasil deteksi ditampilkan	Diterima
U-016-002	Sebuah <i>field</i> terisi dengan format yang benar	2. Mengisi semua atribut yang dibutuhkan 3. Memilih tombol “Simpan”	Data gejala diabetes meitus (gejala pengguna) dengan format yang benar	Deteksi dimulai	Deteksi dilakukan, hasil deteksi ditampilkan	Deteksi dilakukan, hasil deteksi ditampilkan	Diterima
U-016-003	Terdapat <i>field</i> yang terisi dengan format yang tidak sesuai		Data gejala diabetes meitus (gejala pengguna) dengan format yang tidak sesuai	Deteksi tidak dapat dimulai	Sistem menolak inputan dengan format yang tidak sesuai	Sistem menolak inputan dengan format yang tidak sesuai	Diterima
U-016-004	Mengosongkan <i>field</i>		-	Deteksi tidak dapat dimulai	Muncul pesan “Harap isi bidang ini”	Muncul pesan “Harap isi bidang ini”	Diterima

Lampiran 5. Hasil Pengujian Fungsi Deteksi JST LVQ

Tabel L.5 Data Hasil Pengujian Fungsi Deteksi JST LVQ

No.	id dataset	umur	glun	gpost	upost	actn	urn	actpp	ldl	hdl	chol	tg	class target	class hasil deteksi
1.	1	53	77	161	0	0	0	0	26	77	208	176	Yes	Yes
2.	2	61	114	152	0	0	0	0	29	87	114	159	Yes	Yes
3.	3	57	88	83	0	0	0	0	31	28	87	139	Yes	No
4.	7	55	234	364	1000	0	1000	0	35	77	120	262	Yes	Yes
5.	11	70	161	286	1000	0	1000	0	40	74	118	169	Yes	Yes
6.	13	65	214	301	50	0	0	0	42	70	209	196	Yes	Yes
7.	16	58	187	437	3	0	0	0	43	69	223	165	Yes	Yes
8.	24	51	124	125	0	50	0	0	49	72	92	163	Yes	Yes
9.	43	66	123	154	0	5	0	0	55	72	138	207	Yes	Yes
10.	44	61	500	691	3	0	3	0	55	50	175	349	Yes	Yes
11.	54	71	82	120	0	0	0	0	8	51	89	151	No	No
12.	56	84	75	107	0	0	0	0	25	20	83	169	No	No
13.	58	52	84	93	0	0	0	0	31	65	115	137	No	Yes
14.	59	69	103	111	0	0	0	0	32	21	89	169	No	No
15.	60	51	61	105	50	0	100	0	35	21	124	138	No	No
16.	66	52	109	127	3	0	2	0	41	23	109	125	No	No
17.	68	62	157	394	300	0	0	0	42	62	111	34	No	Yes
18.	74	72	253	373	2	0	0	0	47	25	96	120	No	No
19.	90	72	191	224	50	0	300	0	54	31	126	207	No	No
20.	97	58	216	268	3	0	0	0	56	28	166	361	No	No