



**PENERAPAN LEARNING VECTOR QUANTIZATION (LVQ)
UNTUK MENGKLASIFIKASIKAN TENAGA AHLI IT
(STUDI KASUS: PT. CITA KREASI LATENA)**

Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Menempuh Ujian Akhir
Memperoleh Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH

NAMA : ALWI RAHMAN DAMANIK
NPM : 1724370945
PROGRAM STUDI : SISTEM KOMPUTER

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2020**

ABSTRAK

ALWI RAHMAN DAMANIK

**Penerapan Learning Vector Quantization (LVQ) Untuk Mengklasifikasikan
Tenaga Ahli IT (Studi Kasus: PT. Cita Kreasi Latena)
2020**

Skripsi ini membahas tentang klasifikasi Tenaga Ahli IT dengan menggunakan Jaringan Syaraf Tiruan dengan metode *Learning Vector Quantization*. Tenaga ahli ialah orang pribadi yang mempunyai keahlian khusus yang dalam memberikan jasa berdasarkan keahliannya tersebut tidak terikat oleh hubungan kerja (melakukan pekerjaan bebas/memberikan professional services), misalnya akuntan, dokter, pengacara, notaris, programmer, konsulen pajak, arsitek, designer dan sebagainya. Tenaga ahli tersebut memberikan jasa yang lazim disebut jasa professional, sedang pekerjaan yang dilakukannya tanpa hubungan kerja lazim disebut juga pekerjaan bebas. Maka dilakukannya penelitian dengan aplikasi Jaringan Syaraf Tiruan metode *Learning Vector Quantization* untuk mengklasifikasikan Tenaga Ahli IT. Dengan tujuan untuk mengidentifikasi dan mengenali Tenaga Ahli IT.

Kata Kunci: *Jaringan Syaraf Tiruan, Learning Vector Quantization, Tenaga Ahli IT*

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	iv
DAFTAR TABEL	vi

BAB I PENDAHULUAN

1.1	Latar Belakang	1
1.2	Rumusan Masalah	2
1.3	Batasan Masalah	2
1.4	Tujuan Penelitian.....	3
1.5	Manfaat Penelitian	3
1.6	Metodelogi Peneltian.....	3
1.7	Sistematika Penulisan.....	5

BAB II LANDASAN TEORI

2.1	<i>Learning Vector Quantization</i>	6
2.2	Algoritma Pembelajaran LVQ.....	7
2.3	Kelebihan dan Kekurangan LVQ	9
2.4	Jaringan Syaraf Tiruan.....	9
2.5	Aplikasi Jaringan Syaraf Tiruan	12
2.6	Arsitektur Jaringan Syaraf Tiruan	13
2.7	Algoritma.....	15
2.8	Bahasa Pemrograman	17
2.9	<i>Microsoft Visual Studio</i>	19
2.10	Pengertian UML (Unified Modeling Language).....	20
2.11	<i>Use Case Diagram</i>	21
2.12	<i>Flowchart</i>	23
2.13	Pengertian Informasi.....	27
2.14	Pengertian Visual Studio .NET	28

BAB III METODE PENELITIAN

3.1	Tahapan Penelitian	32
3.2	Analisis Sistem Sedang Berjalan.....	33
3.3	Analisa Sistem Yang Diusulkan	33
3.4	Metode Pengembangan Sistem.....	38
3.5	Perancangan Berorientasi Objek	39
3.5.1	Desain Sistem Secara Detail.....	43
3.5.1.1	<i>Activity Diagram</i>	43
3.5.1.2	<i>Sequence Diagram Login</i>	48
3.5.1.3	<i>Sequence Diagram Tenaga Ahli</i>	49
3.5.1.4	<i>Sequence Diagram Data Penilaian</i>	50
3.5.1.5	<i>Sequence Diagram Hasil Penilaian</i>	50

3.5.1.6	<i>Sequence Diagram Logout</i>	51
3.6	Struktur Program	51
3.7	Perancangan Antarmuka.....	52

BAB IV HASIL DAN PEMBAHASAN

4.1	Kebutuhan sistem	55
4.2	Implementasi Sistem	56

BAB V PENUTUP

5.1	Simpulan	61
5.2	Saran	61

DAFTAR PUSTAKA
BIOGRAFI PENULIS
LAMPIRAN-LAMPIRAN

DAFTAR GAMBAR

	Halaman
Gambar 2.1 <i>Learning Vector Quantization</i>	6
Gambar 2.2 Pola Jaringan Syaraf Tiruan	12
Gambar 2.3 Arsitektur Lapisan Tunggal.....	14
Gambar 2.4 Arsitektur Lapisan <i>Multilayer</i>	14
Gambar 2.5 Arsitektur Lapisan Kompetitif	15
Gambar 2.6 Tampilan <i>Toolbox</i>	30
Gambar 3.1 Tahapan Penelitian	32
Gambar 3.2 <i>Use Case Diagram</i>	40
Gambar 3.3 <i>Activity Diagram</i>	41
Gambar 3.4 <i>Sequence Diagram</i>	41
Gambar 3.5 <i>Class Diagram</i>	42
Gambar 3.6 <i>Activity Diagram Login</i>	43
Gambar 3.7 <i>Activity Diagram Tenaga Ahli</i>	44
Gambar 3.8 <i>Activity Diagram Penilaian</i>	45
Gambar 3.9 <i>Activity Diagram Hasil Klasifikasi</i>	46
Gambar 3.10 <i>Activity Diagram Laporan</i>	47
Gambar 3.11 <i>Activity Diagram Log Out</i>	48
Gambar 3.12 <i>Sequence Diagram Login</i>	48
Gambar 3.13 <i>Sequence Diagram Data Tenaga Ahli</i>	49
Gambar 3.14 <i>Sequence Diagram Data Penilaian</i>	50
Gambar 3.15 <i>Sequence Diagram Hasil Klasifikasi</i>	50
Gambar 3.16 <i>Sequence Diagram Log Out</i>	51
Gambar 3.17 Struktur Navigasi Data Ahli.....	52
Gambar 3.18 Rancangan Halaman Judul.....	52
Gambar 3.19 Rancangan Halaman Menu Utama.....	53
Gambar 3.20 Rancangan Halaman Data Pegawai.....	53
Gambar 3.21 Rancangan Halaman Proses Iterasi	54
Gambar 4.1 Tampilan <i>Form Login</i>	57

Gambar 4.2 Tampilan Menu Utama	57
Gambar 4.3 Tampilan <i>Form Input</i> Penilaian	58
Gambar 4.4 Tampilan <i>Form</i> Hasil Perhitungan LVQ.....	59

DAFTAR TABEL

	Halaman
Tabel 2.1 Simbol-simbol <i>Use Case Diagram</i>	22
Tabel 2.2 Simbol-simbol <i>Flowchart</i>	25
Tabel 2.3 <i>Toolbox Visual Studio</i>	31
Tabel 3.1 Tabel Data <i>Training</i>	34
Tabel 4.1 Tabel Pengujian <i>Black Box</i>	60

BAB I

PENDAHULUAN

1.1 Latar Belakang

Tenaga ahli ialah orang pribadi yang mempunyai keahlian khusus yang dalam memberikan jasa berdasarkan keahliannya tersebut tidak terikat oleh hubungan kerja (melakukan pekerjaan bebas/memberikan professional services), misalnya akuntan, dokter, pengacara, notaris, programmer, konsulen pajak, arsitek, designer dan sebagainya. Tenaga ahli tersebut memberikan jasa yang lazim disebut jasa professional, sedang pekerjaan yang dilakukannya tanpa hubungan kerja lazim disebut juga pekerjaan bebas.

Learning Vector Quantization (LVQ) adalah suatu metode pelatihan untuk melakukan pembelajaran pada lapisan kompetitif yang terawasi (supervised learning) yang arsitektur jaringannya berlapis tunggal (single layer). Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara vektor-vektor input. Jika dua vektor input mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor input tersebut ke dalam kelas yang sama. LVQ merupakan metode klasifikasi pola masing-masing unit keluaran mewakili kategori atau kelas tertentu (beberapa unit keluaran seharusnya digunakan untuk masing-masing kelas). Keunggulan dari metode LVQ adalah kemampuannya untuk memberikan pelatihan terhadap lapisan-lapisan kompetitif sehingga secara otomatis dapat mengklasifikasikan vektor input yang diberikan (Nugroho, 2011).

Pada penelitian yang dilakukan oleh Ying dan Mei (2016), yaitu menentukan klasifikasi kualitas air sungai ke dalam keluaran berupa 3 kelas kualitas

air berdasarkan 4 parameter masukan berupa pH, DO, $\text{NH}_3\text{-N}$ dan COD di Sungai Huaihe, China. Penelitian tersebut menggunakan metode jaringan syaraf tiruan, yaitu algoritma Learning Vector Quantization (LVQ) untuk klasifikasi kualitas air sungai. Penggunaan LVQ terbukti memiliki akurasi yang tinggi dalam melakukan klasifikasi kualitas air sungai dengan tingkat akurasi sebesar 92%. Dari uraian di atas penulis tertarik memilih judul “**Penerapan *Learning Vector Quantization (LVQ)* untuk mengklasifikasikan Tenaga Ahli IT (Studi Kasus: PT. Cita Kreasi Latena)**”.

1.2 Perumusan Masalah

Sesuai dengan latar belakang yang dipaparkan, maka masalah yang akan dibahas adalah sebagai berikut:

- a. Bagaimana menerapkan algoritma *Learning Vector Quantization (LVQ)* untuk mengklasifikasi tenaga ahli ?
- b. Bagaimana user mengidentifikasi dan mengenali tenaga ahli menggunakan algoritma *Learning Vector Quantization (LVQ)*?

1.3 Batasan Masalah

Adapun batasan masalah yang dibatasi dari penulisan skripsi ini adalah sebagai berikut:

- a. Penulis hanya mengenali ciri-ciri tenaga ahli di bidang IT.
- b. Metode pengenalan pola yang digunakan yaitu LVQ (*Learning Vector Quantization*).

1.4 Tujuan Penelitian

Adapun tujuan dari penulisan skripsi ini adalah sebagai berikut:

- a. Mendapatkan informasi mengenai tenaga ahli IT.
- b. Mengenali jenis tenaga ahli IT.

1.5 Manfaat Penelitian

Adapun manfaat yang dapat diambil dari penulisan skripsi ini adalah sebagai berikut:

- a. Mengerti akan konsep Jaringan Syaraf Tiruan dengan metode *Learning Vector Quantization* sebagai salah satu bidang dari ilmu kecerdasan buatan (Artificial Intelligence).
- b. Memberikan penjelasan tentang metode *Learning Vector Quantization* dalam implementasi pada aplikasi dekstop.

1.6 Metodologi Penelitian

Metode yang digunakan pada penulisan skripsi dan perancangan aplikasi menggunakan metode *waterfall* atau air terjun. Pada metode *waterfall* hal yang dilakukan berdasarkan langkah demi langkah, dimana bila langkah pertama belum terselesaikan maka tidak dapat masuk ke langkah kedua dan begitu seterusnya.

Metode *waterfall* memiliki langkah-langkah sebagai berikut:

1. Analisa Kebutuhan, langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau studi literatur. Sistem analis akan menggali

informasi sebanyak-banyaknya dari user sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh user tersebut. Tahapan ini akan menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan sistem. Dokumen ini lah yang akan menjadi acuan sistem analis untuk menerjemahkan ke dalam bahasa pemrogram.

2. Desain Sistem, tahapan dimana dilakukan penuangan pikiran dan perancangan sistem terhadap solusi dari permasalahan yang ada dengan menggunakan perangkat pemodelan sistem seperti diagram alir data (data flow diagram), diagram hubungan entitas (entity relationship diagram) serta struktur dan bahasan data.
3. Penulisan Kode Program, Penulisan kode program atau coding merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan menerjemahkan transaksi yang diminta oleh user. Tahapan ini lah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.
4. Pengujian Program, Tahapan akhir dimana sistem yang baru diuji kemampuan dan keefektifannya sehingga didapatkan kekurangan dan

kelemahan sistem yang kemudian dilakukan pengkajian ulang dan perbaikan terhadap aplikasi menjadi lebih baik dan sempurna.

5. Penerapan Program dan Pemeliharaan Perangkat Lunak, pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

1.7 Sistematika Penulisan

BAB I : PENDAHULUAN

Pada bab ini memaparkan mengenai latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian hingga sistematika penulisan skripsi ini.

BAB II : LANDASAN TEORI

Bab ini mengkaji pustaka baik dari buku-buku ilmiah, maupun sumber-sumber lain yang mendukung penelitian ini.

BAB III : METODE PENELITIAN

Bab ini membahas terkait analisa sistem hingga perancangan sistem ini dibuat.

BAB IV : IMPLEMENTASI PROGRAM

Bab ini berisi implementasi yang diperoleh dari metode *Learning Vector Quantization* dalam mengklasifikasi tenaga ahli IT.

BAB V : KESIMPULAN DAN SARAN

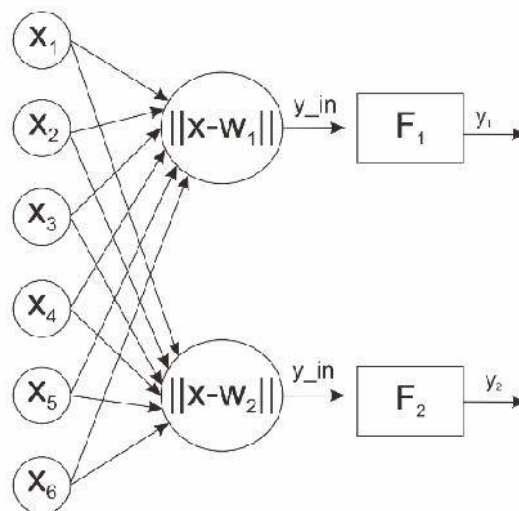
Merupakan bab yang memaparkan kesimpulan beserta saran-saran atas penelitian yang dibuat.

BAB II

LANDASAN TEORI

2.1 Learning Vector Quantization

Menurut Wuryandari dan Afrianto (2012) LVQ merupakan jaringan syaraf dengan tipe arsitektur jaringan lapis-tunggal umpan-maju (*Single Layer Feedforward*) yang terdiri atas unit masukan dan unit keluaran. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vector-vector masukan. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara vektor-vektor masukan. Jika 2 vektor masukan mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor masukan tersebut ke dalam kelas yang sama. Untuk dapat melihat bagaimana Arsitektur jaringan *learning vector quantization* seperti terlihat pada gambar 2.1



Gambar 2.1 *Learning Vector Quantization*
Sumber: Wuryandari dan Afrianto (2012)

Gambar 2.1 memperlihatkan bahwa yang bertindak sebagai dendrit atau data masukan adalah X_1 - X_6 , yang bertindak sebagai sinapsis atau bobot adalah W , sedangkan soma atau badan sel dari jaringan ini adalah perhitungan $\|X-W\|$. Dan yang bertindak sebagai akson atau data keluaran adalah Y .

Learning Vector Quantization (LVQ) adalah metode untuk melakukan pembelajaran pada lapisan kompetitif yang terbimbing. *Learning Vector Quantization (LVQ)* merupakan suatu metode klasifikasi pola masing-masing unit output mewakili kategori atau kelas tertentu. Vektor referensi untuk kelas yang dinyatakan oleh unit tersebut. LVQ mengklarifikasikan vektor input dalam kelas yang sama dengan unit output yang memiliki vektor bobot yang paling dekat dengan vektor input (Widodo, 2015).

2.2 Algoritma Pembelajaran LVQ

Algoritma LVQ bertujuan akhir mencari nilai bobot yang sesuai untuk mengelompokkan vektor-vektor kedalam kelas tujuan yang telah di inialisasi pada saat pembentukan jaringan LVQ. Sedangkan algoritma penguiannya adalah menghitung nilai output (kelas vektor) yang terdekat dengan vektor input, atau dapat disamakan dengan proses pengklasifikasian (pengelompokkan).

Keterangan yang kita gunakan adalah sebagai berikut:

- x : vektor pelatihan (input) ($x_1, \dots, x_i, \dots, x_n$)
- T : kategori yang tepat atau kelas untuk vektor pelatihan
- W_j : bobot vektor untuk unit output ke- j ($w_{1j}, \dots, w_{ij}, \dots, w_{nj}$)

C_j : kategori atau kelas yang ditampilkan oleh unit output ke-j $\|x - w_j\|$ jarak Euclidean antara vektor input dan bobot vektor untuk layer output ke-j

Berikut ini adalah algoritma pembelajaran LVQ:

langkah 0 : inialisasi vektor referensi, inialisasi rating pembelajaran α
(0)

langkah 1 : ketika kondisi berhenti adalah false, lakukan langkah 2 sampai 6

langkah 2 : untuk setiap input pelatihan vektor x lakukan langkah 3 – 4

langkah 3 : temukan j hingga $\|x - w_j\|$ minimum

langkah 4 : perbaharui w_j sebagai berikut : jika $T = C_j$, maka $W_j(\text{baru}) = W_j(\text{lama}) + \alpha[x - w_j(\text{lama})]$; jika $T \neq C_j$, maka $W_j(\text{baru}) = W_j(\text{lama}) - \alpha[x - w_j(\text{lama})]$

langkah 5 : kurang rating pelatihan

langkah 6 : tes kondisi berhenti, yaitu kondisi yang mungkin menetapkan sebuah jumlah tetap dari iterasi atau rating pembelajaran mencapai nilai kecil yang cukup.

Setelah proses pembelajaran *Learning Vector Quantization* (LVQ) maka langkah selanjutnya adalah proses pengambilan hasil *output Learning Vector Quantization* (LVQ). Pengambilan hasil dilakukan hanya pada proses Testing (pengujian). Pada dasarnya, tahapan ini hanya memasukkan input bobot akhir kemudian mencari jarak terdekat dengan perhitungan *Euclidian* (jarak terdekat). (Fausset, (1994) yang dikutip dari Meliawati, dkk (2016).

2.3 Kelebihan dan Kekurangan LVQ

Widodo (2005) *Learning Vektor Quantization (LVQ)* memiliki kelebihan dan kekurangan sebagai berikut:

Kelebihan dari LVQ adalah :

1. Nilai error yang lebih kecil dibandingkan dengan jaringan syaraf tiruan seperti *backpropagation*
2. Dapat meringkas data set yang besar menjadi vektor *codebook* berukuran kecil untuk klasifikasi
3. Dimensi dalam *codebook* tidak dibatasi seperti dalam teknik *nearest neighbour*
4. Model yang dihasilkan dapat diperbaharui secara bertahap

Kekurangan dari LVQ adalah :

1. Dibutuhkan perhitungan jarak untuk seluruh attribute
2. Akurasi model dengan bergantung pada inisialisasi model serta parameter yang digunakan (*learning rate*, literasi dan sebagainya)
3. Akurasi juga dipengaruhi distribusi kelas pada data training
4. Sulit untuk menentukan jumlah *codebook* vektor untuk masalah yang diberikan

2.4 Jaringan Syaraf Tiruan

Menurut Muis (2017:xi) Jaringan syaraf tiruan yang disebut JST merupakan sistem kecerdasan tiruan dengan kemampuan belajar dan menghimpun pengetahuan hasil pembelajaran dalam jaringan selnya (*neuron*) sehingga memungkinkan jaringan secara keseluruhan semakin cerdas merespon

masukan/*input* yang diberikan. Karena kemampuan belajar dan mengakumulasi pengetahuan inilah memungkinkan sistem jaringan syaraf tiruan memiliki kelebihan beradaptasi dengan lingkungan yang memberikan input kepadanya. Dengan keunikan kemampuan demikian, sama halnya cara kerja dan kemampuan olah otak manusia berbeda-beda, JST menjadi bidang penelitian dan pengembangan sangat penting di masa mendatang yang menuntut aspek otomatisasi dan aspek interaktif antara alat dan manusia. Aplikasi JST pada umumnya terkait kuat dengan dunia robotik, yang menuntut unsur kemandirian dan kemampuan beradaptasi dengan lingkungan situasional serta memiliki fleksibilitas tinggi untuk beroperasi di lingkungan keras yang tidak cocok bagi batasan fisik manusia. Contoh aplikasi, misalnya pada robot binatang yang bereaksi terhadap batasan fisik yang mengurungnya, dengan sistem otak JST yang dimiliki, robot JST mampu mencari jalur atau lintasan bebas hambatan untuk melewati tembok dan lainnya yang menghalangi. Contoh lain, misalnya robot peneliti yang beroperasi jauh dalam penjelajahan angkasa luar beradiasi kuat maupun pada kedalaman laut dalam penjelajahan angkasa luar beradiasi kuat maupun pada kedalaman laut dengan tekanan air tidak bersahabat, dan lain-lain.

Jaringan Syaraf Tiruan (JST) atau *artificial neural network (ANN)* adalah sistem komputasi dengan arsitektur dan operasi berdasarkan pengetahuan tentang sel syaraf biologis dalam otak, yang merupakan salah satu representasi buatan dari otak manusia. JST dapat digambarkan sebagai model matematis dan komputasi untuk fungsi aproksimasi *non-linier*, klasifikasi data *cluster*, dan regresi nonparametrik atau sebuah simulasi dari koleksi model syaraf biologi. Selain itu,

JST mampu menangani sistem yang kompleks, rumit, dan tidak linier, serta mampu belajar dengan variabel-variabel keputusan (*decision variables*), seperti halnya pada proses pengeringan. JST sangat baik digunakan untuk pemodelan dinamik karena tidak membutuhkan parameter-parameter model fisik dan mampu belajar dari data eksperimen (Mittal dan Zhang, 2001; Siang, 2005; Hermawan, 2006; Boerietal., 2011).

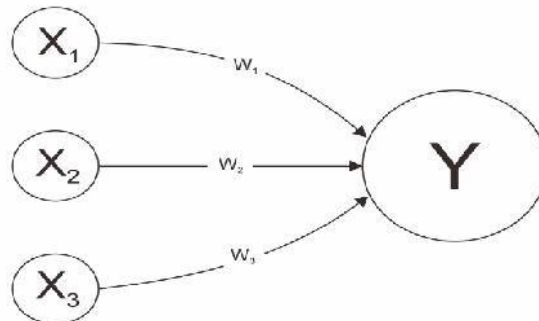
Menurut Siang (2018: 2) Jaringan Syaraf Tiruan (JST) adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi. JST dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan asumsi bahwa :

1. Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron)
2. Sinyal dikirimkan diantara neuron-neuron melalui penghubung-penghubung
3. Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal Untuk menentukan output, setiap neuron menggunakan fungsi aktivasi (biasanya bukan fungsi linier) yang dikenakan pada jumlahan input yang diterima. Besarnya output ini selanjutnya dibandingkan dengan suatu batas ambang.

JST ditentukan oleh 3 hal :

1. Pola hubungan antar *neuron* (disebut arsitektur jaringan)
2. Metode untuk menentukan bobot penghubung (disebut metode *training/learning/algorithm*)
3. Fungsi aktivasi

Sebagai contoh, perhatikan *neuron* Y pada gambar 2.2



Gambar 2.2 Pola Jaringan Syaraf Tiruan
Sumber : Siang (2018)

Y menerima input dari *neuron* X_1 , X_2 dan X_3 dengan bobot hubungan masing-masing adalah w_1 , w_2 dan w_3 . Ketiga impuls neuron yang ada dijumlahkan.

$$\text{net} = X_1W_1 + X_2W_2 + X_3W_3$$

Besarnya impuls yang diterima oleh Y mengikuti fungsi aktivasi $y=f(\text{net})$. Apabila nilai fungsi aktivasi cukup kuat, maka sinyal akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat dipakai sebagai dasar untuk merubah bobot.

2.5 Aplikasi Jaringan Syaraf Tiruan

Menurut Siang (2018: 5) beberapa aplikasi jaringan syaraf tiruan adalah sebagai berikut:

1. Pengenalan Pola (*Pattern Recognition*)

Jaringan syaraf tiruan dapat dipakai untuk mengenali pola (missal huruf, angka, suara atau tanda tangan) yang sudah sedikit berubah. Hal ini mirip dengan otak manusia yang masih mampu mengenali orang yang sudah beberapa waktu tidak dijumpainya (mungkin wajah/ bentuk tubuhnya sudah sedikit berubah)

2. *Signal Processing*

Jaringan syaraf tiruan (model Adaline) dapat dipakai untuk menekan noise dalam saluran telepon

3. Peramalan

Jaringan syaraf tiruan juga dapat dipakai untuk meramalkan apa yang akan terjadi di masa yang akan datang berdasarkan pola kejadian yang ada di masa yang lampau. Ini dapat dilakukan mengingat kemampuan jaringan syaraf tiruan untuk mengingat dan membuat generalisasi dari apa yang sudah ada sebelumnya.

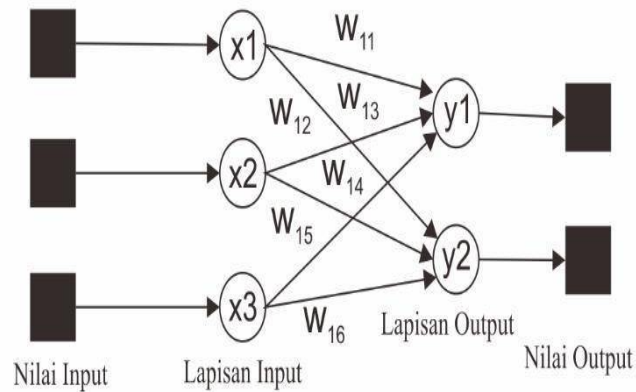
Disamping area-area tersebut, jaringan syaraf tiruan juga dilaporkan dapat menyelesaikan masalah dalam bidang kontrol, kedokteran dan lain-lain. Meskipun banyak aplikasi menjanjikan yang dapat dilakukan oleh jaringan syaraf tiruan, namun jaringan syaraf tiruan juga memiliki beberapa keterbatasan umum. Pertama adalah ketidak akuratan hasil yang diperoleh. Jaringan syaraf tiruan bekerja berdasarkan pola yang terbentuk pada inputnya.

2.6 **Arsitektur Jaringan Syaraf Tiruan**

Lesnussa dkk. (2015) JST memiliki beberapa arsitektur jaringan yang sering digunakan dalam berbagai aplikasi. Arsitektur JST tersebut, antara lain (Hermawan A., 2006):

1. Jaringan Lapisan Tunggal (*Single Layer Network*) Jaringan dengan lapisan tunggal terdiri dari 1 lapisan input dan 1 lapisan *output*. Setiap neuron yang

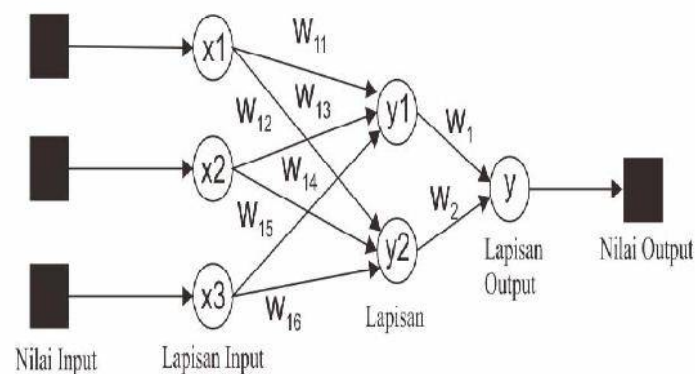
terdapat di dalam lapisan *input* selalu terhubung dengan setiap neuron yang terdapat pada lapisan *output*.



Gambar 2.3 Arsitektur lapisan tunggal

Sumber : Lesnusa dkk. (2015)

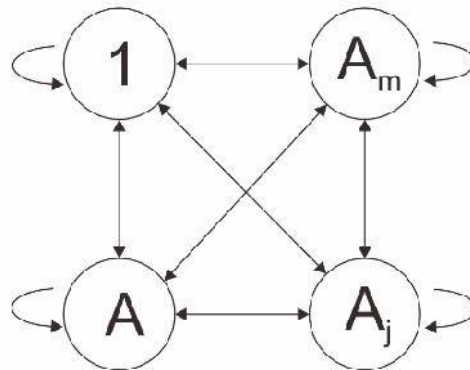
2. Jaringan Banyak Lapisan (*Multilayer Net*) Jaringan dengan lapisan jamak memiliki ciri khas tertentu yaitu memiliki 3 jenis lapisan yakni lapisan *input*, lapisan *output*, dan lapisan tersembunyi. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan jaringan dengan lapisan tunggal. Namun, proses pelatihan sering membutuhkan waktu yang cenderung lama.



Gambar 2.4 Arsitektur Lapisan *Multilayer*

Sumber : Lesnusa dkk. (2015)

3. Jaringan Lapisan Kompetitif (*Competitive Layer*) Pada jaringan ini sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif. Contoh algoritma yang menggunakan jaringan ini adalah LVQ.



Gambar 2.5 Arsitektur Lapisan Kompetitif
Sumber : Lesnussa at al. (2015)

2.7 Algoritma

Penyelesaian permasalahan dengan menggunakan alat bantu sistem computer paling tidak akan melibatkan lima tahapan, yaitu:

1. Analisis masalah
2. Merancang algoritma
3. Membuat program computer
4. Menguji hasil program computer
5. Dokumentasi

(Edhy.2004)

Poin kedua menerangkan bahwa dalam perancangan sebuah sistem computer dibutuhkan adanya perancangan algoritma. Sehingga setelahnya dapat dilanjutkan ke tahap-tahap berikutnya hingga dokumentasi.

Algoritma adalah Sistem kerja komputer memiliki brainware, hardware, dan software. Tanpa salah satu dari ketiga sistem tersebut, komputer tidak akan berguna. Kita akan lebih fokus pada software komputer. Software terbangun atas susunan program (silahkan baca mengenai pengertian program) dan syntax (cara penulisan/pembuatan program). Untuk menyusun program atau syntax, diperlukannya langkah-langkah yang sistematis dan logis untuk dapat menyelesaikan masalah atau tujuan dalam proses pembuatan suatu software. Maka, Algoritma berperan penting dalam penyusunan program atau syntax tersebut.

Pengertian Algoritma adalah susunan yang logis dan sistematis untuk memecahkan suatu masalah atau untuk mencapai tujuan tertentu. Dalam dunia komputer, Algoritma sangat berperan penting dalam pembangunan suatu software. Dalam dunia sehari-hari, mungkin tanpa kita sadari Algoritma telah masuk dalam kehidupan kita.

Pengertian Algoritma adalah susunan yang logis dan sistematis untuk memecahkan suatu masalah atau untuk mencapai tujuan tertentu.

Algoritma adalah kunci dari bidang ilmu komputer, dan pada dasarnya setiap hari kita melakukan aktivitas algoritma. Kata algoritma berasal dari sebutan Algorizm (Abu Abdullah Muhammad Ibn Musa Al Khwarizmi, ahli matematika Uzbeki)

1. Algoritma adalah urutan langkah-langkah berhingga untuk memecahkan masalah logika atau matematika
2. Algoritma adalah logika, metode dan tahapan (urutan) sistematis yang digunakan untuk memecahkan suatu permasalahan.

3. Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis.
4. Algoritma adalah urutan logis pengambilan keputusan untuk pemecahan masalah.

Pembuatan algoritma harus selalu dikaitkan dengan:

- a. Kebenaran algoritma
- b. Kompleksitas (lama dan jumlah waktu proses dan penggunaan memori)

Kriteria Algoritma yang baik:

1. Tepat, benar, sederhana, standar dan efektif
2. Logis, terstruktur dan sistematis
3. Semua operasi terdefinisi
4. Semua proses harus berakhir setelah sejumlah langkah dilakukan
5. Ditulis dengan bahasa yang standar dengan format pemrograman

agar mudah untuk diimplementasikan dan tidak menimbulkan arti ganda.

2.8 Bahasa Pemrograman

Menurut Jusuf Wahyudi (2013) Bahasa pemrograman adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas tertentu. Bahasa pemrograman merupakan sebuah instruksi untuk memerintah komputer agar bisa menjalankan fungsi tertentu, namun hanya instruksi standar saja. Bahasa pemrograman juga memiliki perhimpunan dari aturan sintaks dan semantik yang tugasnya untuk mendefinisikan program komputer. Bahasa pemrograman komputer yang kita kenal antara lain adalah *Java*, *Visual Basic*, *C++*,

C, PHP, dan bahasa pemrograman lainnya. Namun tentu saja kebutuhan bahasa ini harus disesuaikan dengan fungsi dan perangkat yang menggunakannya.

Menurut generasi bahasa pemrograman digolongkan menjadi 4 generasi, yaitu:

1. Generasi ke-1: *machine language*
2. Generasi ke-2: *assembly language: Assembler*
3. Generasi ke-3: *high level programming language*, contoh: C dan Pascal
4. Generasi ke-4: *4 GL (fourth-generation language)*, contoh: SQL
5. Generasi ke-5: *Programming Language Based Object Oriented & Web Development*

Secara umum bahasa pemrograman dibagi menjadi 4 kelompok, yaitu :

1. *Object Oriented Language* : Seperti bahasa *Visual C, Delphi, Visual dBase, Visual FoxPro*.
2. *Low Level Language* : Bahasa *Assembly*.
3. *Middle Level Language* : Bahasa *C*.
4. *High Level Language* : Bahasa *Basic* dan *Pascal*.

Menurut tingkat kedekatannya dengan mesin komputer, bahasa pemrograman terdiri dari:

- a. Bahasa Mesin, yaitu memberikan perintah kepada komputer dengan memakai kode bahasa biner, contohnya 01100101100110.
- b. Bahasa Tingkat Rendah, atau dikenal dengan istilah bahasa rakitan (bah.Inggris *Assembly*), yaitu memberikan perintah kepada komputer

dengan memakai kode-kode singkat (kode *mnemonic*), contohnya *MOV, SUB, CMP, JMP, JGE, JL, LOOP*, dsb.

- c. Bahasa Tingkat Menengah, yaitu bahasa komputer yang memakai campuran instruksi dalam kata-kata bahasa manusia (lihat contoh Bahasa Tingkat Tinggi di bawah) dan instruksi yang bersifat simbolik, contohnya {, }, ?, <<, >>, &&.
- d. Bahasa Tingkat Tinggi, yaitu bahasa komputer yang memakai instruksi berasal dari unsur kata-kata bahasa manusia, contohnya *begin, end, if, for, while, and, or, dsb*. Komputer dapat mengerti bahasa manusia itu diperlukan program *compiler* atau *interpreter*.

Fungsi dari bahasa pemrograman adalah untuk memerintahkan sebuah komputer agar dapat mengolah data yang sesuai dengan di inginkan. *Output* dari bahasa pemrograman ini dapat berupa aplikasi ataupun program khusus.

2.9 Microsoft Visual Studio

Menurut Ninuk Wiliani (2017) Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi. Baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya dalam bentuk aplikasi *console*, aplikasi *Windows*, ataupun aplikasi *Web*. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic.NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di *Windows*) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas *.NET Framework*). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi *Silverlight*, aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*).

Visual studio ini dapat digunakan untuk membuat aplikasi yang berbasis desktop yang merupakan platform windows, namun juga dapat dijalankan dalam bentuk *Microsoft Intermediate Language* diatas *.Net Framework*. Selain itu Visual Studio juga dapat digunakan untuk membuat aplikasi yang dapat dijalankan diatas windows mobile yang berjalan diatas *.Net Compact Framework*.

2.10 Pengertian UML (Unified Modeling Language)

Menurut Winda Aprianti (2016) *Unified Modeling Language (UML)* adalah standarisasi bahasa pemodelan untuk membangun perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek. Diagram diagram yang digunakan pada *UML* antara lain adalah *class diagram*, *object diagram*, *use case diagram*, *activity diagram*, dan *sequence diagram*.

UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga Merupakan alat untuk mendukung pengembangan sistem. *UML* saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

Berikut beberapa tujuan atau fungsi dari penggunaan *UML*, yang diantaranya:

1. Dapat memberikan bahasa permodelan *visual* kepada pengguna dari berbagai macam pemrograman maupun proses rekayasa.
2. Dapat menyatukan praktek-praktek terbaik yang ada dalam permodelan.
3. Dapat memberikan model yang siap untuk digunakan, merupakan bahasa permodelan *visual* yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.
4. Dapat berguna sebagai *blue print*, sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program.
5. Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak (*software*) saja.
6. Dapat menciptakan suatu bahasa permodelan yang nantinya dapat dipergunakan oleh manusia maupun oleh mesin.

2.11 Use Case Diagram

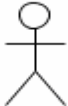
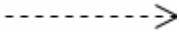


Menurut Yunahar Heriyanto (2018) *Use Case Diagram* adalah sesuatu atau proses merepresentasikan hal-hal yang dapat dilakukan oleh aktor dalam menyelesaikan sebuah pekerjaan.






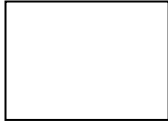
Diagram use case merupakan permodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu

atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Berikut adalah simbol-simbol yang ada pada *diagram use case* :

Tabel 2.1 Simbol-Simbol Use Case Diagram

NO	Simbol	Deskripsi
1		Orang/Actor, proses himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		Dependency, hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		Generalization, hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		Include, menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .

NO	Simbol	Deskripsi
5		Extend, menspesifikasikan bahwa <i>use case</i> target, memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		Association, merupakan Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		Collaboration, interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
8		<i>Use Case</i> , deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Note</i> , Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi
10		System, Menspesifikasikan paket yang menampilkan sistem secara terbatas.

Sumber: Yunahar Heriyanto (2018)

2.12 Flowchart

Menurut Mohamad Natsir (2017) *Flow chart* atau *diagram* alir merupakan sebuah diagram dengan symbol-simbol grafis yang menyatakan aliran algoritma



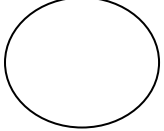


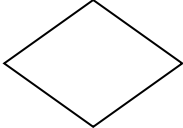


atau proses yang menampilkan langkah-langkah yang disimbolkan dalam bentuk kotak, beserta urutannya dengan menghubungkan masing-masing langkah tersebut menggunakan tanda panah. Diagram ini bisa memberi solusi selangkah demi selangkah untuk penyelesaian masalah yang ada di dalam proses atau algoritma tersebut.

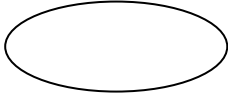


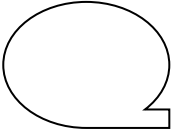
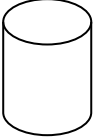
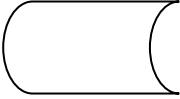



Flowchart digunakan dalam merancang dan mendokumentasikan proses yang kompleks atau program. Seperti jenis lain *diagram*, mereka membantu memvisualisasikan apa yang terjadi dan dengan demikian membantu pengunjung untuk memahami proses, dan mungkin juga menemukan kelemahan, kemacetan, dan ketidakjelasan lain di dalamnya. Ada berbagai jenis diagram alur yang masing-masing memiliki repertoire kotak sendiri dan ketentuan notasinya. Dua jenis yang paling umum dari kotak di *flowchart* adalah:

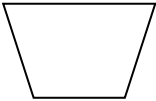
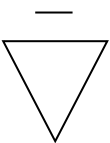

1. Langkah pengolahan, biasanya disebut aktivitas, dan dilambangkan sebagai persegi panjang.
2. Keputusan, biasanya dilambangkan sebagai belah ketupat.

Flowchart memberikan solusi langkah demi langkah dari sebuah masalah yang ingin dipecahkan. Kegunaan *flowchart* adalah untuk menganalisis, merancang, mendokumentasikan serta mengelola suatu proses atau program di berbagai bidang. Sebuah proses atau *action* direpresentasikan dalam sebuah kotak, dan tanda panah yang menghubungkan kotak-kotak ini mewakili aliran atau arah aliran data. Berikut ini adalah simbol-simbol *flowchart* yang sering digunakan beserta deskripsinya:

Tabel 2.2. Simbol-simbol *flowchart*.

SIMBOL	NAMA SIMBOL / ARTI
	INPUT / OUTPUT Mempresentasikan input data atau output data yang diproses atau informasi
	PROSES Mempresentasikan operasi
	PENGHUBUNG Keluar atau masuk dari bagian lain flowchart khususnya halaman yang sama
	ANAK PANAH Mempresentasikan alur kerja
	PENJELASAN Digunakan untuk komentar tambahan
	KEPUTUSAN Menyatakan keputusan dalam program
	PREDEFINED PROCESS Rincian operasi berada di tempat lain.
	PREPARATION Pemberian harga awal

	<p>TERMINAL POINTS Awal / akhir flowchart</p>
	<p>PUNCHED CARD Input / output yang menggunakan kartu berulang</p>
	<p>DOKUMEN Input / output dalam format yang dicetak</p>
	<p>MAGNETIC TAPE Input / output yang menggunakan pita magnetic</p>
	<p>MAGNETIC DISK Input / Output yang menggunakan disk magnetic</p>
	<p>ON-LINE STORAGE Input / output yang menggunakan penyimpanan akses langsung</p>
	<p>PUNCHED TAPE Input / output yang menggunakan pita kertas berlubang</p>
	<p>MANUAL INPUT Input yang dimasukkan secara manual dari keyboard</p>
	<p>DISPLAY Output yang ditampilkan pada terminal</p>

	MANUAL OPERATION Operasi manual
	OFF – LINE STORAGE Penyimpanan yang tidak dapat diakses oleh komputer secara langsung
	COMMUNICATION LINK Transmisi data melalui channel komunikasi, Seperti telepon

Sumber : Mohamad Natsir (2017)

2.13 Pengertian Informasi

Secara Etimologi, kata informasi ini berasal dari kata bahasa Perancis kuno *informacion* (tahun 1387) mengambil istilah dari bahasa Latin yaitu *informationem* yang berarti “konsep, ide atau garis besar”. Informasi ini merupakan kata benda dari *informare* yang berarti aktivitas dalam “pengetahuan yang dikomunikasikan”.

Informasi bisa menjadi fungsi penting dalam membantu mengurangi rasa cemas pada seseorang. Menurut pendapat Notoatmodjo (2018) bahwa semakin banyak memiliki informasi dapat memengaruhi atau menambah pengetahuan terhadap seseorang dan dengan pengetahuan tersebut bisa menimbulkan kesadaran yang akhirnya seseorang itu akan berperilaku sesuai dengan pengetahuan yang dimilikinya.

Informasi adalah data yang telah diolah melalui proses tertentu menjadi sesuatu yang menambah pengetahuan atau temuan yang mempunyai arti baru bagi pemakainya (Melina, 38 : 2014).

Adapun fungsi-fungsi informasi adalah sebagai berikut:

1. Untuk meningkatkan pengetahuan bagi si pemakai.
2. Untuk mengurangi ketidakpastian dalam proses pengambilan keputusan pemakai.
3. Menggambarkan keadaan yang sebenarnya dari sesuatu hal. Informasi yang berkualitas harus akurat, tepat dan relevan.

Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu metode untuk menghasilkan informasi. Data dapat berbentuk simbol-simbol semacam huruf, angka, bentuk suara, sinyal, gambar, dan sebagainya.

2.14 Pengertian Visual Studio .NET

Visual Studio .Net merupakan salah satu *tool development Microsoft* yang dapat digunakan untuk membuat aplikasi di lingkungan kerja berbasis sistem operasi *Windows*. *Visual Studio .NET* menyediakan tools bagi para *developer* untuk membangun aplikasi yang berjalan di *.Net Framework* (Safik : 2015 : 2).

Visual Studio (Beginners All-Purpose Symbolic Instruction Code) merupakan Bahasa pemrograman *Integrated Development Environment (IDE)*, yaitu bahasa pemrograman *visual* yang digunakan untuk membuat program aplikasi atau *software* berbasis sistem operasi *Microsoft Windows*, dengan menggunakan model pemrograman "*Common Object Model (COM)*".

Visual Studio merupakan turunan bahasa pemrograman *STUDIO* yang menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat. Dengan menggunakan bahasa pemrograman VB, para programmer dapat membangun aplikasi dengan menggunakan komponen-komponen yang di sediakan VB.

Microsoft Visual Studio (sering disingkat sebagai VB saja) merupakan sebuah bahasa pemrograman yang menawarkan *Integrated Development Environment* (IDE) visual untuk membuat program perangkat lunak berbasis sistem operasi Microsoft Windows dengan menggunakan model pemrograman (*COM*), *Visual Studio* merupakan turunan bahasa pemrograman *STUDIO* dan menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat, Beberapa bahasa skrip seperti *Visual Studio for Applications* (VBA) dan *Visual Studio Scripting Edition* (VBScript), mirip seperti halnya *Visual Studio*, tetapi cara kerjanya yang berbeda.

Para *programmer* dapat membangun aplikasi dengan menggunakan komponen-komponen yang disediakan oleh *Microsoft Visual Studio* Program-program yang ditulis dengan *Visual Studio* juga dapat menggunakan *Windows API*, tapi membutuhkan deklarasi fungsi luar tambahan.

Dalam pemrograman untuk bisnis, *Visual Studio* memiliki pangsa pasar yang sangat luas. Dalam sebuah survey yang dilakukan pada tahun 2005, 62% pengembang perangkat lunak dilaporkan menggunakan berbagai bentuk *Visual Studio*, yang diikuti oleh *C++*, *JavaScript*, *C#*, dan *Java*.

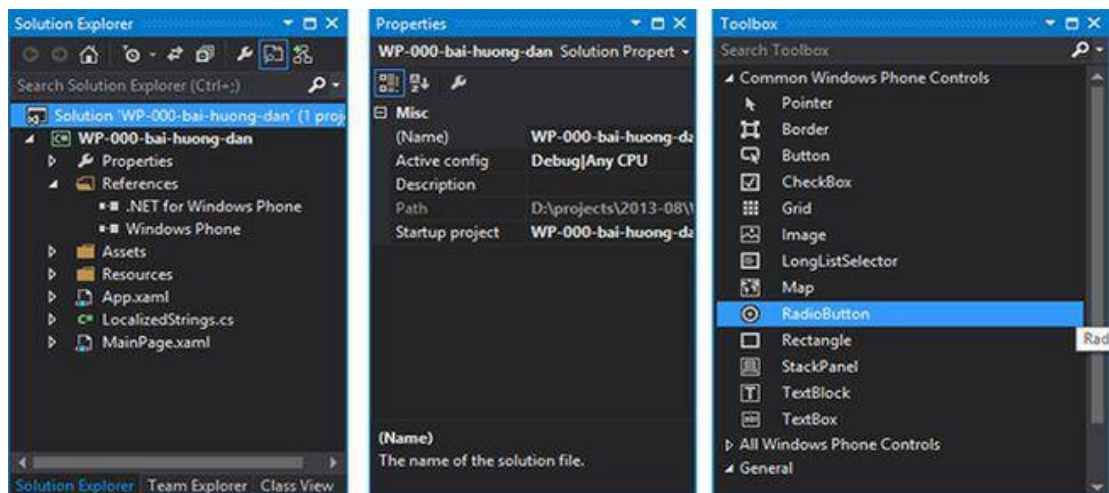
1. Komponen kerja

Beberapa komponen kerja program *visual Studio 2015* telah ditampilkan sebagai tampilan standard. Masih banyak lagi komponen yang masih tersembunyi sehingga memerlukan perintah tertentu untuk menampilkannya. Kita dapat mengatur komponen di dalam program visual Studio 2015 sesuai dengan yang kita butuhkan. Berikut ini adalah beberapa komponen kerja dari visual Studio 2015 adalah :

a. *Toolbox*

Toolbox adalah sebuah panel yang menampung tombol-tombol yang berguna untuk membuat suatu desain mulai dari tombol *label*, *pointer*, *button*, dan lain-lain. Berikut ini adalah gambaran *toolbox* pada *visual Studio 2015* :

Berikut ini adalah *table* yang berisi nama tombol yang terdapat didalam *toolbox* beserta fungsinya.



Gambar 2.6. Tampilan *Toolbox*

Sumber : (Safik : 2015 : 2)

Table 2.3 Toolbox Visual Studio

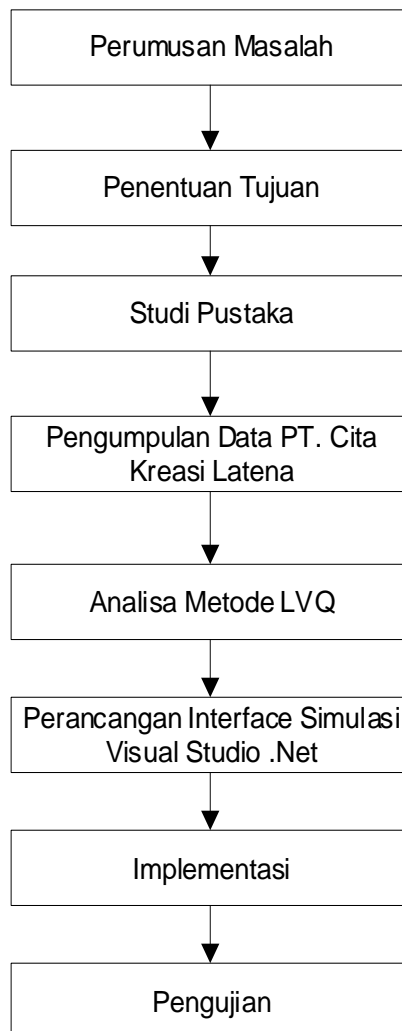
Nama tombol	fungsi
<i>Pointer</i>	Memilih, mengatur ukuran dan memindahkan posisi yang terpasang di bagian form.
<i>Bindingsources</i>	Untuk mengkoneksikan program ke database
<i>Label</i>	Menampilkan teks, dimana pengguna program tidak bisa mengubah teks tersebut
<i>GroupBox</i>	Untuk mengelompokkan item yang ada di form
<i>Checkbox</i>	Membuat kotak periksa, dimana pengguna program dapat memilih sekaligus
<i>Listbox</i>	Membuat daftar pilihan
<i>Timer</i>	Membuat control waktu dan interval yang diperlukan
<i>Image</i>	Menampilkan gambar pada form dalam format <i>bitmap</i> , <i>icone</i> , atau <i>metafile</i>
<i>PictureBox</i>	Menampilkan gambar dari sebuah file
<i>Textbox</i>	Membuat teks, dimana teks tersebut dapat diubah oleh pembuat program
<i>Button</i>	Membuat tombol perintah
<i>Combobox</i>	Menambahkan control kotak combo yang merupakan control gabungan antara <i>textbox</i> dan <i>listbox</i>

Sumber: (Safik : 2015 : 2).

BAB III METODE PENELITIAN

3.1 Tahapan Penelitian

Adapun tahapan penelitian yang dilakukan oleh penulis dengan judul Penerapan *Learning Vector Quantization* (LVQ) Untuk Mengklasifikasikan Tenaga Ahli IT adalah sebagai berikut:



Gambar 3.1 Tahapan Penelitian

3.2 Analisa Sistem Yang Berjalan

Analisa sistem dilakukan dengan membahas tentang proses yang berkaitan dengan sistem yang akan digunakan dalam penelitian. Dalam hal ini, penulis akan merumuskan pengembangan informasi yang didapat dari wawancara, observasi dan studi pustaka. Analisa sistem dibagi menjadi dua bagian, yaitu analisa sistem lama dan analisa sistem baru. Analisa berperan penting dalam memahami permasalahan yang terjadi dalam penggunaan sistem lama sehingga memudahkan dalam perancangan sistem baru.

3.3 Analisa Sistem Yang Diusulkan

Dalam tahap ini, dilakukan analisa metode LVQ (Learning Vector Quantization) yang akan diterapkan dalam kasus penentuan klasifikasi tenaga ahli IT. Tahap awal yang dilakukan adalah :

1. X adalah vektor-vektor pelatihan untuk data latih ($X_1, \dots, X_i, \dots, X_n$).
2. T adalah kategori atau kelas yg benar untuk vektor-vektor pelatihan.
3. W_j adalah vektor bobot pada unit keluaran ke- j ($W_{1j}, \dots, W_{ij}, \dots, W_{nj}$).
4. C_j adalah kategori atau kelas yang merepresentasikan oleh unit keluaran ke- j
5. learning rate (α), α didefinisikan sebagai tingkat pembelajaran. Jika α terlalu besar, maka algoritma akan menjadi tidak stabil sebaliknya jika α terlalu kecil, maka prosesnya akan terlalu lama. Nilai α adalah $0 < \alpha < 1$.
6. Nilai pengurangan learning rate, adalah penurunan tingkat pembelajaran.

7. Nilai minimal learning rate (α), adalah minimal nilai tingkat pembelajaran yang diperbolehkan. Pengurangan nilai α yang digunakan pada penelitian klasifikasi Tenaga Ahli IT ini adalah sebesar $0.1 * \alpha$.
8. Nilai window (ϵ), yaitu nilai yang digunakan sebagai daerah yang harus dipenuhi untuk memperbaharui vektor referensi pemenang dan runner-up jika berada di kelas yang berbeda.
9. Jarak vektor referensi terdekat pertama dengan vektor x adalah d_1 dan Jarak vektor referensi terdekat kedua dengan vektor x adalah d_2 .
10. Vektor referensi dapat diperbaharui jika masuk ke dalam daerah yang disebut window (ϵ).

Tabel 3.1 Tabel Data Training

No	X1	X2	X3	X4	target
1	0	1	1	0	0
2	0	0	1	1	1
3	1	1	1	1	0
4	1	0	0	1	1

pada contoh di atas, penulis menggunakan 4 data sebagai data training beserta target yang bertujuan untuk mendapatkan bobot yang akan digunakan pada proses klasifikasi. Bobot awal adalah $\{ 1, 1, 1, 0 \}$ dan $\{ 1, 0, 1, 1 \}$ dengan learning rate 0,05 dengan fungsi pembelajaran = 0,1.

A. Iterasi ke 1

1. Data ke 1 $\{ 0, 1, 1, 0 \}$ dengan target 0, bobot = $\{ \{ 1, 1, 1, 0 \}, \{ 1, 0, 1, 1 \} \}$
 - a. menghitung bobot untuk masing masing output :

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

$$\begin{aligned} \text{kelas 1} &= \sqrt{((0-1)^2)+((1-0)^2)+((1-1)^2)+((0-1)^2)} \\ &= 1.7320508075689 \end{aligned}$$

- b. menentukan kelas pemenang :

$$\text{output} = \min(\text{kelas 0}, \text{kelas 1}) = \text{kelas 0}$$

- c. update bobot :

karena target 0 sama dengan output 0, maka update bobot :

$$W_{11} = 1 + (0.05*(0 - 1)) = 0.95$$

$$W_{12} = 1 + (0.05*(1 - 1)) = 1$$

$$W_{13} = 1 + (0.05*(1 - 1)) = 1$$

$$W_{14} = 0 + (0.05*(0 - 0)) = 0$$

maka diperoleh bobot baru = $\{\{ 0.95, 1, 1, 0\}, \{ 1, 0, 1, 1\}\}$ untuk digunakan pada perhitungan data selanjutnya.

2. Data ke 2 $\{ 0, 0, 1, 1\}$ dengan target 1, bobot = $\{\{ 0.95, 1, 1, 0\}, \{ 1, 0, 1, 1\}\}$

- a. menghitung bobot untuk masing masing output :

$$\begin{aligned} \text{kelas 0} &= \sqrt{((0-0.95)^2)+((0-1)^2)+((1-1)^2)+((1-0)^2)} \\ &= 1.7036725037401 \end{aligned}$$

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

- b. menentukan kelas pemenang :

$$\text{output} = \min(\text{kelas 0}, \text{kelas 1}) = \text{kelas 1}$$

- c. update bobot :

karena target 1 sama dengan output 1, maka update bobot :

$$W_{21} = 1 + (0.05*(0 - 1)) = 0.95$$

$$W_{22} = 0 + (0.05*(0 - 0)) = 0$$

$$W_{23} = 1 + (0.05 * (1 - 1)) = 1$$

$$W_{24} = 1 + (0.05 * (1 - 1)) = 1$$

maka diperoleh bobot baru = $\{\{ 0.95, 1, 1, 0\}, \{ 0.95, 0, 1, 1\}\}$ untuk digunakan pada perhitungan data selanjutnya

3. Data ke 3 $\{ 1, 1, 1, 1\}$ dengan target 0, bobot = $\{\{ 0.95, 1, 1, 0\}, \{ 0.95, 0, 1, 1\}\}$

- a. menghitung bobot untuk masing masing output :

$$\text{kelas 0} = \sqrt{((1-0.95)^2) + ((1-1)^2) + ((1-1)^2) + ((1-0)^2)}$$

$$= 1.001249219725$$

$$\text{kelas 1} = \sqrt{((1-0.95)^2) + ((1-0)^2) + ((1-1)^2) + ((1-1)^2)}$$

$$= 1.001249219725$$

- b. menentukan kelas pemenang :

output = min(kelas 0, kelas 1) = kelas 0, karena output antara kedua kelas sama besar, bisa dipilih salah satu sebagai output

- c. update bobot :

karena target 0 sama dengan output 0, maka update bobot :

$$W_{11} = 0.95 + (0.05 * (1 - 0.95)) = 0.9525$$

$$W_{12} = 1 + (0.05 * (1 - 1)) = 1$$

$$W_{13} = 1 + (0.05 * (1 - 1)) = 1$$

$$W_{14} = 0 + (0.05 * (1 - 0)) = 0.05$$

maka diperoleh bobot baru = $\{\{ 0.9525, 1, 1, 0.05\}, \{ 0.95, 0, 1, 1\}\}$ untuk digunakan pada perhitungan data selanjutnya

4. Data ke 4 { 1, 0, 0, 1 } dengan target 1, bobot = { { 0.9525, 1, 1, 0.05 }, { 0.95, 0, 1, 1 } }

a. menghitung bobot untuk masing masing keluaran :

$$\begin{aligned} \text{kelas 0} &= \sqrt{((1-0.9525)^2)+((0-1)^2)+((0-1)^2)+((1-0.05)^2)} \\ &= 1.704334547 \end{aligned}$$

$$\begin{aligned} \text{kelas 1} &= \sqrt{((1-0.95)^2)+((0-0)^2)+((0-1)^2)+((1-1)^2)} \\ &= 1.001249219725 \end{aligned}$$

b. menentukan kelas pemenang :

$$\text{output} = \min(\text{kelas 0, kelas 1}) = \text{kelas 1}$$

c. update bobot :

karena target 1 sama dengan output 1, maka update bobot :

$$W_{21} = 0.95 + (0.05*(1 - 0.95)) = 0.9525$$

$$W_{22} = 0 + (0.05*(0 - 0)) = 0$$

$$W_{23} = 1 + (0.05*(0 - 1)) = 0.95$$

$$W_{24} = 1 + (0.05*(1 - 1)) = 1$$

maka diperoleh bobot baru = { { 0.9525, 1, 1, 0.05 }, { 0.9525, 0, 0.95, 1 } } untuk

digunakan pada perhitungan data selanjutnya setelah semua data selesai dihitung,

update learning rate :

learning rate baru = fungsi pembelajaran * learning rate;

$$= 0.1 * 0.05 = 0.005$$

karena dalam satu iterasi semua data sudah diklasifikasikan dengan benar, maka

iterasi pelatihan bisa dihentikan dan memakai bobot terakhir untuk digunakan

dalam proses klasifikasi / prediksi. Jika dalam satu iterasi ada yang outputnya belum

tepat, maka dilanjutkan ke iterasi berikutnya sampai dengan data diklasifikasikan dengan benar atau telah mencapai maksimal iterasi yang telah ditentukan.

B. Klasifikasi / Prediksi

Setelah didapat bobot hasil pelatihan yaitu $\{ \{ 0.9525, 1, 1, 0.05 \}, \{ 0.9525, 0, 0.95, 1 \} \}$, maka kita coba melakukan uji coba klasifikasi terhadap dua data yaitu $\{ 1, 1, 1, 0 \}$ dan $\{ 1, 0, 1, 1 \}$.

1. Data ke 1 { 1, 1, 1, 0}

$$\begin{aligned} \text{kelas 0} &= \sqrt{((1-0.9525)^2)+((1-1)^2)+((1-1)^2)+((0-0.05)^2)} \\ &= 0.06896557 \end{aligned}$$

$$\begin{aligned} \text{kelas 1} &= \sqrt{((1-0.9525)^2)+((1-0)^2)+((1-0.95)^2)+((0-1)^2)} \\ &= 1.41589415 \end{aligned}$$

output = kelas 0

2. Data ke 2 { 1, 0, 1, 1}

$$\begin{aligned} \text{kelas 0} &= \sqrt{((1-0.9525)^2)+((0-1)^2)+((1-1)^2)+((1-0.05)^2)} = \\ &1.380129070051 \end{aligned}$$

$$\begin{aligned} \text{kelas 1} &= \sqrt{((1-0.9525)^2)+((0-0)^2)+((1-0.95)^2)+((1-1)^2)} = \\ &0.068965571120 \end{aligned}$$

output = kelas 1

Keterangan:

Kelas 0 = Tenaga Ahli IT Programmer, Kelas 1= Asisten Tenaga Ahli

3.4 Metode Pengembangan Sistem

Adapun metode pengembangan yang peneliti rancang sebagai berikut:

1. Analisis Kebutuhan

Tahap ini bertujuan untuk menganalisa terhadap kebutuhan sistem, pengumpulan data dalam tahap ini bias dilakukan malalui sebuah penelitian, wawancara atau pencarian referensi terkait

2. Desain Sistem

Tahap ini merupaka proses perancangan dimana menerjemahkan kebutuhan sistem sebelum dimasukkan kedalam *coding*/Bahasa pemograman. Tujuan dari desain sistem ini adalah untuk melihat gambaran awal sistem ini akan dibuat.

3. *Coding* dan Testing

Pada tahap ini akan dilakukan penerjemahan dari desain sistem dalam bentuk Bahasa pemograman. Pada proses ini lebih diketahui mengenai metode apa saja yang digunakan. Testing merupakan tahapan dimana apabila terjadi kesalahan maka dapat diperbaiki. Pada sistem ini nantinya akan menggunakan metode *blackbox* untuk menguji fitur-fitur yang ada sehingga diketagui aakahan fitur tersebut sesuai dengan *output* yang diinginkan.

4. Pemeliharaan dan perawatan

Tahap dimana dimulai melakukan pengoperasian sistem, jika dipelukan melakukan perbaikan kecil atas kesalahan yang ada.

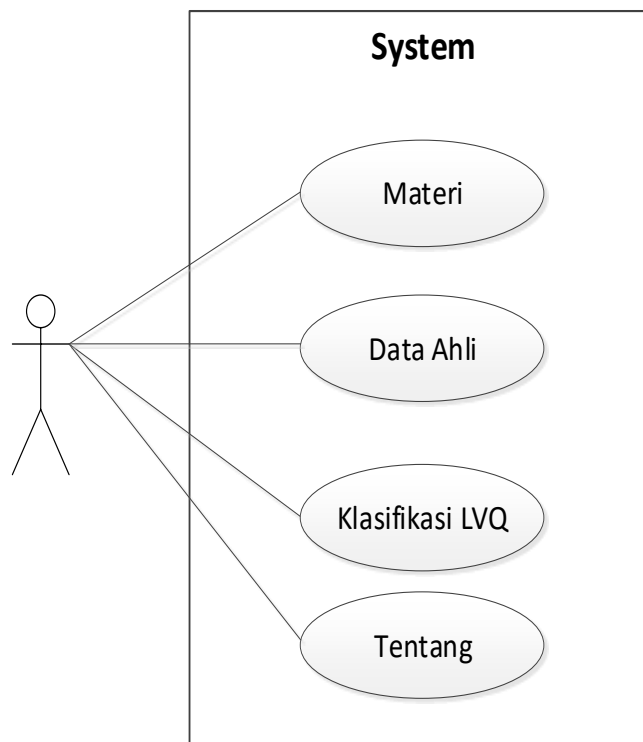
3.5 Perancangan Berorientasi Objek

Perancangan atau Pemodelan Berorientasi Ojek merupakan proses mendapatkan informasi dari model dan menampilkannya secara grafik dengan menggunakan sebuah standar elemen grafik. Tujuan dari perancangan berorientasi

objek ini memungkinkan adanya komunikasi yang lebih berkualitas antara pengguna, pengembang penganalisis, tetster, manajer dan siapapun yang terlibat dalam proyek pengembangan sistem informasi.

1. Use case Diagram

Berikut adalah *use case diagram* yang menggambarkan kegiatan.



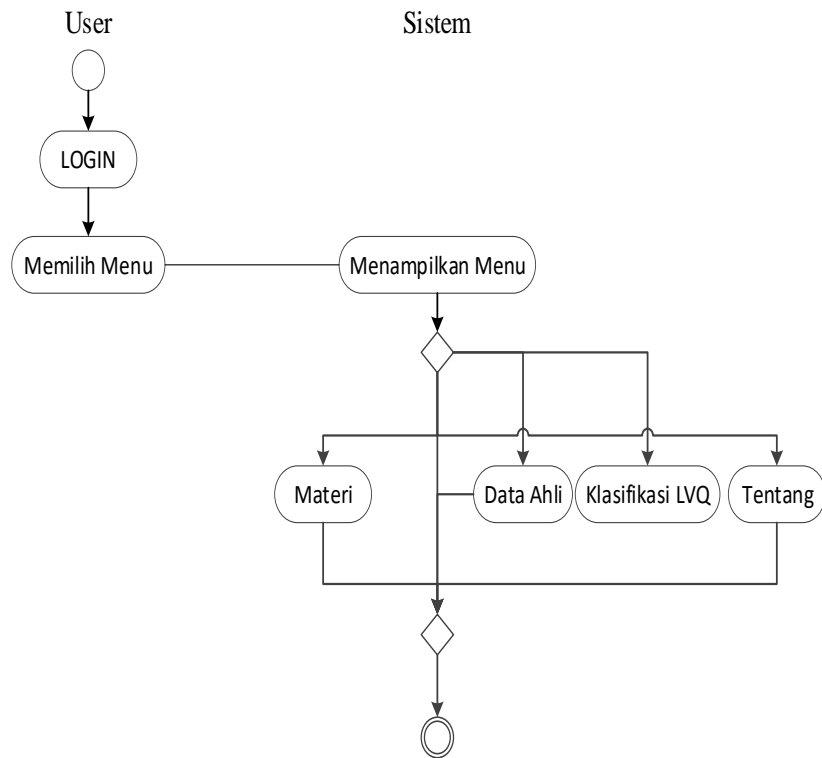
Gambar 3.2 *Use Case Diagram*

Keterangan:

Dalam *use case diagram* di atas, *user/pengguna* sebagai *actor* yang mempunyai *use case* Materi, Data Ahli, Klasifikasi dan Tentang.

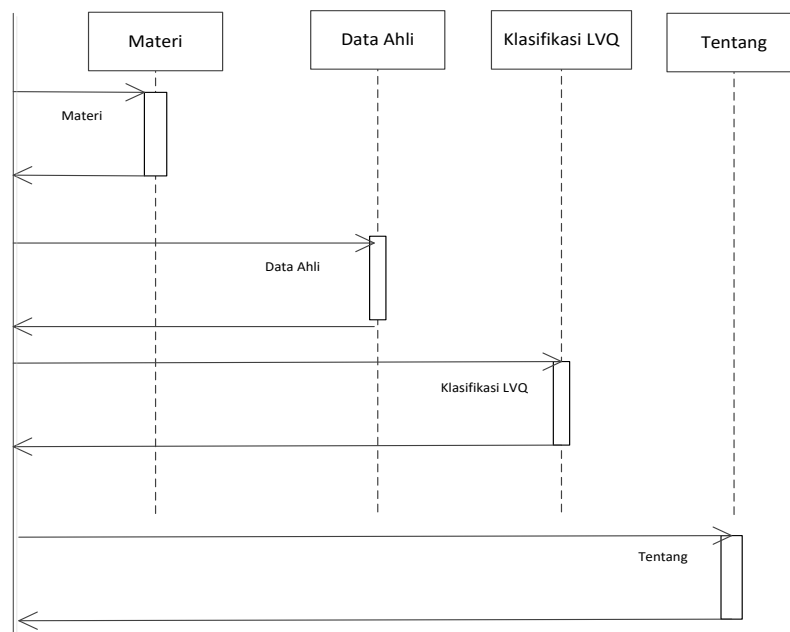
2. Pembuatan Activity Diagram

Activity diagram menggambarkan aktifitas-aktifitas yang terjadi dalam aplikasi dari aktivitas dimulai sampai aktivitas berhenti.



Gambar 3.3 Activity Diagram

3. Sequence Diagram



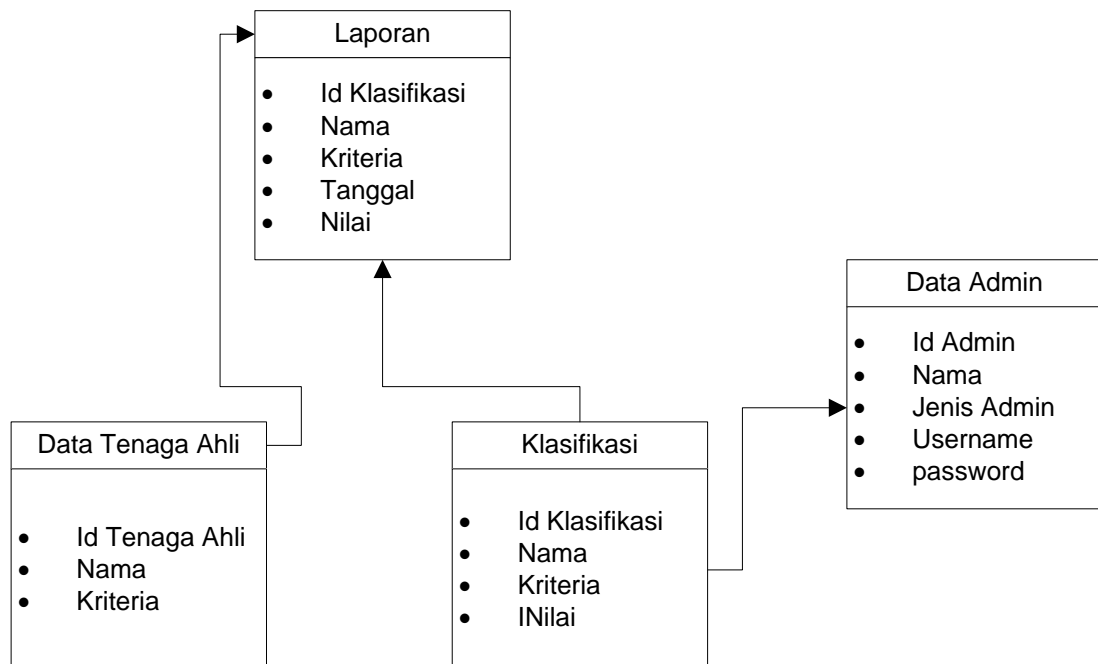
Gambar 3.4 Sequence Diagram

Keterangan Gambar :

1. Dalam diagram di atas menjelaskan bahwa user memilih materi kemudian Sistem menampilkan materi.
2. User merequest Data Ahli kemudian Sistem menampilkan menu Data Ahli
3. User merequest Klasifikasi LVQ kemudian Sistem menampilkan menu Klasifikasi LVQ
4. User merequest Menu Tentang kemudian Sistem menampilkan Form Tentang.

4. Class Diagram

Adapun class diagram sistem usulan yang dilakukan oleh penulis adalah sebagai berikut:

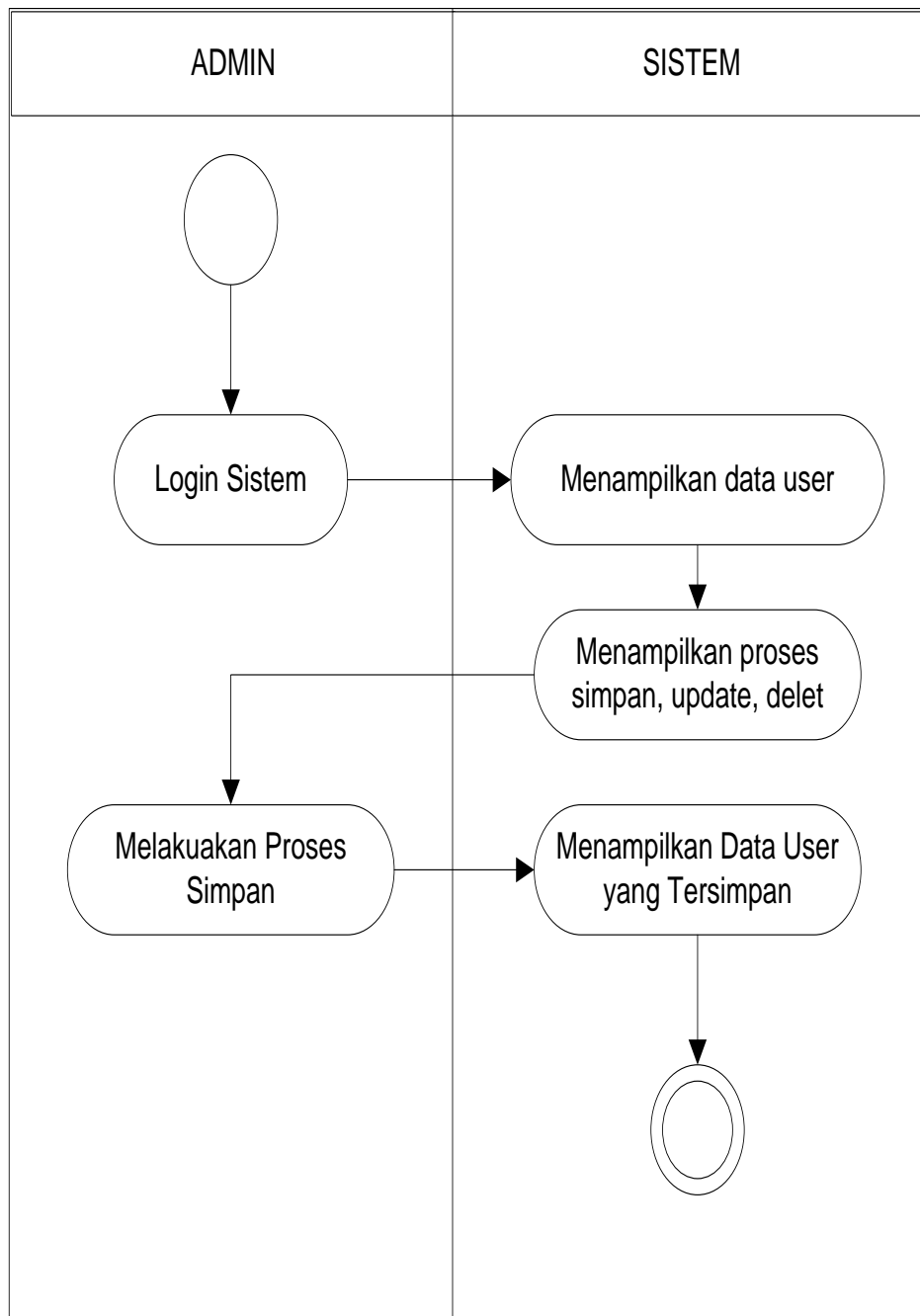


Gambar 3.5 Class Diagram

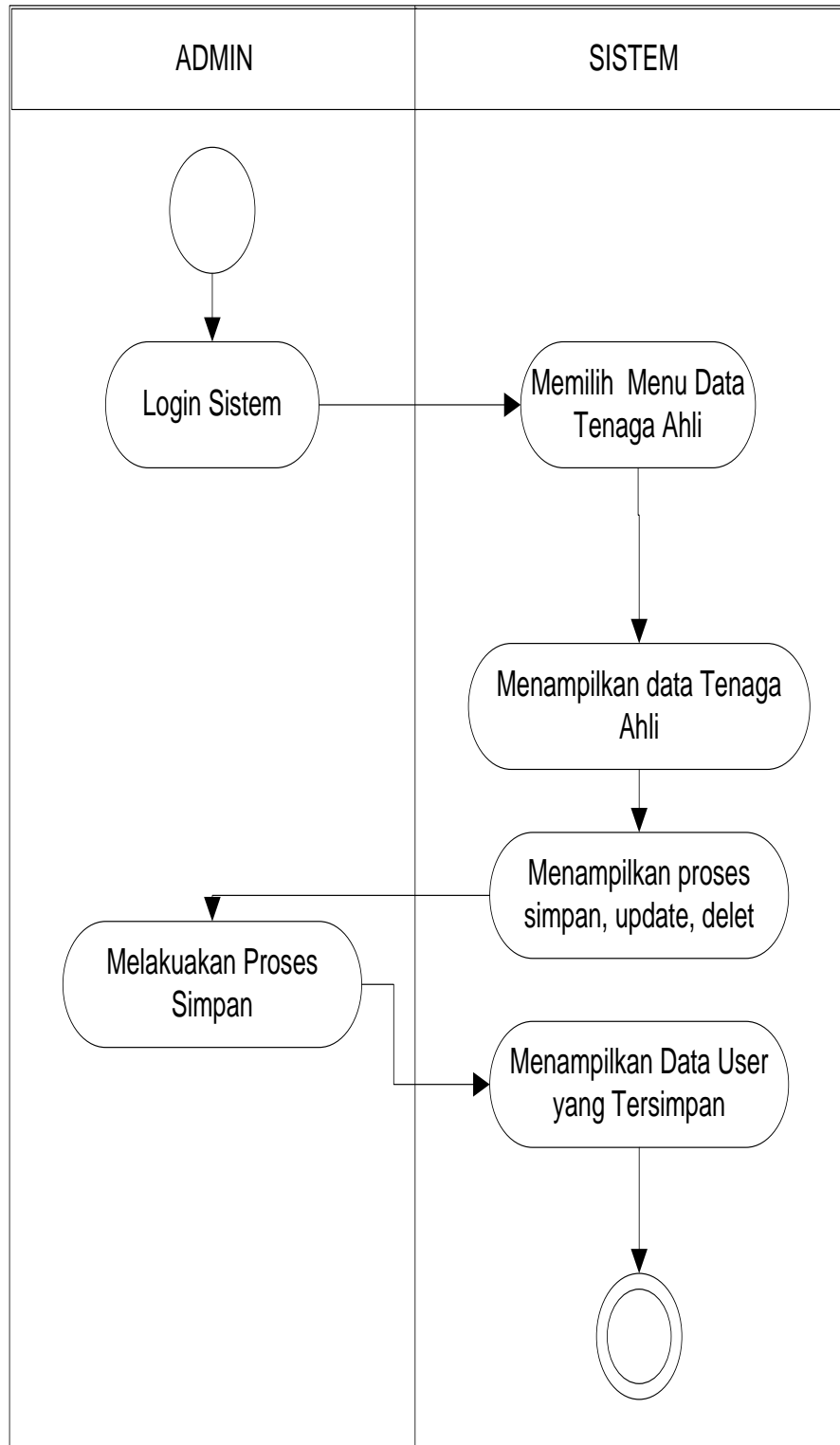
3.5.1 Desain Sistem Secara Detail

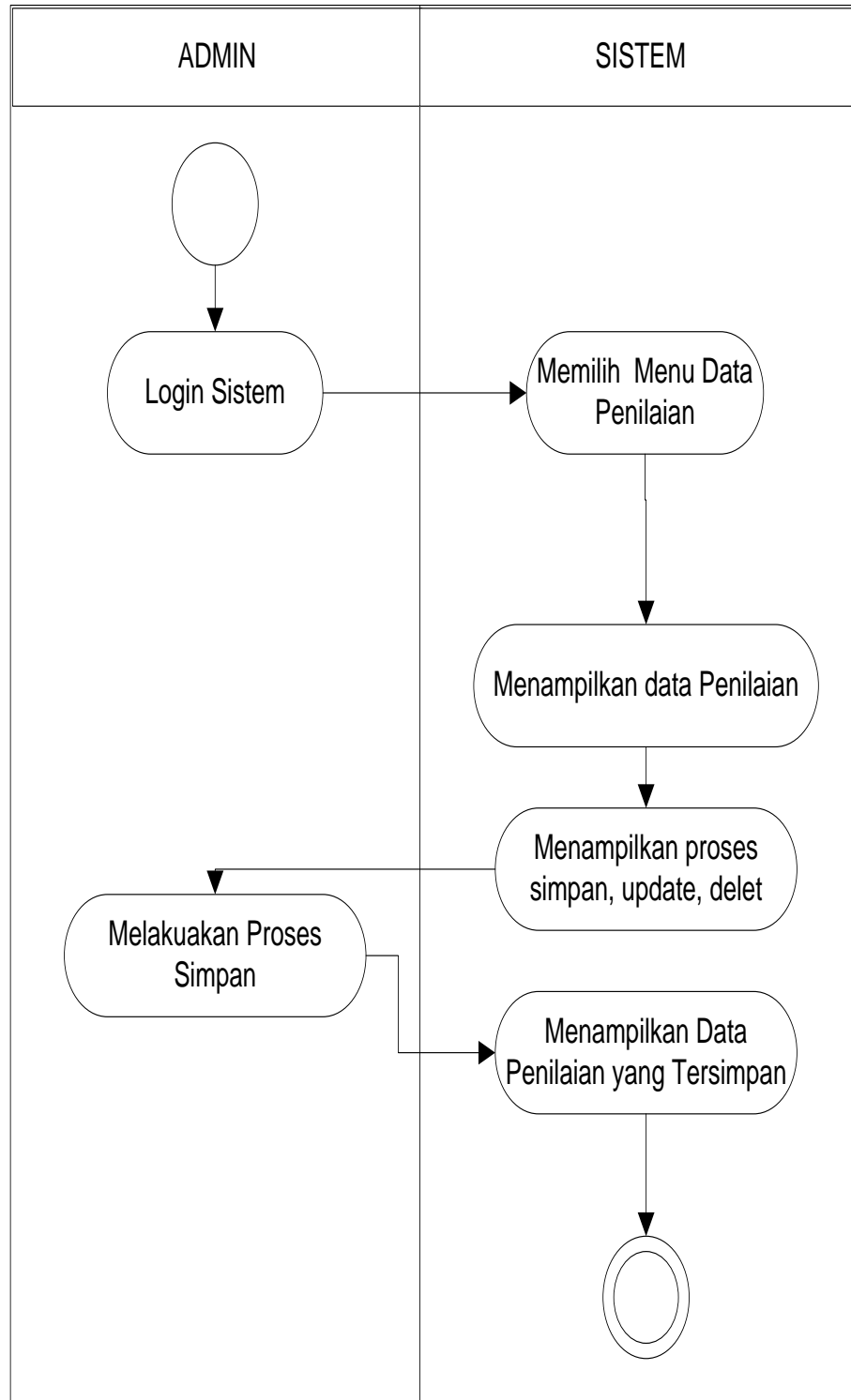
3.5.1.1 Activity Diagram

a. Activity Diagram Login

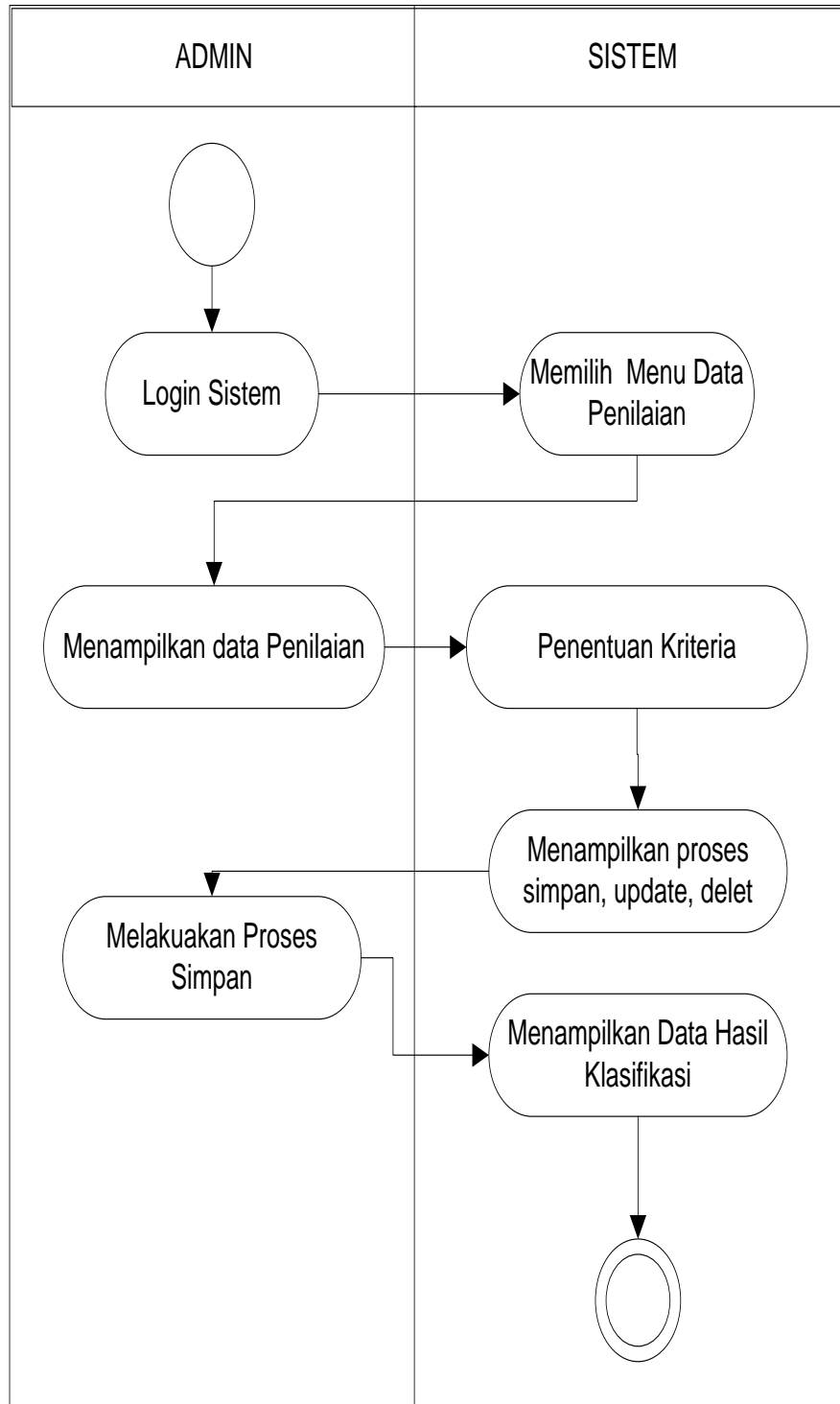


Gambar 3.6 Activity Diagram Login

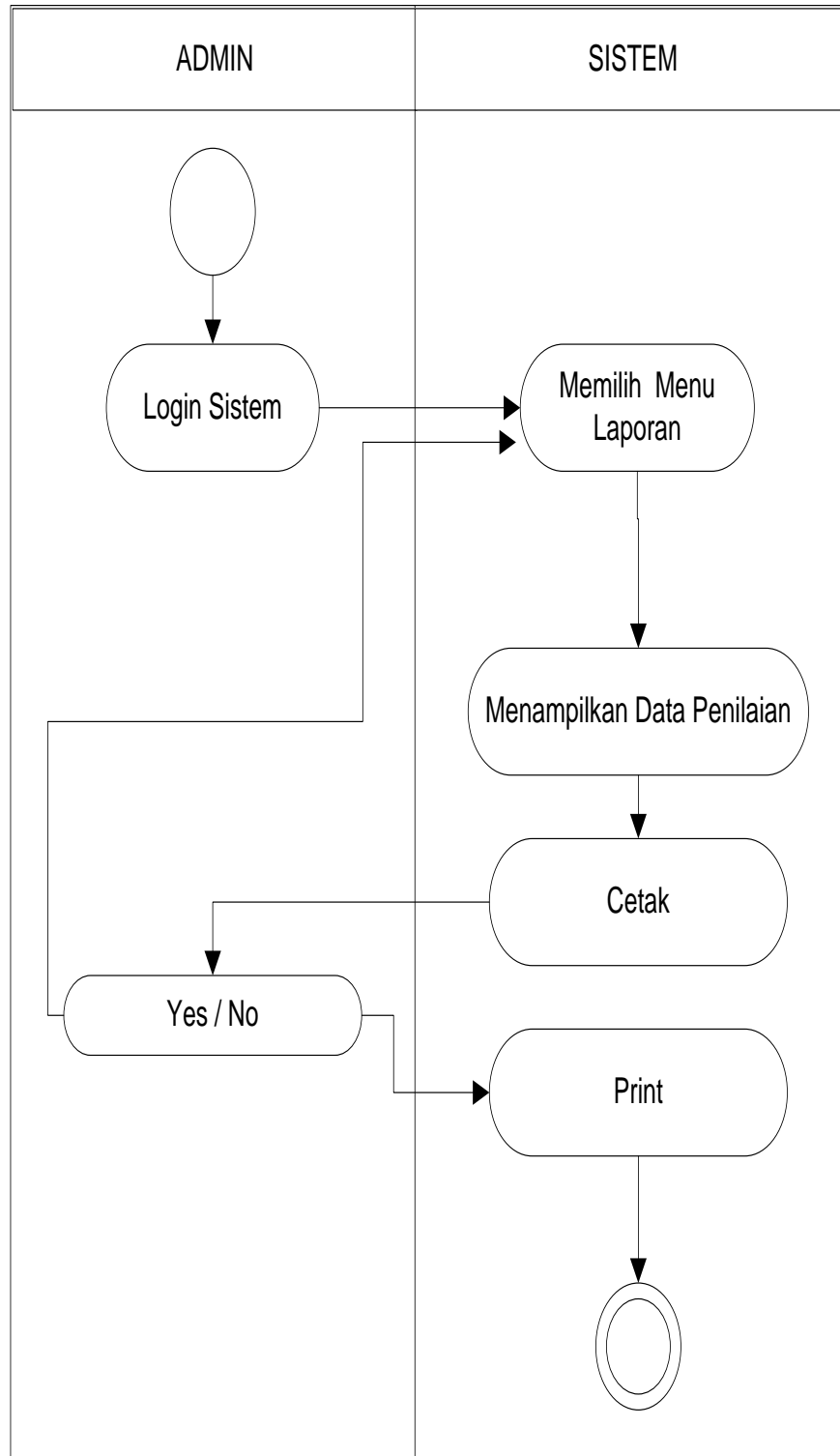
b. Activity Diagram Data Tenaga Ahli**Gambar 3.7** Activity Diagram Tenaga Ahli

c. Activity Diagram Data Penilaian**Gambar 3.8** Activity Diagram Penilaian

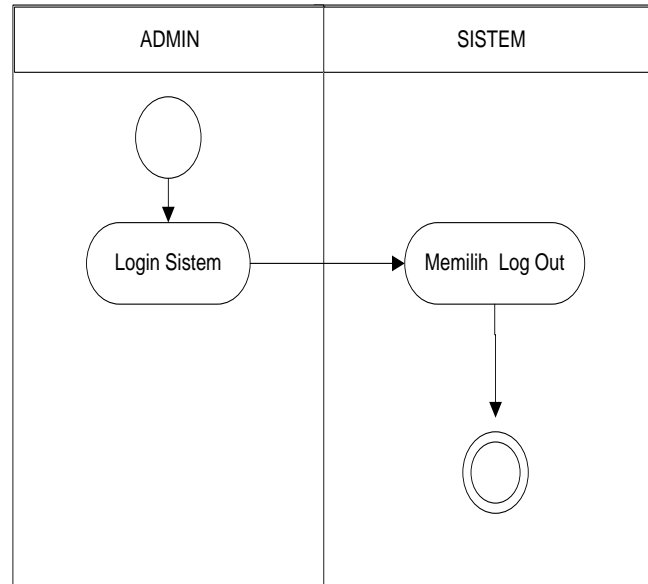
d. Activity Diagram Klasifikasi



Gambar 3.9 Activity Diagram Hasil Klasifikasi

e. Activity Diagram Laporan**Gambar 3.10** Activity Diagram Laporan

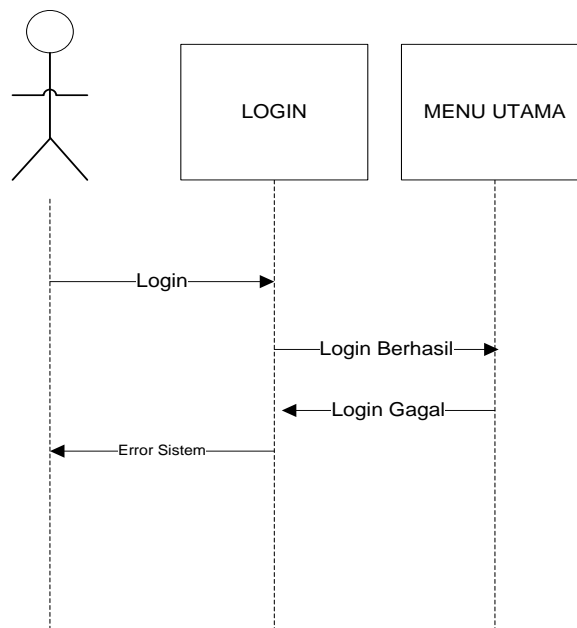
e. **Activity Diagram Log Out**



Gambar 3.11 Activity Diagram Log Out

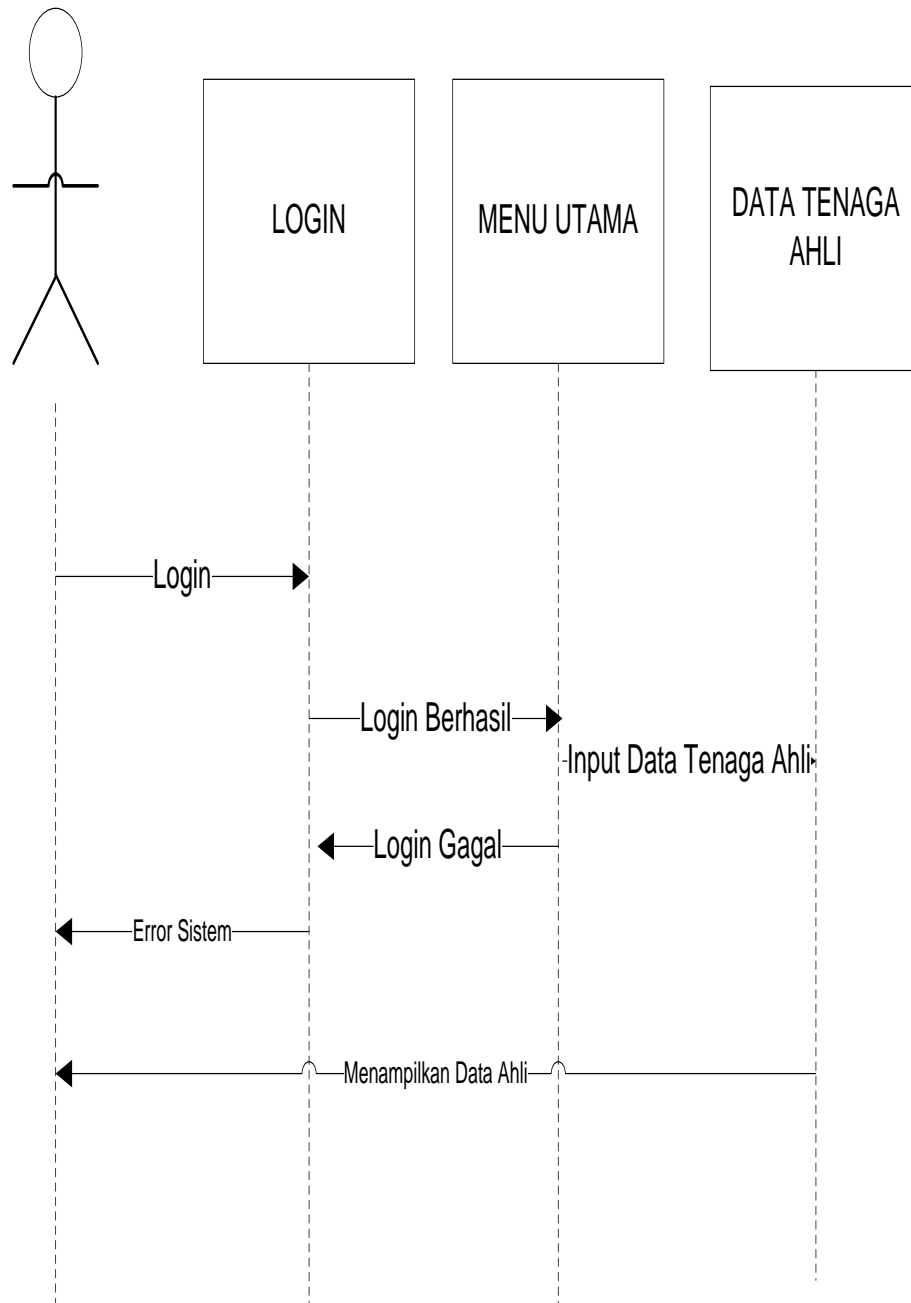
Adapaun sequence diagram sistem usulan yang dilakukan oleh penulis adalah sebagai berikut:

3.5.1.2 Sequence Diagram Login



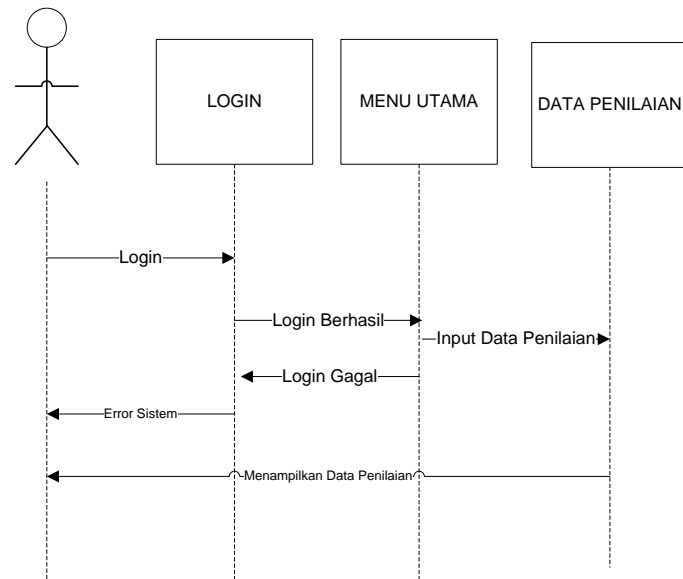
Gambar 3.12 Sequence Diagram Login

3.5.1.3 Sequence Diagram Data Tenaga Ahli



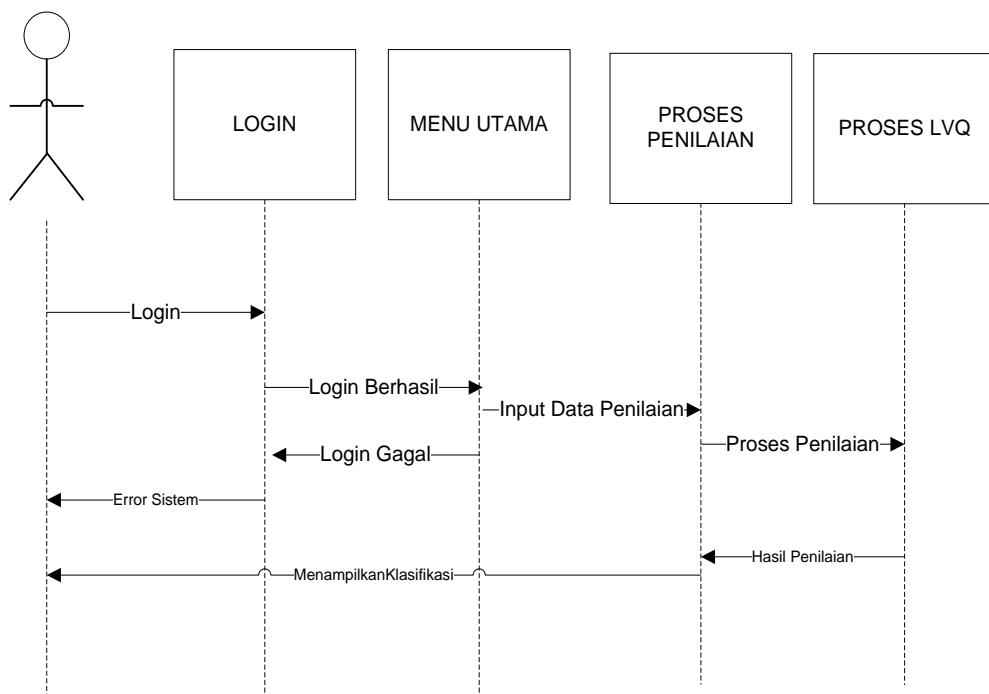
Gambar 3.13 Sequence Diagram Data Tenaga Ahli

3.5.1.4 Sequence Diagram Data Penilaian



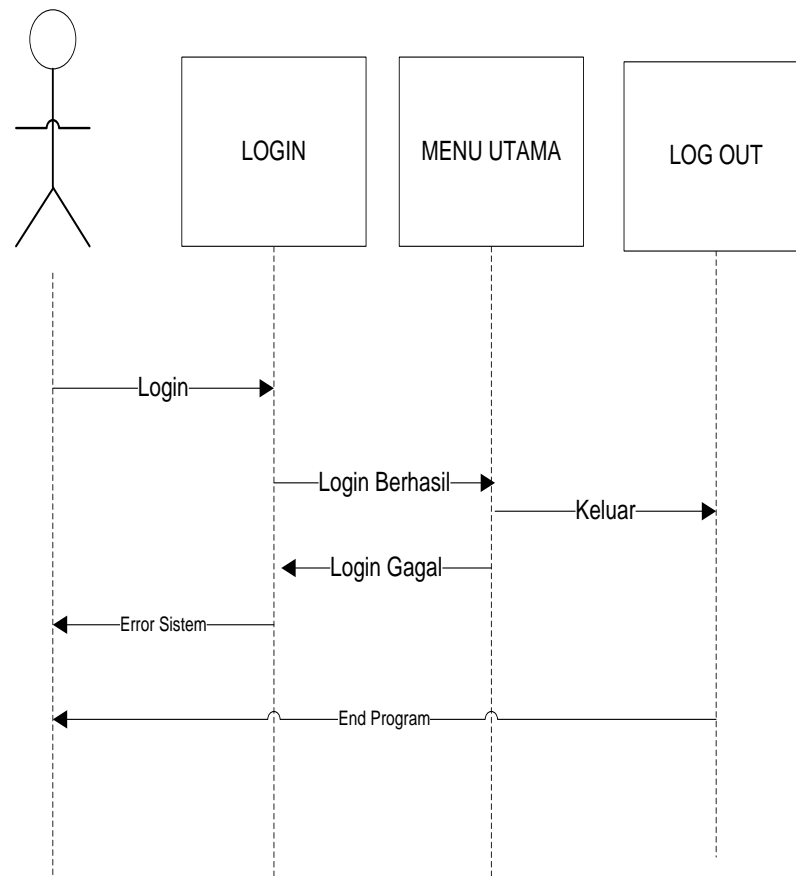
Gambar 3.14 Sequence Diagram Data Penilaian

3.5.1.5 Sequence Diagram Hasil Klasifikasi



Gambar 3.15 Sequence Diagram Hasil Klasifikasi

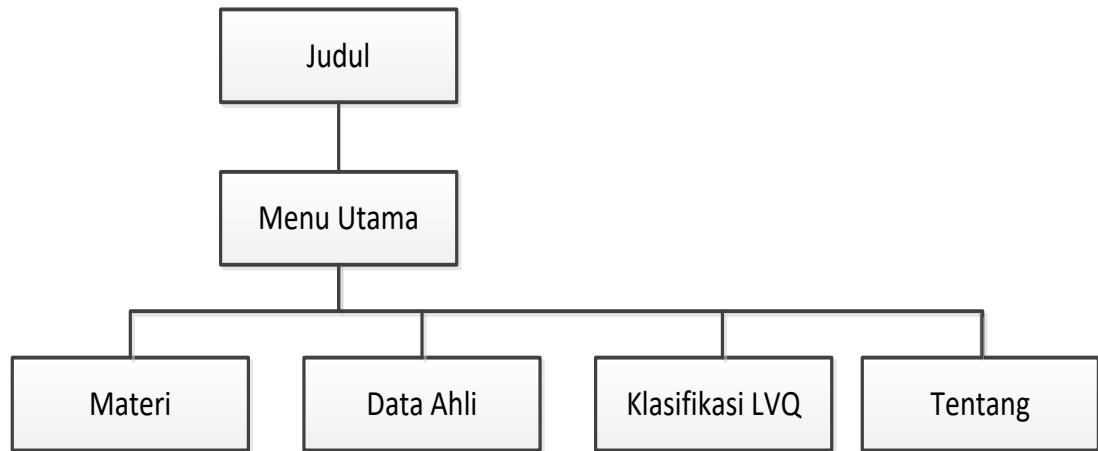
3.5.1.6 Sequence Diagram Log Out



Gambar 3.16 Sequence Diagram Log Out

3.6 Struktur Program

Struktur program mempresentasikan organisasi komponen program (modul) serta mengimplementasikan suatu hirarki kontrol. Hirarki kontrol tidak mengimplementasikan aspek prosedural dari perangkat lunak seperti urutan proses, kejadian atau urutan dari keputusan atau perulangan operasi.

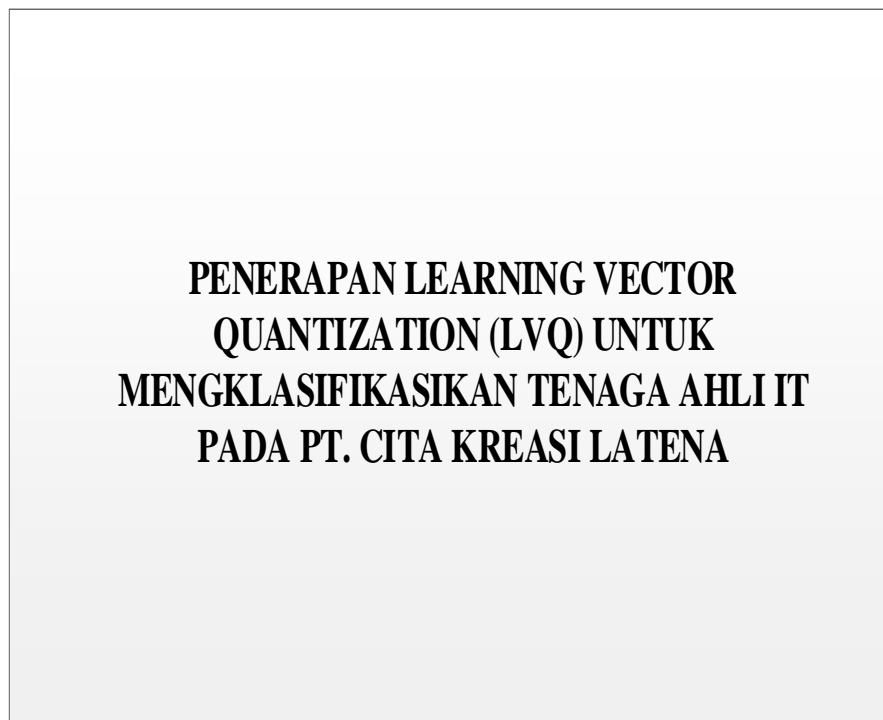


Gambar 3.17 Struktur Navigasi Data Ahli

3.7 Perancangan Antarmuka

1. Rancangan Halaman Judul

Halaman judul merupakan halaman yang pertama muncul pada saat program dijalankan



Gambar 3.18 Rancangan Halaman Judul

Pada rancangan di atas akan menampilkan judul yang kemudian akan pindah ke form menu utama dengan menggunakan timer.

2. Rancangan Halaman Menu Utama

Form ini berisi tombol-tombol seperti menu Materi, Data Ahli, Klasifikasi LVQ, tentang, dan Keluar.

Learning Vector Quantization			
File	Proses	Laporan	Keluar
—			

Gambar 3.19 Rancangan Halaman Menu Utama

3. Rancangan Halaman Data Pegawai

Form ini berisi form untuk menginputkan data pegawai.

Data Pegawai				
Kode Pegawai	<input type="text"/>			
Nama Pegawai	<input type="text"/>			
Alamat Pegawai	<input type="text"/>			
No Telp Pegawai	<input type="text"/>			
<input type="button" value="Baru"/>	<input type="button" value="Simpan"/>	<input type="button" value="Batal"/>	<input type="button" value="Edit"/>	<input type="button" value="Hapus"/>
<input type="button" value="Keluar"/>				

Gambar 3.20 Rancangan Halaman Data Pegawai

4. Rancangan Halaman Proses Iterasi

Form ini berisi form untuk proses data pegawai yang akan menghasilkan proses klasifikasi dari data pegawai menggunakan metode LVQ.

Data Pegawai Yang Dipilih						

Gambar 3.21 Rancangan Form Proses Iterasi

BAB IV

HASIL DAN PEMBAHASAN

4.1 Kebutuhan Sistem

Pembuatan program aplikasi adalah perancangan *interface* dan penulisan kode program sesuai dengan sistem yang telah dirancang. Untuk membuat program sistem komputerisasi pada sistem klasifikasi tenaga ahli IT ini menggunakan *software* pendukung yaitu:

1. *Microsoft Visual Studio 2008*

Microsoft Visual Studio 2008 digunakan sebagai tempat merancang *form-form* untuk Klasifikasi dalam menentukan Pegawai sehingga dapat menampilkan tampilan yang menarik serta memiliki keakuratan data. Rancangan *form-form* yaitu terdiri dari *Form Login*, Menu Utama, Menu *File*, Menu Proses, Menu Laporan, *Form Data Pegawai*, *Form Nilai Preferensi*, *Form Penilaian*, *Form Hasil Perhitungan LVQ*, *Form Laporan Hasil Perhitungan*.

2. *Microsoft Access 2007*

Microsoft Access digunakan sebagai media/tempat pembentukan *database* yang berisikan tabel-tabel yang diperlukan untuk pembentukan Klasifikasi dalam menentukan Pegawai.

3. Sistem Operasi *MS-Windows 2007*

Sistem Operasi merupakan syarat untuk dapat menjalankan atau instalasi program yang dirancang.

4. Spesifikasi *Hardware*

Sistem informasi yang telah terkomputerisasi ini dapat dijalankan apabila telah dilakukan beberapa hal, yaitu proses instalasi sudah dilakukan serta *hardware* yang mendukung dalam menjalankan program ini. Spesifikasi *hardware* yang dibutuhkan untuk mengimplementasikan sistem agar dapat berjalan dengan baik adalah sebagai berikut:

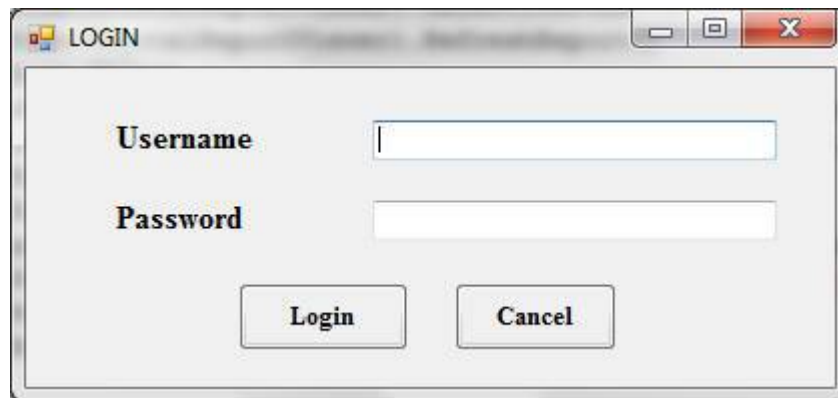
- Minimal *Pentium IV* 2.66 GHz
- RAM dengan kapasitas minimal 1 Gb
- *Harddisk* kapasitas minimal 60 Gb
- *Mainboard P4*
- *Monitor SVGA* dengan resolusi layar minimal 1024 x 768
- *Keyboard* dan *Mouse*
- *CD Room*
- *Printer* sebagai perangkat untuk mencetak laporan

4.2 Implementasi Sistem

Adapun implementasi sistem program dari Klasifikasi dalam menentukan Tenaga Ahli IT menggunakan metode LVQ adalah sebagai berikut :

1. Form Login

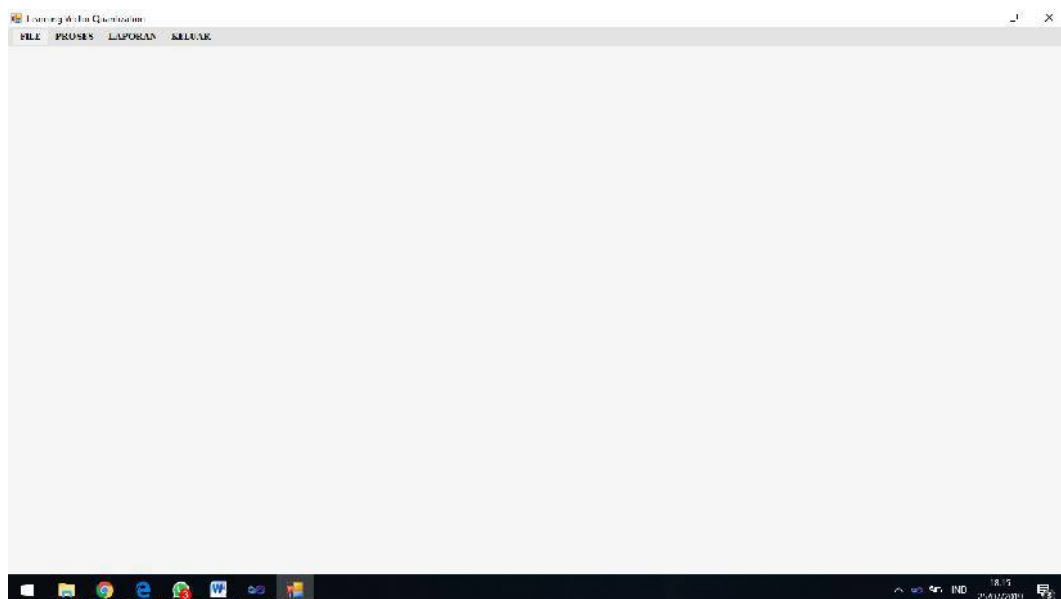
Form login digunakan sebagai *security* untuk sistem yang dibangun, mengantisipasi agar sistem tidak dapat di lihat orang lain. Tampilan *Login* ini dapat dilihat pada gambar 4.1 di bawah ini.



Gambar 4.1 Tampilan *Form Login*

2. Menu Utama

Tampilan menu utama dirancang sebagai *interface* untuk membantu *user* dalam melakukan eksekusi sistem yang dibangun, sehingga mempermudah dalam proses pengolahan data. Menu utama terdiri dari Menu *File*, Menu *Proses*, dan *Laporan* dimana dalam masing-masing menu terdapat *form-form* yang dibutuhkan. Tampilan menu utama dapat dilihat pada gambar 4.2 di bawah ini:



Gambar 4.2 Tampilan Menu Utama

3. Menu *File*

Menu *File* merupakan menu yang akan menampilkan parameter- parameter yang dibutuhkan untuk proses yang terjadi pada sistem yang dibangun, Menu *File* digunakan sebagai induk parameter yang terdiri dari *Form Input* Data Pegawai,

4. Menu Proses

Menu Proses digunakan sebagai *Interface* untuk menampilkan *Form Input* Penilaian dan *Interface* untuk menampilkan *Form* Hasil Perhitungan LVQ.

1. *Form* Penilaian

Form Input Penilaian digunakan sebagai parameter penilaian terhadap kriteria yang sudah ditentukan berdasarkan data Pegawai yang akan disimpan untuk proses perhitungan LVQ. Tampilan *form* ini dapat dilihat pada gambar 4.5 di bawah ini:

NIP	Nama	Programming	Database	Design	Analisa Sistem
123456	Wita	90	90	100	100
1	Pina	90	80	90	90
12344	Monel	100	100	100	100

Gambar 4.3 Tampilan *Form Input* Penilaian

2. *Form* Hasil Perhitungan LVQ

Form Hasil Perhitungan LVQ digunakan untuk menampilkan parameter

penilaian terhadap kriteria yang sudah ditentukan dan sebagai tampilan proses perhitungan LVQ. Tampilan *form* ini dapat dilihat pada gambar 4.6 di bawah ini:

TABEL NILAI PEGAWAI

NIP	Nama Pegawai	Programming	Database	Design	Analisa Sistem
1	Rina	90	80	90	90
1234	Anjur OCTavia	90	90	90	90
12344	Monel	80	90	80	90
123456	Wita	100	100	100	100

HASIL KLASIFIKASI

NIP	Nama Pegawai	Nilai Hasil	Keterangan
1	Rina	0.3686952	Asisten Tenaga Ahli IT
1234	Anjur OCTavia	0.4704748	Tenaga Ahli IT (Programmer)
12344	Monel	0.2225733	Asisten Tenaga Ahli IT
123456	Wita	1	Tenaga Ahli IT (Team Leader)

Gambar 4.4 Tampilan *Form* Hasil Perhitungan LVQ

5. Kelemahan dan Kelebihan Sistem

Adapun kelemahan dari Klasifikasi dalam menentukan tenaga ahli IT menggunakan metode LVQ yang diusulkan adalah:

1. Sistem yang diusulkan hanya dapat menghasilkan laporan hasil perhitungan
2. Sistem hanya dapat diakses oleh satu *user* yaitu Pimpinan
3. Sistem tidak dapat menambah kriteria

Adapun kelebihan Klasifikasi dalam menentukan tenaga ahli IT yang diusulkan adalah:

1. Menjadi sumber informasi yang akurat untuk Pimpinan.
2. Proses klasifikasi lebih efektif dan efisien.

6. Pengujian Black Box

Untuk dapat menggunakan aplikasi ini dengan baik, dibutuhkan seperangkat komputer dengan spesifikasi minimal. Black Box pengujian adalah metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi yang bertentangan dengan struktur internal atau kerja. Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: unit, integrasi, fungsional, sistem dan penerimaan.

Tabel 4.1. Tabel Pengujian Black Box

No	Rancangan Proses	Hasil Yang Diharapkan	Hasil	Keterangan
1	Halaman Utama Interaktif dan Mudah Digunaka	Halaman (Awal)	Sesuai	-
2	Proses Data Tenaga Ahli Mudah Digunakan	Halaman Data Tenaga Ahli	Sesuai	-
3	Proses Data Kriteria Mudah Digunakan	Halaman Kriteria	Sesuai	-
4	Proses Klasifikasi	Halaman Hasil	Sesuai	-

BAB V

PENUTUP

5.1 Kesimpulan

Setelah dilakukan implementasi program dan pengujian pada bab sebelumnya, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Metode LVQ merupakan suatu metode dalam klasifikasi untuk melakukan perhitungan yang cukup akurat didalam proses penentuan klasifikasi tenaga ahli.
2. Dengan adanya aplikasi klasifikasi untuk proses pemilihan tenaga ahli ini dapat mempermudah pihak pimpinan PT. Cita Kreasi Latena untuk melakukan seleksi tenaga ahli.

5.2 Saran

Beberapa saran yang ingin disampaikan untuk melakukan pengembangan terhadap sistem yang ingin dibangun dimasa yang akan datang terhadap program aplikasi ini sebagai berikut :

1. Sistem ini hendaknya bukan hanya melakukan perhitungan didalam klasifikasi tenaga ahli saja, tetapi harus dalam ruang lingkup yang lebih kompleks dan lebih luas lagi.
2. Program yang sudah dirancang atau yang sudah dibangun harus dikembangkan secara terus menerus dan harus mengikuti perkembangan zaman yang sudah serbacanggih seperti sekarang ini yang sudah serba web, gadget dan android.

3. Hendaknya program aplikasi ini tidak hanya digunakan oleh satu user saja tetapi harus dikembangkan menjadi berbasis multi-user artinya banyak pengguna atau kalangan yang dapat menggunakannya.

DAFTAR PUSTAKA

- Afrianto, I. (2012). Perbandingan Metode Jaringan Syaraf Tiruan Backpropagation Dan Learning Vector Quantization Pada Pengenalan Wajah. *KOMPUTA: Jurnal Komputer dan Informatika*, 1(1).
- Fachri, barany, agus perdana windarto, and ikhsan parinduri. "penerapan backpropagation dan analisis sensitivitas pada prediksi indikator terpenting perusahaan listrik." *jepin (jurnal edukasi dan penelitian informatika)* 5.2 (2019): 202-208.
- Fachri, b., windarto, a. P., & parinduri, i. (2019). Penerapan backpropagation dan analisis sensitivitas pada prediksi indikator terpenting perusahaan listrik. *Jepin (jurnal edukasi dan penelitian informatika)*, 5(2), 202-208.
- Fachri, barany; windarto, agus perdana; parinduri, ikhsan. Penerapan backpropagation dan analisis sensitivitas pada prediksi indikator terpenting perusahaan listrik. *Jepin (jurnal edukasi dan penelitian informatika)*, 2019, 5.2: 202-208.
- Fansuri, M. R. (2011). *Klasifikasi Genre Musik Menggunakan Learning Vector Quantization (LVQ)*. Institut Petanian Bogor, Bogor, 4.
- Hamdi, nurul. "model penyiraman otomatis pada tanaman cabe rawit berbasis programmable logic control." *jurnal ilmiah core it: community research information technology* 7.2 (2019).
- Lesnussa, Y. A., Latuconsina, S., & Persulesy, E. R. (2015). Aplikasi Jaringan Saraf Tiruan Backpropagation untuk Memprediksi Prestasi Siswa SMA (Studi kasus: Prediksi Prestasi Siswa SMAN 4 Ambon). *Jurnal Matematika Integratif*, ISSN, 1412-6184.
- Meliawati, Soesanto, Kartini (2016). Penerapan Metode Learning Vector Quantization (LVQ) pada Prediksi jurusan di SMA PGRI 1 Banjarbaru. *Kumpulan Jurnal Ilmu Komputer (KLIK)*. Volume 04, No. 01 Februari 2016. ISSN: 2406-7857.
- Mohamad Natsir (2017). Pengembangan Prototype Sistem Kriptografi Untuk Enkripsi Dan Dekripsi Data Office Menggunakan Metode Blowfish Dengan Bahasa Pemrograman Java. *Jurnal Format*. Volume 06, No. 01 Tahun 2017. ISSN: 2089-5615.

- Nugroho, F. F. Kurniawan, Arif, Y. M. & Dermawan, D. A., 2011. Simulasi Multi Atribut di Dasarkan pada Agen untuk Keandalan Distribusi Energi Listrik menggunakan Metode LVQ. Seminar on Electrical, Informatics.
- Permana, aminuddin indra. "kombinasi algoritma kriptografi one time pad dengan generate random keys dan vigenere cipher dengan kunci em2b." (2019).
- Putra, randi rian. "sistem informasi web pariwisata hutan mangrove di kelurahan belawan sicanang kecamatan medan belawan sebagai media promosi." jurnal ilmiah core it: community research information technology 7.2 (2019).
- Putra, randi rian, et al. "decision support system in selecting additional employees using multi-factor evaluation process method." (2019).
- Putra, randi rian. "implementasi metode backpropagation jaringan saraf tiruan dalam memprediksi pola pengunjung terhadap transaksi." jurti (jurnal teknologi informasi) 3.1 (2019): 16-20.
- Saputra, muhammad juanda, and nurul hamdi. "rancang bangun aplikasi sejarah kebudayaan aceh berbasis android studi kasus dinas kebudayaan dan pariwisata aceh." journal of informatics and computer science 5.2 (2019): 147-157
- Sidik, a. P., efendi, s., & suherman, s. (2019, june). Improving one-time pad algorithm on shamir's three-pass protocol scheme by using rsa and elgamal algorithms. In journal of physics: conference series (vol. 1235, no. 1, p. 012007). Iop publishing.
- Sitepu, n. B., zarlis, m., efendi, s., & dhany, h. W. (2019, august). Analysis of decision tree and smooth support vector machine methods on data mining. In journal of physics: conference series (vol. 1255, no. 1, p. 012067). Iop publishing.
- Syadid Zamb, Ninuk Wliani (2017). Rancang Bangun Aplikasi Kasir Tiket Nonton Bola Bareng Pada X Kasir Di Suatu Lokasi X Dengan Visual Basic 2010 Dan Mysql. *Jurnal Rekayasa Informasi, Vol. 6. No.2, Oktober 2017*. ISSN: 2252-7354.
- Tasril, v., wijaya, r. F., & widya, r. (2019). Aplikasi pintar belajar bimbingan dan konseling untuk siswa sma berbasis macromedia flash. *Jurnal informasi komputer logika*, 1(3).
- Winda Aprianti, Umi Maliha (2016). Sistem Informasi Kepadatan Penduduk Kelurahan Atau Desa Studi Kasus Pada Kecamatan Bati-Bati Kabupaten Tanah Laut. *Jurnal Sains dan Informatika*, Volume 2, Nomor 1, Juni 2016. ISSN: 2460-173X.

Yunahar Heriyanto (2018). Perancangan Sistem Informasi Rental Mobil Berbasis Web Pada Pt.Apm Rent Car. *Jurnal Intra-Tech*, Volume 2, Nomor 2, Oktober 2018. ISSN: 2549-0222.

Ying, Zhang & Mei, Li, 2016. An Evaluation Model of Water Quality Based of Learning Vector Quantization Neural Network. Proceedings of the 35th Chinese Control Conference, Pp.3658-3689.