JARINGAN SYARAF TIRUAN
FUZZY LEARNING VECTOR QUANTIZATION (FLVQ)
UNTUK MENGIDENTIFIKASI BEBERAPA DISTRIBUSI PELUANG

(Skripsi)

Oleh

GERRY ALFA DITO

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDARLAMPUNG
2016

ABSTRACT

NEURAL NETWORK FUZZY LEARNING VECTOR QUANTIZATION (FLVQ) TO IDENTIFY PROBABILITY DISTRIBUTIONS

By

GERRY ALFA DITO

Statistical model is built based on probability distribution. Classically, probability distribution is identified by nonparametric goodness of fits test. In this study will be discussed FLVQ model to identify probability distribution, whereas this model is a merger between neural network and fuzzy set. Result obtained FLVQ model in identify probability distribution is good enough.

Keyword: neural network, fuzzy set, goodness of fits test, and fuzzy sets

ABSTRAK

JARINGAN SYARAF TIRUAN *FUZZY LEARNING VECTOR QUANTIZATION* (FLVQ) UNTUK MENGIDENTIFIKASI BEBERAPA DISTRIBUSI PELUANG

Oleh

GERRY ALFA DITO

Model Statistika dibangun berdasarkan distribusi peluang. Secara klasik, distribusi peluang diidentifikasi dengan uji *goodness of fits* nonparametrik. Dalam penelitian ini akan dibahas dengan model FLVQ untuk mengklasifikasikan distribusi peluang, dimana model FLVQ merupakan penggabungan antara konsep jaringan syaraf tiruan dan himpunan *fuzzy*. Hasil yang diperoleh model FLVQ dalam mengklasifikasikan distribusi peluang cukup baik.

Kata kunci: jaringan syaraf tiruan, distribusi peluang, uji *goodness of fits*, dan himpunan *fuzzy*

JARINGAN SYARAF TIRUAN
FUZZY LEARNING VECTOR QUANTIZATION (FLVQ)
UNTUK MENGIDENTIFIKASI BEBERAPA DISTRIBUSI PELUANG

Oleh

GERRY ALFA DITO

Skripsi
Sebagai Salah Satu Syarat untuk Memperoleh Gelar
SARJANA SAINS
Pada
Jurusan Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Lampung

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
2016

Judul Skripsi	:	**JARINGAN SYARAF TIRUAN *FUZZY LEARNNG VECTOR QUANTIZATION* (FLVQ) UNTUK MENGIDENTIFIKASI BEBERAPA DISTRIBUSI PELUANG**
Nama Mahasiswa	:	Gerry Alfa Dito
Nomor Pokok Mahasiswa	:	1217031029
Jurusan	:	Matematika
Fakultas	:	Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI,

1. Komisi Pembimbing

Warsono, Ph.D.
NIP. 19630216 198703 1 003

Dian Kurniasari, M.Sc.
NIP.19690305 199603 2 001

2. Ketua Jurusan Matematika

Drs. Tiryono Ruby, M.Sc., Ph.D.
NIP. 19620704 198803 1 002

MENGESAHKAN

1. Tim Penguji

   Ketua : **Warsono, Ph.D.** .....

   Sekretaris : **Dian Kurniasari, M.Sc.** .....

   Penguji
   Bukan Pembimbing : **Mustofa Usman, Ph.D.** .....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Prof. Warsito, S.Si., DEA., Ph.D.
NIP. ~~19710212~~ 199512 1 001

Tanggal Lulus Ujian Skripsi: **28 April 2016**

PERNYATAAN

Nama : Gerry Alfa Dito

Nomor Pokok Mahasiswa : 1217031029

Program Studi : Matematika

Jurusan : Matematika

Dengan ini saya menyatakan bahwa skripsi ini adalah hasil karya saya sendiri. Skripsi ini juga tidak berisi materi yang dipublikasikan atau ditulis oleh orang lain atau telah dipergunakan dan diterima sebagai persyaratan penyelesaian studi pada Universitas Lampung atau institusi lain.

Bandar Lampung, Mei 2016

Gerry Alfa Dito

NPM. 1217031032

RIWAYAT HIDUP

Penulis dilahirkan di Bagelen, Kecamatan Gedong Tataan,, Pesawaran pada 2 November 1993, Sebagai anak pertama dari dua bersaudara.

Pendidikan Sekolah Dasar (SD) diselesaikan penulis pada tahun 2006 di SD Fransiskus Pringsewu, Pringsewu, Lampung. Sekolah Menengah Pertama (SMP) diselesaikan pada tahun 2009 di SMP Xaverius Pringsewu, dan Sekolah Menengah Atas (SMA) diselesaikan pada tahun 2012 di SMA Xaverius Pringsewu.

Pada pertengahan tahun 2012 penulis terdaftar sebagai mahasiswa Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung. Selama menjadi mahasiswa, penulis tergabung dalam organisasi HIMATIKA (Himpunan Mahasiswa Jurusan Metematika) FMIPA Unila sebagai anggota bidang keilmuan pada periode 2013/2014, Kepala Bidang Keilmuan 2014/2015 dan Dewan Pertimbangan Organisasi (DPO).

Pada awal tahun 2015, penulis melaksanakan Praktek Kerja Lapangan (PKL) di Bank Indonesia Kantor Perwakilan Provinsi Lampung. Pada pertengahan tahun 2015 penulis melaksanakan Kuliah Kerja Nyata (KKN) di Desa Mulya Sari, Kecamatan Gunung Agung, Kabupaten Tulang Bawang Barat.

PERSEMBAHAN

Alhamdulillahirabbil 'alamiin dengan penuh rasa syukur kepada Allah SWT, kupersembahkan karya kecilku ini untuk orang-orang yang selalu mengasihi, menyayangi, dan memotivasiku dalam segala hal.

Bapak dan Ibu tercinta yang telah membesarkanku dan menyayangiku dengan penuh kasih sayang yang tak terhingga serta selalu mendoakan dan memberi motivasi kepadaku.

Dosen pembimbing dan penguji yang tiada henti-hentinya memberikan ilmu dan pelajaran berharga kepadaku.

Sahabat-sahabatku yang selalu berbagi kebahagiaan, keceriaan, saling mendukung, dan menyemangati.

Be better than yesterday

Dream what you dare to dream, go where you want to go, Be what you want to be

Ilmu itu lebih baik daripada harta. Ilmu menjaga engkau dan engkau menjaga harta. Ilmu itu penghukum (hakim) dan harta terhukum. Harta itu kurang apabila dibelanjakan tapi ilmu bertambah bila dibelanjakan
(Ali bin Abi Thalib)

SANWACANA

Puji syukur penulis ucapkan atas kehadirat Allah SWT karena berkat rahmat dan hidayah-Nya skripsi yang berjudul "Jaringan Syaraf Tiruan *Fuzzy Learning Vector Quantization* (FLVQ) untuk Mengidentifikasi Beberapa Distribusi Peluang" dapat diselesaikan tepat pada waktunya. Selain itu, sholawat serta salam selalu tercurahkan kepada Nabi Muhammad SAW.

Penulis menyadari bahwa banyak pihak yang telah membantu dalam menyelesaikan penulisan skripsi ini. Oleh karena itu, doa dan ucapan terimakasih penulis sampaikan, terutama kepada :

1. Bapak Warsono, Ph.D. selaku dosen pembimbing pertama yang telah memberikan bimbingan dan arahan serta motivasi kepada penulis dalam menyelesaikan skripsi ini.
2. Ibu Dian Kurniasari, M.Sc selaku pembimbing kedua dan sekaligus pembimbing akademik yang telah memberikan bimbingan dan motivasi dalam menyelesaikan skripsi ini serta selalu memberi saran, dukungan dan motivasi selama masa perkuliahan.
3. Bapak Mustofa Usman, Ph.D selaku pembahas yang telah memberikan kritik dan saran yang membangun dalam penulisan skripsi ini.

4. Bapak Drs. Tiryono Ruby, M.Sc., Ph.D. selaku Ketua Jurusan Matematika FMIPA Universitas Lampung.
5. Bapak Prof. Warsito, S.Si., DEA., Ph.D. selaku Dekan FMIPA Universitas Lampung.
6. Orang tua tercinta yang telah membesarkan, mendidik, dan memberi kasih sayang yang begitu besar serta telah menjadi inspiratorku.
7. Yeftanus Antonio yang selalu memberi kritik dan saran.
8. Keluarga besar HIMATIKA FMIPA Unila yang telah memberikan dukungan dan semangatnya
9. Matematika 2012 atas keceriaan dan kebersamaannya selama ini.
10. Seluruh rekan-rekan yang tidak dapat disebutkan satu persatu oleh penulis.

Akhir kata, Penulis menyadari bahwa penulisan skripsi ini masih jauh dari kesempurnaan. Oleh karena itu, kritik dan saran yang membangun sangat Penulis harapkan.

Bandar Lampung, Mei 2016

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR xiv

DAFTAR TABEL xvi

I. PENDAHULUAN 1

1.1 Latar Belakang dan Masalah 1

1.2 Batasan Masalah 2

1.3 Tujuan Penelitian 3

1.4 Manfaat Penelitian 3

II. TINJAUAN PUSTAKA 4

2.1 Jaringan Syaraf Tiruan (*Artificial Neural Network*) 4

2.1.1 Model Neuron 4

2.1.2 Arsitektur Jaringan Syaraf Tiruan 6

2.1.3 Operasi dari Jaringan Syaraf Tiruan 8

2.1.4 Sifat-Sifat Jaringan Syaraf Tiruan 8

2.2 Proses Pembelajaran (*Learning Process*) 9

2.2.1 Komponen dari Pembelajaran (*Learning*) 9

2.2.2 Jenis-Jenis Paradigma Pembelajaran 10

2.3 *Generalization* (Generalisasi) 12

2.4 *Learning Vector Quantization* (FLVQ) 15

2.4.1 *Vector Quantization* 15

2.4.2 *Competitive Learning* (Pembelajaran Kompetitif) 16

2.4.3 Algoritma *Learning Vector Quantization* (LVQ) 17

2.5 Himpunan *Fuzzy* 21

2.5.1 Definisi-Definisi Dasar 21

2.5.2 Operasi pada Himpunan *Fuzzy* 25

2.6 Himpunan *Fuzzy* 29

2.7 Ukuran Kemiripan *Fuzzy* 34

2.8 Aritmatika *Fuzzy* 35

2.9 Peubah Acak *Fuzzy* 36

2.10 K-Nearest Neighboor (KNN) 39

2.11 Metode Monte Carlo 40

III. METODOLOGI PENELITIAN 41

3.1 Waktu dan Tempat Penelitian 41

3.2 Metode Penelitan 41

IV. HASIL DAN PEMBAHASAN 44

4.1 *Fuzzy Learning Vector Quantization* (FLVQ) 44

4.1.1 Data *Input* 44

4.1.2 Vector *Codebook* 46

4.1.3 Pemrosesan Elemen pada Model FLVQ 47

4.1.4 Pembelajaran FLVQ 50

4.2 Penerapan FLVQ untuk Mengidentifikasi Distribusi Peluang 54

4.2.1 Algoritma FLVQ untuk Mengidentifikasi Distribusi Peluang 54

4.2.2 Hasil Identifikasi Distribusi Peluang 59

V. KESIMPULAN..... 63

5.1 Kesimpulan 63

5.2 Saran 63

DAFTAR PUSTAKA 64

LAMPIRAN 66

DAFTAR GAMBAR

Halaman

Gambar 2.1 Model neuron 5

Gambar 2.2 a. *Feedforward network* b. *Recurrent network* c. *Two-dimensional lattice network* d. *layered feedforward network with lateral connection* e. *cellular network* 7

Gambar 2.3 Proses *learning* (pembelajaran) 10

Gambar 2.4 Contoh *overfitting* 13

Gambar 2.5 Arsitektur J-K *neural network* 17

Gambar 2.6 Wilayah “window” pada algoritma LVQ2 20

Gambar 2.7 Himpunan *fuzzy* A dan B 22

Gambar 2.8 *Fuzzy singleton* 23

Gambar 2.9 Himpunan *fuzzy* tidak konveks 24

Gambar 2.10 α*-cut* pada himpunan *fuzzy* 26

Gambar 2.11 Komplemen himpunan *fuzzy* 27

Gambar 2.12 Gabungan dari dua himpunan *fuzzy* 27

Gambar 2.13 Irisan dari dua himpunan *fuzzy* 28

Gambar 2.14 Inklusi dari dua himpunan *Fuzzy* 28

Gambar 2.15 Interval tertutup yang diinterpretasikan dengan bilangan *fuzzy* 31

Gambar 2.16 Bilangan *fuzzy* trapesium 32

Gambar 2.17 Bilangan *fuzzy* segitiga 33

Gambar 2.18 Bilangan *fuzzy* Gaussian 34

Gambar 2.19 Nilai dari ‘persepsi tentang biaya registrasi’ dari suatu konfrensi (dalam euro) 39

Gambar 4.1 Bilangan *fuzzy* yang menyatakan bilangan disekitar 5 45

Gambar 4.2 Representasi vektor *fuzzy* dalam bentuk grafik 46

Gambar 4.3 Representasi vektor *codebook fuzzy* dalam bentuk grafik.....47

Gambar 4.4 Memperbaharui vektor *codebook* ketika klasifikasi tepat.....52

Gambar 4.5 Diagram alir identifikasi distribusi peluang dengan FLVQ 55

Gambar 4.6 Diagram tahap *training* FLVQ 57

Gambar 4.7 Diagram alir algoritma *training* FLVQ 58

Gambar 4.8 Diagram alir algoritma *testing* FLVQ 60

DAFTAR TABEL

Halaman

Tabel 4.1 Kebenaran klasifikasi berdasarkan distribusi peluang 61

Tabel 4.2 Kekuatan uji model FLVQ 62

I. PENDAHULUAN

1.1 Latar Belakang dan Masalah

Dalam analisis data statistika, sering kali ingin diketahui distribusi peluang dari suatu data yang akan diteliti. Hal ini dikarenakan jika ingin dibangun model dari data dan ingin melakukan uji statistik pada data tersebut, maka perlu diketahui distribusi peluang dari data tersebut dan mencocokan apakah sesuai dengan asumsi distribusi peluang pada model atau uji statistik tersebut. Uji klasik untuk mengetahui distribusi peluang dari suatu data adalah uji statistika nonpararmetrik *goodness of fit* seperti chi-*square* dan Kolmogorov Smirnov. Namun uji klasik ini kurang akurat untuk mengidentifikasi distribusi peluang jika ukuran dari data sampel kecil (Su & Chou, 2007). Proses dalam mengidentifikasi distribusi peluang dari suatu data masuk dalam wilayah kajian "pengenalan pola".

Jaringan syaraf tiruan telah banyak digunakan untuk masalah pengenalan pola dan telah menunjukan kehandalannya dalam mengenali pola (Samarasinghe, 2007). Su & Chou (2007) menunjukkan bahwa jaringan syaraf tiruan dapat menjadi alat yang baik untuk mengidentifikasi distribusi peluang dari suatu data. Penelitan Su & Chou (2007), menunjukkan bahwa jaringan syaraf tiruan LVQ (*Learning Vector Quantization*) tingkat akurasi yang tinggi dibandingkan

dengan uji statistika nonparametrik untuk mengidentifikasi distribusi peluang dari data. Pada penelitian tersebut digunakan delapan distribusi peluang untuk membangun jaringan syaraf tiruan, yaitu distribusi normal, distribusi eksponensial, distrbusi Weibull, distribusi seragam, distribusi chi-*square*, distribusi t, distribusi F dan distribusi lognormal.

Dewasa ini teori himpunan *fuzzy* banyak digunakan dalam berbagai bidang keilmuan. Hal ini dikarenakan himpunan *fuzzy* dapat menjelaskan ketidakpastian dan ketidakjelasan yang banyak terjadi di berbagai masalah yang dihadapi. Himpunan *fuzzy* juga dapat dipandang sebagai perluasan (*generalized*) dari himpunan biasa yang ditunjukkan dengan nilai keanggotaanya (Klir & Yuan, 1995). Teori himpunan *fuzzy* ternyata dapat digabungkan dengan jaringan syaraf tiruan. Sakuraba *et al* (1991) menggabungkan jaringan syaraf tiruan LVQ dengan teori himpunan *fuzzy* yang kemudian diberi nama *Fuzzy* LVQ (FLVQ). Dalam penelitian Sakuraba *et al* (1991) telah ditunjukkan bahwa FLVQ memiliki tingkat keakuratan yang lebih tinggi daripada LVQ dalam pengenalan pola. Oleh karena itu, dalam penelitian ini mengkaji jaringan syaraf tiruan FLVQ untuk mengidentifikasi distribusi peluang dari suatu data.

1.2 Batasan Masalah

Penelitian ini difokuskan untuk membangun model jaringan syaraf tiruan *Fuzzy Learning Vector Quantization* (FLVQ) dengan algoritma LVQ1 untuk mengidentifikasi distribusi peluang dari data sampel. Adapun distribusi

peluang yang digunakan pada peneitian ini adalah distribusi normal, distribusi eksponensial, distrbusi Weibull, distribusi seragam, distribusi chi-*square*, distribusi t, distribusi F, distribusi lognormal, dan distribusi gamma

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah membangun model *Fuzzy Learning Vector Quantization* dengan algoritma LVQ1 untuk mengidentifikasi distribusi peluang dari data sampel.

1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan metode alternatif untuk mengidentifikasi distribusi peluang dari suatu data sampel dengan tingkat akurasi yang lebih baik

II. TINJAUAN PUSTAKA

2.1 Jaringan Syaraf Tiruan (*Artificial Neural Network*)

Menurut Samarasinghe (2007), *Artificial Neural Network* atau biasa disebut *Neural Network* atau Jaringan Syaraf Tiruan (JST) adalah sekumpulan dari neuron yang terinterkoneksi yang secara bertahap belajar dari lingkunganya (data), sehingga memberikan prediksi yang reliabel untuk situasi baru yang memuat bahkan informasi parsial dan *noisy*. Neuron merupakan satuan komputasi dasar yang melakukan pemrosesan data didalam JST. Neuron ini membentuk jaringan pararel yang besar, dimana fungsi ditentukan dengan struktur JST (yaitu, bagaimana neuron diatur dan dihubungkan satu sama lain), kekuatan koneksi antara neuron dan proses yang dilakukan di neuron.

2.1.1 Model Neuron

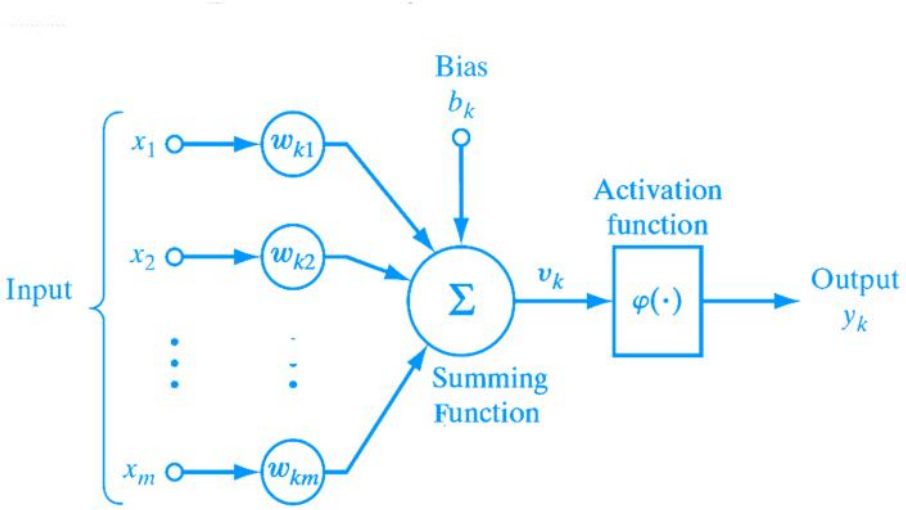
Neuron merupakan satuan pemrosesan informasi yang menjadi dasar dari JST. Gambar 2.1 menunjukkan model dari suatu neuron. Berikut adalah 3 elemen dasar dari model neuron:

1. Suatu himpunan dari sinapsis atau *conneting links*, yang masing-masing ditandai dengan suatu bobotnya sendiri. Secara khusus, suatu sinyal x_j pada *input* dari sinapsis j dihubungkan ke neuron k dikalikan dengan bobot sinapsis

w_k . Bobot sinapsis dari suatu neuron mungkin terletak pada rentang yang meliputi nilai-nilai negatif dan positif.

2. Fungsi penjumlah (*Summing function*) digunakan untuk menjumlahkan sinyal *input*, diboboti oleh masing-masing bobot sinapsis dari neuron. Operasi yang dijelaskan disini merupakan *linear combiner*.

3. Suatu fungsi aktifasi (*activation function*) untuk membatasi ampitudo *output* dari suatu neuron. Fungsi aktifasi disebut juga sebagai *squasing function*.



Gambar 2.1 Model neuron

Model Neuron pada Gambar 2.1 juga terdapat bias, yang dilambangkan dengan b_k. Bias tersebut mempunyai efek untuk menaikan atau menurunkan *input* dari fungsi aktifasi, tergantung dari bias tersebut bernilai positif ataupun negatif. Secara Matematis, model neuron pada Gambar 2.1 dapat ditulis dengan sepasang persamaan:

$$v_k = \sum_{j-1}^{m} w_k \; x_j \qquad (2.1)$$

dan

$$y_k = \varphi(v_k + b_k) \qquad (2.2)$$

dimana $x_1, x_2, \dots, x_m$ merupakan *input*; $w_{k1}, w_{k2}, \dots, w_k$ merupakan bobot dai neuron; v_k merupakan fungsi penjumlah; b_k merupakan bias; $\varphi(.)$ adalah fungsi aktifasi; $u_k = v_k + b_k$ dan y_k adalah *output*.

(Haykin, 2009)

2.1.2 Arsitektur Jaringan Syaraf Tiruan

Arsitektur Jaringan Syaraf Tiruan (JST) secara garis besar dapat dibagi menjadi JST *feedforward neural network*, *recurrent neural network*, dan campurannya. Topologi JST yang sering digunakan anatara lain *fully connected layered feedforward networks, recurrent networks, lattice networks, layered feedforward networks with lateral connection*, dan *celular network*, seperti yang diperlihatkan pada Gambar 2.2. Penjelasan tentang topologi JST pada Gambar 2.2 adalah sebagai berikut :

1. *Feedforward network*

Dalam *feedforward network*, koneksi-koneksi antara neuron berjalan dalam satu arah. Suatu feedforward network biasanya disusun dalam bentuk *layer*. Seperti *layer*ed feedforward network, tidak ada koneksi anatara neuron dalam *layer* yang sama, dan tidak ada *feedback* (umpan balik) antara *layer*. Dalam *fully connected layered feedforward network*, setiap node pada *layer* dihubngkan ke setiap node pada *forward layer* yang berdekatan.

2. *Recurrent Network*

Dalam *recurrent network*, terdapat paling tidak satu koneksi *feedback*.

3. *Lattice Network*

Suatu *lattice network* terdiri dari satu, dua atau lebih *dimensional array* dari neuron. Setiap array memiliki himpunan dari node *input* yang bersesuaian.

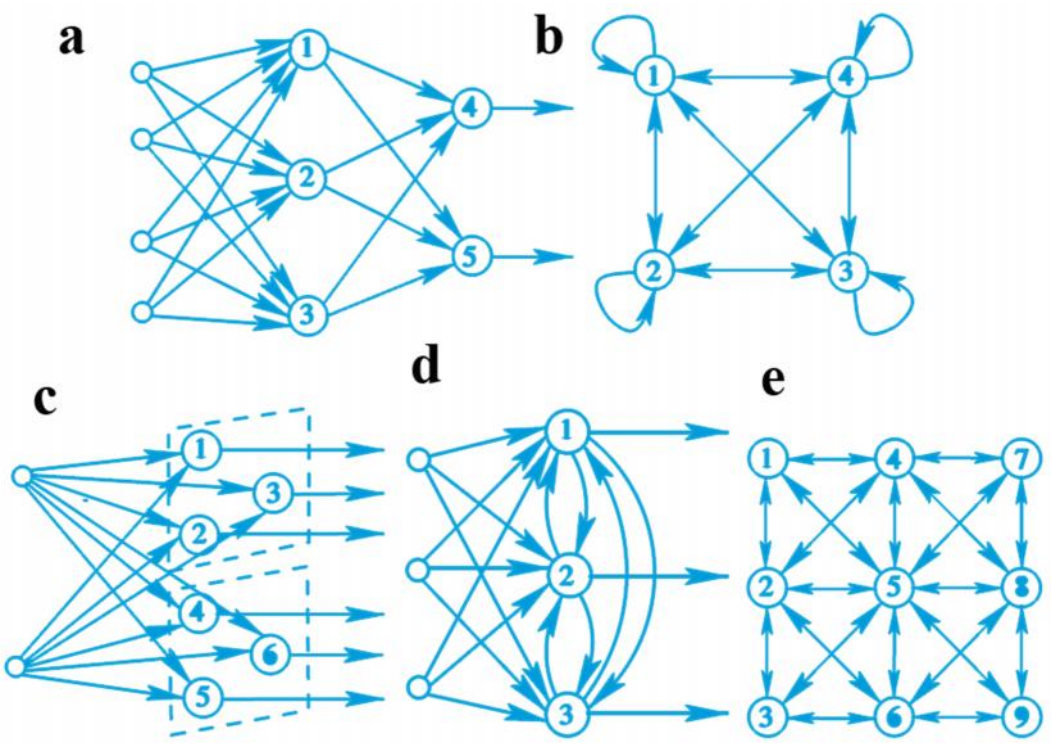
4. *Layered Feedforward Network with Lateral Connection*

Layered feedforward network with lateral connection memiliki koneksi lateral antara unit-unit pada *layer* yang sama dari arsetektur *layered feedforward network*.

5. *Cellular network*

Suatu *cellular network* terdiri dari neuron dengan jarak teratur, disebut *cell,* yang hanya berkonikasi dengan neuron terdekatnya. *Cell* yang bersebelahan terhubung dengan interkoneksi yang sama.

(Du & Swamy, 2014)



Gambar 2.2 a. *Feedforward network* b. *Recurrent network* c. *Two-dimensional lattice network* d. *Layered feedforward network with lateral connection* e. *cellular network*

2.1.3 Operasi dari Jaringan Syaraf Tiruan

Operasi dari jaringan syaraf tiruan dibagi menjadi dua tahap, yaitu *learning* (pembelajaran) dan *generalization* (generalisasi). Pembelajaran jaringan syaraf tiruan biasanya dilakukan dengan contoh-contoh, dan parameter-parameter JST yang disesuaikan dengan menggunakan *learning algorithm* (algoritma pembelajaran). Setelah JST dilatih untuk melakukan kinerja yang diinginkan, proses pembelajaran dihentikan dan JST bisa langsung digunakan.

2.1.4 Sifat-Sifat Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan memiliki keunggulan sebagai berikut:

1. *Adaptive Learning* (Pembelajaran Adatif)

Jaringan Syaraf Tiruan dapat menyesuaikan diri dengan mengubah parameter-parameter pada JST.

2. Generalisasi

Jaringan syaraf tiruan yang telah terlatih memiliki kemamupuan generalisasi yang baik

3. Self-organizing

Beberapa jaringan syaraf tiruan memiliki sifat mengelompokan sendiri.

4. *Robustness* dan *fault tolerance*

Suatu jaringan syaraf tiruan dapat dengan mudah menangani infromasi yang tidak akurat, kabur, *noisy* dan probabilistik.

(Du & Swamy, 2014)

2.2. Proses Pembelajaran (*Learning* Process)

Pembelajaran adalah kemampuan dasar dari jaringan syaraf tiruan. Aturan pembelajaran merupakan algoritma untuk menemukan bobot w dan/atau parameter lainya yang sesuai dari suatu JST. Pembelajaran dari suatu JST dapat dipandang sebagai masalah optimisasi nonlinear untuk menemukan himpunan dari parameter-parameter JST yang meminimumkan *cost function* untuk suatu contoh. Pendugaan parameter untuk jenis ini juga disebut dengan algoritma pembelajaran atau latihan.

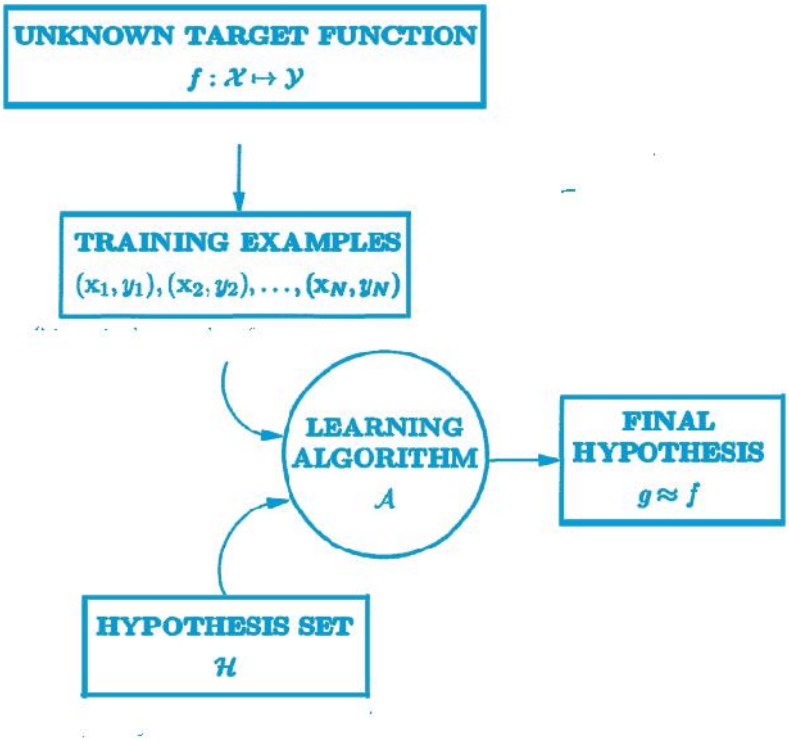
Jaringan syaraf tiruan biasanya dilatih dengan *epoch*. *Epoch* adalah *run* lengkap ketika semua contoh pelatihan telah diberikan ke JST dan diproses menggunakan algoritma pembelajaran hanya sekali. Setelah pembelajaran, suatu JST menampilkan suatu hubungan yang kompleks dan memiliki kemampuan untuk generalisasi.

2.2.1 Komponen dari Pembelajaran (*Learning*)

Misalkan x merupakan suatu *input*, $f: X \rightarrow Y$ merupakan fungsi target, dimana X adalah himpunan semua *input* x yang mungkin dan Y merupakan himpunan semua *output* yang mungkin. Terdapat himpunan data D dari contoh *input-output* $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ dimana $y_n = f(x_n)$ untuk $n = 1,2, \dots, N$. Terakhir, terdapat algoritma pembelajaran yang menggunakan data D untuk memilih suatu rumus $g: X \rightarrow Y$ yang mengaproksimasi f. Algoritma memilih g dari kandidat

himpunan formula yang dipertimbangkan, yang disebut himpunan hipotesis $\mathcal{H}$. Gambar 2.3 mengilustrasikan komponen dari pembelajaran

(Mostafa *et al*, 2012)



Gambar 2.3 Proses *learning* (pembelajaran)

2.2.2 Jenis-Jenis Paradigma Pembelajaran

Ada beberapa jenis dari paradigma pembelajaran yang akan dibahas, antara lain:

1. *Supervised Learning* (Pembelajaran Terawasi)

Ketika data pelatihan (*training data*), berisi contoh-contoh ekspilisit dari hasil *output* yang benar untuk suatu *input* yang diberikan, maka hal ini merupakan supervised *learning*, Pembelajaran diawasi sama seperti beberapa "*supervisor*" mengawasi setiap *input* dan menentukan *output* yang benar. *Supervised learning* menyesuaikan parameter JST dengan perbandingan langsung antara hasil *output* dari JST dan *output* yang diinginkan (nilai target). Nilai target biasanya didapat

dari *supervisor*. Ukuran *error*, dimana menunjukan perbedaan antara *output* JST dan *output* dari *training* sampel, digunakan sebagai petunjuk untuk proses pembelajaran. Ukuran dari *error* biasanya didifinisikan dengan *Mean Square of Error* (MSE)

$$E = \frac{1}{n}\sum_{p-1}^{n}\left|y_p - \hat{y}_p\right|^2 \qquad (2.3)$$

Dimana n adalah banyaknya pasangan *patern* dalam sampel, y_p adalah nilai target dari pasangan pattern ke-p dan $\hat{y}_p$ adalah *output* JST yang sesuai dengan pasangan pattern p. E dihitung lagi pada akhir setiap *epoch.* Proses pembelajaran dihentikan ketika MSE cukup kecil atau kriteria kesalahan sudah seperti yang diinginkan

2. *Unsupervised Learning* (Pembelajaran tak Terawasi)

Dalam *unsupervised learning*, data pelatihan (*training data*) tidak memuat informasi *output* sama sekali. Pada pembelajaran ini hanya diberikan contoh *input* $x_1, x_2, \dots, x_N$. *Unsupervised Learning* bisa dipandang sebagai tugas secara spontan untuk menemukan pola dan struktur data *input*. Sebagai contoh, misalnya kita bertugas untuk mengelompokan buku-buku, dan kita hanya menggunakan sifat-sifat umum dari bermacam-macam buku, kita dapat mengidentifikasi buku-buku tersebut yang memiliki kesamaan sifat dan menruhnya pada satu kategori yang sama tanpa memberi nama pada kategori tersebut

Suatu Kriteria dibutuhkan untuk menghentikan proses pembelajaran. Tanpa Kriteria *stopping*, proses pembelajaran akan terus berlanjut bahkan ketika pola, yang tidak termasuk dalam data pelatihan, ditampilkan oleh JST. Hebbian *learning*, *Competitve learning* dan SOM (*Self-Organized Maps*) merupakan jenis-jenis *unsupervised learning* yang cukup dikenal.

3. *Reinforcement Learning* (Pembelajaran dengan Dukungan)

Ketika data pelatihan (*training data*) tidak memuat secara eksplisit *output* yang benar dari suatu *input*, maka pembelajaran ini tidak termasuk lagi dalam *supervised learning*. Misalkan seorang balita belajar untuk tidak menyentuh secangkir teh panas. Pengalaman balita tersebut biasanya akan terdiri satu himpunan peristiwa ketika balita dihadapkan secangkir teh dan dihadapkan dengan keputusan menyentuh atau tidak menyentuhnya. Setiap kali ia menyentuhnya, hasilnya adalah rasa sakit tingkat tinggi dan ketika ia tidak menyentuhnya, rasa yang yang lebih rendah hasilnya (yaitu keingintahuan yang tidak terpuaskan). Akhirnya, balita belajar bahwa dia lebih baik tidak menyentuh cangkir panas.

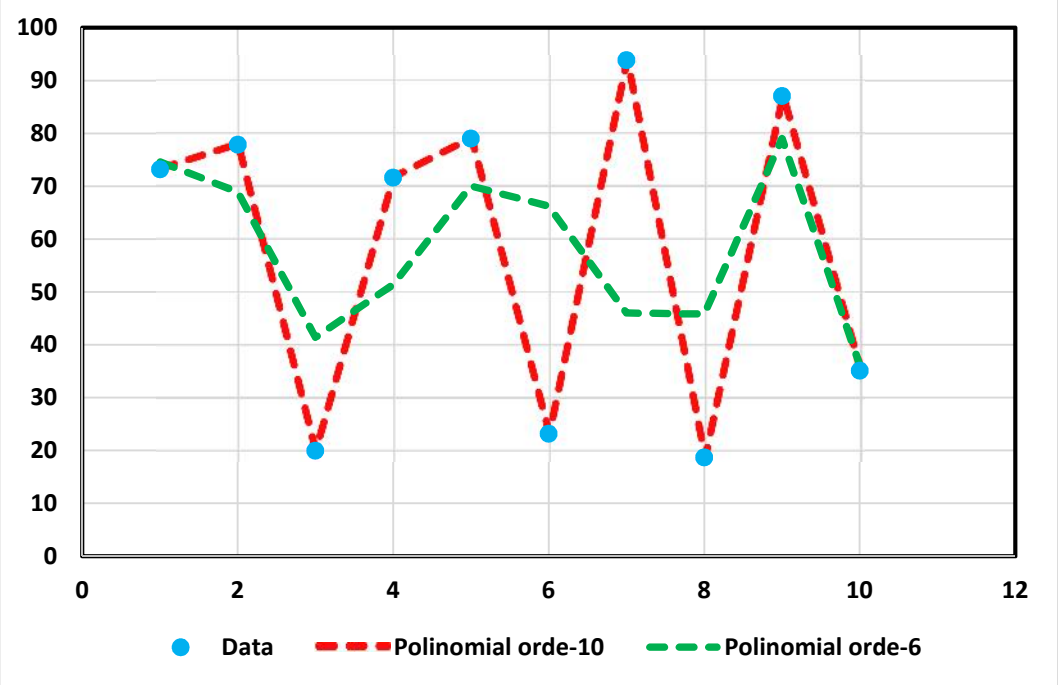
Dalam *reinforcement learning*, dimana contoh pelatihan tidak mengandung target *output*, tetapi berisi tentang beberapa kemungkinan *output* bersama dengan ukuran seberapa bagus *output* tersebut. Berbeda dengan supervised *learning* dimana contoh pelatihan berbetuk (*input*, *correct*, *output*), contoh-contoh dalam reinforcement *learning* berbentuk (*input*, *output*, tingkatan untuk *output*)
(Mostafa *et al*, 2012)

2.3 *Generalization* (Generalisasi)

Secara matematis, proses pembelajaran adalah proses *curve-fitting nonlinear*, sedangkan generalisasi adalah interpolasi dan ekstrapolasi data *input*. Tujuan dari melatih jaringan syaraf tiruan bukanlah untuk belajar dengan tepat bedasarkan data pelatihan itu sendiri, tetapi untuk membangun model statistika. Ketika JST dilatih secara berlebihan dengan terlalu banyak contoh, parameter atau *epoch*, JST

mungkin menghasilkan hasil yang baik untuk data pelatihan, tetapi memiliki kemampuan yang buruk untuk generalisasi. Hal ini merupakan fenomena *overfitting* dan diilustrasikan pada Gambar 2.4. Pada Gambar 2.4, *overfitting* terjadi pada polynomial orde-10 sedangkan pada polinomial orde-6 *overfitting* tidak terjadi. Dalam statistik, *overfitting* terjadi ketika model terlalu memiliki banyak parameter dan *fits noise* data. Umumnya, kemampuan generalisasi dari JST oleh ukuran dari *training pattern set*, kompleksitas masalah dan arsitektur JST. Untuk generalisasi yang baik, ukuran dari training set, N, harus paling tidak lebih besar daripada kapasitas JST, yaitu $N > \frac{N_w}{N_y}$, dimana N_w adalah jumlah dari bobot atau *free* parameter, dan N_y adalah jumlah dari komponen *output*.

(Du & Swamy, 2014)



Gambar 2.4 contoh *overfitting*

Untuk mengatasi *overfitting*, ada beberapa cara yang bisa dilakukan. Menurut Samarasinghe (2007), overfitting bias dihindari jika fleksibelitas JST dapat dikurangai. Fleksibilitas JST berasal dari *hidden neuron*, dan semakin banyak *hidden neuron* maka bobot (*weight*) akan semakin banyak. Semakin banyak bobot (*weight*), semakin besar pula tingkat Fleksibilats JST. Berikut adalah cara-cara untuk mengatasi overfitting

1. *Exhaustive Search*

Walaupun membutuhkan banyak waktu, cara termudah adalah mencari jumlah optimum dari neuron dengan *trial and error*. Dengan menggunakan *validation set*, dimana jumlah neuron yang memberikan *error* terkecil pada *validation set* merupakan jumlah neuron yang optimum.

2. Early Stopping

Bobot (*weight*) disebut parameter bebas. Hal ini dikarenakan bobot dapat berubah selama pelatihan (*training*). Jika bobot diperbolehkan cukup bekembang dalam pelatihan dan kemudian pelatihan berhenti pada titik ini, maka hal ini dapat mengatasi *overtfitting*,

3. Regularization

Pendekatan lain untuk mengatasi *overfitting* adalah *regularization*. Dalam *regularization*, suatu *regularized performance index* diminimumkan daaripada meminimumkan MSE aslinya. *Regularized performance index* didapatkan dengan menambahkan jumlah kuadart bobot pada MSE asli. Dalam JST, *regularization* sering juga disebut *weight decay*. Persamaan *regularized performance index* dapat dilihat pada persamaan berikut

$$W = M \quad + \delta \sum_{j=1}^{m} w_j^2 \tag{2.4}$$

dengan δ adalah parameter *regularization* dan w adalah bobot (*weight*)

2.4 *Learning Vector Quantization* (LVQ)

Setelah mengetahui dasar-dasar tentang jaringan syaraf tiruan beserta proses pembelajaranya, pada bagian selanjutnya akan dijelaskan mengenai salah satu model dari jarigan syaraf tiruan yang diperkenalkan oleh Kohonen (1990) yaitu *Learning Vector Quantization* (LVQ). Pertama-tama akan dijelaskan tentang *Vector Quantization* (VQ) dan pembelajaran kompetitif (*Competitive Learning*) yang merupakan dasar dari LVQ. Kemudian akan dijelaskan LVQ beserta jenis-jenis algoritma pada LVQ.

2.4.1 *Vector Quantization*

Vector Quantization merupakan metode klasik yang menghasilkan suatu aproksimasi ke *probability density function* $p(x)$ kontinu dari vektor variabel x □ $\mathbb{R}^n$ menggunakan sejumlah vektor *codebook* terbatas w_i □ $\mathbb{R}^n, i = 1, 2, \dots, k$ Himpunan terbatas prototipe disebut sebagai *codebook*. Setelah *codebook* ditentukan, aproksimasi dari x berarti menemuan vektor *codebook* w_c yang paling dekat dengan x sedemikian sehingga

$$\|x - w_c\| = \min_i \|x - w_i\|$$

Atau

$$w_c = \arg\min_i \|x - w_i\| \tag{2.5}$$

Skema pendekatan iteratif untuk menemukan vector *codebook* adalah

$$w_c(t+1) = w_v + \alpha(t)(x - w_v) \tag{2.6}$$

(Kohonen, 1995)

2.4.2 *Competitive Learning* (Pembelajaran Kompetitif)

Pembelajaran kompetitif dapat diimplementasikan menggunakan J-K *neural network. Layer output* disebut dengan *layer* kompetisi dimana neuron-neuron terhubung penuh ke *input*. Dalam *layer* kompetisi, koneksi lateral digunakan untuk menunjukkan inhibisi lateral. Arsitektur dari pembelajaran kompetitif ditunjukkan pada Gambar 2.5. Untuk *input* x, JST memilih satu dari K bobot w_i dengan mengatur $y_i = 1$ dan $y_j = 0, j \neq i$

Prinsip dasar pada pembelajaran kompetitif adalah masalah statistika matematika yang disebut analisis kluster. Pembelajaran kompetitif biasanya didasarkan pada minimisasi dari suatu fungsional sedemikian sehingga

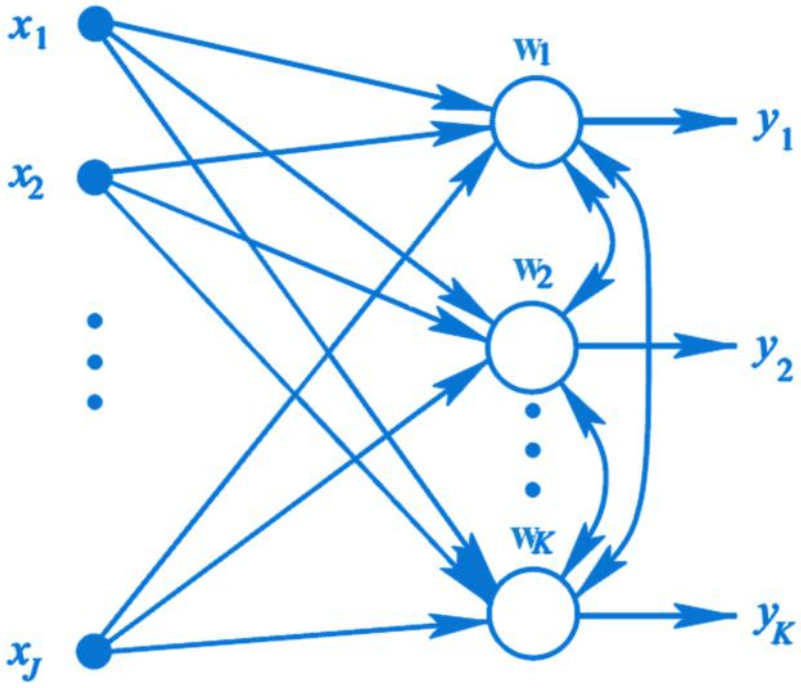
$$E = \frac{1}{N}\sum_{p=1}^{N}\sum_{k=1}^{K} u_k \left\|x_p - w_k\right\|^2 \tag{2.7}$$

Dimana *N* adalah ukuran dari himpunan *pattern* dan u_k adalah bobot koneksi yang ditugaskan ke w_k berkaitan dengan x_p, melambangkan keanggotaan dari *pattern* p dalam kluster k. Untuk pembelajaraan kompetitif sederhana dengan asumsi bahwa bobot-bobot diperoleh berdasarkan kondisi codebook vector terdekat

$$u_k = \begin{cases} 1 & k = \arg m \left\|x_p - w_k\right\| \\ 0 & s \end{cases} \tag{2.8}$$

Salah satu aturan dalam pembelajaran kompetitif adalah WTA (*winner-takes-all*). Dalam WTA, *agent* (yaitu sesuatu yang melakukan) dalam grup saling berkompetisi dan hanya satu dengan *input* tertinggi akan tetap aktif ketika yang lainya tidak aktif. Kemudian aturan lain yang penting dalam kompetitif learning adalah *winner-kills-loser*. Dalam aturan ini, *winner* tidak hanya *takes all*, tetapi juga membunuh *losers*. Dengan kata lain, *winner* belajar sampel *training*, dan *losers* dihapus dari JST.

(Kohonen, 1995)



Gambar 2.5 Arsitektur J-K *neural network*

2.4.3 Algoritma *Learning* Vector Quantization (LVQ)

LVQ secara luas digunakan untuk klasifikasi. LVQ menggunakan arsitektur jaringan pada Gambar 2.5 dengan setiap neuron *output* dikhususkan dengan suatu keanggotaan kelas. Kohonen (1990) mengusulkan keluarga dari algoritma LVQ

yaitu LVQ1, LVQ2 dan LVQ3. Berikut adalah penjelasan dari masing-masing keluarga algoritma LVQ:

1. LVQ1

Misalkan beberapa vector *codebook* ditugaskan ke setiap kelas dari nilai x, dan x ditentukan berada pada kelas yang sama dengan w_i yang terdekat. Misalkan

$$w_c = \arg\min_i \|x - w_i\|$$

Didefinisikan sebagai indeks dari w_i yang terdekat dengan x. Catat bahwa c merupakan indeks "pemenang" bergantung pada x dan semua w_i. Misalkan $x(t)$ adalah *input* dan $w_i(t)$ menyatakan nilai terurut dari w_i dalam domain waktu diskrit, $t = 0, 1, 2, \dots, n$. Mulai dengan nilai awal yang sesuai, persamaan berikut mendefinisikan tentang proses dasar *Learning Vector Quantization* (LVQ); algoritma ini disebut dengan LVQ1

$$\begin{aligned} w_c(t+1) &= w_c(t) + \alpha(t)[x(t) - w_c(t)], && x, w_c \in C \\ w_c(t+1) &= w_c(t) - \alpha(t)[x(t) - w_c(t)], && x \in C, w_c \in C \\ w_i(t+1) &= w_i(t), && i \neq c \end{aligned} \quad (2.9)$$

Dengan $\alpha(t)$ adalah *learning rate* dan C merupakan kelas yang bersesuaian dengan x pada LVQ. Dimana $\alpha(t) \in (0,1)$ dan $\alpha(t)$ biasanya dibuat monoton turun dengan waktu (t)

Algoritma dasar LVQ1 akan dimodifikasi sedimikian sehingga *learning rate* α ditugaskan pada setiap w_i. Algoritma kemudian disebut dengan OLVQ1 (*Optimized-Learning Rate* LVQ1). Berikut adalah algoritmaOLVQ1

$$\begin{aligned} w_c(t+1) &= w_c(t) + \alpha_c(t)[x(t) - w_c(t)], && x, w_c \in A \\ w_c(t+1) &= w_c(t) - \alpha_c(t)[x(t) - w_c(t)], && x \in A, w_c \in A \\ w_i(t+1) &= w_i(t), && i \neq c \end{aligned} \quad (2.10)$$

dengan

$$\alpha_c(t) = \frac{\alpha_c(t-1)}{1 + s(t)\alpha_c(t-1)} \tag{2.11}$$

Untuk $s(t) = 1$ jika klasifikasi benar dan $s(t) = -1$. Karena $\alpha_i(t)$ juga dapat naik, sangat penting bahwa $\alpha_i(t)$ tidak sampai nilai diatas 1. .Hal ini bisa dibatasi dengan algoritma itu sendiri. Tingkat konvergensi OLVQ1 mungkin akan lebih cepat daripada LVQ1

2. LVQ2

Keputusan klasifikasi dalam algoritma ini identik dengan LVQ1. Dalam pembelajaran, dua vektor *codebook* w_i dan w_j yang terdekat dengan x sekarang diperbaharui secara simultan. Salah satu dari vektor *codebook* pasti ada pada kelas yang benar dan lainya ada pada kelas yang salah. Selain itu, x harus jatuh dalam wilayah yang disebut dengan "*window*" yang didefinisikan sebagai pertengahan dari w_i dan w_j. Gambar 2.6 mengilustrasikan wilayah *window*. Misalkan d_i dan d_j adalah jarak Euclidean x dari w_i dan w_j. Maka x didefinisikan masuk dalam "*window*" dengan lebar relatif m jika

$$\min\left(\frac{d_i}{d_j}, \frac{d_j}{d_i}\right) > s, di \qquad s = \frac{1-m}{1+m} \tag{2.12}$$

Lebar relatif "*window*" yang direkomendasikan adalah 0.2 samapi 0.3. Versi dari Algoritma LVQ2 pada persamaan disebut juga LVQ2.1 merupakan perbaikan dari algoritma LVQ2 asli yang dalam pengertian algoritma ini mengizinkan w_i dan w_j menjadi vektor *codebook* terdekat dengan x, sedangkan dalam LVQ2 asli, w_i harus yang paling dekat. Berikut adalah algoritma LVQ2.1

$$w_i(t+1) = w_i(t) - \alpha(t)[x(t) - w_i(t)], \tag{2.13}$$

$$w_j(t+1) = w_j(t) + \alpha(t)[x(t) - w_i(t)],$$

dimana w_i dan w_j adalah dua vektor *codebook* terdekat dengan x, dengan x dan w_j ada dalam kelas yang sama, ketika x dan w_i ada pada kelas berbeda. Selain itu x harus berada di dalam "*window*".

3. LVQ3

Algoritma LVQ3 adalah perbaikan dari LVQ2.1, yaitu

$$w_i(t+1) = w_i(t) - \alpha(t)[x(t) - w_i(t)],$$

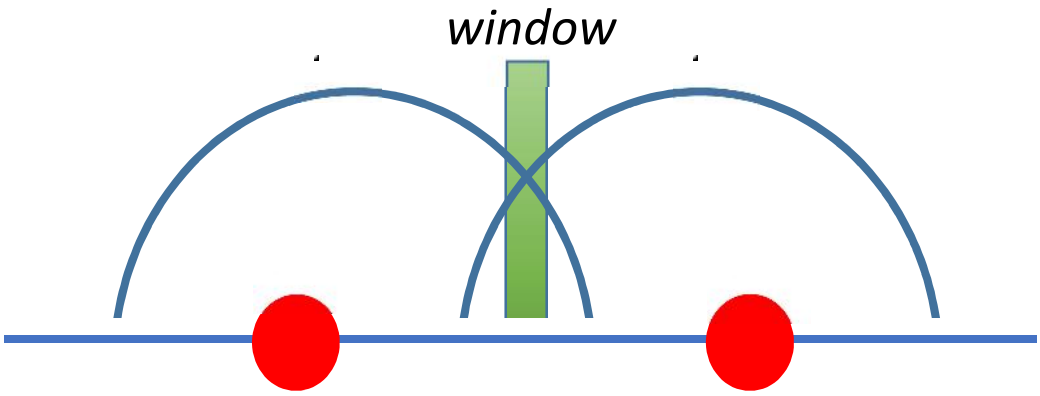
$$w_j(t+1) = w_j(t) + \alpha(t)[x(t) - w_i(t)],$$

dimana w_i dan w_j adalah dua vektor *codebook* terdekat dengan x, dengan x dan w_j ada dalam kelas yang sama, ketika x dan w_i ada pada kelas berbeda. Selain itu x harus berada di dalam "*window*";

$$w_k(t+1) = w_k(t) + \eta\ (t)[x(t) - w_k(t)] \qquad (2.14)$$

Untuk $k \in \{i, j\}$, jika x, w_i dan w_j ada pada kelas yang sama. Nilai optimum dari η bergantung pada ukuran dari *window*, menjadi semakin kecil untuk *window* yang sempit. Dalam rangkaian percobaan, nilai yang dapat diaplikasikan dari η antara 0.1 dan 0.5.

(Kohonen, 1995)



Gambar 2.6 Wilayah "*window*" pada algortima LVQ2

2.5 Himpunan *Fuzzy*

Konsep dari himpunan *fuzzy* bisa dipandang sebagai generalisasi dari himpunan biasa atau himpunan *crisp*. Teori himpunan *fuzzy* dan dasar-dasar dari logika *fuzzy* dikembangkan oleh Zadeh (1965) berdasarkan pada teori himpunan klasik dan logika klasik.

2.5.1 Definisi-Definisi Dasar

Menurut Klir & Yuan (1995), himpunan *fuzzy* dapat dipandang sebagai perluasan dari himpunan biasa (*crisp*). Oleh karena itu, perlu diketahui hal-hal dasar yang membedakan himpunan *fuzzy* dan himpunan biasa serta bagaimana hubungan himpunan *fuzzy* dengan himpunan biasa. Menurut Rutkowska (2002), ada beberapa hal-hal dasar yang perlu diketahui tentang himpunan *fuzzy*.

Definisi 2.1 Himpunan *Fuzzy*

Misalkan X merupakan suatu ruang dari titik-titik (objek-objek), dengan elemen dari X dilambangkan dengan x. Suatu himpunan *fuzzy* A dalam X ditandai dengan fungsi keanggotaan $\mu_A(x)$ yang berhubungan dengan setiap bilangan *real* x dalam interval $[0,1]$ menunjukkan dereajat keanggotaan dari x dalam A

$$A = \{(x, u_A(x)); x \square X\}$$

dimana

$$\mu_A(x): X \rightarrow [0,1]$$

Semakin lebih dekat nilai dari $\mu_A(x)$ ke *unity* (satu), derajat keanggotaan dari x di A senakin tinggi derajat keanggotaanya. Jika $\mu_A(x) = 1$, maka x sepenuhnya

berada pada A. Jika $\mu_A(x) = 0$, maka x tidak berada pada A. Ruang X disebut dengan *universe of discourse.*

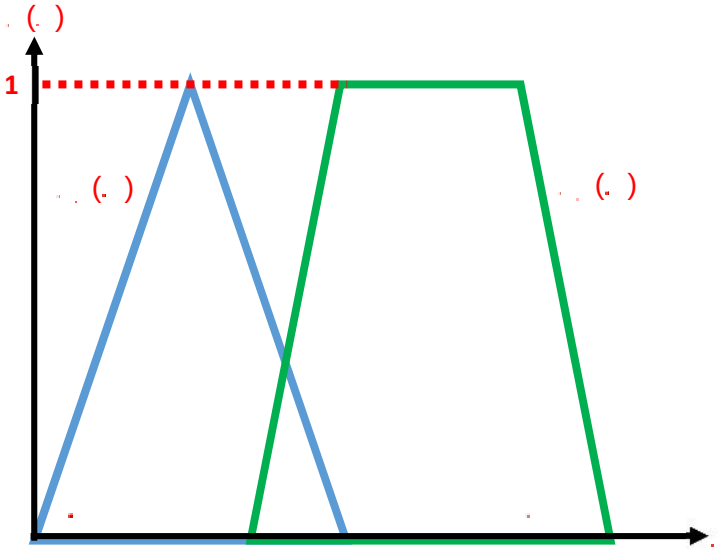
Suatu himpunan *fuzzy* A dinyatakan dengan himpunan pasangan terurut. Ketika *universe of discourse* adalah himpunan terhigga, yaitu $X = \{x_1, x_2, \dots, x_n\}$, suatu himpunan *fuzzy* dapat dinyatakan sebagai

$$\boldsymbol{A} = \sum_{i=1}^{n} \mu_A(x_i)/x_i = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \cdots + \mu_A(x_n)/x_n \qquad (2.15)$$

Ketika *universe of discourse* X merupakan himpunan tak terbatas, suatu himpunan *fuzzy* A dapat dinyatakan sebagai

$$\boldsymbol{A} = \int_X \mu_A(x_i)/x_i = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \cdots + \mu_A(x_n)/x_n \qquad (2.16)$$

Simbol $\sum$, $\int$, + dalam formula (2.15) dan (2.16) mengacu pada gabungan himpunan daripada penjumlahan aritmatika. Demikian juga, tidak ada pembagian aritmatika pada formula-formula ini. Notasi simbol ini (/) digunakan untuk menghubungkan suatu elemen dengan nilai keanggotaanya. Gambar 2.7 adalah contoh dari himpunan *fuzzy*.



Gambar 2.7 Himpunan *fuzzy* A dan B

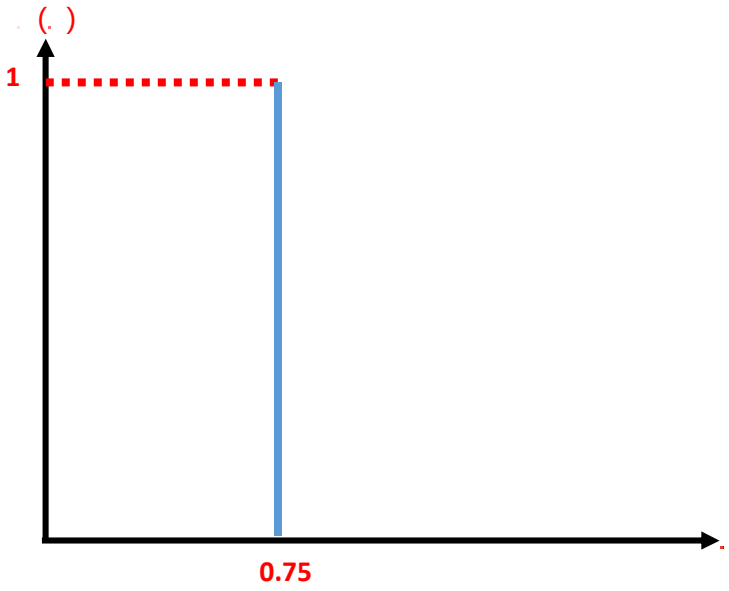
Definisi 2.2 Pendukung (*support*) dari suatu himpunan *fuzzy* A, dilambangkan dengan s A, merupakan himpunan dari titik-titik di X dimana fungsi keanggotaan $u_A(x)$ adalah positif

$$s \quad A = \{x \in X;\ \mu_A(x) > 0\} \tag{2.17}$$

Definisi 2.3 Suatu *fuzzy singleton* adalah suatu himpunan A dimana pendukung merupakan suatu poin tunggal dalam *universe of discourse X*

Gambar 2.8 mengilustrasikan tentang *fuzzy singleton*. Perlu dicatat bahwa *fuzzy singleton* A dimana *support*-nya adalah suatu titik x dapat ditulis, berdasarkan formula (2.15) , seperti

$$\boldsymbol{A} = \mu_A(x)/x \tag{2.18}$$



Gambar 2.8 *Fuzzy singleton*

Definisi 2.4 Inti dari himpunan *fuzzy* A yang didefinisikan dalam *universe of discourse* X, dilambangkan dengan c (A), disebut juga kernel atau nucleus, adalah himpunan dari titik-titik di X dimana fungsi keanggotaan $\mu_A(x)$ sama dengan 1, yaitu

$$c_{\square}\quad(A) = \{x \square X;\ \mu_A(x) = 1\} \tag{2.19}$$

Definisi 2.5 Tinggi (*height*) dari suatu himpunan *fuzzy* A yang didefinisikan dalam *universe of discourse* X, dilambangkan dengan $hg\ (A)$, adalah nilai maksimum dari fungsi keanggotaan $\mu_A(x)$, yaitu

$$hg\ (A) = \sup_{x \square X} \mu_A(x) \tag{2.20}$$

Definisi 2.6 Suatu himpunan *fuzzy* A disebut himpunan *fuzzy* normal jika dan hanya jika nilai maksimum dari fungsi keanggotaanya sama dengan 1, yang berarti $hg\ (A) = 1$

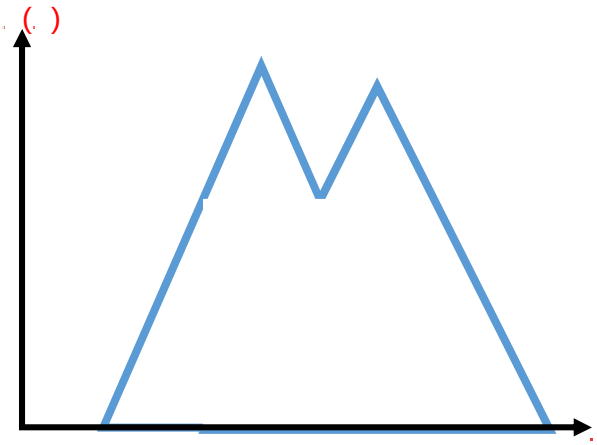
Definisi 2.7 Suatu himpunan *fuzzy* A didefinisikan dalam *universe of discourse* X, dimana diasumsikan menjadi suatu ruang *real* Euclidean berdimensi-N

, adalah konveks jika dan hanya jika

$$\mu_A(\lambda x_1 + (1 - \lambda)x_2) \geq \min[\mu_A(x_1), \mu_A(x_2)] \tag{2.21}$$

Untuk setiap x_1 dan x_2 dalam X dan setiap λ dalam [0,1]

Gambar 2.7 merupakan contoh himpunan *fuzzy* konveks dan Gambar 2.9 merupakan himpunan *fuzzy* tidak konveks



Gambar 2.9 Himpunan *fuzzy* tidak konveks

Definisi 2.8 Suatu himpunan *fuzzy* A yang didefinisikan dalam *universe of discourse* X merupakan himpunan kosong, dilambangkan dengan $A = \emptyset$, jika dan hanya jika fungsi keanggotaanya $\mu_A(x) = 0$ untuk semua $x \in X$

Definisi 2.9 Dua himpunan *fuzzy* A dan B sama, ditulis $A = B$, jika dan hanya jika fungsi keanggotaan dua himpunan *fuzzy* tersebut sama, yaitu $\mu_A(x) = \mu_B(x)$ untuk semua x dalam *universe of discourse X*

Definisi 2.10 Elemen-elemen dari himpunan biasa (himpunan *crisp*) berada pada himpunan *fuzzy* A dalam X paling tidak untuk derajat dari α yang disebut α-*level set* atau α-*cut* dan didefinisikan dengan

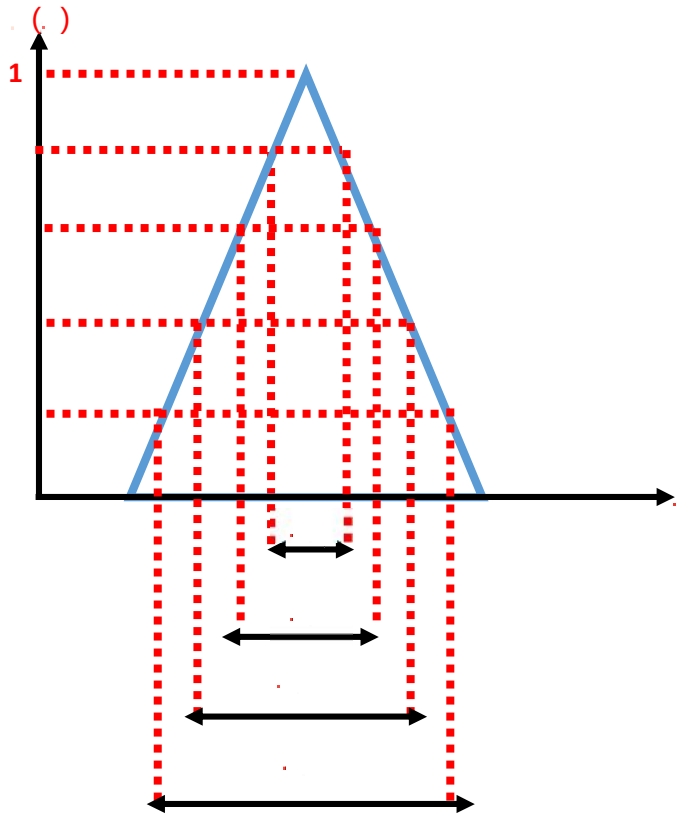
$$A_\alpha = \{x \in X : \mu_A(x) \geq \alpha\} \quad \forall \alpha \in [0,1] \tag{2.22}$$

Jika kondisi $\mu_A(x) \geq \alpha$ pada (2.22) diganti dengan $\mu_A(x) > \alpha$, himpunan A_α akan disebut dengan s*trong* $\alpha - c$. Gambar 2.9 mengilustrasikan tentang α-*cut* pada himpunan *fuzzy* dimana $X = \mathbb{R}$, dengan α-*cut* yang berbeda-beda $\alpha_1, \alpha_2, \alpha_3, \alpha_4$. Dapat dilihat bahwa $\alpha_1 > \alpha_2 > \alpha_3 > \alpha_4 \rightarrow A_{\alpha_1} \subseteq A_{\alpha_2} \subseteq A_{\alpha_3} \subseteq A_{\alpha_4}$

2.5.2 Operasi pada Himpunan *Fuzzy*

Menurut Bede (2013) operasi dasar pada teori himpunan *fuzzy* adalah komplemen, gabungan, irisan dan inklusi. Operasi-operasi ini dilakukan pada fungsi keanggotaan. Untuk itu dibawah ini akan dibahas tentang definisi operasi dasar himpunan *fuzzy*.

Misalkan $\mathcal{F}(X)$ melambangkan kumpulan dari himpunan *fuzzy* pada suatu *universe of discourse* X.



Gambar 2.10 α-*cut* pada himpunan *fuzzy*

Definisi 2.11 Misalkan $A \in \mathcal{F}(X)$, komplemen dari himpunan *fuzzy* A, dilambangkan dengan $\bar{A}$, didefinisikan dengan

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{2.23}$$

Untuk semua $x \in X$

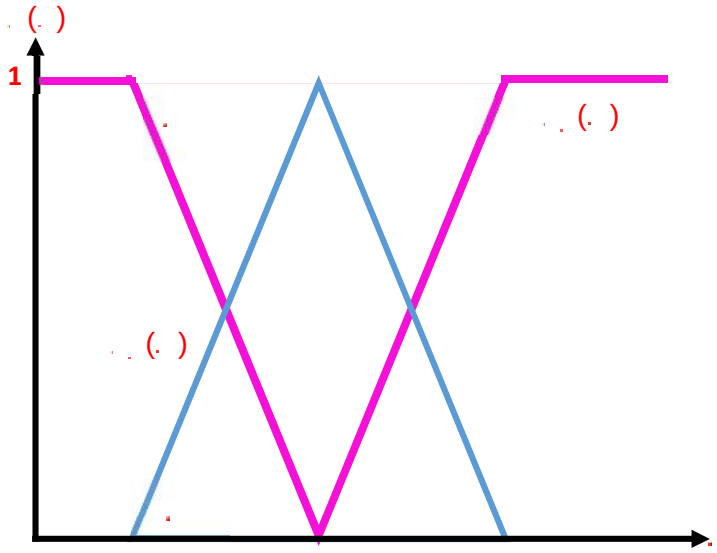
Gambar 2.10 mengilusatrasikan komplemen dari himpunan *fuzzy*, dengan mengasumsikan bahwa A adalah hiimpunan *fuzzy* normal.

Definisi 2.12 Gabungan dari dua himpunan *fuzzy* A dan B dengan fungsi keanggotaan $\mu_A(x)$ dan $\mu_B(x)$ merupakan himpunan *fuzzy* dilambangkan dengan $A \cup B$ dimana fungsi keanggotaanya ditentukan oleh

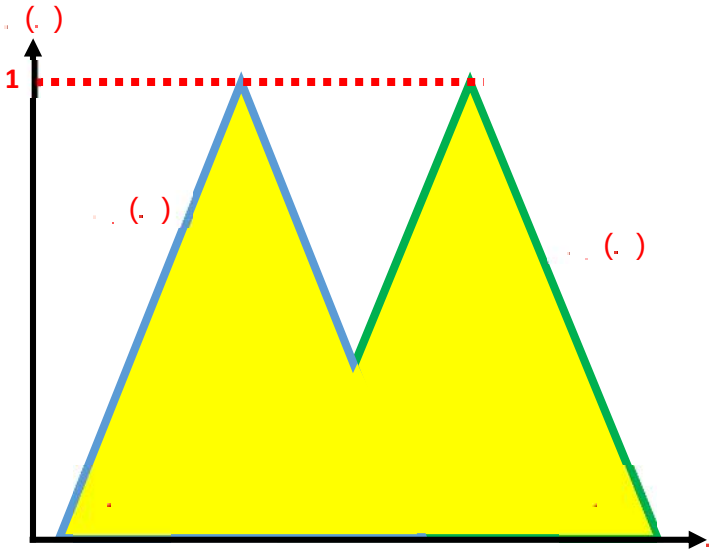
$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \quad \forall x \in X \tag{2.24}$$

atau dengan bentuk singkat

$$\mu_{A\square B}(x) = \mu_A(x) \square \mu_B(x) \tag{2.25}$$



Gambar 2.11 Komplemen himpunan *fuzzy*



Gambar 2.12 Gabungan dari dua himpunan *fuzzy*

Definisi 2.13 Irisan dari dua himpunan *fuzzy* A dan B dengan fungsi keanggotaan $\mu_A(x)$ dan $\mu_B(x)$ merupakan himpunan *fuzzy* dilambangkan dengan $A \cap B$ dimana fungsi keanggotaanya ditentukan oleh

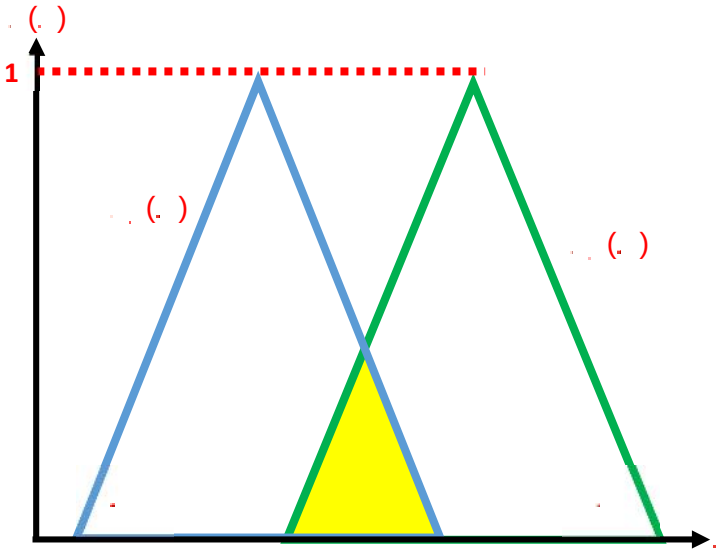
$$\mu_{A\cap B}(x) = \min[\mu_A(x), \mu_B(x)] \quad \forall x \in X \tag{2.26}$$

atau dengan bentuk singkat

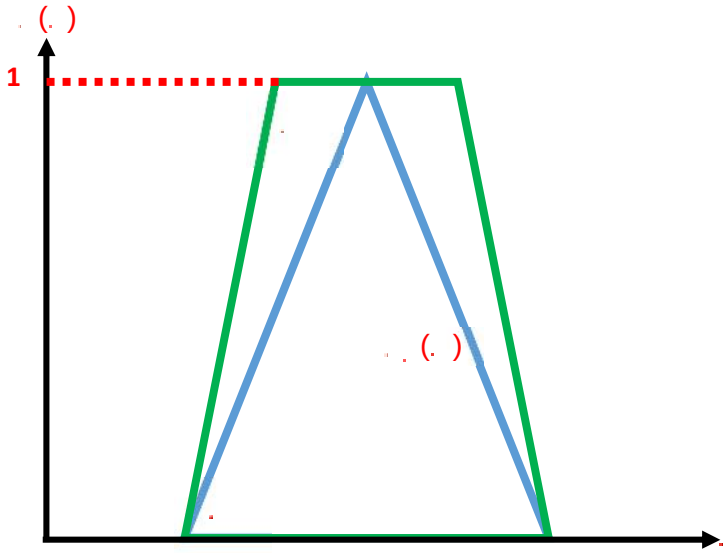
$$\mu_{A\cap B}(x) = \mu_A(x) \cap \mu_B(x) \tag{2.27}$$

Definisi 2.14 Suatu himpunan *fuzzy* A termasuk (*included*) dalam B dengan fungsi keanggotaan $\mu_A(x)$ dan $\mu_B(x)$ dilambangkan dengan $A \subseteq B$ jika

$$\mu_A(x) \leq \mu_B(x) \quad \forall x \in X \tag{2.28}$$



Gambar 2.13 Irisan dari dua himpunan *fuzzy*



Gambar 2.14 Inklusi dari dua himpunan *fuzzy*

2.6 Bilangan *Fuzzy*

Bilangan *fuzzy* merupakan generalisasi dari bilangan *real* dan bisa dikatakan suatu bilanngan *fuzzy* adalah suatu himpunan bagian *fuzzy* dari garis *real* yang mempunyai sifat-sifat tambahan. Konsep bilangan *fuzzy* merupakan dasar dari analisis *fuzzy* dan merupakan alat yang berguna dalam beberapa penerapan dari himpunan *fuzzy* dan logika *fuzzy*. Berikut dibawah ini adalah definisi bilangan *fuzzy* menurut Rutkowska (2002).

Definisi 2.11 Suatu himpunan *fuzzy* A merupakan bilangan *fuzzy* jika *universe of discourse* X adalah $\mathbb{R}$ dan kriteria-kriteria berikut terpenuhi:

a. Himpunan *fuzzy* A merupakan konveks

b. Himpunan *fuzzy* A merupakan himpunan *fuzzy* normal

c. Fungsi keanggotaan dari himpunan *fuzzy* $\mu_A(x)$ merupakan *piecewise continuous*

d. Inti dari himpunan *fuzzy* hanya terdiri dari satu nilai saja

Gambar 2.15 merupakan contoh dari bilangan *fuzzy*.dan $\mathbb{R}_{\mathcal{F}}$ melambangkan ruang dari bilangan *fuzzy*.

Definisi 2.12 Suatu himpunan *fuzzy* A merupakan selang *fuzzy* jika *universe of discourse* X adalah $\mathbb{R}$ dan kriteria-kriteria berikut terpenuhi:

a. Himpunan *fuzzy* A merupakan konveks

b. Himpunan *fuzzy* A merupakan himpunan *fuzzy* normal

c. Fungsi keanggotaan dari himpunan *fuzzy* $\mu_A(x)$ merupakan *piecewise continuous*

Gamber 2.14 merupakan contoh dari selang *fuzzy*. Harus dicatat bahwa selang *fuzzy* merupakan himpunan *fuzzy* dengan kriteria yang sama seperti bilangan *fuzzy*, namun dengan pengecaulian inti dari himpunan *fuzzy* tidak terbatas pada satu titik saja. Untuk bilangan *fuzzy* dan selang *fuzzy*, *universe of discourse* merupakan bilangan *real*. Jadi bilangan *fuzzy* dan selang *fuzzy* merupakan kasus khusus dari himpunan *fuzzy*. Kadang-kadang selang *fuzzy* juga dianggap sebagai bilangan *fuzzy*.

Menurut Bede (2013), bilangan *real* merupakan bilangan *fuzzy*, yaitu $\mathbb{R} \subset \mathbb{R}_{\mathcal{F}}$. Disini $\mathbb{R} \subset \mathbb{R}_f$ dimengerti sebagai

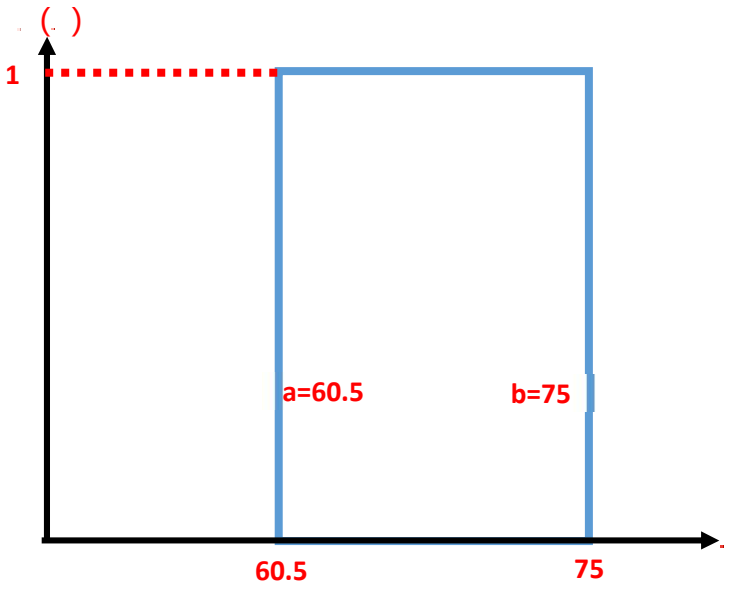
$$\mathbb{R} = \{\mathcal{X}_{(x)}; x \text{ merupakan bilangan } real \text{ biasa }\}$$

$\mathcal{X}_{(x)}$ merupakan bilangan *fuzzy* singleton untuk sebarang $x \in \mathbb{R}$ dan $\mathcal{X}_{(x)}$ bisa diidentifikasi dengan $x \in \mathbb{R}$ (Gambar 2.8 mengilustrasikan hal ini). Dan juga, bilangan *fuzzy* menggeneralisasi interval tertutup. Jika $\mathbb{I}$ melambangkan himpunan dari semua interval *real*, maka $\mathbb{I} \subset \mathbb{R}_{\mathcal{F}}$, dimana

$$\mathbb{I} = \{\mathcal{X}_{[a,b]}; [a, b] \text{ merupakan interval } real \text{ biasa }\}$$

Gambar 2.14 menglustrasikan interval tertutup yang dinterpretasikan sebagai bilangan *fuzzy*.

Bilangan *fuzzy* L-R penting dalam teori himpunan *fuzzy*. Bilangan *fuzzy* L-R dan kasus-kasus khusunya seperti bilangan *fuzzy* segitiga dan trapezium, sangat berguna dalam berbagai macam aplikasi. Berikut adalah definisi dari bilangan *fuzzy* L-R menurut Bede (2013).



Gambar 2.15 Interval tertutup yang diinterpretasikan dengan bilangan *fuzzy*

Definisi 2.13 Misalkan L, R [0,1] → [0,1] merupakan dua fungsi kontinu naik Memenuhi $L(0) = R(0) = 0, L(1) = R(1) = 1$. Misalkan $a_0^- \leq a_1^- \leq a_1^+ \leq a_0^+$ merupakan bilangan *real*. Himpunan u ℝ → [0,1] merupakan suatu bilangan *fuzzy* $L - R$ jika

$$u(x) = \begin{cases} 0 & x < a_0^- \\ L\left(\dfrac{x - a_0^-}{a_1^- - a_0^-}\right), & a_0^- < x < a_1^- \\ 1 & a_1^- < x < a_1^+ \\ R\left(\dfrac{a_0^+ - x}{a_0^+ - a_1^+}\right) & a_1^+ < x < a_0^+ \\ 0 & x \geq a_0^+ \end{cases} \tag{2.29}$$

Dengan menggunakan symbol (2.29) dapat ditulis menjadi

$$u = (a_0^-, a_1^-, a_1^+, a_0^+) \tag{2.30}$$

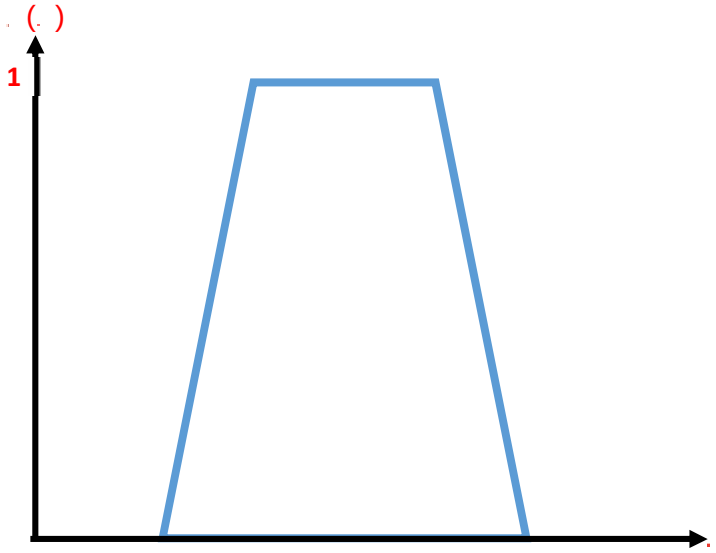
Dimana $[a_1^-, a_1^+]$ merupakan inti dari A dan $\underline{a} = a_1^- - a_0^-$, $\overline{a} = a_1^+ - a_0^+$ disebut dengan penyebaran kiri dan kanan. Jika A merupakan bilangan *fuzzy* $L - R$ maka tingkat himpunanya bisa dihitung sebagai

$$u_\alpha = [a_0^- + L^{-1}(\alpha)\underline{a}\,, a_0^+ - R^{-1}(\alpha)\overline{a}] \tag{2.31}$$

Dalam kasus tertentu, bilangan *fuzzy* trapesium didapat ketika fungsi L dan R linear. Suatu bilangan *fuzzy* trapesium A dapat dinyatakan dengan $(a, b, c, d) \in \mathbb{R}^4, a \leq b \leq c \leq d$,

$$u(x) = \begin{cases} 0 & x < a \\ \dfrac{x-a}{b-a} & a \leq x < b \\ 1 & b \leq x \leq c \\ \dfrac{d-t}{d-c} & c < x \leq d \\ 0 & d < x \end{cases} \tag{2.32}$$

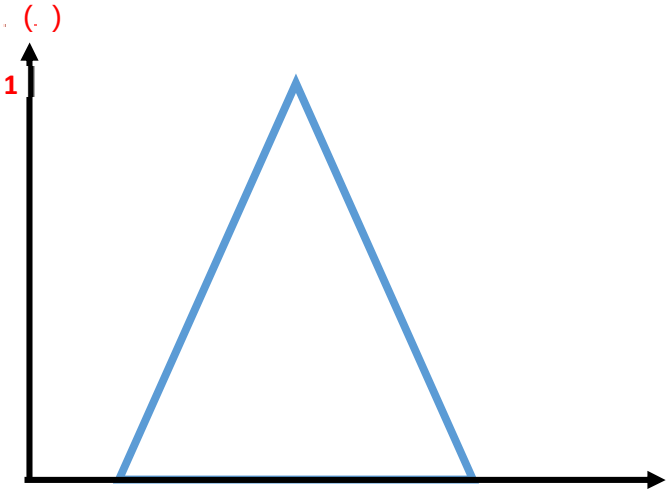
Gambar 2.15 mengilustrasikan bilangan *fuzzy* trapesium.



Gambar 2.16 Bilangan *fuzzy* trapesium

Kemudian jika bilangan *fuzzy* trapesium memiliki $b = c$, maka bilangan *fuzzy* tersebut dinamakan bilangan *fuzzy* segitiga. Bilangan *fuzzy* segitiga dapat dinyatakan dengan $(a, b, c) \in \mathbb{R}^3, a \leq b \leq c$,

$$u(x) = \begin{cases} 0 & t < a \\ \dfrac{x-a}{b-a} & a \le t < b \\ 1 & b \\ \dfrac{x-t}{c-b} & b < t \le c \\ 0 & t > c \end{cases} \tag{2.33}$$



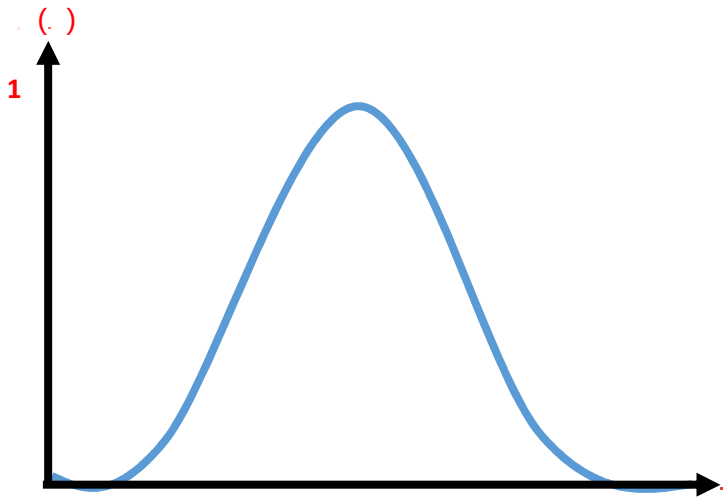
Gambar 2.17 Bilangan *fuzzy* segitiga

Bilangan *fuzzy* lainya yang sering digunakan adalah bilangan *fuzzy* Gaussian. Suatu bilangan *fuzzy* Gaussian mememliki fungsi keanggotaan

$$u(x) = \begin{cases} 0, & x < x_1 - a\sigma_l \\ \exp\left(-\dfrac{(x-x_1)^2}{2\sigma_l^2}\right), & x_1 - a\sigma_l \le x \le x_1 \\ \exp\left(-\dfrac{(x-x_1)^2}{2\sigma_r^2}\right), & x_1 \le x \le x_1 + a\sigma_r \\ 0, & x_1 + a\sigma_r \le x \end{cases} \tag{2.34}$$

Dimana x_1 adalah inti dari bilangan *fuzzy*, σ_l, σ_r adalah standar deviasi kiri dan kanan dan $a > 0$ adalah nilai toleransi. Gambar 2.15 merupakan ilustrasi dari bilangan *fuzzy* Gaussian. Ketiga bilangan *fuzzy* di atas banyak digunakan dalam penerapan, khususnya pada proses *fuzzification.* *Fuzzification* adalah proses kunatitas *crisp* menjadi *fuzzy*. Konversi nilai *fuzzy* ditunjukkan dengan fungsi

keanggotaan. Proses *fuzzification* melibatkan penugasan nilai kenggotaan untuk kuantitas *crisp* yang diberikan



Gambar 2.18 Bilangan *fuzzy* Gaussian

2.7 Ukuran Kemiripan *Fuzzy*

Ukuran kemiripan *fuzzy* digunakan untuk membandingkan dua objek yang berbeda. Semakin besar nilai kemiripan dari objek maka semakin besar pula kesamaan antara dua objek tersebut. Berikut adalah definisi dari ukuran kemiripan *fuzzy* menurut Baccour *et al* (2014).

Definisi 2.13 Misalkan $\mathcal{F}(X)$ melambangkan koleksi dari himpunan *fuzzy* dari suatu *universe of discourse* X. Ukuran kemiripan *fuzzy* antara himpunan *fuzzy* A dan B dilambangkan dengan $\sigma(A,B)$, jika pemetaan

$$\sigma: \mathcal{F}(X) \times \mathcal{F}(X) \rightarrow \mathcal{F}(X), \qquad (A,B) \rightarrow \sigma(A,B)$$

memenuhi sifat-sifat berikut

a. $\sigma(A,B) = \sigma(B,A) \quad \forall A,B \in \mathcal{F}(X)$

b. $0 \leq \sigma(A,B) \leq 1 \quad \forall A,B \in \mathcal{F}(X)$

c. $\sigma(A,B) = 1 \leftrightarrow A = B \quad \forall A,B \in \mathcal{F}(X)$

d. $\forall A,B,C \in \mathcal{F}(X), A \subseteq B \subseteq C \rightarrow \sigma(A,B) \geq \sigma(A,C) \wedge \sigma(B,C) \geq \sigma(A,C)$

2.8 Aritmatika *Fuzzy*

Sebelum dibahas tentang aritmatika *fuzzy* terlebih dahulu akan dibahas tetang Prinsip Perluasan Zadeh yang menjadi dasar dari aritmatika hinpunan *fuzzy*.

Definisi 2.14 Diberikan sebarang fungsi $f: X \rightarrow Y$, dimana X dan Y merupakan himpunan biasa, hal ini dapat diperluas menjadi suatu fungsi $F: \mathcal{F}(X) \rightarrow \mathcal{F}(Y)$ (suatu fungsi *fuzzy*) sedemikian sehingga $v = F(u)$, dimana

$$v(y) = \begin{cases} \sup\{u(x): x \in X, f(x) = y\}, & whe \quad f^{-1}(y) \neq \emptyset \\ 0, & o\ he \end{cases}$$

Fungsi F disebut sebagai perluasan Zadeh dari f

Selanjutnya akan didefinisi penjumlahan dan perkalian scalar pada himpunan *fuzzy* pada definisi berikut ini.

Definisi 2.15 Untuk $u, v \in \mathbb{R}_{\mathcal{F}}$ dan $\lambda \in \mathbb{R}$, berdasarkan prinsip perluasan Zadeh, dapat didefinisikan jumlah dari dua bilangan *fuzzy* $u + v$ dan perkalian antara bilangan real dan bilangan *fuzzy* $\lambda \cdot u$ sedemikian sehingga

$$(u + v)_\alpha = \{x + y | x \in u_\alpha, y \in v_\alpha\} = u_\alpha + v_\alpha, \quad \text{.....(2.35)}$$

$$(\lambda \cdot u)_\alpha = \{\lambda \ | x \in u_\alpha\} = \lambda u_\alpha \quad \text{.....(2.36)}$$

dimana $u_\alpha + v_\alpha$ merupakan penjumahan dari dua interval (sebagai himpunan bagian dari $\mathbb{R}$) dan λu_α merupakan perkalian biasa dari bilangan dan himpunan bagian dari $\mathbb{R}$. Jadi, aritmatika *fuzzy* merupakan perluasan dari aritmatika interval.

Kemudian definisi dari perkalian antara dua himpunan *fuzzy*

Definisi 2.16 Untuk $u, v \in \mathbb{R}_{\mathcal{F}}$, berdasarkan prinsip perluasan Zadeh, dapat didefinisikan perkalian dari dua bilangan *fuzzy* $u \cdot v$ sedemikian sehingga

$$(u \cdot v)_{\alpha}^{-} = \min\{u_{\alpha}^{-}v_{\alpha}^{-}, u_{\alpha}^{-}v_{\alpha}^{+}, u_{\alpha}^{+}v_{\alpha}^{-}, u_{\alpha}^{+}v_{\alpha}^{+}\} \quad (2.35)$$

dan

$$(u \cdot v)_{\alpha}^{+} = \min\{u_{\alpha}^{-}v_{\alpha}^{-}, u_{\alpha}^{-}v_{\alpha}^{+}, u_{\alpha}^{+}v_{\alpha}^{-}, u_{\alpha}^{+}v_{\alpha}^{+}\} \quad (2.36)$$

(Bede, 2013)

2.9 Peubah Acak *Fuzzy*

Sebelum dijelaskan tentang peubah acak *fuzzy*, terlebih dahulu akan dijelaskan tentang dasar-dasar dari probabilitas menurut menurut Hoel *et al* (1971).

Definisi 2.17 Suatu himpunan bagian tak kosong $\boldsymbol{S}$ dari $\boldsymbol{\Omega}$ disebut sebagai suatu *σ-field* dari himpunan bagian dari $\boldsymbol{\Omega}$ dengan memenuhi sifat-sifat sebagai berikut:

a. Jika $\boldsymbol{A}$ berada di $\boldsymbol{S}$, maka $\boldsymbol{A^c}$ juga berada di $\boldsymbol{S}$

b. Jika $\boldsymbol{A}$ berada di $\boldsymbol{S}$, $n = 1, 2, \ldots$, maka $\bigcup_{n-1}^{\infty} \boldsymbol{A_n}$ dan $\bigcap_{n-1}^{\infty} \boldsymbol{A_n}$ keduanya berada di $\boldsymbol{S}$

Definisi 2.18 Suatu ukuran probabilitas *P* dalam suatu *σ-field* dari himpunan bagian $\boldsymbol{S}$ dari suatu himpunan $\boldsymbol{\Omega}$ merupakan suatu fungsi bernilai real yang memiliki domain $\boldsymbol{S}$ dengan memenuhi sifat-sifat sebagai berikut:

a. $P(\boldsymbol{\Omega}) = 1$

b. $P(\mathbf{A}) \geq 0 \; \forall \; \boldsymbol{A} \in \mathbf{S}$

c. Jika $\boldsymbol{A_n}$, $n = 1, 2, \ldots$, merupakan himpunan yang tak beririsan di $\boldsymbol{\Omega}$, maka

$$P\left(\bigcup_{n=1}^{\infty} \boldsymbol{A_n}\right) = \sum_{n=1}^{\infty} P(\boldsymbol{A_n})$$

Definisi 2.19 Suatu ruang probabilitas, dilambangkan dengan $(\boldsymbol{\Omega}, \mathbf{S}, \mathrm{P})$, merupakan suatu himpunan **Ω,** suatu *σ-field* himpunan bagian **S,** dan suatu ukuran probabilitas P

Setelah mengetahui dasar-dasar dari peluang selajutnya akan dijelaskan definisi dari peubah acak *fuzzy*. Menurut Gill *et al* (2006) definisi peubah acak *fuzzy* adalah sebagai berikut.

Definisi 2.20 Diberikan suatu ruang probabilitas $(\boldsymbol{\Omega}, \mathbf{S}, \mathrm{P})$, suatu pemataan

$$\mathcal{X}: \boldsymbol{\Omega} \to \mathbb{R}_{\mathcal{F}}$$

disebut suatu peubah acak *fuzzy* jika $\forall \alpha \in [0,1]$ dua nilai real memetakan

$$\inf \mathcal{X}_\alpha: \boldsymbol{\Omega} \to \mathbb{R} \qquad\qquad \sup \mathcal{X}_\alpha: \boldsymbol{\Omega} \to \mathbb{R}$$

(didefinisikan sedemikian sehingga $\forall \omega \in \boldsymbol{\Omega}$ terdapat

$\mathcal{X}_\alpha(\omega) = [\inf(\mathcal{X}(\omega))_\alpha, \sup(\mathcal{X}(\omega))_\alpha]$) merupakan peubah acak bernilai real

Contoh 2.1 Misalkan suatu percobaan acak dimana seorang peneliti memilih secara acak suatu konfrensi tentang logika *fuzzy* dan diselenggarakan dalam periode 2004-2005.

Asumsikan bahwa peubah acak $\mathcal{X}$ adalah persepsi tentang biaya registrasi dari konfrensi tertentu untuk anggota EUSFLAT. Situasi ini dapat dimodelkan dengan konsep peubah acak *fuzzy*. Jadi, ruang probabilitas $(\boldsymbol{\Omega}, \mathbf{S}, \mathrm{P})$ adalah

Ω = konfrensi yang diadakan dari 2004 sampai 2005 yang membahas tentang logika *fuzzy*

S = himpunan kuasa dari Ω

dan, karena konfrensi dipilih secara acak,

$$P(\omega) = \frac{1}{|\Omega|}, \ \square\omega \ \square \ \mathbf{\Omega}.$$

Di sisi lain, persepsi biaya regristrasi untuk anggota EUSFLAT dapat dipandang sebagai peubah acak *fuzzy* $\mathcal{X}$ yang berkaitan dengan $(\mathbf{\Omega}, \mathbf{S}, \mathrm{P})$ dan bernilai

$\tilde{x}_1$ = sangat murah

$\tilde{x}_2$ = murah

$\tilde{x}_3$ = agak murah

$\tilde{x}_4$ = moderat

$\tilde{x}_5$ = agak mahal

$\tilde{x}_6$ = mahal

$\tilde{x}_7$ = sangat mahal

dimana dapat dijelaskan dengan himpunan *fuzzy* trapesium seperti pada Gambar 2.16. Dalam contoh ini misalkan $\alpha = 0$ maka

$\mathcal{X}_0(\omega)$= biaya registrasi yang sebenarnya dari konfrensi dalam **Ω.**

Jika, dalam basis biaya regristrasi dari konfrensi dalam **Ω,** maka didapat distribusi peluang

$$P(\{\omega \ \square \ \Omega | \mathcal{X}_0(\omega) = \tilde{x}_1\}) = P(\mathcal{X}_0 = [0{,}150]) = 0.04$$

$$P(\{\omega \ \square \ \Omega | \mathcal{X}_0(\omega) = \tilde{x}_2\}) = P(\mathcal{X}_0 = [50{,}350]) = 0.14$$

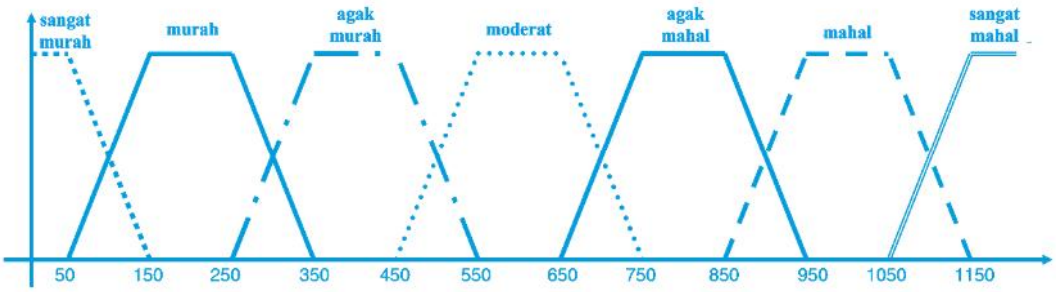
$$P(\{\omega \ \square \ \Omega | \mathcal{X}_0(\omega) = \tilde{x}_3\}) = P(\mathcal{X}_0 = [250{,}550]) = 0.25$$

$$P(\{\omega \ \square \ \Omega | \mathcal{X}_0(\omega) = \tilde{x}_4\}) = P(\mathcal{X}_0 = [450{,}750]) = 0.34$$

$$P(\{\omega \ \square \ \Omega | \mathcal{X}_0(\omega) = \tilde{x}_5\}) = P(\mathcal{X}_0 = [650{,}950]) = 0.19$$

$$P(\{\omega \ \square \ \Omega | \mathcal{X}_0(\omega) = \tilde{x}_6\}) = P(\mathcal{X}_0 = [850{,}1150]) = 0.03$$

$$P(\{\omega \ \square \ \Omega | \mathcal{X}_0(\omega) = \tilde{x}_7\}) = P(\mathcal{X}_0 = [1050{,}1200]) = 0.01$$



Gambar 2.19 Nilai dari 'persepsi tentang biaya registrasi' dari suatu konfrensi (dalam euro)

2.10 K-*Nearest Neighbor* (KNN)

Menurut Rencher & Christensen (2012), Prosedur KNN sangat sederhana. Pertama-tama dihitung jarak antara suatu observasi y_i dan semua titik y_j dengan menggunakan rumus jarak (Euclidean atau Mahalanobis). Untuk mengelompokan y_i kedalam satu dari dua grup, k titik terdekat dari y_i diperikasa, dan jika mayoritas dari k titik dimiliki oleh G_1, maka y_i termasuk pada G_1; selainya y_i termasuk pada G_2. Jika dilambangkan jumlah titik dari G_1 sebagai k_1, dengan sisanya k_2 titik dari G_2, dimana $k = k_1 + k_2$,maka aturan diatas dapat dinyatakan sebagai: Tugaskan y_i ke G_1 jika $k_1 > k_2$ dan ke G_2 selainnya. Jika ukuran sampel n_1 dan n_2 berbeda, digunakanlah proporsi: Tugaskan y_i ke G_1 jika $\frac{k_1}{n_1} > \frac{k_2}{n_2}$ dan ke G_2 selainnya.

Aturan ini mudah untuk diperluas untuk lebih dari 2 grup. Sebagai contoh: Tugaskan suatu observasi ke grup yang memiliki proporsi tertinggi $\frac{k_i}{n_i}$, dimana k_i merupakan jumlah observasi dari G_i diantara k nearest neighbor dari observasi

tersebut. Nilai dari k harus ditentukan terlebih dahulu, dimana disarankan memilih niai k didekat $\sqrt{n_i}$ untuk suatu n_i.

2.11 Metode Monte Carlo

Motode Monte Carlo berhubungan dengan konsep pengambilan sampel. Metode pengambilan sampel adalah proses memilih nilai tertentu dari suatu variabel acak yang berdistribusi terntentu. Sebagai contoh, misalkan suatu vaiabel acak berdistribusi eksponensial. Maka, pengambilan sanpel dari variabel tersebut menghasilkan data berdistribusi eksponensial juga. Banyak teknik-teknik yang biasa digunakan untuk membangkitkan nilai dari variabel acak yang memiliki distribusi tertetu. Metode yang paling sering adalah membangkitkan nilai dari suatu variabel acak x dengan distribusi tertentu yang didasarkan *pseudo random number generator* yang dapat menghasilkan barisan panjang *pseudo random number* yang berdistribusi Uniform dengan parameter (0,1).
(Dunn & Shultis, 2012)

Pada praktikanya banyak paket program yang telah menyediakan *pseudorandom number generator* untuk distribusi tertentu yang didasarkan pada prinsip pengambilan sampel pada Monte Carlo.

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan pada semester ganjil tahun akademik 2015/2016, bertempat di Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung.

3.2 Metode Penelitian

Adapun langkah-langkah yang dilakukan dalam penelitian ini adalah sebagai berikut :

1. Melakukan kajian tentang jaringan syaraf tiruan FLVQ beserta algoritmanya. Adapun kajian yang dilakukan adalah sebagai berikut:

a. Proses *fuzzification* data dan bobot pada model FLVQ

b. Ukuran Kemiripan *fuzzy* yang akan digunakan pada model FLVQ

c. Pembelajaran dalam FLVQ

2. Melakukan pembelajaran FLVQ untuk mengidentifikasi distribusi peluang dengan langkah-langkah sebagai berikut:

a. Melakukan pengambilan sampel data yang memiliki distribusi normal, distribusi eksponensial, distrbusi Weibull, distribusi seragam, distribusi chi-*square*, distribusi t, distribusi F, distribusi lognormal, dan distribusi gamma dengan

metode Monte Carlo yang berukuran 5, 10, 20, 30 dan 100. Untuk banyaknya data sampel pada masing-masing ukuran adalah 300.

b. Membagi data sampel menjadi *training* data dan *testing* data secara acak. Untuk banyaknya sampel pada *training* data dan *testing* data adalah 1800 dan 900 pada masing-masing ukuran sampel

c. Menentukan nilai awal vektor *codebook* dengan metode K-*Nearest Neighbor* (KNN) pada masing-masing ukuran sampel.

d. Melakukan *fuzzification* pada vektor *codebook* awal, *training* data dan *testing* data

e. Menentukan *learning rate* dan maksimum *epoch* pada *Fuzzy Learning Vector Quantization* (FLVQ)

f. Melakukan pelatihan pada FLVQ dengan menggunakan *training data* yang sudah dalam bentuk data *fuzzy*

g. Melakukan pengujian terhadap FLVQ yang telah selesai melakukan pelatihan dengan menggunakan *testing data* untuk melihat tingkat generalisasi FLVQ.

h. Menghitung kekuatan uji dari FLVQ dengan melakukan pengujian pada 1000 pengulangan pada masing-masing distribusi

V. KESIMPULAN

5.1 Kesimpulan

Adapun kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut:

1. Ukuran kemiripan *fuzzy* $M(A,B) = \frac{\sum_{i=1}^{n} \mathrm{m} \quad (\mu_A^i, \mu_B^i)}{\sum_{i=1}^{n} \mathrm{m} \quad (\mu_A^i, \mu_B^i)}$ terbukti memenuhi keempat aksioma pada definisi ukuran kemiripan *fuzzy*.
2. Pembelajaran *Fuzzy Learning Vector Quantization* memiliki dua tahap yaitu, tahap mendekatkan atau menjauhkan vektor *codebook* dari data input dan memperbesar atau memperkecil tingkat *fuzziness* vektor *codebook.*
3. Model FLVQ baik untuk mengidentifikasi distribusi peluang dari suatu data yang ditunjukkan dari hasil peluang kebenaran klasifikasi yang rata-rata dibatas 0.9.
4. Semakin Besar ukuran sampel maka model FLVQ akan semakin baik dalam mengidentifikasi distribusi peluang

5.2 Saran

Pada penelitian ini peniliti menggunakan jaringan syaraf tiruan FLVQ untuk mengidentifikasi distribusi peluang. Oleh karena itu, dapat dapat dilakukan penelitian yang sama dengan menggunakan model jaringan syaraf tiruan yang berbeda.

DAFTAR PUSTAKA

Baccour, L., Alimi, A. M., & John, R. I. (2014). Some Notes on Fuzzy Similarity Measures and Applications to Classification of Shapes, Recognition of Arabic Sentences and Mosaic. *IAENG*.

Bede, B. (2013). *Mathematics of Fuzzy Set and Fuzzy Logic.* New York: Springer.

Du, K.-L., & Swamy, M. (2014). *Neural Network and Statistical Learning.* London: Springer.

Dunn, W. L., & Shultis, J. K. (2012). *Exploring Monte Carlo Methods.* Oxford: Elsevier.

Gill, M. A., Lopez-Diaz, M., & Ralescu, D. A. (2006). Overview on the Development of Fuzzy Random Variables. *Fuzzy Sets and Systems*, 2546-2557.

Haykin, S. S. (2009). *Neural Network and Machine Learning* (3rd ed.). New Jersey: Pearson Prentice Hall.

Klir, G. J., & Yuan, B. (1995). *Fuzzy Sets and Fuzzy Logic : Theory and Application.* New Jersey: Prentice Hall.

Kohonen, T. (1990). Improved Version of Learning Vector Quantization. *International Joint Conference on Neural Network.* Baltimore MD.

Kohonen, T. (1995). *Self-Organizing Maps.* Berlin: Springer.

Rencher, A. C., & F., C. W. (2012). *Methods of Multivariate Analysis* (3rd ed.). Canada: John Wiley & Sons.

Russel, S. J., & Norvig, P. (2010). *Artificial Intelligence A Modern Approach* (3rd ed.). New Jersey: Pearson.

Rutkowska, D. (2002). *Neuro-Fuzzy Architectures and Hybird Learning.* Berlin: Springer.

Sakuraba, Y., Nakamoto, T., & Moriizumi, T. (1991). New Method of Learning Vector Quantization. *Systems and Computers in Japan*.

Samarasinghe, S. (2007). *Neural Networks for Applied Science and Engineering.* New York: Auerbach Publications.

Su, C.-T., & C.-J, C. (2007). A Neural Network-Based Approach for Statistical Distribution Recognition. *Quality Engineering*, 293-297.