

Tesis - TE142599

**KLASIFIKASI KEMAMPUAN KOGNITIF PEMBELAJAR
PADA PERMAINAN MATEMATIKA BERBASIS
METODE *BACKPROPAGATION NEURAL NETWORK***

Heny Yuniarti

2212205017

DOSEN PEMBIMBING

Prof. Dr. Ir. Mauridi Hery Purnomo, M. Eng.
Dr. I Ketut Eddy Purnama, S.T., M.T.

PROGRAM MAGISTER

BIDANG KEAHLIAN JARINGAN CERDAS MULTIMEDIA

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNOLOGI INDUSTRI

INSITITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2016



Tesis - TE142599

**STUDENT COGNITIVE SKILL CLASSIFICATION
IN MATHEMATIC GAME BASED ON
BACKPROPAGATION NEURAL NETWORK
METHOD**

Heny Yuniarti

2212205017

SUPERVISOR

Prof. Dr. Ir. Mauridi Hery Purnomo, M. Eng.
Dr. I Ketut Eddy Purnama, S.T., M.T.

MAGISTER PROGRAM

INTELLIGENT NETWORK EXPERTISE MULTIMEDIA

DEPARTMENT OF ELECTRICAL ENGINEERING

FACULTY OF INDUSTRIAL TECHNOLOGY

SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY

SURABAYA

2016

Tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T)
di
Institut Teknologi Sepuluh Nopember

Oleh :
Heny Yuniarti
NRP. 2212205017

Tanggal Ujian : 20 Juni 2016
Periode Wisuda : September 2016

Disetujui oleh :

1. Prof. Dr. Ir. Mauidhi Hery Purnomo, M.Eng. (Pembimbing I)
NIP. 195809161986011001
2. Dr. I Ketut Eddy Purnama, ST., MT. (Pembimbing II)
NIP. 19690730 199512 1 001
3. Dr. Ir. Yoyon Kusnendar Suprpto, M.Sc. (Penguji)
NIP. 195409251978031001
4. Dr. Surya Sumpeno, ST., M.Sc. (Penguji)
NIP. 19690613 199702 1 003
5. Dr. Supeno Mardi Susiki Nugroho, ST., MT. (Penguji)
NIP. 19700313 199512 1 001

Direktur Program Pasca Sarjana,

Prof. Ir. Djauhar Manfaat, M.Sc., Ph.D.
NIP. 19601202 198701 1 001
PROGRAM
PASCASARJANA

**KLASIFIKASI KEMAMPUAN KOGNITIF PEMBELAJAR PADA
PERMAINAN MATEMATIKA BERBASIS METODE
*BACKPROPAGATION NEURAL NETWORK***

Nama Mahasiswa : Heny Yuniarti
NRP : 2212205017
Dosen Pembimbing : Prof. Dr. Ir. Mauridhi Hery P., M.Eng
Dr. I Ketut Eddy Purnama, S.T., M.T.

ABSTRAK

Mengklasifikasikan profil pembelajar siswa merupakan suatu kebutuhan dalam dunia pendidikan. Klasifikasi tersebut bisa meliputi kemampuan kognitif maupun gaya belajar dari seorang siswa. Kemampuan kognitif siswa dalam belajar di sekolah sangat penting untuk diketahui perkembangannya agar seorang pengajar dapat mengevaluasi kegiatan belajar mengajar yang dilakukan telah sesuai dengan yang diharapkan atau belum. Dengan menerapkan metode jaringan saraf tiruan, diharapkan hasil klasifikasi dapat menunjukkan angka galat yang sedikit. Penelitian ini mengklasifikasikan profil pembelajar dari segi kemampuan kognitifnya. Klasifikasi ini diharapkan dapat memudahkan pengajar dalam menilai kemampuan anak didiknya serta mengevaluasi kegiatan belajar mengajar di dalam kelas. Dengan cara membandingkan keluaran sistem dengan keluaran hasil tes tulis siswa, didapatkan nilai selisih akar kuadrat (RMSE) senilai 0.295.

Kata kunci : permainan matematika, klasifikasi profil pembelajar, *artificial neural network*

[halaman ini sengaja dikosongkan]

STUDENT COGNITIVE SKILL CLASSIFICATION IN MATHEMATIC GAME BASED ON *BACKPROPAGATION NEURAL NETWORK METHOD*

Student Name : Heny Yuniarti
NRP : 2212205017
Advisor : Prof. Dr. Ir. Mauridhi Hery P., M.Eng
Dr. I Ketut Eddy Purnama, S.T., M.T.

ABSTRACT

Classify student learner profile is a necessity in education sector. Such classification includes cognitive ability and learning style of the student. The student cognitive skill in learning at school is very important to note its development so that a teacher can evaluate the studying and teaching methods that have been done. By applying Artificial Neural Network method, expected results of classification can show a small error. This research classify student cognitive ability. This classify is expected to help teacher in assessing student ability and also evaluate the studying and teaching methods at class. Student cognitive ability can be classified in this research with RMSE value 0.295.

Keywords: Mathematic game, student profile clasification, artificial neural network

[halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis ke hadirat Allah S.W.T atas berkat limpahan rahmat dan hidayah-Nya sehingga Tesis yang berjudul: *KLASIFIKASI KEMAMPUAN KOGNITIF PEMBELAJAR PADA PERMAINAN MATEMATIKA BERBASIS METODE BACKPROPAGATION NEURAL NETWORK* dapat diselesaikan dengan baik dengan tak lupa mengucapkan syukur kehadirat-Nya. Semoga tesis ini dapat memberikan manfaat pada perkembangan ilmu pengetahuan terutama bidang komputasi cerdas dan visualisasi serta dapat memberikan kontribusi bagi peneliti selanjutnya.

Dengan selesai dan tersusunnya laporan tesis ini, maka penulis mengucapkan terima kasih atas bantuan dan dukungan dari berbagai pihak baik moral atau materiil dalam pembuatan tesis ini, antara lain:

1. Prof. Dr. Ir. Mauridhi Hery P., M.Eng selaku dosen pembimbing utama atas kesabaran membimbing dan dukungan yang diberikan hingga terselesaikannya tesis ini.
2. Dr. I Ketut Eddy Purnama, S.T., M.T. selaku dosen pembimbing kedua atas kesabaran membimbing dan dukungan yang diberikan hingga terselesaikannya tesis ini.
3. Bapak Yoyon, Pak Surya, dan Pak Uki selaku penguji tesis yang telah memberikan arahan.
4. Bapak dan Ibu dosen pascasarjana Teknik Elektro ITS yang telah bersedia dengan sabar mengajar dan memberi bimbingan selama masa kuliah.
5. Suami hebat, Muhammad Mundzir, yang selalu memberikan dukungan dan semangatnya dalam menyelesaikan tesis ini.
6. Kedua orang tua tercinta dan terhormat, atas segala doa, bimbingan, kasih sayang, perhatian, semangat, kerja keras, dan pengorbanannya demi kesuksesan putra-putrinya. Serta terimakasih juga kepada saudara kandung saya, Heru Saputra dan Hesti Melinda yang selalu mengingatkan untuk menyelesaikan tesis ini.
7. Teman baik saya, Nurissaidah Ulinnuha yang memberi semangat untuk mengerjakan tesis.
8. Teman-teman Game Tech angkatan 2012 yang berjuang bersama-sama selama dua tahun terakhir, atas dukungan dan motivasinya.
9. Teman-teman satu pengajian dan murabbi saya, yang mengisi ruhiyah dan senantiasa mengingatkan dalam lingkaran kebaikan.
10. Beasiswa *Fresh Graduate* ITS yang telah memberikan bantuan dana dalam studi S2 saya.

11. Juga tak lupa kepada semua pihak yang belum sempat disebutkan satu per satu disini yang telah membantu penulis dalam menyelesaikan studi S2 Teknik Elektro ITS.

Semoga Allah S.W.T. membalas semua kebaikan tersebut dengan pahala yang berlimpah, Aamiin.

Surabaya, Juni 2016

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	
PERNYATAAN KEASLIAN TESIS	
ABSTRAK.....	i
ABSTRACT.....	iii
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	xi
DAFTAR ISTILAH.....	xiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan dan Manfaat Penelitian	2
1.5 Sistematika Penulisan.....	2
BAB 2 DASAR TEORI DAN KAJIAN PUSTAKA	5
2.1 Taksonomi Bloom	5
2.1.1 Pengetahuan (<i>Knowledge</i>).....	6
2.1.2 Pemahaman (<i>Comprehension</i>)	6
2.1.3 Penerapan (<i>Application</i>).....	7
2.1.4 Analisis (<i>Analysis</i>).....	7
2.1.5 Sintesis (<i>Synthesis</i>).....	7
2.1.6 Evaluasi (<i>Evaluation</i>).....	7
2.2 Pedagogi.....	8
2.3 Jaring Saraf Tiruan	10
2.3.1 Fungsi aktivasi.....	12
2.3.2 Fungsi biaya	13
2.3.3 Algoritma Backpropagation	13
2.4 K-Fold Cross Validation	15

BAB 3 METODE PENELITIAN	17
3.1. Langkah-langkah Penelitian.....	17
3.2 Studi Literatur	17
3.3 Perencanaan Sistem.....	18
3.4 Uji Coba Sistem	23
3.5 Analisis Hasil	24
3.6 Penyusunan Laporan Akhir.....	24
3.7 Perancangan Data.....	24
3.7.1 Topik Matematika.....	26
3.7.2 Teknik Penilaian	27
3.8 Perancangan Metode	25
3.8.1 Desain Secara Umum.....	26
3.8.2 Tahap Normalisasi	27
3.8.3 Tahap Proses	27
3.9 Implementasi Kode	29
3.9.1 Tahap Praproses.....	29
3.9.2 Tahap Proses	30
3.9.3 Tahap Pengacakan Data.....	31
3.9.4 Tahap Proses (<i>Training</i> dan <i>Testing</i>).....	33
3.9.5 Tahap Inisialisasi Parameter Input dan Bobot	34
3.9.6 Tahap <i>Feedforward</i>	36
3.9.7 Tahap <i>Backpropagation</i>	38
3.9.8 Tahap Penghitungan RMSE.....	41
BAB 4 HASIL DAN PEMBAHASAN	43
4.1 Pengujian Sistem.....	45
4.2 Perbandingan dengan metode fuzzy.....	50
BAB 5 KESIMPULAN DAN SARAN	55
DAFTAR PUSTAKA	57
BIOGRAFI	

DAFTAR GAMBAR

Gambar 2.1. Enam Aspek Taksonomi Bloom Ranah Kognitif	6
Gambar 2.2. Struktur JST dengan satu lapisan tersembunyi	11
Gambar 2.3. Fungsi Sigmoid Logistik.....	12
Gambar 2.4. Fungsi tangen hiperbolik.....	12
Gambar 2.5. Pseudocode algoritma backpropagation.....	13
Gambar 2.6. Pseudocode k-fold cross validation.....	14
Gambar 2.7. Ilustrasi data dengan 4-fold cross validation.....	15
Gambar 3.1 Diagram Alur Kerja Langkah Penelitian	17
Gambar 3.2. Blok diagram perencanaan sistem.....	18
Gambar 3.3. Masukan, Proses dan Keluaran Pengolahan Data.....	18
Gambar 3.4. Tes Server apakah <i>online</i> atau tidak	20
Gambar 3.5. Pemain diminta menulis namanya	20
Gambar 3.6. Contoh Soal pada permainan	21
Gambar 3.7. Salah satu hadiah yang didapat ketika benar	22
Gambar 3.8. Permainan selesai	22
Gambar 3.9. <i>Output data game</i>	23
Gambar 3.10. Bank Soal pada Permainan Matematika	25
Gambar 3.11 Perancangan perangkat lunak model prediksi kognitif secara umum.....	26
Gambar 3.12 Diagram Alir Tahap <i>training</i> JST.....	28
Gambar 3.13 Diagram Alir Tahap <i>testing</i> JST.....	29
Gambar 3.14. Kode program tahap normalisasi data.....	30
Gambar 3.15. Kode program tahap proses k-fold cross validation.....	31
Gambar 3.16. Kode program mengacak baris data.....	32
Gambar 3.17. Kode program tahap proses training dan testing.....	33
Gambar 3.18. Kode program utama inialisasi.....	34
Gambar 3.19. Kode program inialisasi parameter jaringan.	35
Gambar 3.20. Kode program inialisasi bobot, bias dan delta pengubahannya.	36
Gambar 3.21. Kode program tahap feedforward jaringan	37
Gambar 3.22. Kode program penghitungan nilai aktivasi.	38
Gambar 3.23 Kode program tahap backpropagation.	39
Gambar 3.24 Kode program perubahan nilai bobot dan bias pada lapisan tersembunyi ke lapisan output.	40
Gambar 3.25 Kode program perubahan nilai bobot dan bias pada lapisan input ke lapisan tersembunyi.	41

Gambar 3.26 Kode program penghitungan RMSE.....	42
Gambar 4.1. Foto saat siswa memainkan <i>game</i>	43
Gambar 4.2. Data dengan username yang sama	44
Gambar 4.3. Data dengan username yang tidak sesuai.....	44
Gambar 4.4. Data jumlah skor permainan	45
Gambar 4.5. Data jumlah waktu permainan	46
Gambar 4.6. Grafik skor tes tulis siswa	46
Gambar 4.7. Perbandingan <i>output</i> prediksi dengan target sebelum normalisasi	50
Gambar 4.8. Fungsi keanggotaan <i>input</i> skor	50
Gambar 4.9 Fungsi keanggotaan <i>input</i> waktu.....	51
Gambar 4.10 Fungsi keanggotaan <i>output</i>	51
Gambar 4.11 Perbandingan <i>output</i> fuzzy dengan hasil tes tulis	54

DAFTAR TABEL

Tabel 3.1. Jenis soal pada permainan.....	18
Tabel 4.1. Data permainan dengan sembilan variabel soal.....	45
Tabel 4.2. Data permainan dengan sembilan variabel waktu	45
Tabel 4.3. Data setelah normalisasi	47
Tabel 4.4. Tabel Epoch dan <i>hidden layer</i>	48
Tabel 4.5. Tabel Pencarian <i>Learning Rate</i>	48
Tabel 4.6. Tabel Pencarian Momentum.....	48
Tabel 4.7. <i>Rule Base</i>	51
Tabel 4.7. Hasil <i>output</i> dengan metode fuzzy.....	52

[halaman ini sengaja dikosongkan]

DAFTAR ISTILAH

Taksonomi Bloom	: struktur hierarki yang mengidentifikasi <i>skills</i> mulai dari tingkat yang rendah hingga yang tinggi
Kognitif	: berisi perilaku yang menekankan aspek intelektual, seperti pengetahuan, dan keterampilan berpikir
Pedagogi	: suatu ilmu dan seni dalam mengajar anak-anak
JST	: Jaring Saraf Tiruan yaitu salah satu teknik klasifikasi yang akurat dan banyak digunakan pada aplikasi di berbagai sektor
<i>Feedforward</i>	: sinyal input diberikan ke dalam jaringan. Komputasi dilakukan oleh neuron pada setiap lapisan sehingga menghasilkan output
<i>Backpropagation</i>	: semua bobot diperbaiki berdasarkan <i>error</i> jaringan. Bobot diperbaiki untuk membuat output yang keluar semakin mendekati output yang diinginkan
Momentum	: untuk mempercepat proses belajar pada JST sekaligus menghindari ketidakstabilan algoritma
<i>K-fold cross validation</i>	: teknik yang menggunakan semua data yang tersedia sebagai <i>training</i> dan testing
RMSE	: <i>Root Mean Square Error</i> yaitu nilai rata-rata dari jumlah kuadrat kesalahan, juga dapat menyatakan ukuran besarnya kesalahan yang dihasilkan oleh suatu model prakiraan
δ_k	: <i>error</i> output
o_k	: keluaran dari lapisan output
t_k	: output yang diharapkan
δ_h	: <i>error</i> pada lapisan tersembunyi
o_h	: keluaran dari lapisan tersembunyi
w_{kh}	: bobot antara lapisan output dan lapisan tersembunyi
$\Delta w_{i,j}$: selisih bobot saat ini dengan bobot sebelumnya
$W_{i,j}$: semua bobot dan bias pada jaringan saraf
η	: <i>learning rate</i>
$x_{i,j}$: keluaran dari lapisan
α	: momentum

[halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

Pada Bab 1 dibahas mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, dan manfaat dari penelitian yang telah dilakukan, serta sistematika penulisan.

1.1 Latar Belakang

Perkembangan dunia permainan telah maju dengan pesat seiring perkembangan teknologi. Secara umum permainan dapat dibedakan menjadi dua jenis, yaitu *fun game* dan *serious game*. *Fun games* adalah permainan seperti : *tennis*, basket, catur, sepak bola, tetris, balap mobil atau motor, dan semua permainan yang sifatnya menghibur. Permainan semacam ini terlihat mudah dari segi grafiknya tetapi biasanya sulit dalam algoritma.

Serious game memiliki beberapa manfaat dibandingkan permainan konvensional pada umumnya. Tipe ini merupakan konsep *game* dengan tujuan untuk kepentingan *training*, *advertising*, simulasi, dan edukasi. Intinya adalah untuk kepentingan yang sifatnya *non-entertainment*. Implementasinya, *serious game* ini bisa diterapkan untuk berbagai tingkatan usia serta dengan berbagai genre dan teknologi *game*. Inti utama dari tujuan *serious game* adalah menumbuhkan, mengedukasi, dan memotivasi pemain untuk satu tujuan tertentu. Tujuan lain bisa juga untuk kepentingan *marketing* dan *advertising*. Di Amerika sendiri, *serious game* ini banyak diterapkan di kalangan pemerintahan serta para profesional kesehatan. Dalam penelitian ini, diharapkan permainan jenis ini dapat membantu siswa dalam memahami pelajaran – pelajaran yang dianggap sulit di sekolah.

Untuk mengetahui kemampuan kognitif seorang siswa, biasanya dilakukan metode pengujian yang konvensional yaitu *paper based*. Secara psikologis, siswa yang mendengar kata ujian atau ulangan menjadi takut. Dengan menggunakan permainan, diharapkan hasil ujian dapat diketahui secara maksimal karena siswa dapat mengerjakannya tanpa perasaan takut atau stres.

Profil siswa dalam belajar di sekolah sangat penting untuk diketahui agar seorang pengajar dapat mengetahui dan mengevaluasi apakah kegiatan belajar mengajar di dalam kelas yang dilakukan telah sesuai dengan yang diharapkan.. Dalam penelitian ini, profil siswa yang dimaksud adalah kemampuan kognitif pembelajar.

Melalui Permainan matematika yang sudah ada, diharapkan dapat membantu siswa mengetahui profil pembelajar seperti apa mereka. Pada penelitian ini, akan dicoba penerapan metode *backpropagation neural network* untuk memperjelas profil pembelajar.

1.2 Perumusan Masalah

Dalam penelitian ini, masalah yang akan diselesaikan, dirumuskan sebagai berikut:

Seorang guru perlu mengetahui kemampuan kognitif siswa agar dapat memberikan materi yang sesuai dengan kemampuan siswa. Pengetahuan akan kemampuan kognitif pada tiap siswa perlu dilakukan guru setiap awal pembelajaran, sehingga seorang guru perlu melakukan serangkaian uji untuk mendapatkan informasi tentang kemampuan kognitif siswanya.

1.3 Batasan Masalah

Pada penelitian ini, hal yang dibatasi adalah siswa yang akan diklasifikasikan merupakan siswa SD kelas 5.

1.4 Tujuan dan Manfaat Penelitian

Tujuan penelitian ini adalah melakukan klasifikasi kemampuan kognitif pembelajar pada permainan matematika yang sudah ada sehingga dapat membantu seorang guru dalam mengetahui kemampuan kognitif siswa tanpa harus melakukan ujian secara konvensional. Penelitian ini dilakukan berbasis metode *backpropagation neural network*. Hasil dalam penelitian ini telah menunjukkan bahwa klasifikasi dapat dilakukan dengan nilai RMSE 0,295.

1.5 Sistematika Penulisan

Agar dapat memahami lebih lanjut tentang penelitian ini, dilakukan dengan cara mengelompokkan materi menjadi beberapa sub bab dengan sistematika penulisan sebagai berikut:

BAB I : PENDAHULUAN

Bab ini menjelaskan tentang informasi umum yaitu latar belakang penelitian, perumusan masalah, batasan masalah, tujuan dan manfaat penelitian, dan sistematika penelitian.

BAB II : LANDASAN TEORI

Bab ini berisikan teori yang diambil dari beberapa kutipan buku, yang berupa pengertian dan definisi. Bab ini juga menjelaskan tentang gaya dan strategi pembelajaran individu, termasuk gaya visual dan gaya verbal. Sub bab kedua berisi tentang fuzzy termasuk struktur dasar sistem fuzzy.

BAB III : METODE PENELITIAN

Bab ini berisikan pembahasan mengenai langkah penelitian. Langkah penelitian dibagi menjadi lima tahap yakni studi literature, perencanaan sistem, ujicoba sistem, analisis hasil dan penyusunan laporan akhir.

BAB IV : HASIL DAN PEMBAHASAN

Bab ini menjelaskan tentang pengujian sistem dan validasi hasil. Pengujian sistem meliputi pengambilan data permainan matematika dan hasil klasifikasi. Validasi hasil meliputi pembahasan hasil perbandingan antara pengujian dengan sistem dan pengujian secara konvensional.

BAB V : Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang berkaitan dengan analisa dan pengujian sistem berdasarkan yang telah diuraikan pada bab-bab sebelumnya.

BAB 2

DASAR TEORI DAN KAJIAN PUSTAKA

Bab 2 berisi pembahasan mengenai dasar-dasar teori dan kajian pustaka yang digunakan dalam penelitian ini. Dasar teori dan kajian pustaka yang dibahas pada bab ini dibagi menjadi dua subbab. Subbab pertama berisi tentang taksonomi bloom. Subbab kedua berisi tentang pedagogi, subbab ketiga berisi tentang jaringan saraf tiruan.

2.1 Taksonomi Bloom

Taksonomi Bloom adalah struktur hierarkhi yang mengidentifikasi *skills* mulai dari tingkat yang rendah hingga yang tinggi (Utari, Retno 2013). Tentunya untuk mencapai tujuan yang lebih tinggi, level yang rendah harus dipenuhi lebih dulu. Dalam kerangka konsep ini, tujuan pendidikan ini oleh Bloom dibagi menjadi tiga ranah kemampuan intelektual (*intellectual behaviors*) yaitu kognitif, afektif dan psikomotorik. Ranah kognitif berisi perilaku yang menekankan aspek intelektual, seperti pengetahuan, dan keterampilan berpikir. Ranah afektif mencakup perilaku terkait dengan emosi, misalnya perasaan, nilai, minat, motivasi, dan sikap. Sedangkan ranah psikomotorik berisi perilaku yang menekankan fungsi manipulatif dan keterampilan motorik / kemampuan fisik, berenang, dan mengoperasikan mesin. Para *trainer* biasanya mengaitkan ketiga ranah ini dengan *Knowledge, Skill and Attitude* (KSA). Kognitif menekankan pada *Knowledge*, Afektif pada *Attitude*, dan Psikomotorik pada *Skill*. Sebenarnya di Indonesia pun, kita memiliki tokoh pendidikan, Ki Hajar Dewantara yang terkenal dengan doktrinnya Cipta, Rasa dan Karsa atau Penalaran, Penghayatan, dan Pengamalan. Cipta dapat diidentikkan dengan ranah kognitif, rasa dengan ranah afektif dan karsa dengan ranah psikomotorik. Ranah kognitif mengurutkan keahlian berpikir sesuai dengan tujuan yang diharapkan. Proses berpikir menggambarkan tahap berpikir yang harus dikuasai oleh siswa agar mampu mengaplikasikan teori kedalam perbuatan. Ranah kognitif ini terdiri atas enam level, yaitu: (1) *knowledge* (pengetahuan), (2) *comprehension* (pemahaman atau persepsi), (3) *application* (penerapan), (4) *analysis* (penguraian atau penjabaran), (5) *synthesis* (pemaduan), dan (6) *evaluation* (penilaian). Jika digambarkan, maka hierarki dari taksonomi bloom pada ranah kognitif adalah sebagai berikut:



Gambar 2.1 Enam Aspek Taksonomi Bloom Ranah Kognitif

2.1.1 Pengetahuan (*Knowledge*)

Berisikan kemampuan untuk mengenali dan mengingat peristilahan, definisi, fakta-fakta, gagasan, pola, urutan, metodologi, prinsip dasar, dsb. Sebagai contoh, ketika diminta menjelaskan manajemen kualitas, orang yang berada di level ini bisa menguraikan dengan baik definisi dari kualitas, karakteristik produk yang berkualitas, standar kualitas minimum untuk produk.

2.1.2 Pemahaman (*Comprehension*)

Berisikan kemampuan mendemonstrasikan fakta dan gagasan mengelompokkan dengan mengorganisir, membandingkan, menerjemahkan, memaknai, memberi deskripsi, dan menyatakan gagasan utama

- Terjemahan
- Pemaknaan
- Ekstrapolasi

Pertanyaan seperti: Membandingkan manfaat mengkonsumsi apel dan jeruk terhadap kesehatan

2.1.3 Penerapan (*Application*)

Di tingkat ini, seseorang memiliki kemampuan untuk menerapkan gagasan, prosedur, metode, rumus, teori, dsb di dalam kondisi kerja. Sebagai contoh, ketika diberi informasi tentang penyebab meningkatnya reject di produksi, seseorang yang berada di tingkat aplikasi akan mampu merangkum dan menggambarkan penyebab turunnya kualitas dalam bentuk fish bone diagram.

2.1.4 Analisis (*Analysis*)

Di tingkat analisis, seseorang akan mampu menganalisis informasi yang masuk dan membagi-bagi atau menstrukturkan informasi ke dalam bagian yang lebih kecil untuk mengenali pola atau hubungannya, dan mampu mengenali serta membedakan faktor penyebab dan akibat dari sebuah skenario yang rumit. Sebagai contoh, di level ini seseorang akan mampu memilah-milah penyebab meningkatnya reject, membandingkan tingkat keparahan dari setiap penyebab, dan menggolongkan setiap penyebab ke dalam tingkat keparahan yang ditimbulkan.

2.1.5 Sintesis (*Synthesis*)

Satu tingkat di atas analisis, seseorang di tingkat sintesa akan mampu menjelaskan struktur atau pola dari sebuah skenario yang sebelumnya tidak terlihat, dan mampu mengenali data atau informasi yang harus didapat untuk menghasilkan solusi yang dibutuhkan. Sebagai contoh, di tingkat ini seorang manajer kualitas mampu memberikan solusi untuk menurunkan tingkat reject di produksi berdasarkan pengamatannya terhadap semua penyebab turunnya kualitas produk.

2.1.6 Evaluasi (*Evaluation*)

Dikenali dari kemampuan untuk memberikan penilaian terhadap solusi, gagasan, metodologi, dengan menggunakan kriteria yang cocok atau standar yang ada untuk memastikan nilai efektivitas atau manfaatnya. Sebagai contoh, di tingkat ini seorang manajer kualitas harus mampu menilai alternatif solusi yang sesuai untuk dijalankan berdasarkan efektivitas, urgensi, nilai manfaat, nilai ekonomis, dan sebagainya (Bloom, B. S. ed. et al. 1956).

2.2 Pedagogi

Pedagogi berasal dari bahasa Yunani yang berarti pergaulan dengan anak-anak. Paedagogos berasal dari kata “paid” yang artinya “anak” dan “agogos” yang artinya “memimpin atau membimbing”. Dari kata ini maka lahir istilah paedagogi yang diartikan sebagai suatu ilmu dan seni dalam mengajar anak-anak. Dalam perkembangan selanjutnya istilah paedagogi berubah menjadi ilmu dan seni mengajar. Paedagogi juga merupakan kajian mengenai pengajaran, khususnya pengajaran dalam pendidikan formal. Dengan kata lain, ia adalah sains dan seni mengenai cara mengajar di sekolah. Secara umumnya pedagogi merupakan mata pelajaran yang wajib bagi mereka yang ingin menjadi guru di sekolah. Sebagai satu bidang kajian yang luas, pedagogi melibatkan kajian mengenai proses pengajaran dan pembelajaran, pengurusan bilik darjah, organisasi sekolah dan juga interaksi guru-pelajar (Sadulloh, Uyoh, dkk. 2011).

Berbagai defenisi teoritis paedagogi telah muncul seperti oleh Danilov (1978). Beliau mendefinisikan istilah paedagogis sebagai proses interaksi terus-menerus dan saling berasimilasi antara pengetahuan ilmiah dan pengembangan siswa. Asimilasi yang dimaksud adalah pengetahuan oleh siswa berkaitan dengan antusiasme mereka untuk mengetahui diverifikasi dalam proses kerja yang intensif dan aktif. Perlu diperhatikan adanya penekanan pada aspek pengajaran terus-menerus dari proses asimilasi yang merupakan upaya intelektual yang intensif pada diri siswa. Menjaga proses pendidikan dan pengajaran secara keseluruhan dan bermuara pada pembentukan kepribadian siswa adalah fungsi dari paedagogi.

Menurut Prof. Dr. J. Hoogveld (Belanda) pedagogik adalah ilmu yang mempelajari masalah membimbing anak ke arah tujuan tertentu, yaitu supaya ia kelak “mampu secara mandiri menyelesaikan tugas hidupnya”. Jadi pedagogik adalah ilmu untuk mendidik anak.

Langeveld (1980), membedakan istilah “pedagogik“ dengan istilah “ pedagogi”. Pedagogik diartikan dengan ilmu mendidik, lebih menitik beratkan kepada pemikiran, perenungan tentang pendidikan. Suatu pemikiran bagaimana kita membimbing anak, dan mendididk anak. Sedangkan istilah pedagogi berarti pendidikan, yang lebih menekankan kepada praktik, menyangkut kegiatan mendidik, kegiatan membimbing anak.

Menurut Addine (2001), di antara prinsip-prinsip paedagogis terdapat kesatuan karakter ilmiah dan ideologis dari proses paedagogis. Karakter ilmiah dan ideologis ini menyoroti bahwa setiap proses paedagogis harus terstruktur berdasarkan temuan yang

paling maju di bidang sains kontemporer dan dalam korespondensi total dengan ideologi kita. Selain itu, prinsip hubungan sekolah dan kehidupan didasarkan pada dua aspek penting: kaitan antara kehidupan dan pekerjaan sebagai kegiatan yang mendidik manusia. Prinsip lain yang berorientasi pada proses tersebut adalah salah satu yang mengombinasikan karakter kolektif dan individual pendidikan, serta penghormatan terhadap kepribadian siswa. Prinsip berikutnya merujuk pada kesatuan pengajaran, pendidikan dan perkembangan proses, karena didasarkan pada kesatuan dialektis antara pendidikan dan pengajaran yang harus terkait dengan kegiatan pembangunan pada umumnya. Prinsip terakhir dari proses paedagogis adalah bahwa masing-masing subsistem aktivitas, komunikasi, dan kepribadian saling terkait satu sama lain.

Paedagogi juga kadang-kadang merujuk pada penggunaan yang tepat dari strategi mengajar. Sehubungan dengan strategi mengajar itu, filosofi mengajar diterapkan dan dipengaruhi oleh latar belakang pengetahuan dan pengalamannya, situasi pribadi, lingkungan, serta tujuan pembelajaran yang dirumuskan oleh peserta didik dan guru. Salah satu contohnya adalah aliran pemikiran Socrates. Sumbangsih Socrates yang terpenting bagi pemikiran barat adalah metode penyelidikannya, yang dikenal sebagai metode elenchos, yang banyak diterapkan untuk menguji konsep moral yang pokok. Karena itu, Socrates dikenal sebagai bapak dan sumber etika atau filsafat moral, dan juga filsafat secara umum. Kata yang berhubungan dengan paedagogi, yaitu pendidikan, sekarang digunakan untuk merujuk pada keseluruhan konteks pembelajaran, belajar, dan berbagai kegiatan yang berhubungan dengan hal tersebut.

Pendidikan adalah usaha sadar dan terencana untuk mewujudkan suasana belajar dan proses pembelajaran agar peserta didik secara aktif mengembangkan potensi dirinya untuk memiliki kekuatan spiritual keagamaan, pengendalian diri, kepribadian, kecerdasan, akhlak mulia, serta keterampilan yang diperlukan dirinya sendiri dan masyarakat.

Pendidikan dalam arti luas merupakan usaha manusia untuk meningkatkan kesejahteraan hidupnya, yang berlangsung sepanjang hayat. Menurut Henderson, pendidikan merupakan suatu proses pertumbuhan dan perkembangan, sebagai hasil interaksi individu dengan lingkungan sosial dan lingkungan fisik, berlangsung sepanjang hayat sejak manusia lahir. Warisan sosial merupakan bagian dari lingkungan masyarakat, merupakan alat bagi manusia untuk pengembangan manusia yang terbaik dan inteligen, untuk meningkatkan kesejahteraan hidupnya (Usman, Moh. Uzer, 1994).

Penilaian aspek kognitif biasanya dilakukan oleh seorang guru dengan cara mengambil nilai saat Ulangan Harian, baik Ulangan Harian 1, 2, ataupun 3 sesuai dengan kebutuhan dan kebijakan masing – masing institusi pendidikan. Gambar 2.2 merupakan contoh nilai raport bayangan siswa SMA, nilai kognitif diambil dari Ulangan Harian 1 dan Ulangan Harian 2.

**LEGER NILAI RAPORT BAYANGAN
SMA PELITA TIGA NO. 3 JAKARTA
TAHUN 2011 / 2012**

NO	NIS	NAMA SISWA	KELAS	BIDANG STUDI							
				BAHASA INDONESIA				MATEMATIKA			
				NILAI MID				NILAI MID			
				NILAI KOGNITIF		PRAKTIK	AFEKTIF	NILAI KOGNITIF		PRAKTIK	AFEKTIF
UH1	UH2	UH1	UH2								
1	4601	AMRAN RABBANI ZUBAIDI	X.D	60	44	61	3	45	46	-	3
2	4602	ANDRO BRAMANTIO	X.D	61	50	60	3	58	34	-	3
3	4603	ARIEF GHAZALI ANWAR	X.D	66	62	64	3	60	46	-	3
4	4604	BUNGA AYU ANGGIA	X.D	71	76	70	4	66	62	-	4
5	4605	DEFIKA RIALLISTYANTI	X.D	65	64	65	3	60	38	-	3

Gambar 2.2 Contoh penilaian kognitif siswa

2.3 Jaring Saraf Tiruan

Jaring saraf tiruan (JST) adalah salah satu teknik klasifikasi yang akurat dan banyak digunakan pada aplikasi di berbagai sektor. Beberapa karakter yang berbeda dari JST membuat teknik ini menjadi pengklasifikasi yang baik.

Pertama, JST adalah model yang beradaptasi dengan karakteristik data. Kedua, setelah melakukan pembelajaran pada data, JST sering memprediksi kelas dengan tepat, bahkan jika data sampel mengandung gangguan informasi. Ketiga, JST mendekati fungsi universal. Telah terbukti bahwa jaringan dapat memperkirakan setiap fungsi kontinu untuk mencapai akurasi yang diinginkan. Terakhir, JST bersifat non linier (Khashei, Hejazi, & Bijari, 2008).

JST merupakan pengembangan sistem komputer yang meniru operasi otak manusia. JST disusun oleh neuron yang merupakan elemen pemrosesan dalam jaringan. Setiap pola-pola informasi input dan output yang diberikan ke dalam JST diproses dalam neuron. Neuron-neuron tersebut dikelompokkan pada lapisan-lapisan.

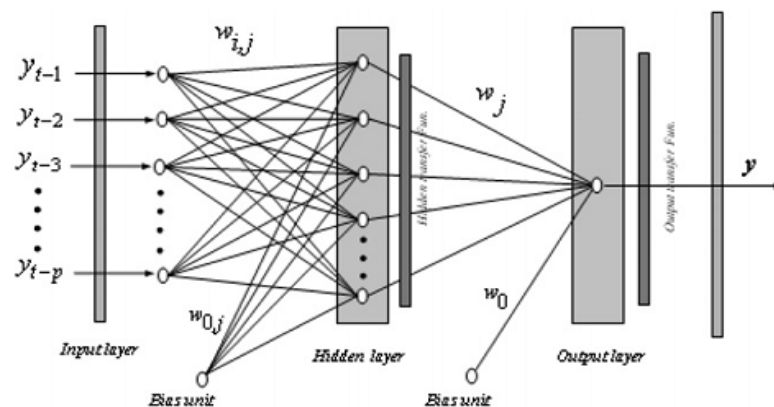
Terdapat tiga jenis lapisan yaitu lapisan input, lapisan tersembunyi dan lapisan output. Lapisan input memuat unit-unit yang menerima pola inputan dari luar. Pola

inputan ini menggambarkan permasalahan. Lapisan tersembunyi menghasilkan output yang tidak dapat secara langsung diamati. Lapisan output menghasilkan output yang berbentuk hasil prediksi dari data yang ada.

Selain itu, ada bobot sebagai tempat menyimpan informasi yang telah dilatih. Bobot yang optimal memungkinkan sistem dapat menghasilkan keluaran yang diinginkan (Khashei, Hamadani, & Bijari, 2011).

JST dengan satu lapisan tersembunyi adalah bentuk yang paling banyak digunakan untuk pemodelan, peramalan dan klasifikasi. Model ini terdiri dari tiga lapisan unit pemroses sederhana yang dihubungkan dengan *link* tak bersiklus (Khashei, Hamadani, & Bijari, 2011).

Untuk semua neuron pada jaringan, kecuali neuron pada lapisan input, total input yang masuk adalah jumlah dari bobot output yang dikalikan dengan neuron pada lapisan sebelumnya. Tiap input yang masuk ke neuron kemudian diaktivasi dengan fungsi aktivasi (Amini, 2008). Gambar 2.2 menjelaskan struktur JST dengan satu lapisan tersembunyi.



Gambar 2.3 Struktur JST dengan satu lapisan tersembunyi (Khashei, Hamadani, & Bijari, 2011).

Input dan output pada JST memiliki persamaan sebagai berikut:

$$\text{Input} \quad X_i = \sum_{j=0}^p w_{i,j} O_j + b_i \quad (2.1)$$

$$\text{Output} \quad O_j = f(X_i) \quad (2.2)$$

w_{ij} adalah bobot koneksi antara node i ke node j , O_j adalah nilai neuron, b_i adalah nilai numerik yang disebut bias dan f adalah fungsi aktivasi. Tahap perhitungan di atas merupakan tahap *feedforward* yang pada akhirnya menghasilkan output prediksi dari lapisan output.

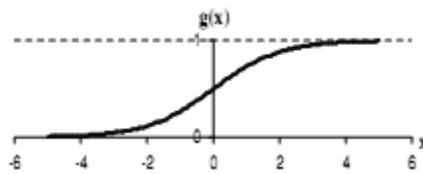
2.3.1 Fungsi aktivasi

Fungsi aktivasi adalah fungsi yang digunakan untuk memproses input menjadi output yang diinginkan. Penggunaan fungsi aktivasi ini bergantung pada kebutuhan dan output yang diinginkan. Fungsi logistik dan hiperbolik sering digunakan pada lapisan tersembunyi dan lapisan output (Khashei, Hamadani, & Bijari, 2011). Kedua fungsi tersebut ditunjukkan pada Persamaan (2.3) dan (2.4).

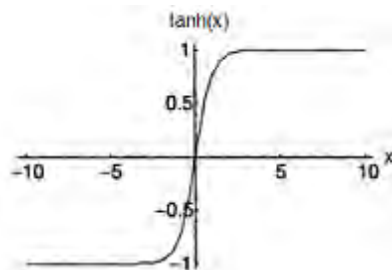
$$Sig(x) = \frac{1}{1 + exp^{-x}} \quad (2.3)$$

$$Tanh(x) = \frac{1 - exp^{-2x}}{1 + exp^{-2x}} \quad (2.4)$$

x adalah nilai keluaran komputasi bobot dengan lapisan sebelumnya sedangkan exp adalah bilangan natural. Gambar 2.3 dan Gambar 2.4 masing-masing menunjukkan grafik fungsi logistik dan hiperbolik.



Gambar 2.4 Fungsi sigmoid logistik (Karlik & Olgac, 2011).



Gambar 2.5 Fungsi tangen hiperbolik (Karlik & Olgac, 2011).

Pada model hibrida, fungsi yang digunakan dalam struktur JST adalah fungsi logistik baik di lapisan tersembunyi maupun di lapisan output.

2.3.2 Fungsi biaya

Setelah struktur jaringan ditentukan, jaringan tersebut siap untuk dilatih. Hal ini dilakukan agar fungsi biaya dari JST dapat diminimalkan. Fungsi biaya adalah ukuran akurasi secara keseluruhan seperti rata-rata *error* kuadrat (Khashei, Hamadani, & Bijari, 2011) berikut:

$$E = \frac{1}{N} \sum_{n=1}^N (e_i)^2 \quad (2.5)$$

N adalah jumlah term *error* dan e adalah selisih *error* nilai keluaran yang diharapkan dan hasil output prediksi.

2.3.3 Algoritma Backpropagation

Pada dasarnya, algoritma *backpropagation* terdiri dari dua tahap, yaitu *feed forward* dan *backpropagation*. Pada tahap *feedforward*, sinyal input diberikan ke dalam jaringan. Komputasi dilakukan oleh neuron pada setiap lapisan sehingga menghasilkan output. Pada tahap ini, bobot pada jaringan tidak mengalami perubahan. Sebaliknya, pada tahap *backpropagation*, semua bobot diperbaiki berdasarkan *error* jaringan. Bobot diperbaiki untuk membuat output yang keluar semakin mendekati output yang diinginkan (Khashei, Hamadani, & Bijari, 2011).

Minimisasi fungsi biaya dilakukan dengan mengubah $w_{i,j}$, dengan sejumlah $\Delta w_{i,j}$, menurut persamaan berikut (Khashei, Hamadani, & Bijari, 2011):

$$\Delta w_{i,j} = \eta \frac{\partial E}{\partial w_{i,j}} \quad (2.6)$$

Parameter η adalah kecepatan *learning* yang berkisar antara 0 sampai 1 dan $\partial E / \partial w_{i,j}$ adalah turunan parsial dari fungsi E dari Persamaan 2.6 terhadap bobot $w_{i,j}$.

Turunan ini dihitung dalam tahap *feed forward* dan *backpropagation*. Di *feedforward*, vektor input dari *training* set diterapkan untuk unit input jaringan dan dilewatkan dari lapisan ke lapisan sehingga menghasilkan output akhir. Selama *backpropagation*, output dari jaringan dibandingkan dengan output yang diinginkan untuk mendapatkan nilai *error*. *Error* yang dihasilkan digunakan untuk mengubah bobot (Mitchel, 1997).

Untuk mempercepat proses *learning*, sekaligus menghindari ketidakstabilan algoritma, perlu adanya momentum α , sehingga aturan *learning* memiliki persamaan sebagai berikut (Khashei, Hamadani, & Bijari, 2011):

$$\Delta w_{i,j}(t+1) = \eta \frac{\partial E}{\partial w_{i,j}} + \alpha \Delta w_{i,j}(t) \quad (2.7)$$

(t) menunjukkan fungsi waktu, sehingga $\Delta w_{i,j}(t)$ adalah nilai bobot saat ini sedangkan $\Delta w_{i,j}(t+1)$ adalah lambang untuk *update* bobot, yaitu nilai bobot selanjutnya. Momentum juga dapat membantu untuk mencegah proses *learning* agar tidak terjebak ke dalam kondisi lokal minima. Nilai momentum biasanya dipilih pada interval [0, 1] (Khashei, Hamadani, & Bijari, 2011). Akhirnya, model estimasi dievaluasi menggunakan sampel terpisah yang tidak termasuk dalam data *training*.

Gambar 2.5 (Mitchel, 1997) menjelaskan *pseudocode* algoritma *backpropagation*.

- Buat sebuah network dengan n_{in} unit input, n_{hidden} unit tersembunyi dan n_{out} unit output.
 - Inisialisasi semua bobot dan bias ke angka random yang kecil (antara -0.05 dan 0.05)
 - Sampai kondisi terminasi dicapai, Lakukan
- Untuk tiap data *training*, Lakukan
1. Masukkan data *training* sebagai input dan hitung output jaringan O_u
 2. Untuk tiap unit output k, hitung term *error* δ_k

$$\delta_k \leftarrow o_k (1-o_k) (t_k-o_k)$$
 3. Untuk tiap unit tersembunyi hitung term *error* δ_h

$$\delta_h \leftarrow o_h (1-o_h) \sum_{k \in output} w_{h,k} \delta_k$$
 4. Ubah tiap bobot dan bias $w_{i,j}$

$$W_{i,j} \leftarrow w_{i,j} + \Delta w_{i,j} \text{ dimana}$$

$$\Delta w_{i,j}(n) \leftarrow \eta \delta_j x_{i,j} + \alpha \Delta w_{i,j}(n-1)$$

Gambar 2.6 *Pseudocode* algoritma *backpropagation*.

Keterangan:

- δ_k adalah *error* output
- o_k adalah keluaran dari lapisan output
- t_k adalah output yang diharapkan
- δ_h adalah *error* pada lapisan tersembunyi
- o_h adalah keluaran dari lapisan tersembunyi
- w_{kh} adalah bobot antara lapisan ouput dan lapisan tersembunyi
- $\Delta w_{i,j}$ adalah selisih bobot saat ini dengan bobot sebelumnya
- $W_{i,j}$ adalah semua bobot dan bias pada jaringan saraf
- η adalah *learning rate*
- $x_{i,j}$ adalah keluaran dari lapisan
- α adalah momentum

2.4 K-Fold Cross Validation

K-fold cross-validation adalah teknik yang menggunakan semua data yang tersedia sebagai *training* dan testing (Bengio & Grandvalet, 2004). Teknik ini dilakukan untuk mengestimasi kinerja dari model *learning*. Selain itu, teknik ini juga dapat membandingkan kinerja dua algoritma yang berbeda atau lebih untuk menemukan algoritma yang terbaik pada data yang tersedia (Refaeilzadeh, Tang, & Liu, 2008).

Kelebihan dari metode ini adalah semua bagian dari data pernah digunakan sebagai data *training* dan testing sehingga setiap bagian mendapatkan perlakuan yang sama (Russell & Norvig, 2002).

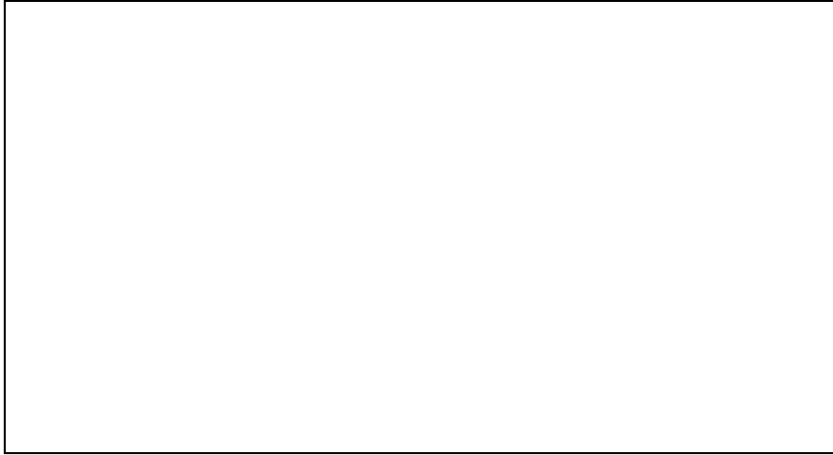
Uji coba pada model hibrida dilakukan dengan memakai metode *k-fold cross validation* dengan empat *fold*. Masukkan data input, kemudian bagi ke dalam empat sub atau bagian.

```
Input: data S, integer k
Prosedur:
partisi S ke k subset berukuran sama  $S_1, \dots, S_k$ 
for i = 1 to i = k
    set T = S tanpa  $S_i$ 
    jalankan algoritma learning dengan T sebagai training set
    testing hasil klasifier pada  $S_i$ 
```

Gambar 2.7 Pseudocode *k-fold cross validation*.

Tiap sub diberi indeks antara angka satu sampai empat. Sub data ke-1 bertindak sebagai data testing sedangkan sisanya adalah data *training*. Langkah lebih lengkap tentang *k-fold cross validation* dapat dilihat pada *pseudocode* (Bengio & Grandvalet, 2004) di Gambar 2.6.

Setelah itu, tahap *training* dilakukan untuk membangun model, dilanjutkan testing untuk pengujian model dengan menghitung nilai *error*. *Error* tersebut disimpan sebagai *error* pada *fold* ke-1. Gambar 2.7 menunjukkan ilustrasi *k-fold cross validation* dengan $k = 4$.



Gambar 2.8 Ilustrasi data dengan *4-fold cross validation*.

Selanjutnya dilakukan tahap *training* dan testing lagi dengan sub data ke-2 bertindak sebagai data testing, sedangkan sisanya adalah data *training*. Demikian seterusnya sampai tahap *training* dan testing dilakukan sampai empat kali dan semua data pernah merasakan sebagai data *training* dan data testing.

Error yang tersimpan adalah sebanyak empat sesuai dengan jumlah *fold*-nya. Untuk mendapatkan *error* secara keseluruhan pada model, dilakukan perhitungan rata-rata *error* dari *error* pada semua *fold*. Rata-rata *error* tersebut mewakili *error* atau tingkat kesalahan prediksi pada model.

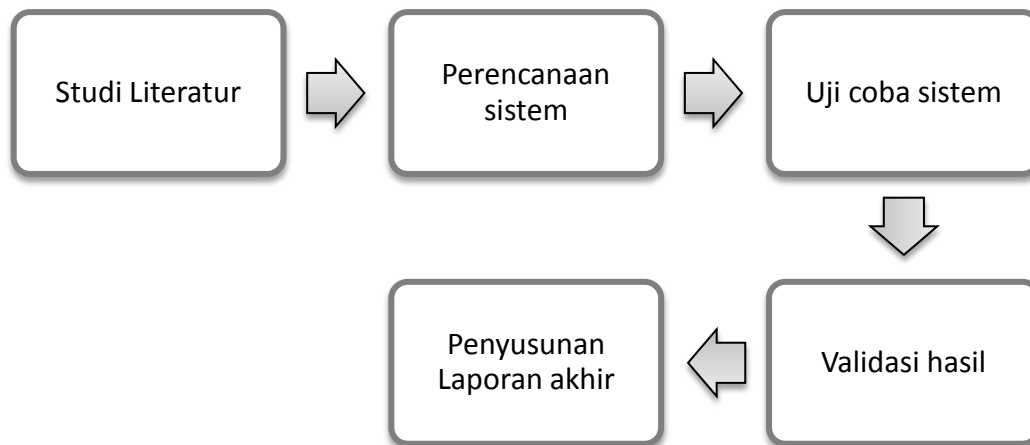
BAB 3

METODE PENELITIAN

Bab 3 berisi pembahasan mengenai langkah penelitian. Langkah penelitian dibagi menjadi lima tahap yakni studi literatur, perencanaan sistem, uji coba sistem, analisis hasil dan penyusunan laporan akhir.

3.1. Langkah-langkah Penelitian

Pada penelitian ini, terdapat beberapa tahapan penyelesaian yang akan dilakukan. Tahapan tersebut dijelaskan pada Gambar 3.1 dan penjelasannya akan dijelaskan pada subbab selanjutnya..



Gambar 3.1 Diagram Alur Kerja Langkah Penelitian

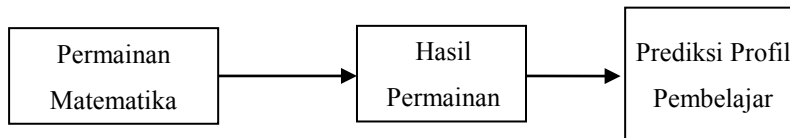
3.2 Studi Literatur

Proses ini meliputi pencarian referensi-referensi pendukung yang sesuai dan dapat dipertanggungjawabkan, baik dari buku, jurnal, maupun artikel. Informasi ini nantinya berguna dalam mendesain algoritma untuk memecahkan permasalahan yang sedang diteliti. Setelah studi literatur dilakukan, pendefinisian masalah serta pendefinisian dataset dapat diketahui lebih detail. Proses tersebut dilanjutkan dengan pencarian dataset yang tersedia di internet.

3.3 Perencanaan Sistem

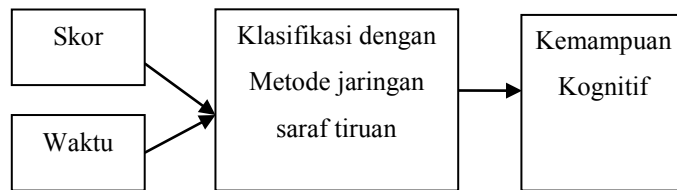
Pada penelitian kali ini, soal yang menjadi masukan berasal dari permainan matematika yang sudah ada. Berdasarkan tingkat kesulitan dan *taksonomi bloom* soal dibedakan menjadi sembilan jenis.

Dalam perencanaan sistem ini, dibuat diagram perencanaan sistem untuk menjelaskan sistem kerja yang akan dilakukan.



Gambar 3.2. Blok diagram perencanaan sistem

Gambar 3.2 menjelaskan bahwa sistem akan dimulai dari Permainan matematika. Pada permainan ini, ada sembilan soal yang harus dijawab tiap pembelajar. Hasil jawaban beserta waktunya akan disimpan untuk menentukan profil pembelajar. *User* melakukan beberapa kali pengujian sehingga didapatkan pula hasil profil pembelajar yang telah diolah dengan metode jaringan saraf tiruan.



Gambar 3.3. Masukan, Proses dan Keluaran Pengolahan Data

Gambar 3.3 menunjukkan ada dua masukan yang dapat dijadikan bahan pengolahan untuk kemudian dikeluarkan. Sebelum data diolah dengan menggunakan metode jaringan saraf tiruan, berikut tabel jenis soal yang akan menjadi masukan dari sistem sebagai keluaran.

Tabel 3.1. Jenis soal pada permainan

Tk. Kesukaran	Mudah (L)	Sedang (M)	Sukar (S)
Taksonomi Bloom			
C1 (Pengetahuan)	C1L	C1M	C1H
C2 (Pemahaman)	C2L	C2M	C2H
C3 (Aplikasi)	C3L	C3M	C3H

Tabel 3.1 menunjukkan urutan – urutan cara soal pertama sampai kesembilan ditampilkan dalam permainan. Soal yang pertama kali muncul adalah C1M yang berarti soal berjenis pengetahuan dengan tingkat kesulitan sedang. Jika pemain menjawab benar, maka level kesulitan akan meningkat sehingga soal yang muncul kemudian adalah C1H, soal berjenis pengetahuan dengan tingkat kesulitan sukar. Jika pemain menjawab soal C1M salah, maka level kesulitan akan diturunkan sehingga soal yang muncul kemudian adalah C1L, soal berjenis pengetahuan dengan tingkat kesulitan mudah. Jika pemain menjawab semua soal dengan benar, maka urutan keluarnya soal adalah C1M, C1H, C1H, C2H, C2H, C2H, C3H, C3H, dan C3H. Dari sembilan jenis soal ini diharapkan kemampuan kognitif siswa dapat diketahui.

3.3.1 Permainan Matematika

Permainan matematika ini berjudul Cupak dan Grantang yang merupakan pertunjukan teater rakyat tradisional khas Bali dan Lombok yang sederhana. Alur cerita dibuat sedemikian rupa sehingga berpadu dengan unsur budaya Jawa. Cerita bermula saat Sang Putri Kediri diculik seorang Raksasa dan dibawa ke hutan. Kemudian Sang Raja Kediri mengadakan sayembara bagi seluruh penduduknya. Barang siapa yang dapat mengalahkan raksasa dan membawa kemabali Tuan Putri, maka hadiah yang besar pun akan menanti. Grantang yang ditemani saudaranya, yaitu Cupak berniat untuk menyelamatkan Tuan Putri. Dimulai dari pertemuan dengan Sang Raja Kediri hingga akhirnya bertemu dengan raksasa, Grantang akan diuji dengan berbagai macam soal. Ketika telah sampai di akhir permainan saat Grantang bertemu dengan raksasa, Tuan Putri dibawa Cupak untuk menuju ke istana sedangkan Grantang yang bertarung melawan raksasa. Disinilah Cupak akan mengkhianati Grantang dengan mengatakan bahwa dirinyalah yang mengalahkan raksasa dan menyelamatkan Tuan Putri. Pesan moral akan diberikan di akhir permainan bahwa sebagai seorang manusia, kita tidak boleh bertindak curang demi mendapatkan sesuatu. Karakter Grantang mengajarkan bahwa seseorang yang rendah hati dan pemaaf akan mulia di kemudian hari.



Gambar 3.4 Tes *server* apakah *online* atau tidak

Pertama kali ketika akan memulai permainan, akan dicek terlebih dahulu apakah *servers* sudah siap atau belum seperti pada Gambar 3.4. Jika sudah ada keterangan *online* maka permainan telah siap. Namun, jika masih ada keterangan *offline* berarti server harus diaktifkan terlebih dahulu. Berikut tampilan gambar awal saat permainan dimulai.



Gambar 3.5 Pemain diminta menuliskan namanya

Saat *server* telah *online*, maka pemain harus menuliskan nama sebagai identitas pemain seperti pada Gambar 3.5. Data ini kemudian akan disimpan untuk selanjutnya dianalisa. Permainan siap dilanjutkan, akan ada beberapa pertanyaan yang diberikan agar Cupak dan Grantang dapat menyelamatkan Tuan Putri. Bentuk soal yang akan tampil seperti pada Gambar 3.6.

Dalam permainan ini, pemain akan diberikan nilai jika menjawab benar dalam setiap soal. Nilai dan lama menjawab soal (waktu) pada permainan ini yang akan digunakan dalam masukan sistem untuk kemudian dapat menghasilkan profil siswa.

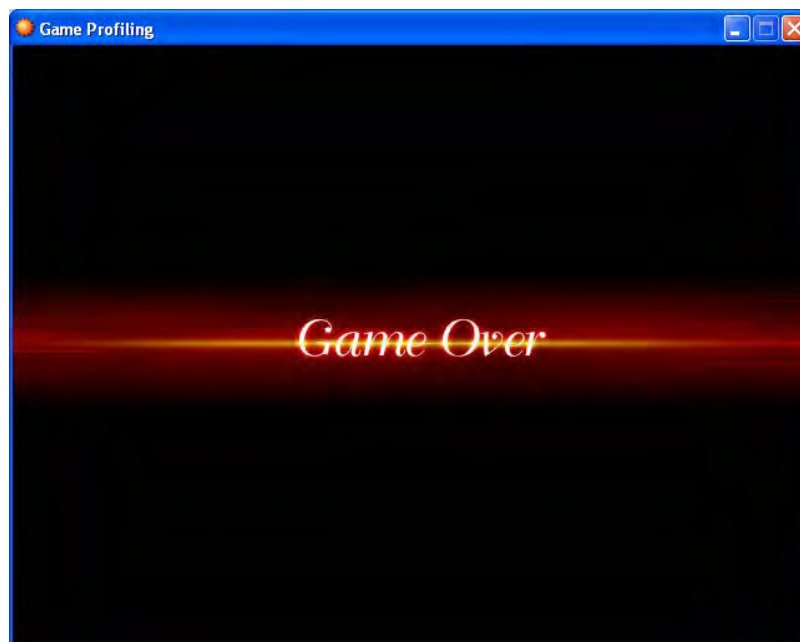


Gambar 3.6 Contoh soal yang akan muncul dalam permainan

Ketika soal yang dijawab benar ataupun salah, maka skor akan disimpan dalam *database* untuk keperluan analisa lebih lanjut. Tidak hanya skor yang dicatat tetapi juga lama menjawab soal akan dianalisa. Ada beberapa hadiah yang didapat ketika soal yang dijawab benar seperti pada Gambar 3.7.



Gambar 3.7 Salah satu hadiah yang didapatkan ketika menjawab soal dengan benar



Gambar 3.8 Permainan akan selesai ketika salah menjawab soal sebanyak tiga kali

Jika sekali melakukan kesalahan dalam menjawab soal, masih diberi toleransi, begitu juga ndengan kesalahan kedua. Nmun, saat tiga kali salah menjawab soal, maka permainan ini selesai.

3.3.2 Hasil Permainan

Output data *game* dijelaskan pada Gambar 3.9. Pada Gambar 3.9, terdapat 18 variabel yang dapat dibagi menjadi dua jenis yakni sembilan variabel soal dan sembilan variabel waktu.

1	username	soal1	soal2	soal3	soal4	soal5	soal6	soal7	soal8	soal9	waktu1	waktu2	waktu3	waktu4	waktu5	waktu6	waktu7	waktu8	waktu9
2	arman1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	arman2	2	0	2	3	0	0	0	0	0	25	15	33	41	85	29	0	0	0
4	ibra1	2	0	2	0	0	1	0	0	0	50	49	29	31	14	37	37	0	0
5	yulhaq	0	1	0	0	0	0	0	0	0	2	3	3	0	0	0	0	0	0
6	yulhaq1	0	0	1	0	0	0	0	0	0	5	1	1	2	10	9	0	0	0
7	deni	2	0	2	3	0	2	0	0	0	22	23	12	16	17	9	20	33	0
8	deni2	2	3	3	3	0	0	0	0	0	3	16	30	16	32	7	0	0	0
9	deni3	2	0	2	0	0	1	0	0	0	10	13	10	7	12	7	8	0	0
10	almas1	0	1	0	0	0	0	0	0	0	14	16	31	0	0	0	0	0	0
11	almas2	0	0	1	2	0	2	3	0	0	3	10	13	13	21	13	33	25	22
12	tajul2	2	3	0	0	0	0	0	0	0	15	6	12	0	0	0	0	0	0
13	fad1	0	0	1	0	0	0	0	0	0	4	3	4	3	5	4	0	0	0
14	wikki1	?	n	n	n	n	n	n	n	n	4	35	13	n	n	n	n	n	n

Gambar 3.9. Output data game

Sembilan variabel soal (soal1, soal2 dan seterusnya) menunjukkan siswa menjawab soal dengan benar atau tidak. Ketika menghasilkan angka 1, maka siswa menjawab benar di level low. Sedangkan ketika menghasilkan angka 2, maka siswa menjawab benar di level medium. Ketika menghasilkan angka 3, maka siswa menjawab benar di level high. Jika yang muncul adalah angka 0, artinya siswa menjawab tidak benar.

Sembilan variabel waktu menunjukkan lama siswa menjawab masing-masing soal. Variabel waktu1 menunjukkan lama siswa mengerjakan soal 1, variabel waktu2 menunjukkan lama siswa mengerjakan soal ke-2 dan seterusnya.

3.3.3 Prediksi Profil Pembelajaran

Setelah hasil permainan didapatkan, langkah selanjutnya adalah prediksi profil pembelajar. Keluaran dari proses ini adalah data prediksi profil kognitif siswa yang berupa data ordinal dengan skala 0 – 4. Proses prediksi ini dikerjakan dengan menggunakan algoritma jaringan saraf tiruan.

3.4 Implementasi Sistem

Sistem ini diimplementasikan dengan mengambil data di lokasi yang telah ditentukan. Lokasi pengambilan data permainan matematika dilakukan di MI Ma'arif NU As Saadah Gresik. Pengujian dilakukan terhadap anak kelas 5 SD sebanyak tiga kelas.

3.5 Validasi Hasil

Validasi bertujuan untuk mengetahui hasil prediksi apakah sesuai dengan kondisi siswa sebenarnya dengan membandingkan hasil prediksi dan tes tulis. Output tes tulis adalah integer ordinal yakni 0, 1, 2, 3 dan 4. Semakin tinggi nilai output tes tulis siswa menunjukkan semakin tinggi pula kemampuan kognitifnya.

3.6 Penyusunan Laporan Akhir

Pada tahap ini dilakukan pendokumentasian dan laporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan dan hasil-hasil yang telah didapatkan selama pengerjaan penelitian. Laporan akhir yang akan disusun bertujuan untuk memberikan gambaran dari pengerjaan penelitian dan diharapkan dapat berguna sebagai literatur bagi pembaca yang tertarik melakukan pengembangan lebih mendalam.

3.7 Perancangan Data

Perancangan data merupakan hal penting untuk diperhatikan. Dibutuhkan data yang tepat agar perangkat lunak dapat beroperasi secara benar. Lokasi pengambilan data permainan matematika dilakukan di MI Ma'arif NU As Saadah Gresik. Perancangan dimulai dari topik matematika yang digunakan untuk soal dalam *game*, teknik penilaian (*scoring*), skenario permainan, pengujian *game*, dan melakukan tes tulis. Pengambilan data dilakukan melalui dua tahap pengujian. Pengujian pertama melalui uji coba *game*. Pengujian kedua dilakukan tes tulis untuk menguji kognitif siswa. Pengujian dilakukan terhadap anak kelas 5 SD sebanyak tiga kelas.

3.7.1 Topik Matematika

Dalam pembuatan soal matematika pada permainan yang digunakan dipenelitian ini melibatkan tim olimpiade matematika tingkat sekolah dasar di Bali. Soal yang akan diacak terdiri dari 45 soal dengan tingkat kesulitan mudah, sedang, dan sukar. Soal juga dibedakan menurut tingkat *taksonomi bloom* nya, tingkat pengetahuan, pemahaman, dan aplikasi. Soal yang disediakan disesuaikan dengan kurikulum untuk siswa SD kelas 5 dengan pembahasan mengenai bangun dua dimensi jajar genjang. Gambar 3.10 merupakan tampilan contoh soal yang akan keluar secara acak pada permainan.

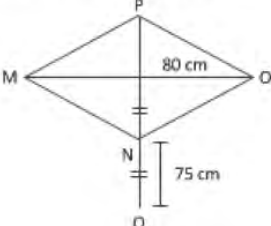
1. Sebuah segiempat merupakan jajargenjang jika memiliki

- Sepasang sisi yang berdekatan sama panjang
- Sepasang sisi yang sejajar
- Sepasang sisi yang sama panjang
- Dua pasang sisi yang berhadapan sejajar

2. Sebuah segiempat merupakan jajargenjang jika

- Dua pasang sudut yang berhadapan kongruen
- Sepasang sudut yang berdekatan jumlahnya 90°
- Sepasang sudut yang berdekatan sama besar
- Sudut-dudut yang berdekatan saling tegak lurus

21. MNOP adalah jajargenjang. Panjang \overline{PQ} adalah



- 75 cm
- 80 cm
- 150 cm
- 225 cm

22. Jajargenjang dapat dibangun dari segiempat dengan dua pasang sudut berhadapan yang masing-masing besarnya

- 40° dan 50°
- 65° dan 45°
- 75° dan 60°
- 35° dan 145°

Gambar 3.10 Bank soal pada permainan matematika

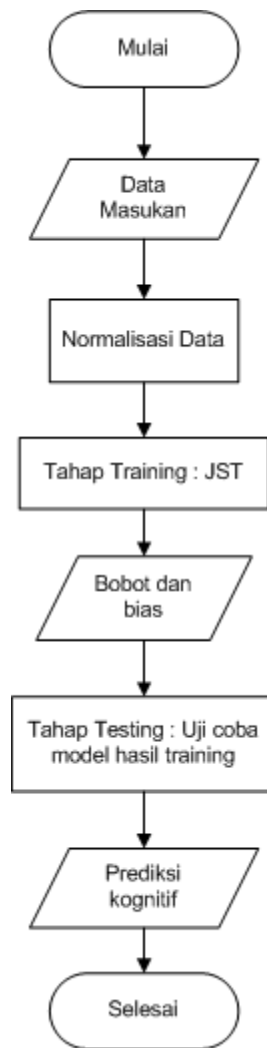
3.7.2 Teknik Penilaian

Untuk dapat mengetahui kemampuan kognitif seorang siswa, penelitian ini menggunakan dua variable masukan, yaitu skor dan waktu. Teknik penilaian dalam permainan ini adalah sebagai berikut:

- Jika siswa menjawab benar dalam soal bertipe mudah, akan diberikan nilai satu.
- Jika siswa menjawab benar dalam soal bertipe sedang, akan diberikan nilai dua.
- Jika siswa menjawab benar dalam soal bertipe sukar, akan diberikan nilai tiga.
- Jika siswa menjawab salah dalam setiap jenis soal, maka nilai sama dengan nol dan tingkat kesulitan soal akan diturunkan.

3.8 Perancangan Metode

Perancangan metode dilakukan untuk mengetahui alur dalam penerapan algoritma yang nantinya dipakai.



Gambar 3.11 Perancangan perangkat lunak model prediksi kognitif secara umum

3.8.1 Desain Secara Umum

Gambar 3.11 menjelaskan perancangan perangkat lunak model prediksi secara umum. Perangkat lunak dimulai dengan tahap normalisasi data dilanjutkan tahap *training* dan testing yang akan menghasilkan prediksi kognitif siswa. Hasil normalisasi tersebut digunakan pada tahap proses, yang terdiri dari tahap *training* dan testing.

Model prediksi menerapkan pengujian dengan metode *k-fold cross validation*. Sebelum dilakukan *k-fold cross validation*, data diacak pada indeks barisnya. Kemudian dilanjutkan dengan pembagian data untuk *training* dan testing sesuai jumlah *fold* pada *cross validation* yang telah ditetapkan. *Training* digunakan untuk membangun model

JST. Setelah itu, dilakukan testing terhadap data selain data *training* untuk menunjukkan keakuratan model yang telah dibuat.

3.8.2 Tahap Normalisasi

Data memiliki rentang data yang sangat luas. Perlu ada penyesuaian nilai rentang data dengan fungsi aktivasi pada sistem *backpropagation*. Hal ini karena fungsi aktivasi selalu mengeluarkan nilai dalam rentang 0 hingga 1. Penyesuaian ini bertujuan memudahkan mengenali pola data pada saat *training* dan meningkatkan akurasi dari hasil output pada proses *learning* JST. Oleh karena itu, sebelum digunakan sebagai input pada sistem, maka data tersebut dinormalisasi menggunakan persamaan berikut:

$$input = \frac{data - \min(data)}{\max(data) - \min(data)} \quad (3.1)$$

3.8.3 Tahap Proses

Proses prediksi kognitif dibagi menjadi 2 tahap, yaitu tahap *training* dan tahap testing.

a. Tahap Training

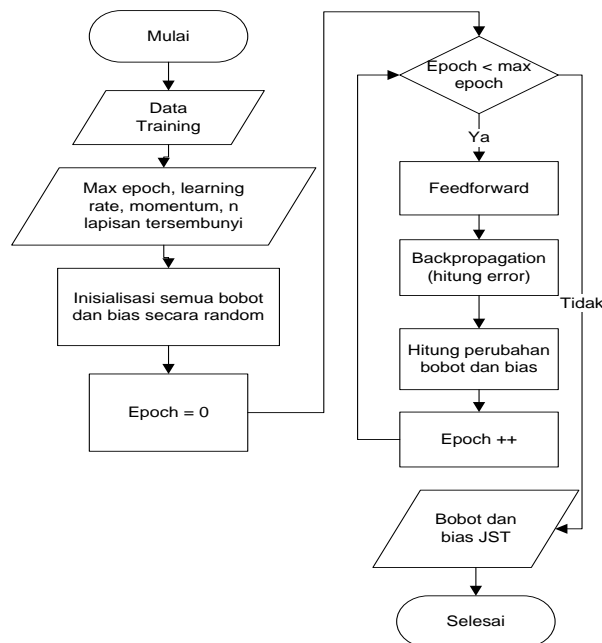
Tahap *training* adalah tahap pembangunan model klasifikasi. Tahap training pada JST menghasilkan bobot dan bias yang telah dilatih. Langkah yang dilakukan adalah sebagai berikut:

1. Tentukan nilai parameter jaringan seperti *learning rate* (kecepatan pembelajaran), momentum dan jumlah lapisan tersembunyi. Selain itu tentukan pula berapa *epoch* maksimum suatu jaringan. Nilai tersebut diatur sedemikian rupa untuk mendapatkan model jaringan yang optimum.
2. Inisialisasi semua bobot dan bias JST dengan random dalam rentang $[-0.05, 0.05]$. Rentang tersebut didapatkan dari pseudocode algoritma *backpropagation*.
3. Set *epoch* = 0 yang menunjukkan nilai iterasi awal sebelum nilai iterasi mencapai *epoch* maksimum.
4. Lakukan proses *feedforward* untuk mendapatkan nilai output prediksi. Setiap kali selesai perhitungan output neuron baik pada lapisan tersembunyi maupun lapisan output, Selanjutnya, lakukan fungsi aktivasi.
5. Lakukan proses *backpropagation* yang menyesuaikan bobot dan bias dengan *error* pada output prediksi.

6. Lakukan langkah 4 dan 5 secara berulang-ulang sampai *epoch* menemukan nilai maksimumnya.

Hasil dari *training* yaitu koefisien regresi, bobot dan bias disimpan. Model dengan hasil *training* tersebut kemudian diujicobakan dengan data testing. Hal ini bertujuan untuk menguji model yang telah dilakukan *training*.

Tahap-tahap *training* tersebut secara garis besar digambarkan dalam diagram alir pada Gambar 3.12. Pada proses *training*, sebelum memasuki pembelajaran JST, dilakukan inisialisasi parameter jaringan, inisialisasi bobot dan bias JST dengan random dan inisialisasi *epoch* = 0. *Max epoch* adalah jumlah *training* yang dilakukan untuk melatih jaringan. Setiap kali jaringan dilatih, bobot dan bias terus diubah agar pada *training* berikutnya, *error* output prediksi semakin kecil.

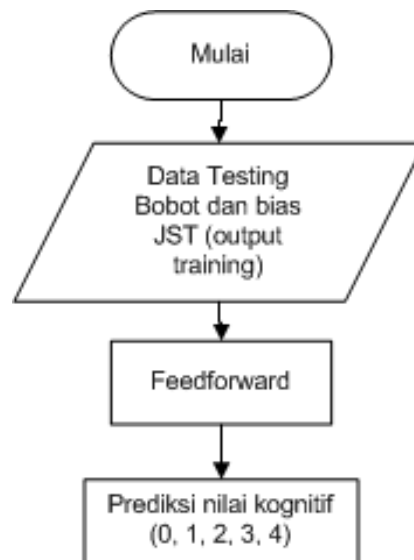


Gambar 3.12 Diagram alir tahap *training* JST

b. Tahap Testing

Tahap testing dilakukan untuk menguji kehandalan model. Testing memberikan input yang berbeda dengan input pada tahap *training*. Tahap-tahap testing secara garis besar digambarkan pada Gambar 3.13. Pada tahap ini, dilakukan penghitungan step *feedforward*.

Tahap ini menghasilkan output prediksi yang dilakukan fungsi aktivasi sigmoid sebelumnya. Output prediksi ini kemudian dibandingkan dengan nilai kelas yang asli untuk menghitung RMSE. Hal ini bertujuan mengetahui tingkat error model yang telah dilatih.



Gambar 3.13 Diagram alir tahap testing.

3.9 Implementasi Kode

Pada subbab ini diuraikan mengenai implementasi perangkat lunak yang meliputi kode program yang terdapat dalam perangkat lunak. Pada tiap fungsi dijelaskan mengenai parameter input, output dan penjelasan kode yang berhubungan dengan program dan teori.

3.9.1 Tahap Praproses

Pada tahap praproses, dilakukan tahap normalisasi data seperti yang ditunjukkan pada Gambar 3.14. Proses normalisasi terdapat pada kode baris ke-17. Kode pada Gambar 3.14 memiliki variabel dengan penjelasan sebagai berikut:

1. `min` menyimpan data yang nilainya paling rendah dalam satu kolom fitur.
2. `max` menyimpan data yang nilainya paling tinggi dalam satu kolom fitur.
3. `input [i][j]` adalah array dua dimensi yang menyimpan *record* sebuah data dengan fiturnya.

Nama File : `NeuralNetwork.java`

Nama Fungsi : `normalizationData`

Input : `read (object dari readFile.java yang menyimpan dataset dari file)`

1	<code>public void normalizationData(ReadFile read) {</code>
2	<code> double min;</code>
3	<code> double max;</code>
4	<code> ...</code>
5	<code> for (int j = 0; j < input[0].length; j++)</code> <code> {</code>
6	<code> min = 1000000000;</code>
7	<code> max = 0.00000;</code>
8	<code> for (int i = 0; i < (input.length); i++) {</code>
9	<code> if (min > input[i][j]) {</code>
10	<code> min = input[i][j];</code>
11	<code> }</code>
12	<code> if (max < input[i][j]) {</code>
13	<code> max = input[i][j];</code>
14	<code> }</code>
15	<code> }</code>
16	<code> for (int i = 0; i < (input.length); i++) {</code>
17	<code> input[i][j] = (input[i][j] - min) / (max -</code> <code> min);</code>
18	<code> ...}}}</code>

Gambar 3.14 Kode program tahap normalisasi data.

3.9.2 Tahap Proses

Fungsi utama yang digunakan pada tahap proses terdapat pada fungsi `crossValidation`.

Nama File : `NeuralNetwork.java`

Nama Fungsi : `crossValidation`

Input : `fold` (jumlah `k` dalam *k-fold cross validation*)

1	<code>public void crossValidation(int fold) {</code>
2	<code> swap();</code>
3	<code> rmse = new double [fold];</code>
4	<code> int batasAtas, batasBawah;</code>
5	<code> for (int i = 0; i < fold; i++) {</code>
6	<code> batasBawah = (int) (Math.floor((double)i /</code> <code> (double)fold * output.length));</code>
7	<code> batasAtas = (int) (Math.floor((double)(i +</code> <code> 1) / (double)fold * output.length));</code>
8	<code> process(batasBawah, batasAtas);</code>
9	<code> rmse[i] = calcParseMissclass(batasBawah,</code> <code> batasAtas, testResult, output);</code>
10	<code> }</code>
11	<code>}</code>

Gambar 3.15 Kode program tahap proses *k-fold cross validation*.

Gambar 3.15 menunjukkan kode fungsi `crossValidation`, dengan penjelasan fungsi dan variabel sebagai berikut:

a. Fungsi

1. `swap` yang merandom data di kode baris ke-2.
2. `process` yang melakukan tahap *training* dan testing dengan menerapkan metode *k-fold cross validation* di kode baris ke-8.
3. `calcParseMissclass` yang menghitung jumlah prediksi yang tidak sesuai dengan kelas sebenarnya di setiap *fold*. Fungsi ini terletak di kode baris ke-9.

b. Variabel

1. `rmse` berbentuk array satu dimensi yang menyimpan keluaran dari fungsi `calcParseMissclass`. Hasil keluaran berupa nilai RMSE di tiap *fold*.
2. `batasAtas`, yang menyimpan indeks paling tinggi pada data testing yang ditetapkan dari *k-fold cross validation*.
3. `batasBawah`, kebalikan dari `batasAtas`, menyimpan indeks paling rendah pada data testing.

3.9.3 Tahap Pengacakan Data

Tahap ini mengacak isi data dengan cara menukar dua baris yang indeksinya didapatkan dari fungsi `Random` di Java. Penukaran ini dilakukan sebanyak jumlah baris

pada data. Hasilnya, indeks data yang digunakan pada proses prediksi tidak sama dengan data input sebelum diacak.

Nama File : NeuralNetwork.java

Nama Fungsi : swap

1	public void swap() {
2	Random rand = new Random();
3	int indeksA, indeksB;
4	double[] tempIn = new double[input[0].length];
5	double tempOut;
6	for (int i = 0; i < input.length; i++) {
7	indeksA = rand.nextInt(input.length);
8	indeksB = rand.nextInt(input.length);
9	while (indeksA == indeksB) {
10	indeksB = rand.nextInt(input.length);
11	}
12	tempIn = input[indeksA];
13	input[indeksA] = input[indeksB];
14	input[indeksB] = tempIn;
15	tempOut = output[indeksA];
16	output[indeksA] = output[indeksB];
17	output[indeksB] = tempOut;
18	}
19	}

Gambar 3.16 Kode program mengacak baris data.

Gambar 3.16 menunjukkan kode fungsi swap, dengan penjelasan variabel sebagai berikut:

1. rand, yang memiliki fungsi Random pada Java. Variabel ini diinisialisasi pada kode baris ke-2.
2. indeksA dan indeksB sebagai indeks yg akan ditukar. Pada kode baris ke-9 sampai ke-10 menerangkan jika indeks yang akan ditukar sama, maka nilai indeksB akan dirandom terus sampai indeksA tidak sama dengan indeksB.
3. input menyimpan data fitur.
4. output menyimpan data kelas aktual.
5. tempIn bertugas sebagai tempat sementara menyimpan data fitur yang ditukar indeksnya. tempIn berbentuk array satu dimensi.

6. tempOut bertugas sebagai tempat sementara menyimpan data kelas aktual yang ditukar indeksinya. tempOut berbentuk array satu dimensi.

3.9.4 Tahap Proses (*Training dan Testing*)

Fungsi ini mencakup proses *training* dan testing pada JST.

Nama File : NeuralNetwork.java

Nama Fungsi : process

Input :

- batasBawah (batas indeks bawah untuk melakukan proses testing)
- batasAtas (batas indeks atas untuk melakukan proses testing)

1	public void process(int batasBawah, int batasAtas) {
2	initAll();
3	//training
4	for (int j = 0; j < batasBawah; j++) {
5	//feedforward
6	network.computeOutputs(input[j]);
7	//backpropagation
8	network.backPropagation(output[j]);}
9	for (int k = batasAtas; k < output.length; k++) {
10	//feedforward
11	network.computeOutputs(input[k]);
12	//backpropagation
13	network.backPropagation(output[k]);}
14	//testing
15	for (int j = batasBawah; j < batasAtas; j++)
	{
16	testResult[j] =
	network.computeOutputs(input[j]); }}

Gambar 3.17 Kode program tahap proses *training* dan testing.

Gambar 3.17 menunjukkan kode fungsi process, dengan penjelasan fungsi dan variabel sebagai berikut:

a. Fungsi

1. initAll melakukan inisialisasi parameter input, bobot, bias dan delta pengubahannya di kode baris ke-2.

2. `computeOutputs` yang menghitung output pada fase *feedforward* JST di kode baris ke-6, 12 (*training*) dan 19 (*testing*).
3. `backpropagation` mengoreksi bobot dan bias JST. Fungsi ini terletak pada kode baris ke-8 dan 14.

b. Variabel

1. `testResult` menyimpan hasil prediksi.

3.9.5 Tahap Inisialisasi Parameter Input dan Bobot

Fungsi `initAll` dipanggil setiap kali membangun model JST baru untuk menginisialisasi parameter input jaringan dan bobot.

Nama File : `NeuralNetwork.java`

Nama Fungsi : `initAll`

1	<code>public void initAll()</code>
2	<code>{</code>
3	<code> initParameter();</code>
4	<code> network = new Network((input[0].length + 1), hiddenLayers, 1, learnRate, momentum);</code>
5	<code>}</code>

Gambar 3.18 Kode program utama inisialisasi.

Gambar 3.18 menunjukkan kode fungsi `initAll`. Di dalam fungsi `initAll`, terdapat fungsi `initParameter` pada kode baris ke-3 yang melakukan inisialisasi parameter input secara *default*. Parameter input tersebut adalah `learnRate` yang menyatakan kecepatan pembelajaran, `momentum` untuk mempercepat akurasi, `maxEpoch` sebagai batas iterasi *training* serta `hiddenLayers` yang menyatakan jumlah lapisan tersembunyi pada model jaringan. Nilai parameter bisa berubah jika ada masukan *user* pada form. Gambar 3.19 menunjukkan kode fungsi `initParameter`.

Nama File : NeuralNetwork.java

Nama Fungsi : initParameter

1	<code>public void initParameter() {</code>
2	<code> learnRate = 0.3;</code>
3	<code> momentum = 0.2;</code>
4	<code> maxEpoch = 50;</code>
5	<code> hiddenLayers = (input[0].length + 2) / 2;</code>
6	<code>}</code>

Gambar 3.19 Kode program inialisasi parameter jaringan.

Setelah dilakukan inialisasi parameter, langkah berikutnya adalah inialisasi bobot dan bias secara random serta penetapan delta pengubahan bobot dan bias dengan angka nol pada semua lapisan. Fungsi `initWeight` dipanggil saat memanggil *constructor* dari kelas `Network`.

Gambar 3.20 menunjukkan kode fungsi `initWeight`, dengan penjelasan variabel sebagai berikut:

1. `biasHidden`, menyimpan bias pada lapisan tersembunyi.
2. `deltaBiasHidden`, menyimpan delta pengubahan bias pada lapisan tersembunyi yang diset mula-mula nol pada kode baris ke-5.
3. `biasOutput`, menyimpan bias pada lapisan output.
4. `deltaBiasOutput`, menyimpan delta pengubahan bias pada lapisan output yang diset mula-mula nol pada kode baris ke-10.
5. `inputCount`, merupakan jumlah neuron pada lapisan input.
6. `hiddenCount`, merupakan jumlah neuron pada lapisan tersembunyi.
7. `outputCount`, merupakan jumlah neuron pada lapisan output.
8. `weightInputHidden`, menyimpan bobot yang terletak di antara lapisan input dan lapisan tersembunyi.
9. `weightHiddenOutput`, menyimpan bobot yang terletak di antara lapisan tersembunyi dan lapisan output.

Nama File : Network.java

Nama Fungsi : initWeight

1	private void initWeight() {
2	Random r = new Random();
3	for (int i = 0; i < biasHidden.length; i++) {
4	biasHidden[i] = r.nextDouble() * .1 - .05;
5	deltaBiasHidden[i] = 0;
6	}
7	for (int i = 0; i < biasOutput.length; i++)
8	{
9	biasOutput[i] = r.nextDouble() * .1 - .05;
10	deltaBiasOutput[i] = 0;
11	}
12	for (int i = 0; i < inputCount; i++) {
13	for (int j = 0; j < hiddenCount; j++){
14	weightInputHidden[i][j] = r.nextDouble()
15	* .1 - .05;
16	deltaWeightInHid[i][j] = 0;
17	}
18	}
19	for (int i = 0; i < hiddenCount; i++) {
20	for (int j = 0; j < outputCount; j++) {
21	weightHiddenOutput[i][j] = r.nextDouble()
22	*.1 - .05;
	deltaWeightHidOut[i][j] = 0;
	}}}

Gambar 3.20 Kode program inisialisasi bobot, bias dan delta pengubahannya.

3.9.6 Tahap *Feedforward*

Tahap ini menghitung output pada model JST. Output tersebut didapatkan dengan penjumlahan dari perkalian bobot dengan masing-masing neuron pada lapisannya.

Nama File : Network.java

Nama Fungsi : computeOutputs

Input : input (input pada lapisan input)

Output : neuronOutput (matriks satu kolom yang menyatakan nilai akhir neuron pada lapisan output)

1	<code>public double computeOutputs(double input[]) {</code>
2	<code> for (int i = 0; i < input.length; i++) {</code>
3	<code> neuronInput[i] = input[i];</code>
4	<code> }</code>
5	
6	<code> double sum;</code>
7	<code> for (int i = 0; i < hiddenCount; i++) {</code>
8	<code> sum = 0;</code>
9	<code> for (int j = 0; j < inputCount; j++)</code>
10	<code> {</code>
11	<code> sum += neuronInput[j] *</code>
	<code> weightInputHidden[j][i];</code>
12	<code> }</code>
13	<code> sum += biasHidden[i];</code>
14	<code> neuronHidden[i] = sigmoid(sum);</code>
15	<code> }</code>
16	<code> for (int i = 0; i < outputCount; i++) {</code>
17	<code> sum = 0;</code>
18	<code> for (int j = 0; j < hiddenCount; j++)</code>
	<code> {</code>
19	<code> sum += neuronHidden[j] *</code>
	<code> weightHiddenOutput[j][i];</code>
20	<code> }</code>
21	<code> sum += biasOutput[i];</code>
22	<code> neuronOutput[i] = sigmoid(sum);</code>
23	<code> }</code>
24	<code> return neuronOutput[0];</code>
25	<code>}</code>

Gambar 3.21 Kode program tahap *feedforward* jaringan.

Gambar 3.21 menunjukkan kode fungsi `computeOutputs`, dengan penjelasan fungsi dan variabel sebagai berikut:

a. Fungsi

1. `sigmoid` melakukan fungsi aktivasi. Fungsi ini terletak pada kode di baris ke-22.

b. Variabel

1. `neuronInput` menyimpan nilai neuron pada lapisan input dan berupa array satu dimensi.
2. `neuronHidden` menyimpan nilai neuron pada lapisan tersembunyi dan berupa array satu dimensi.

3. `neuronOutput` menyimpan nilai neuron pada lapisan output dan berupa array satu dimensi.
4. `sum` menyimpan hasil penjumlahan dari perkalian bobot dengan neuron dan penambahan bias.

Setelah mengalikan bobot dengan lapisan dan penambahan bias, dipanggil fungsi `sigmoid` untuk menghitung nilai aktivasi logistik. Gambar 3.22 menunjukkan fungsi `sigmoid`.

Nama File : `Network.java`

Nama Fungsi : `sigmoid`

Input :

- `sum` (output hasil perkalian bobot dengan lapisan dan penambahan bias)

Output : nilai aktivasi dari fungsi `sigmoid`

1	<code>public double sigmoid(double sum) {</code>
2	<code>return 1.0 / (1 + Math.exp(-sum));</code>
3	<code>}</code>

Gambar 3.22 Kode program penghitungan nilai aktivasi.

3.9.7 Tahap *Backpropagation*

Pertama, tahap ini menghitung *error* dari selisih nilai kelas prediksi dan kelas aktual. Kedua, dilanjutkan dengan melakukan koreksi bobot dan bias berdasarkan informasi *error*. Dengan adanya koreksi bobot dan bias, *error* yang diperoleh akan semakin kecil pada iterasi selanjutnya.

Nama File : `Network.java`

Nama Fungsi : `backPropagation`

Input :

- `idealOutput` (keluaran kelas yang diharapkan)

1	<code>public void backPropagation(double idealOutput) {</code>
2	<code>for (int j = 0; j < outputCount; j++) {</code>
3	<code>errorOutput[j] = (idealOutput - neuronOutput[j]) * (neuronOutput[j]) * (1 - neuronOutput[j]);</code>
4	<code>}</code>
5	<code>for (int j = 0; j < hiddenCount; j++) {</code>
6	<code>double sum = 0.0;</code>
7	<code>for (int k = 0; k < outputCount; k++)</code>
8	<code>{</code>
9	<code>double term = errorOutput[k] * weightHiddenOutput[j][k];</code>
10	<code>sum += term;</code>
11	<code>}</code>
12	<code>errorHidden[j] = sum * (neuronHidden[j]) * (1 - neuronHidden[j]);</code>
13	<code>}</code>
14	<code>updateWeightHiddenOutput();</code>
15	<code>updateWeightInputHidden();</code>
16	<code>}</code>

Gambar 3.23 Kode program tahap *backpropagation*.

Gambar 3.23 menunjukkan kode fungsi `backPropagation`, dengan penjelasan fungsi dan variabel sebagai berikut:

a. Fungsi

1. `updateWeightHiddenOutput` mengubah bobot dan bias antara lapisan tersembunyi dengan lapisan output. Fungsi ini terletak pada kode baris ke-14.
2. `updateWeightInputHidden` mengubah bobot dan bias antara lapisan tersembunyi dengan lapisan input. Fungsi ini terletak pada kode baris ke-15.

b. Variabel

1. `errorOutput` menyimpan nilai *error* dari keluaran neuron pada lapisan output sesuai *pseudocode* algoritma *backpropagation*.
2. `errorHidden` menyimpan nilai *error* dari keluaran neuron pada lapisan tersembunyi sesuai *pseudocode* algoritma *backpropagation*.

Setelah mendapatkan informasi *error*, dipanggil fungsi `updateWeightHiddenOutput` yang ditunjukkan pada Gambar 3.24 dan

updateWeightInputHidden untuk mengubah semua bobot dan bias pada model jaringan yang ditunjukkan pada Gambar 3.25.

Pengubahan bobot terjadi pada Gambar 3.24 di kode baris ke-5 dan juga pada Gambar 3.25 di kode baris ke-5.

Nama File : Network.java

Nama Fungsi : updateWeightHiddenOutput

1	private void updateWeightHiddenOutput()
2	{
3	for (int i = 0; i < hiddenCount; i++) {
4	for (int j = 0; j < outputCount; j++) {
5	double weightChange = learnRate * errorOutput[j] * neuronHidden[i] + momentum * deltaWeightHidOut[i][j];
6	weightHiddenOutput[i][j] += weightChange;
7	deltaWeightHidOut[i][j] = weightChange;
8	weightChange = learnRate * errorOutput[j] + momentum * deltaBiasOutput[j];
9	biasOutput[j] = +weightChange;
10	deltaBiasOutput[j]=weightChange;
11	}
12	}
13	}

Gambar 3.24 Kode program perubahan nilai bobot dan bias pada lapisan tersembunyi ke lapisan output.

Nama File : Network.java

Nama Fungsi : updateWeightInputHidden

1	<code>private void updateWeightInputHidden()</code>
2	<code>{</code>
3	<code>for (int i = 0; i < inputCount; i++) {</code>
4	<code>for (int j = 0; j < hiddenCount; j++) {</code>
5	<code>double weightChange = learnRate * <u>errorHidden</u>[j] * neuronInput[i] + momentum * deltaWeightInHid[i][j];</code>
6	<code>weightInputHidden[i][j] += weightChange;</code>
7	<code>deltaWeightInHid[i][j] = weightChange;</code>
8	<code>weightChange = learnRate * errorHidden[j] + momentum * deltaBiasHidden[j];</code>
9	<code>biasHidden[j] = +weightChange;</code>
10	<code>deltaBiasHidden[j] = weightChange;</code>
11	<code>}</code>
12	<code>}</code>
13	<code>}</code>

Gambar 3.25 Kode program perubahan nilai bobot dan bias pada lapisan input ke lapisan tersembunyi.

3.9.8 Tahap Penghitungan RMSE

Tahap ini menghitung nilai RMSE. Gambar 3.26 menunjukkan kode fungsi `calcParseMissclas`.

Nama File : `NeuralNetwork.java`

Nama Fungsi : `calcParseMissclass`

Input :

- `batasBawah` (batas indeks bawah untuk melakukan proses testing)
- `batasAtas` (batas indeks atas untuk melakukan proses testing)
- `target` (nilai kelas prediksi)
- `y` (nilai kelas aktual)

BAB 4

HASIL DAN PEMBAHASAN

Bab ini akan dibagi menjadi dua bagian yaitu Pengujian Sistem dan Validasi Hasil. Pengujian Sistem meliputi pengambilan data permainan matematika, dan hasil klasifikasi. Validasi hasil meliputi pembahasan hasil dan nilai RMSE.

4.1 Pengujian Sistem

Lokasi pengambilan data permainan matematika dilakukan di MI Ma'arif NU As Saadah Gresik. Pengujian dilakukan terhadap anak kelas 5 SD sebanyak tiga kelas. Siswa dipersilakan untuk memainkan *game* sebanyak lima kali. Dari kelima kali percobaan ini, diambil data mereka yang paling lengkap. Pada saat bermain, mereka cukup antusias dalam menyelesaikan berbagai soal yang ada dalam permainan.



Gambar 4.1 Foto saat siswa memainkan *game*

Sebelum mendapatkan data yang bagus, dilakukan preproses data. Langkah preproses data dijelaskan sebagai berikut:

1. Dari kelima percobaan *game* oleh username yang sama, dipilih data yang lengkap. Pada Gambar 4.2, terdapat lima *username* gilang. Data yang dipilih adalah data dengan username Gilang4 karena data tersebut berhasil menjawab

lebih banyak dibandingkan *username* yang lain. Hal ini dapat dilihat melalui variabel soal1 sampai soal9.

Jika terdapat dua data yang memiliki skor soal yang sama, maka kemudian dilihat dari sedikitnya waktu yang digunakan dalam menjawab soal.

1	username	soal1	soal2	soal3	soal4	soal5	soal6	soal7	soal8	soal9
90	GILANG1	2	3	0	0	0	0	0	0	0
91	GILANG2	2	0	2	0	2	0	0	0	0
92	GILANG3	2	0	2	0	2	0	0	0	0
93	GILANG4	2	3	3	3	3	3	3	0	0
94	GILANG5	0	1	2	0	0	0	0	0	0
95	FIKI1	2	0	0	0	0	0	0	0	0

Gambar 4.2 Data dengan *username* yang sama

2. Data dengan *username* yang tidak diketahui harus dihapus. Contoh *username* yang tidak sesuai yakni *username* juyer, shadow dan goli. *Username* tersebut tidak mewakili nama siswa. Data *username* yang tidak sesuai dapat dilihat pada Gambar 4.3.

1	username	soal1	soal2	soal3	soal4	soal5	soal6	soal7	soal8	soal9	waktu1	waktu2	waktu3	waktu4	waktu5	waktu6	waktu7	waktu8	waktu9
34	juyer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	shadow	2	0	2	0	0	0	0	0	0	10	17	10	14	9	21	0	0	0
36	goli	0	0	0	0	0	0	0	0	0	6	5	11	0	0	0	0	0	0
37	hendra	2	3	0	0	0	0	0	0	0	13	10	11	0	0	0	0	0	0
38	kian	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
39	Ani1	0	1	2	0	0	0	0	0	0	17	17	16	41	12	0	0	0	0
40	ani2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	aNI3	0	1	2	3	3	0	0	0	0	2	75	2	5	24	22	0	0	0
42	KHODIJAH	0	1	0	0	0	0	0	0	0	6	2	8	0	0	0	0	0	0
43	Agustin	2	0	0	0	0	0	0	0	0	27	24	57	0	0	0	0	0	0

Gambar 4.3. Data dengan *username* yang tidak sesuai

Aturan level pada *game* yaitu level siswa akan meningkat jika siswa menjawab benar dan akan menurun jika siswa menjawab salah. Contoh, jika siswa berhasil menjawab benar pada level 2, maka soal yang keluar adalah level 3. Jika siswa menjawab salah pada level 3, maka soal yang muncul berikutnya adalah soal level 2. Soal yang keluar ketika siswa menghadapi *game* untuk pertama kalinya adalah soal dengan level 2.

Data skor dengan sembilan soal memunculkan sembilan variabel skor. Data waktu dengan sembilan soal memunculkan sembilan variabel waktu. Total variabel menjadi 18 dengan sembilan variabel dari skor dan sembilan variabel dari waktu. Untuk memudahkan proses klasifikasi, maka variabel disederhanakan menjadi dua yaitu skor dan waktu. Variabel skor merupakan skor total permainan dan variabel waktu merupakan jumlah total lama permainan. Data permainan dapat dilihat pada Tabel 4.1 dan Tabel 4.2.

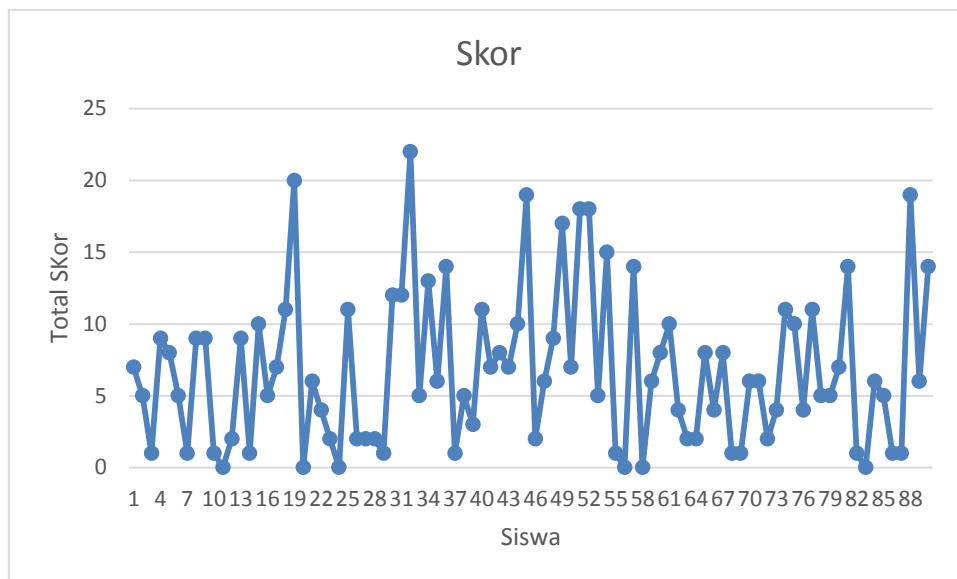
Tabel 4.1 Data permainan dengan sembilan variabel soal

username	soal1	soal2	soal3	soal4	soal5	soal6	soal7	soal8	soal9
arman1	0	0	0	0	0	0	0	0	0
arman2	2	0	2	3	0	0	0	0	0
ibra1	2	0	2	0	0	1	0	0	0
yulhaq	0	1	0	0	0	0	0	0	0
yulhaq1	0	0	1	0	0	0	0	0	0
deni	2	0	2	3	0	2	0	0	0

Tabel 4.2 Data permainan dengan sembilan variabel waktu

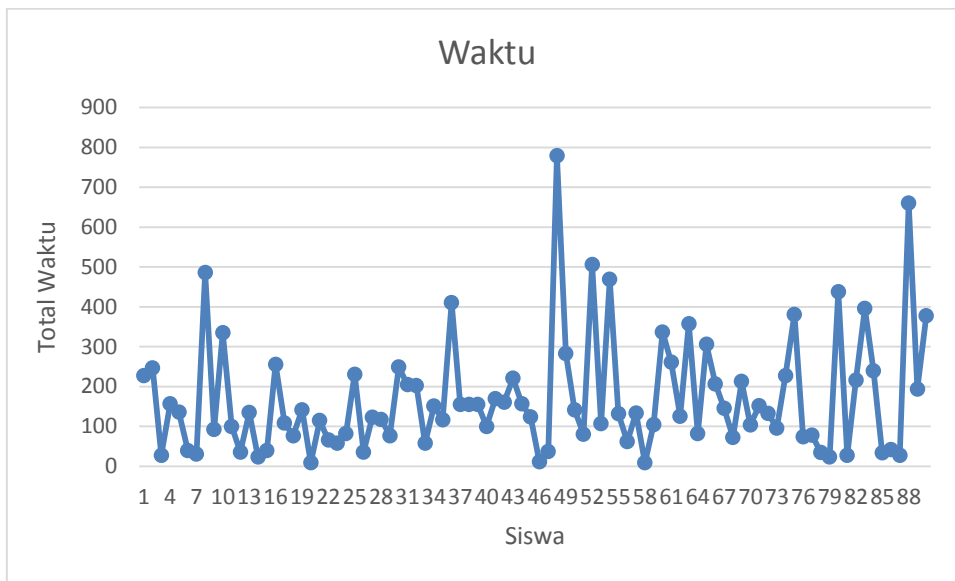
username	waktu1	waktu2	waktu3	waktu4	waktu5	waktu6	waktu7	waktu8	waktu9
arman1	0	0	0	0	0	0	0	0	0
arman2	25	15	33	41	85	29	0	0	0
ibra1	50	49	29	31	14	37	37	0	0
Yulhaq	2	3	3	0	0	0	0	0	0
yulhaq1	5	1	1	2	10	9	0	0	0
Deni	22	23	12	16	17	9	20	33	0

Pemilihan data kemudian dilakukan untuk mendapatkan data yang tepat mewakili siswa. Artinya satu data mewakili satu siswa, sehingga jumlah data yang digunakan dalam proses klasifikasi adalah sejumlah siswa yang menyelesaikan *game* matematika. Data tersebut kemudian ditotal untuk dijadikan sebagai *input* metode.



Gambar 4.4 Data jumlah skor permainan

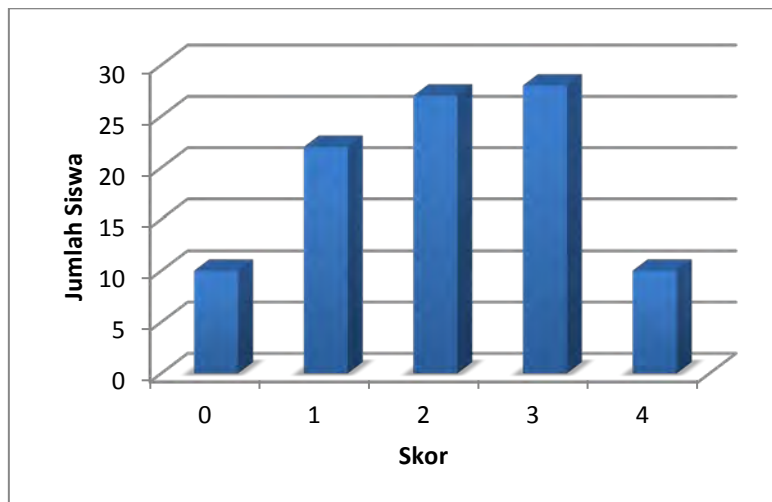
Gambar 4.4 menunjukkan grafik hubungan antara siswa dengan total skor permainan. Arah horizontal merupakan data siswa, sedangkan arah vertikal merupakan data total skor yang dimiliki tiap siswa.



Gambar 4.5 Data jumlah waktu permainan

Pada Gambar 4.5 menunjukkan grafik hubungan antara siswa dengan total waktu permainan. Arah horizontal merupakan data siswa, sedangkan arah vertikal merupakan data total waktu yang dimiliki tiap siswa.

Setelah data *input* diolah, selanjutnya dilakukan pengolahan data *output* target yang didapatkan dari tes tulis. Hasil akhir dari pengerjaan tes tulis dibagi menjadi lima kelompok berdasarkan skor yang dicapai. Sebanyak 10 siswa mendapat skor 0 (nol), 22 siswa mendapat skor 1 (satu), 27 siswa mendapat skor 2 (dua), 28 siswa mendapat skor 3(tiga) dan 10 siswa mendapat skor 4(empat).



Gambar 4.6 Grafik Skor Tes Tulis Siswa

Tahap preproses data dilakukan dengan melakukan normalisasi pada data yang ada menggunakan persamaan 3.1. Tabel 4.3 menunjukkan data yang sudah dinormalisasi.

Tabel 4.3 Data setelah normalisasi

No.	Skor	Waktu	Post Test
1	0.32	0.28	0.75
2	0.23	0.31	0.5
3	0.05	0.02	0.5
4	0.41	0.19	0.25
5	0.36	0.16	0.75
6	0.23	0.04	0.5
7	0.05	0.02	0
8	0.41	0.62	0.5
9	0.41	0.11	0.75
10	0.05	0.42	0.75
11	0	0.11	0.75
12	0.09	0.03	1
13	0.41	0.16	0
14	0.05	0.02	0
15	0.45	0.04	0.75
16	0.23	0.32	0.25
17	0.32	0.13	0.5
18	0.5	0.08	0.5
19	0.91	0.17	0.5
20	0	0	0.75
21	0.27	0.13	0.75
22	0.18	0.07	0.5
23	0.09	0.06	0.25

Tabel 4.3 berisi kolom data skor murid saat mengerjakan *game*. Kolom waktu adalah data – data waktu mengerjakan soal. Data pada kolom *post test* merupakan hasil tes tulis siswa yang akan dijadikan sebagai target dalam penelitian ini. Data skor dan waktu diperoleh dari nilai total skor dan total waktu dalam permainan *serious game*.

Langkah selanjutnya adalah melakukan uji coba untuk mendapatkan model yang optimal dengan mencari tingkat kesalahan prediksi yang paling kecil dari jumlah epoch, *learning rate*, *hidden layer*, dan momentum. Hal ini untuk mengetahui struktur jaringan saraf tiruan yang terbaik.

Tabel 4.4 Tabel epoch dan *hidden layer*

Learning rate = 0.2, momentum = 0.3

Epoch / Hidden Layer	1	2	3	4	5	6	7
25	0.30	0.30	0.30	0.30	0.297	0.30	0.298
50	0.299	0.298	0.30	0.299	0.30	0.298	0.30
100	0.30	0.30	0.30	0.298	0.30	0.30	0.298
200	0.299	0.298	0.299	0.2974	0.2975	0.302	0.30
300	0.298	0.315	0.306	0.299	0.309	0.300	0.303
400	0.309	0.300	0.308	0.299	0.307	0.298	0.300
500	0.306	0.301	0.297	0.306	0.299	0.303	0.298
1000	0.300	0.303	0.296	0.302	0.298	0.299	0.299
2000	0.302	0.298	0.300	0.305	0.306	0.298	0.303

Tabel 4.4 menunjukkan cara pencarian nilai Epoch dan *hidden layer*. *Learning rate* diset mula-mula 0.2 dan nilai momentum 0.3. Nilai epoch dinaikkan perlahan mulai dari 25 hingga 2000. Kemudian nilai *hidden layer* diuji coba mulai dari satu hingga tujuh. Nilai RMSE terkecil didapat ketika *hidden layer* 4, dan nilai epoch 200 yaitu 0.2974. Pencarian berhenti ketika semakin tinggi epoch, semakin besar nilai kesalahan prediksi. Artinya, performa model mulai menurun seiring kenaikan epoch.

Tabel 4.5 Tabel pencarian *learning rate*

Learning rate	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
RMSE	0.298	0.295	0.301	0.313	0.303	0.309	0.299	0.30	0.299	0.30

Pada tabel 4.5 sistem diuji coba dengan mengubah nilai *learning rate* mulai dari 0.1 sampai 1. Nilai RMSE terkecil adalah 0.295 pada saat *learning rate* bernilai 0.2.

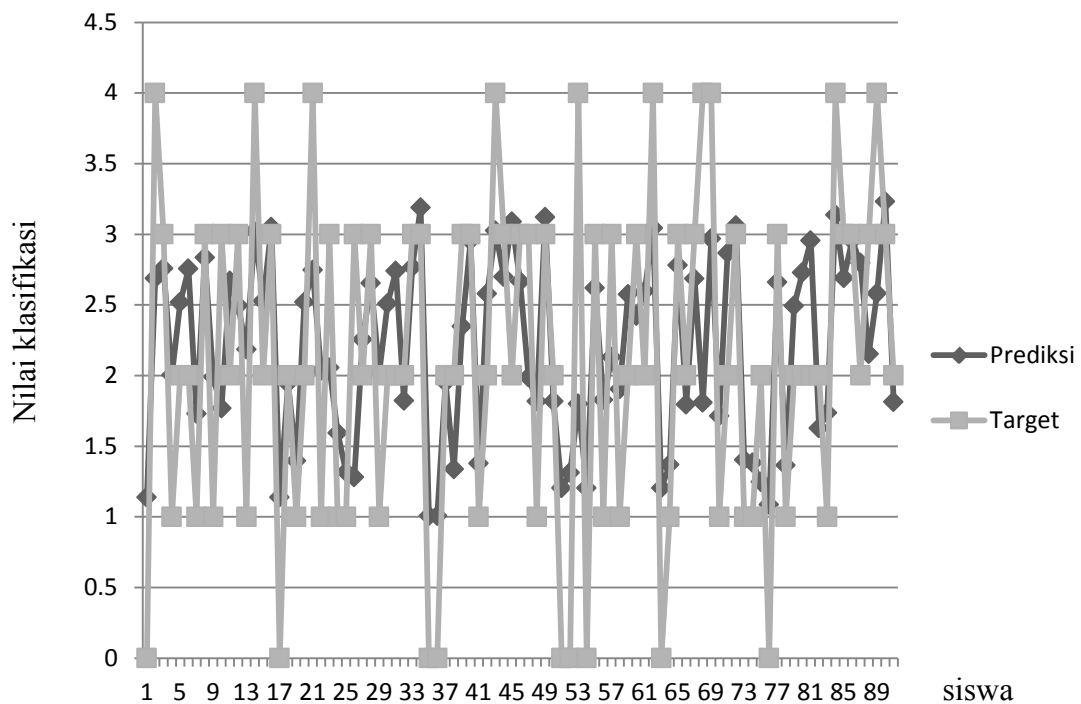
Tabel 4.6 Tabel pencarian momentum

Momentum	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
RMSE	0.298	0.302	0.298	0.295	0.299	0.3	0.304	0.3	0.302	0.298	0.296

Nilai momentum diujicobakan mulai dari nol hingga satu. Pada Tabel 4.6 dapat dilihat bahwa nilai RMSE yang terkecil adalah 0.295 pada saat momentum bernilai 0.3.

Uji coba dilakukan dengan memakai metode *k-fold cross validation*. Jumlah fold yang digunakan adalah 4. Pada setiap percobaan, dihitung error program yang didapatkan dari selisih antara output ideal (nilai kelas yang sebenarnya) dengan output yang dihasilkan oleh jaringan. RMSE dihitung dari RMSE semua percobaan. Semakin kecil nilai RMSE menunjukkan bahwa aplikasi semakin handal dalam memprediksi kelas sesuai dengan yang diharapkan. Data yang contohnya ada dalam tabel 4.3 kemudian diolah dengan *backpropagation neural network*.

Sebagai perbandingan, Gambar 4.7 menunjukkan grafik perbandingan antara hasil output prediksi dengan target sebelum data dinormalisasi. Grafik berwarna hitam menunjukkan hasil prediksi siswa melalui metode JST. Grafik berwarna abu-abu menunjukkan hasil kognitif siswa yang didapatkan dari tes tulis.

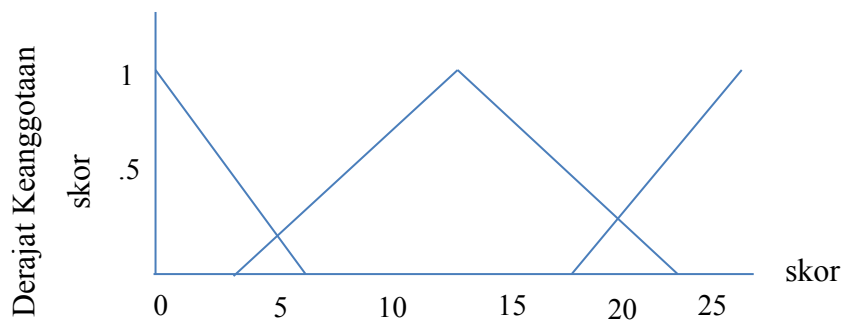


Gambar 4.7 Grafik Perbandingan Output Prediksi dengan Target sebelum Normalisasi Analisis menggunakan data yang belum dinormalisasi memiliki nilai RMSE yang lebih tinggi, yaitu 1.52.

4.2 Perbandingan dengan Metode Fuzzy

Sebagai perbandingan, pengolahan data untuk klasifikasi juga dilakukan dengan metode fuzzy. Fuzzy yang dilakukan adalah dengan cara menjumlah skor dan waktu pada masing – masing data bermain siswa. Skor dibagi menjadi tiga kelompok, yaitu *low*,

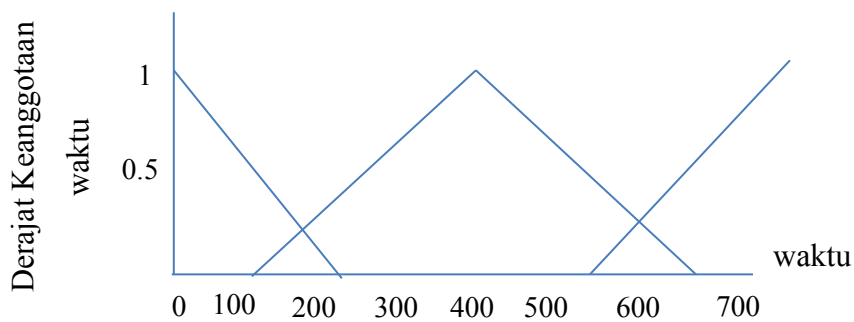
medium, dan *high*. Waktu juga dibagi menjadi tiga kelompok, yaitu lama, sedang, dan cepat. Gambar 4.8 menunjukkan fungsi keanggotaan dari skor.



Gambar 4.8 Fungsi keanggotaan *input* skor

Dari Gambar 4.8 dapat diketahui bahwa:

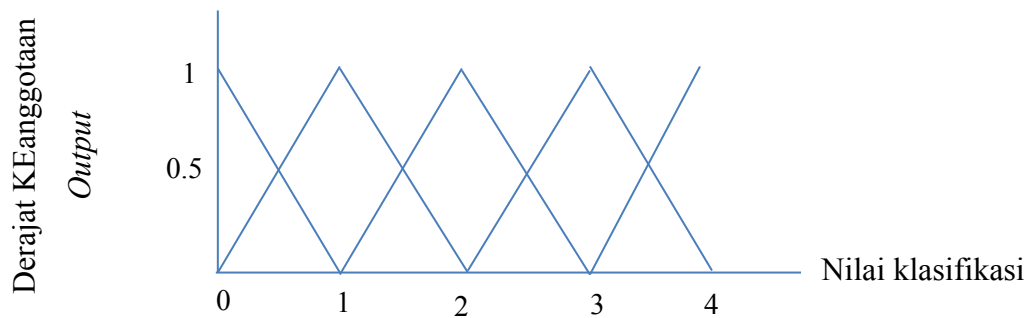
1. skor *low* adalah skor < 7
2. skor *medium* adalah $4 < \text{skor} < 22$
3. skor *high* adalah skor > 18



Gambar 4.9 Fungsi keanggotaan *input* waktu

Dari Gambar 4.9 dapat diketahui bahwa:

- a. waktu cepat adalah waktu < 260
- b. waktu sedang adalah $150 < \text{waktu} < 640$
- c. waktu lama adalah waktu > 550



Gambar 4.10 Fungsi keanggotaan *output*

Berikut merupakan tabel *rule base* untuk proses pengolahan data.

Tabel 4.7 *Rule Base*

	Low	medium	High
Lama	2	3	4
sedang	1	2	3
cepat	0	2	3

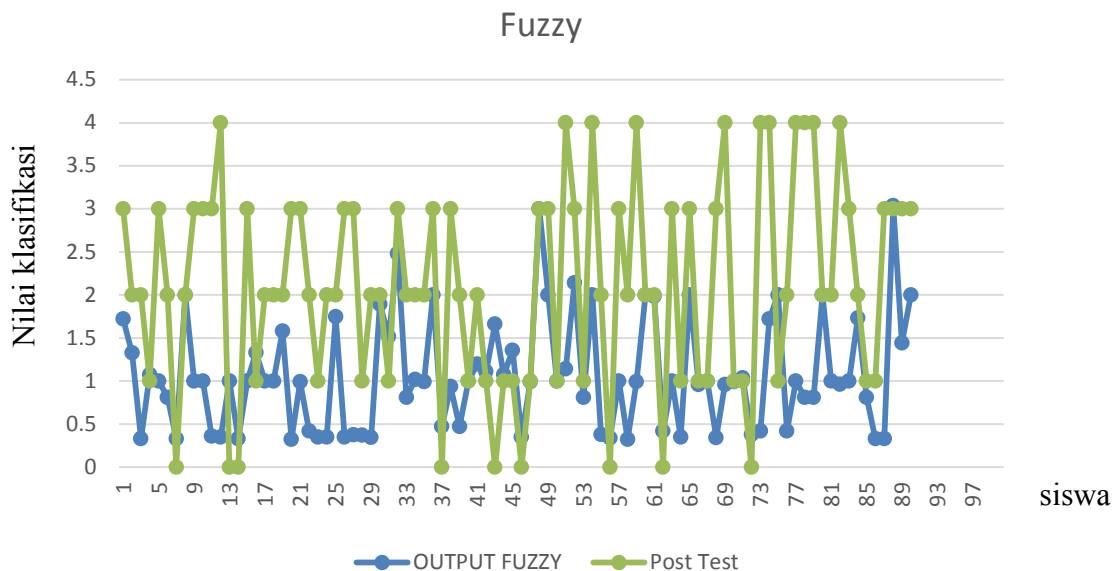
Tabel 4.7 adalah tabel aturan – aturan untuk fuzzy yang dibuat. Ada Sembilan aturan yang dihasilkan dari variasi skor dan waktu permainan siswa dengan 5 jumlah keluaran. Aturan pertama adalah jika skor siswa *low* dan dijawab dengan waktu yang lama, maka *output* = 2. Aturan kedua adalah jika skor siswa *low* dan dijawab dengan waktu yang sedang, maka *output* = 1. Aturan ketiga adalah jika skor siswa *low* dan dijawab dengan waktu yang cepat, maka *output* = 0. Aturan keempat adalah jika skor siswa *medium* dan dijawab dengan waktu yang lama, maka *output* = 3. Aturan kelima adalah jika skor siswa *medium* dan dijawab dengan waktu yang sedang, maka *output* = 2. Aturan keenam adalah jika skor siswa *medium* dan dijawab dengan waktu yang cepat, maka *output* = 2. Aturan ketujuh adalah jika skor siswa *high* dan dijawab dengan waktu yang lama, maka *output* = 4. Aturan kedelapan adalah jika skor siswa *high* dan dijawab dengan waktu yang sedang, maka *output* = 3. Aturan kesembilan adalah jika skor siswa *high* dan dijawab dengan waktu yang cepat, maka *output* = 3. Hasil klasifikasi dengan metode fuzzy dapat dilihat pada Tabel 4.8

Tabel 4.8 Hasil *output* dengan metode fuzzy

username	total skor	total waktu	OUTPUT FUZZY		Post Test
arman2	7	228	1.721393	2	3
ibra1	5	247	1.3252	1	2
yulhaq1	1	28	0.328648	0	2
deni	9	157	1.072666	1	1
almas2	8	136	1	1	3
tajul2	5	40	0.81017	1	2
fad1	1	31	0.328648	0	0
farah4	9	486	2	2	2
maftuh4	9	93	1	1	3
rafli	1	335	1	1	3
abror	0	100	0.357548	0	3
aan	2	36	0.347424	0	4
aNI3	9	135	1	1	0
KHODIJAH3	1	24	0.328648	0	0
amal9	10	40	1	1	3
fiki1	5	256	1.330906	1	1
rafny1	7	109	1	1	2
laily	11	77	1	1	2
GILANG4	20	142	1.5808	2	2
TIVAL1	0	9	0.320845	0	3
kenzu2	6	115	0.992729	1	3
afi1	4	66	0.419025	0	2
irit1	2	58	0.347424	0	1
nina1	0	82	0.347099	0	2
adi4	11	231	1.746093	2	2
arman4	2	36	0.347424	0	3
dinda3	2	123	0.372534	0	3
nur	2	118	0.369137	0	1
rian3	1	77	0.344422	0	2
afi1	12	249	1.891504	2	2
farhat4	22	203	2.480271	2	3
novi8	5	58	0.81017	1	2
tajul1	13	151	1.018124	1	2
ocha3	6	117	0.992729	1	2
firman4	14	411	2	2	3
azra7	1	155	0.471096	0	0
alda4	5	155	0.934175	1	3
wanda5	3	155	0.471096	0	2
andre1	11	100	1	1	1
fifin2	7	170	1.196202	1	2
akbar2	8	161	1.108755	1	1

username	total skor	total waktu	OUTPUT FUZZY		Post Test
ubet1	7	221	1.661457	2	0
vikki2	10	157	1.072666	1	1
andre1	19	124	1.357612	1	1
ary9	2	12	0.347424	0	0
zidan1	6	37	0.992729	1	1
evan4	9	779	3	3	3
nida'6	17	283	2	2	3
zidan3	7	142	1	1	1

Dari data – data pada tabel 4.8, dapat diketahui nilai RMSE adalah sebesar 1.59. Nilai *error* ini lebih besar jika dibandingkan dengan klasifikasi menggunakan jaringan saraf tiruan. Hasil ini dapat terlihat dengan jelas melalui Gambar 4.11.



Gambar 4.11 Perbandingan *output* fuzzy dengan hasil tes tulis

Gambar 4.11 menunjukkan pola perbandingan permainan siswa yang diolah dengan metode fuzzy. Gambar dengan garis biru menunjukkan hasil *output* fuzzy sedangkan garis hijau menunjukkan nilai pos tes siswa. Pola 13 anak pertama menunjukkan bahwa hasil klasifikasi belum mendekati nilai tes tulis siswa. Namun, pada data 41 hingga 89 menunjukkan bahwa hasil klasifikasi dan nilai tes tulis siswa sudah hampir mendekati pola yang sama. Hasil yang dicapai sudah mendekati nilai yang diinginkan namun masih terdapat nilai *error* sehingga hasil penelitian ini dapat dikatakan biasa saja.

BAB 5

KESIMPULAN DAN PENELITIAN SELANJUTNYA

5.1 Kesimpulan

Berdasarkan uji coba dan analisis hasil, maka dapat ditarik beberapa kesimpulan, yaitu:

1. Kemampuan kognitif pembelajar dapat diklasifikasikan dalam penelitian ini menggunakan metode *backpropagation neural network* dengan nilai *error* 1,52.
2. Perbandingan akurasi antara metode NN dengan Fuzzy memiliki nilai *error* yang tidak jauh berbeda. *Error* pada metode jaringan saraf tiruan lebih kecil yaitu 1,52 dibandingkan dengan nilai RMSE menggunakan metode fuzzy sebesar 1,59.
3. Setelah dilakukan normalisasi pada *range* data *input*, hasil klasifikasi metode NN menjadi lebih baik karena nilai *error* yang didapat adalah 0.295.

5.2 Penelitian Selanjutnya

Sebagai saran untuk penelitian selanjutnya, sebaiknya perlu dibuat tes tulis yang lebih banyak untuk mengukur kognitif siswa lebih akurat. Tes tulis ini sangat penting untuk mengetahui kinerja metode klasifikasi.

Penambahan data permainan juga dapat dilakukan agar data *training* menjadi lebih baik. Data dari sekolah atau siswa yang berbeda dapat dijadikan bahan analisis lebih lanjut.

[halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- Amini. (2008). Optimum Learning Rate in Back-Propagation Neural Network for Classification of Satellite Images (IRS-1D). *Scientific Ironoca* , 558-567.
- Bengio, Y., & Grandvalet, Y. (2004). No Unbiased Estimator of the Variance of K-Fold Cross-Validation. *Journal of Machine Learning Research* 5 , 1089-1105.
- Bloom, B. S. ed. et al. (1956). *Taxonomy of Educational Objectives: Handbook 1, Cognitive Domain*. New York: David McKay.
- Kashei, M., & Bijari, M. (2011). Which Methodology is Better for Combining Linear and Nonlinear Models for Time Series Forecasting? *Journal of Industrial and Systems Engineering* , 265-285.
- Karlik, B., & Olgac, A. V. (2011). Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks . *International Journal of Artificial Intelligence And Expert Systems (IJAE)* , 111-122.
- Mitchel, T. M. (1997). *Machine Learning*. Singapura: The McGraw-Hill Companies Inc.
- Refaeilzadeh, P., Tang, L., & Liu, H. (2008). Cross-Validation. *Encyclopedia of Database System*.
- Russell, S. J., & Norvig, P. (2002). *Artificial Intelligence: A Modern Approach (2nd ed)*. New Jersey: Prentice Hall International Inc.
- Sadulloh, Uyoh, dkk. (2011). *Pedagogik*. Bandung. Alfabeta.
- Usman, Moh. Uzer. (1994). *Menjadi Guru Professional*. Bandung. PT Remaja Rosdakarya.
- Utari, Retno (2013). *Taksonomi Bloom Apa dan Bagaimana Menggunakannya?*. Pusklat KNPk.

[halaman ini sengaja dikosongkan]

BIOGRAFI PENULIS



Heny Yuniarti, yang lahir di Palembang pada tanggal 15 Juni 1990, merupakan anak pertama dari tiga bersaudara. Penulis menempuh pendidikan sekolah mulai di SD Xaverius 9 Palembang (1996 – 2002), SMP Negeri 1 Metro (2002-2005), SMK Telkom Sandhy Putra Jakarta (2005-2008), hingga menjadi mahasiswa di PENS (Diploma D4) Surabaya (2008-2012). Pada tahun 2012, Penulis mengikuti program beasiswa *Fresh Graduate* yang memberikan kesempatan untuk melanjutkan studi Magister bebas biaya pendidikan, di Program Pasca Sarjana di jurusan Teknik ElektroITS. Penulis dapat dihubungi melalui e-mail: heny.elka@gmail.com.