

**PREDIKSI JUMLAH PENDAPATAN ASLI DAERAH KABUPATEN
BOYOLALI DENGAN METODE JARINGAN SYARAF TIRUAN
*BACKPROPAGATION***

SKRIPSI

Disusun Oleh :

**Tino Arif Cahya Adinugraha
NIM : 09650119**



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2016**

**PREDIKSI JUMLAH PENDAPATAN ASLI DAERAH
KABUPATEN BOYOLALI DENGAN METODE JARINGAN
SYARAF TIRUAN *BACKPROPAGATION***

SKRIPSI

Diajukan Kepada:

Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh

**TINO ARIF CAHYA ADINUGRAHA
NIM. 09650119**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2016**

HALAMAN PERSETUJUAN

**PREDIKSI JUMLAH PENDAPATAN ASLI DAERAH KABUPATEN
BOYOLALI DENGAN METODE JARINGAN SYARAF TIRUAN
BACKPROPAGATION**

SKRIPSI

Oleh :
Tino Arif Cahya Adinugraha
09650119

Telah Disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Ririen Kusumawati, M.Kom
NIP. 19720309200501200

Fatchurrachman, M.Kom
NIP. 197007312005011002

Malang,

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Malang

Dr. Cahyo Crysdian
NIP. 197404242009011008

HALAMAN PENGESAHAN

PREDIKSI JUMLAH PENDAPATAN ASLI DAERAH KABUPATEN BOYOLALI DENGAN METODE JARINGAN SYARAF TIRUAN *BACKPROPAGATION*

SKRIPSI

Oleh :

Tino Arif Cahya Adi Nugraha

09650119

Telah dipertahankan Di Depan Dewan Penguji
Dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan Untuk
Memperoleh Gelar Sarjana Komputer (S. Kom)

Pada Tanggal,

SUSUNAN DEWAN PENGUJI

TANDA TANGAN

- | | | | |
|-----------------------|-----------------------------------|---|---|
| 1. Penguji Utama | : <u>Zainal Abidin, M.Kom</u> | (|) |
| | NIP. 19760613 200501 1 004 | | |
| 2. Ketua Penguji | : <u>A'la Syaqui, M.Kom</u> | (|) |
| | NIP. 19771201 200801 1 007 | | |
| 3. Sekertaris Penguji | : <u>Ririen Kusumawati, M.Kom</u> | (|) |
| | NIP.19720309 200501 2 002 | | |
| 4. Anggota Penguji | : <u>Fatchurrachman, M.Kom</u> | (|) |
| | NIP.19700731 200501 1 002 | | |

Mengetahui dan Mengesahkan
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian

NIP. 19740424 200901 1 008

HALAMAN PERSEMBAHAN

Karya ini saya persembahkan kepada :

**Kedua orang tua:
Ayahanda Suyatno, M.Ag
Ibunda Emiyati**

**Keluarga:
Adik saya Arifuzaky Septika A
Minasari Putri Kurniawati**

**Special Thank's:
"Aku Dewe"
Anak e Abah Ayub
Kosan Suwante
TI Rongewusongo
Mbliyut Family**

MOTTO



" Berjuang untuk mendapatkan sesuatu bukan menunggu untuk mendapatkannya"

"Yaitu orang-orang yang mengingat Allah sambil berdiri, atau duduk atau dalam keadaan berbaring dan mereka memikirkan tentang penciptaannya langit dan bumi ,seraya berkata "Ya Tuhan kami tiadalah Engkau menciptakan ini sia-sia, Maha Suci Engkau maka peliharalah kami" dari siksa api neraka "

(Q.S. Ali Imran : 191)

"Apa yang kita pikirkan itulah yang akan terjadi. Berfikirlah bisa, niscaya pasti akan bisa, karena Allah bersama sangkaan hamba-Nya"

"Jadikanlah sabar dan sholat sebagai penolongmu. Dan sesungguhnya yang demikian itu sungguh berat, kecuali bagi orang-orang yang khusu' yaitu orang-orang yang mengetahui bahwa mereka akan menemui Tuhannya dan bahwa mereka akan kembali kepada-Nya."

(QS. Al-Baqarah : 45-46)

"Sesungguhnya sesudah kesulitan itu ada kemudahan. Maka apabila kamu Telah selesai (dari sesuatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain. Dan Hanya kepada Tuhanmulah hendaknya kamu berharap."

(QS. Al Insyiroh : 6-8)

KATA PENGANTAR

Bissmillahirrahmanirrahim

Alhamdulillah, puji syukur kehadiran Allah Swt. yang melimpahkan segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang menjadi salah satu syarat mutlak untuk menyelesaikan program studi Teknik Informatika jenjang Strata-1 Universitas Islam Negeri (UIN) Malang.

Dengan segala kerendahan hati, penulis menyadari bahwa dalam menyelesaikan skripsi ini tidak lepas dari peran berbagai pihak yang telah banyak memberikan bantuan, bimbingan dan dorongan. Dalam kesempatan ini penulis ingin mengucapkan terima kasih yang tak terhingga khususnya kepada:

1. Bapak Prof. Dr. H. Mudjia Rahardjo, M.Si , selaku Rektor Universitas Islam Negeri Malang.
2. Bapak Dr. Cahyo Crysdian selaku Ketua Jurusan Teknik Informatika yang selalu memberi saran dan arahan saya dalam menyelesaikan skripsi ini .
3. Ibu Ririen Kusumawati, M.Kom selaku Dosen Pembimbing penulisan skripsi ini yang telah memotivasi, membantu dan memberikan penulis arahan yang baik dan benar dalam menyelesaikan penulisan skripsi ini .
4. Seluruh Dosen Teknik Informatika Universitas Islam Negeri Malang yang telah mengajar penulis, dan memberikan dukungan untuk menyelesaikan penulisan skripsi ini.
5. Bapak Drs. H. Widodo Munir selaku kepala DPPKAD Kab. Boyolali yang telah memberikan izin tempat untuk melaksanakan penelitian ini.

6. Orangtua, Adik tersayang yang telah banyak memberikan doa, motivasi dan dorongan dalam penyelesaian skripsi ini, terutama Bapak H. Suyatno, M.Ag selaku bapak saya yang selalu memompa saya dalam berkuliah , dan Ibu Emiyati selaku ibu saya yang selalu menitipkan namaku dalam tiap doanya.
7. Sahabat-sahabat dan teman-teman di Universitas Islam Negeri (UIN) Malang. Teman-teman TI 2009 , teman-teman Psikologi 2009 , teman-teman BSI 2010 khususnya Radit, Ichwan, Nailur, Roziqin ,yang spesial Carolina, Terima kasih buat kalian semua .
8. Semua pihak yang mungkin belum saya sebutkan dan sahabat-sahabat yang telah membantu penulis hingga terselesaikanya skripsi ini, semoga Allah SWT memberikan balasan yang setimpal atas jasa dan bantuan yang telah diberikan.
Penulis menyadari sepenuhnya bahwa sebagai manusia biasa tentunya tidak akan luput dari kekurangan dan keterbatasan. Maka dengan segenap kerendahan hati, penulis mengharapkan saran dan kritik yang dapat menyempurnakan penulisan ini sehingga dapat bermanfaat dan berguna untuk pengembangan ilmu pengetahuan.

Malang, Mei 2016

Penulis

DAFTAR ISI

HALAMAN JUDUL	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
SURAT PERNYATAAN	v
HALAMAN MOTTO	vi
HALAMAN PERSEMBAHAN	v
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
ABSTRAK	xiv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	4
1.3. Ruang lingkup	5
1.4 Tujuan Penelitian	5
1.5. Manfaat Penelitian	6
BAB II LANDASAN TEORI	7
2.1.Jaringan Syaraf Biologi	7
2.2. Jaringan Syaraf Tiruan	8
2.2.1. Definisi	8
2.2.2. Macam Arsitektur Jaringan Syaraf Tiruan.....	11
2.2.3. Struktur Arsitektur Jaringan Syaraf Tiruan	14

2.2.4. Algoritma Pembelajaran	19
2.3. Backpropagation	21
2.3.1. Defnisi	21
2.3.2. Algoritma Backpropagation	21
2.3.3. Hitung MSE	24
2.4. Pendapatan Asli Daerah (PAD)	26
2.4.1. Definisi	26
2.4.2. Jenis-jenis Pendapatan Asli Daerah.....	27
2.4.3. Dasar Hukum Pendapatan Asli Daerah.....	31
BAB III ANALISA DAN PERANCANGAN SYSTEM.....	33
3.1. Analisa System	33
3.1.1. Deskripsi Umum System	33
3.1.2. Batasan System	36
3.1.3. Analisa Kebutuhan Perangkat Lunak.....	36
3.1.3.1. Analisa Use Case	36
3.1.3.2. Activity Diagram	37
3.2. Perancangan System	38
3.2.1. Prediksi PAD	39
3.2.2. Variabel Data Masukan.....	39
3.2.3. Iniasialisasi Parameter	39
3.2.4 Pembahasan Perhitungan Data Training	43
BAB IV IMPLEMENTASI DAN PEMBAHASAN.....	50
4.1. Lingkungan Uji Coba	50

4.2. Data Uji Coba	51
4.3 .Penjelasan Program	52
4.3.1. Proses Menampilkan Halaman Login.....	52
4.3.2. Proses Prosesing Data.....	52
4.3.3. Proses Learning JST.....	54
4.3.4. Proses Uji Test JST.....	59
4.4 .Hasil Uji Coba	62
4.5 Integrasi Dalil dengan Prediksi PAD	67
BAB V Kesimpulan dan Saran	69
5.1. Kesimpulan	69
5.2. Saran	69
DAFTAR PUSTAKA	
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1	: Sel Syaraf Biologi	8
Gambar 2.2	: Jaringan Syaraf Sederhana	11
Gambar 2.3	: Jaringan Syaraf Tiruan Satu Layer	12
Gambar 2.4	: Jaringan Syaraf Tiruan Multi Layer	13
Gambar 2.5	: Jaringan Syaraf Tiruan Competitive Layer	13
Gambar 2.6	: Jaringan Syaraf Tiruan Hard Limit	14
Gambar 2.7	: Jaringan Syaraf Tiruan Symmetric Hard Limit	15
Gambar 2.8	: Jaringan Syaraf Tiruan Linear	16
Gambar 2.9	: Jaringan Syaraf Tiruan Saturating Linear	17
Gambar 2.10	: Jaringan Syaraf Tiruan Symmetric Saturating Linear	17
Gambar 2.11	: Jaringan Syaraf Tiruan Sigmoid Biner	18
Gambar 2.12	: Jaringan Syaraf Tiruan Sigmoid Bipolar	19
Gambar 2.13	: Contoh Data Pendapatan Asli Daerah Kab.Boyolali	26
Gambar 3.1	: Blok Diagram Proses Umum System	34
Gambar 3.2	: Use Case Diagram System Prediksi	37
Gambar 3.3	: Activity Diagram System Prediksi	38
Gambar 3.4	: Skema Global System	40
Gambar 3.5	: Flowchart System.....	41
Gambar 3.6	: Flowchart Proses Training	42
Gambar 4.1	: Halaman Login	43
Gambar 4.2	: Halaman Proses Perhitungan	49

DAFTAR TABEL

Tabel 1	: Tabel Parameter PAD Data Training	43
Tabel 2	: Tabel Hasil Normalisasi	43
Tabel 3	: Tabel Parameter Data Uji	62
Tabel 4	: Tabel Hasil Prediksi	67



ABSTRAK

Nugraha, Tino.2016. **Prediksi Pendapatan Asli Daerah Kabupaten Boyolali Dengan Menggunakan Jaringan Syaraf Tiruan Backpropagation**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (1)Ririen Kusumawati, M.Kom.(2)Fatchurrachman, M.Kom

Kata Kunci: Prediksi, Pendapatan Asli Daerah,JST,Backpropagation

Pelaksanaan Otonomi daerah memberikan kewenangan kepada daerah untuk mengatur dan mengelola daerahnya sendiri. Masing-masing daerah harus kreatif agar pengelolaan daerahnya lebih terfokus dan mencapai sasaran yang telah ditentukan. perlu dilakukan estimasi Pendapatan Asli Daerah secara benar sehingga dapat diprediksi penerimaan yang sudah ada dan dapat digali, dikembangkan, dan dikelola secara profesional.

Dalam waktu tertentu jumlah penerimaan pendapatan menurut parameternya akan semakin meningkat dan dalam situasi yang lain akan terjadi sebaliknya. Untuk itu perlu adanya proyeksi atau Prediksi terhadap jumlah penerimaan pendapatan untuk tahun selanjutnya.

Penelitian ini membahas mengenai Prediksi Pendapatan Asli Daerah Kabupaten Boyolali menggunakan Jaringan Syaraf Tiruan *Backpropagation*. *Backpropagation* merupakan teknik prediksi berbasis model matematis yang dapat membuat model fungsi dari perubahan jumlah variabel yang dalam penelitian ini adalah Pendapatan Asli Daerah khususnya Kabupaten Boyolali sebagai hasil acuan perencanaan anggaran pada periode selanjutnya.

ABSTRACT

Nugraha, Tino. 2016. **Prediction Of Original Income Area Of Boyolali Regency Using Backpropagation Neural Network** .Thesis . Department of Computer Engineering Faculty of Science and Technology in the Islamic State University Maulana Malik Ibrahim Malang. Supervisor: (1) Ririen Kusumawati, M. Kom. (2) Fatchurrachman, M. Kom

Keywords: Prediction, The Original Income Areas, JST, Backpropagation

The implementation of regional Autonomy provides the authority to the regions to set up and manage their own regions. Each region must be creative in order to make the management of its territory is more focused and achieve the objectives that have been determined. the original Revenue estimate needs to be done correctly so that the Area can be unpredictable acceptance that already exist and can be dug up, developed, and managed professionally.

Certain amount of time in receipt of an income according to its parameters will be increasing and in other situations will happen otherwise. For it is need for projections or Predictions against the sum of the receipt of income for years to come.

This research deals with the original Revenue Prediction Area Kabupaten Neural Network using Boyolali Backpropagation. Backpropagation is the technique-based prediction mathematical model that can model a function of the number of variable changes in this research is Original Income Region in particular as a result of reference of Boyolali district budget planning in the next period.

خلاصة

نوغراها، تينو-٦١٠٢ . التنبؤ بمنطقة الدخل الأصلي ريجنسي بويولالي باستخدام الشبكات العصبية. أطروحة. ومن المؤسف إدارة الكمبيوتر كلية الهندسة للعلوم والتكنولوجيا في جامعة الدولة الإسلامية مولانا مالك إبراهيم. المشرف: (١) ريرين كوسوماواتي، م. كوم. (٢) فاثشوراتشمان، م. كوم

الكلمات الرئيسية : التنبؤ، وفي مجالات الدخل الأصلي، الشبكات العصبية

تنفيذ الحكم الذاتي الإقليمي يوفر السلطة إلى المناطق لإعداد وإدارة مناطقها. يجب أن تكون كل منطقة خلاقة من أجل جعل إدارة أراضيها هو أكثر تركيزاً وتحقيق الأهداف التي تحددت. احتياجات تقدير الإيرادات الأصلي ينبغي القيام به بشكل صحيح حتى تكون المنطقة القبول لا يمكن التنبؤ بها موجودة بالفعل ويمكن أن تكون عمليات حفر، البلدان المتقدمة النمو، وتدار مهنيًا

سيتم زيادة مقدار معين من الوقت في استلام الدخل وفقا لمعاييرها وفي حالات أخرى سوف يحدث خلاف ذلك. أنه لا حاجة للتوقعات أو التنبؤات ضد مجموع تلقي الإيرادات لسنوات قادمة

ويتناول هذا البحث الأصل "الإيرادات التنبؤ كابوبيتين العصبية شبكة المنطقة" بويولالي باستخدام. تقنية الشبكات العصبية على أساس التنبؤ هو نموذج رياضي نموذج لعدد من التغيرات في وظيفة فيايبيل في هذا البحث هو "الأصلي دخل المنطقة" وبخاصة نتيجة للإشارة لميزانية منطقة بويولالي التخطيط في الفترة القادمة

BAB I

PENDAHULUAN

1.1 Latar Belakang

Manusia itu tidak dapat mengetahui dengan pasti apa yang akan diusahakannya besok atau yang akan diperolehnya. Namun demikian manusia diwajibkan berusaha sebagaimana Allah berfirman:

إِنَّ اللَّهَ عِنْدَهُ عِلْمُ السَّاعَةِ وَيُنَزِّلُ الْغَيْثَ وَيَعْلَمُ مَا فِي الْأَرْحَامِ وَمَا تَدْرِي
نَفْسٌ مَّاذَا تَكْسِبُ غَدًا وَمَا تَدْرِي نَفْسٌ بِأَيِّ أَرْضٍ تَمُوتُ إِنَّ اللَّهَ عَلِيمٌ خَبِيرٌ



“Sesungguhnya Allah, hanya pada sisi-Nya sajalah pengetahuan tentang hari Kiamat; dan Dia-lah yang menurunkan hujan, dan mengetahui apa yang ada dalam rahim dan tiada seorangpun yang dapat mengetahui (dengan pasti) apa yang akan diusahakannya besok dan tiada seorangpun yang dapat mengetahui di bumi mana Dia akan mati. Sesungguhnya Allah Maha mengetahui lagi Maha mengenal” (QS. Al Lukman/31 : 34)

Menurut buku Tafsir Al Qur'an jilid VII , Pada ayat ini Allah SWT menerangkan lima perkara gaib yang hanya Allah sendirilah yang mengetahui perkara itu : salah satunya dijelaskan. “Hanya Dia pula yang mengetahui dengan pasti apa yang akan dikerjakan oleh seseorang esok harinya. Sekalipun manusia dapat merencanakan apa yang akan dikerjakannya itu , namun semuanya itu bersifat rencana saja. Jika Allah menghendaki terlaksananya ,terlaksanalah dia ,dalam pada itu tidak sukar bagi Allah untuk menghalangi terlaksananya.”

Ayat tersebut menerangkan supaya dimengerti bahwa manusia tidak mengerti akhir dari semua perkara yang terjadi. Secara umum, semua aktifitas yang dilakukan

manusia didasarkan pada prediksi dan pengambilan keputusan di dalam situasi yang mengandung ketidakpastian. Akan tetapi manusia diwajibkan selalu berusaha dan berhati-hati untuk mengestimasi apa yang akan terjadi esok agar terhindar dari hal-hal yang diinginkan. Termasuk di dalamnya masalah jumlah pendapatan daerah yang akan datang.

Pelaksanaan otonomi daerah memberikan kewenangan kepada daerah untuk mengatur dan mengelola daerahnya sendiri. Masing-masing daerah harus kreatif agar pengelolaan daerahnya lebih terfokus dan mencapai sasaran yang telah ditentukan. Dalam rangka menjalankan fungsi dan kewenangan Pemerintah daerah dalam bentuk pelaksanaan kewenangan fiskal, daerah harus dapat mengenali potensi dan mengidentifikasi sumber-sumber daya yang dimilikinya.

Penerimaan Pendapatan Asli Daerah (PAD) yang salah satu contohnya pajak dapat ditingkatkan dengan cara intensifikasi dan ekstensifikasi. Ekstensifikasi adalah perluasan jenis pajak. Namun beberapa studi menunjukkan bahwa penggalian potensi dengan memperluas jenis pajak tidak merangsang minat bahkan menimbulkan keengganan para investor untuk menanam modal di daerah tersebut. Intensifikasi adalah upaya peningkatan pengoleksian pajak. Upaya ini menuntut kemampuan daerah untuk dapat mengidentifikasi secara benar penerimaan pendapatan daerah dan kemampuan pengoleksian pajak berdasarkan manfaat dan prinsip keadilan.

Dengan demikian, perlu dilakukan estimasi Pendapatan Asli Daerah (PAD) secara benar sehingga dapat diprediksi penerimaan yang sudah ada dan dapat digali, dikembangkan, dan dikelola secara profesional.

Dalam waktu tertentu jumlah penerimaan pendapatan akan semakin meningkat dan dalam situasi yang lain akan terjadi sebaliknya. Untuk itu perlu

adanya proyeksi atau Prediksi terhadap jumlah penerimaan pendapatan untuk tahun selanjutnya. Prediksi atau juga disebut *forecasting* adalah suatu bagian yang dipertimbangkan untuk membuat perencanaan. Salah satu aspek penting perencanaan adalah pembuatan keputusan (*decision making*), proses pengembangan dan penyeleksian sekumpulan kegiatan-kegiatan untuk memecahkan masalah.

Tujuan perencanaan adalah melihat bahwa program-program yang telah dilaksanakan dapat digunakan untuk meningkatkan pembuatan keputusan yang baik. Suatu perencanaan *forecast* yang tepat adalah memperkirakan apa yang akan terjadi di waktu yang akan datang. Perubahan suatu kejadian dapat dinyatakan dengan perubahan nilai variabel.

Hasil penerimaan Pendapatan Asli Daerah (PAD) adalah salah satu contoh nilai variabel yang selalu berubah, apabila nilai variabel ini tetap dari waktu ke waktu maka mudah sekali untuk memprediksi, akan tetapi pada kenyataannya, nilai tersebut selalu berubah. Itulah sebabnya prediksi tentang suatu nilai variabel selalu tidak tepat pada kenyataan. Dalam pemilihan metode *forecast* dibutuhkan ketepatan metode guna meminimumkan kesalahan dalam memprediksi (*forecast error*), tujuannya agar *forecast* bisa mendekati kenyataan.

Terdapat dua metode yang dapat digunakan dalam implementasi prediksi yaitu metode konvensional dan Jaringan Syaraf Tiruan (JST). Metode konvensional berhubungan dengan pengidentifikasian dan pemodelan. Pengidentifikasian melakukan pengamatan besaran yang keluar dari proses untuk masukan yang telah diketahui sebelumnya. Sedangkan pemodelan memerlukan suatu aturan yang menjelaskan tingkah laku dari proses tersebut. Model dapat berupa gambar, warna,

persamaan matematis dan sebagainya. Untuk keperluan prediksi, model matematis lebih diinginkan daripada model-model lainnya.

Seringkali model matematis dari suatu masalah sangat sukar untuk dibuat dan yang ada hanya data. Seperti model fungsi perubahan jumlah dari variabel Pendapatan Asli Daerah itu sendiri yang tidak menentu. Jaringan Syaraf Tiruan *Backpropagation* menyaring informasi tersebut melalui pelatihan. Oleh karena itu, Jaringan Syaraf Tiruan *Backpropagation* tepat untuk menyelesaikan masalah prediksi Pendapatan Asli Daerah.

Berdasarkan atas penjelasan di atas maka dari itu penyusun mengambil judul penelitian “PREDIKSI PENDAPATAN ASLI DAERAH KABUPATEN BOYOLALI MENGGUNAKAN JARINGAN SYARAF TIRUAN BACKPROPAGATION”.

1.2 Rumusan Masalah

Berdasarkan Latar Belakang di atas dapat di rumuskan sebagai berikut,

1. Bagaimana merancang dan membuat perangkat lunak prediksi dengan menggunakan Jaringan Syaraf Tiruan *Backpropagation* pada studi kasus Pendapatan Asli Daerah Kabupaten Boyolali
2. Bagaimana mengetahui tingkat keakurasian berdasarkan hasil yang didapat dari perhitungan Jaringan Syaraf Tiruan *Backpropagation* terhadap hasil sebenarnya, dalam memprediksi nilai Pendapatan Asli Daerah Kabupaten Boyolali.

1.3 Ruang Lingkup

Penelitian ini memiliki ruang lingkup:

1. Prediksi Pendapatan Asli Daerah Kabupaten Boyolali di dapatkan melalui perhitungan Jaringan Syaraf Tiruan dengan menggunakan metode *backpropagation*.
2. Studi kasus dilakukan di Dinas Pendapatan dan Pengelolaan Keuangan Aset Daerah (DPPKAD) Kabupaten Boyolali
3. Aplikasi Jaringan Syaraf Tiruan *backpropagation* dibangun menggunakan perangkat lunak Netbeans 6.5

1.4 Tujuan Penelitian

1. Adapun tujuan dari penelitian ini adalah merancang dan membuat perangkat lunak prediksi dengan menggunakan Jaringan Syaraf Tiruan *Backpropagation* pada studi kasus Pendapatan Asli Daerah Kabupaten Boyolali
2. Untuk mengetahui tingkat keakurasian berdasarkan hasil yang didapat dari perhitungan Jaringan Syaraf Tiruan *Backpropagation* terhadap hasil sebenarnya, dalam memprediksi nilai Pendapatan Asli Daerah Kabupaten Boyolali.

1.5 Manfaat Penelitian

Manfaat yang dapat diambil dari pelaksanaan tugas akhir ini antara lain:

1. Untuk mengetahui penggunaan Jaringan Syaraf Tiruan *backpropagation* pada kasus *forecast* atau prediksi.
2. Untuk mengetahui seberapa tingkat keakurasian prediksi berdasarkan nilai error yang dihasilkan pada perhitungan Jaringan Syaraf Tiruan *backpropagation* ,studi kasus Pendapatan Asli Daerah Kabupaten Boyolali.

3. Sebagai tolok ukur untuk mengetahui berhasil tidaknya pemerintah daerah mengelola sumber daya miliknya khususnya pada Pendapatan Asli Daerah Kabupaten Boyolali.
4. Sebagai salah satu alat pengambil keputusan dalam menentukan anggaran daerah pemerintah Kabupaten Boyolali



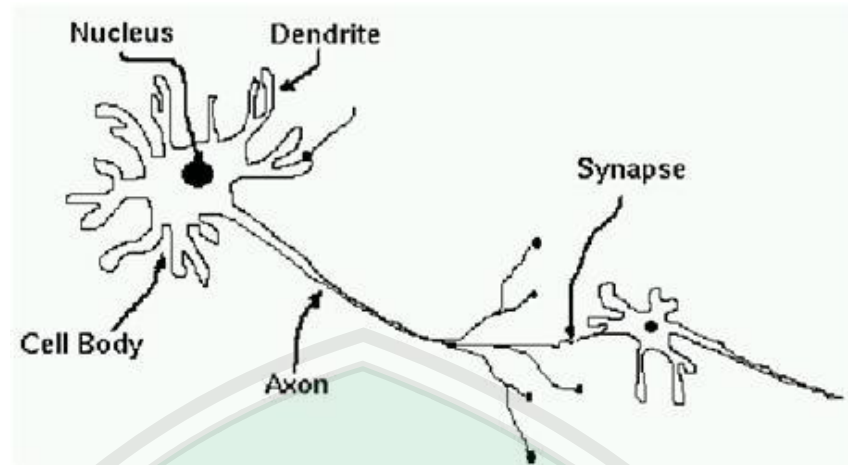
BAB II

LANDASAN TEORI

1.6 Jaringan Syaraf Biologi

Jaringan syaraf adalah merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan disini digunakan karena jaringan syaraf ini di implementasikan dengan menggunakan program computer yang mampu menyelesaikan sejumlah proses pembelajaran (Fausett, 1994).

Otak manusia memiliki struktur yang sangat kompleks dan memiliki kemampuan yang luar biasa. Otak manusia terdiri dari neuron-neuron dan penghubung yang disebut sinapsis. Neuron bekerja berdasarkan sinyal yang diberikan neuron lain dan meneruskannya pada neuron lainnya. Diperkirakan manusia memiliki 10^{11} neuron. Masing-masing neuron saling berhubungan dengan jumlah hubungan tersebut mencapai 10^4 buah per neuron. Jadi jumlah koneksi untuk setiap neuron adalah 10^{15} buah. Neuron-neuron tersebut dapat bekerja secara paralel dengan kecepatan luar biasa. Dengan jumlah yang begitu banyak, maka otak manusia mampu mengenali pola, melakukan perhitungan, serta mengontrol organ-organ tubuh dengan baik. Neuron memiliki 3 komponen utama, yaitu dendrite, badan sel, (soma) dan akson. Dendrite berfungsi menerima sinyal informasi dari satu atau beberapa neuron yang terhubung. Sinyal yang diterima oleh dendrite diteruskan ke badan sel. Jika total sinyal yang diterima oleh badan sel cukup kuat untuk mengaktifkan neuron, maka neuron tersebut akan mengirimkan sinyal ke semua neuron terhubung melalui akson. Jadi semua neuron hanya memiliki dua kemungkinan yaitu mengirimkan sinyal kepada neuron lain atau tidak.



Gambar 2.1 Sel Syaraf Biologi

Sumber: *Fundamentals of Neural Networks*.1994

1.7 Jaringan Syaraf Tiruan

2.2.1 Definisi

Artificial Neural Network (Jaringan Syaraf Tiruan/ANN) adalah sebuah bentuk perhitungan yang meniru bentuk jaringan syaraf pada makhluk hidup. Sama seperti pada jaringan syaraf asli, jaringan syaraf tiruan memiliki neuron untuk memproses input/output. Karena itulah, pada sebagian besar kasus ,ANN merupakan sistem yang adaptif. Karena struktur sistemnya dapat berubah seiring dengan berubahnya informasi internal ataupun eksternal yang diproses pada saat *learning phase* (tahap pembelajaran). Pembuatan sistem ANN dimaksudkan agar computer dapat mengenali suatu pola, bentuk, atau struktur tertentu karena komputer tidak mempunyai intelegensia, meskipun pada kenyataannya, sebuah komputer dapat melakukan operasi, misalnya mengenalo wajah manusia, dalam waktu yang jauh lebih singkat daripada manusia.

Beberapa definisi tentang jaringan saraf tiruan adalah sebagai berikut.

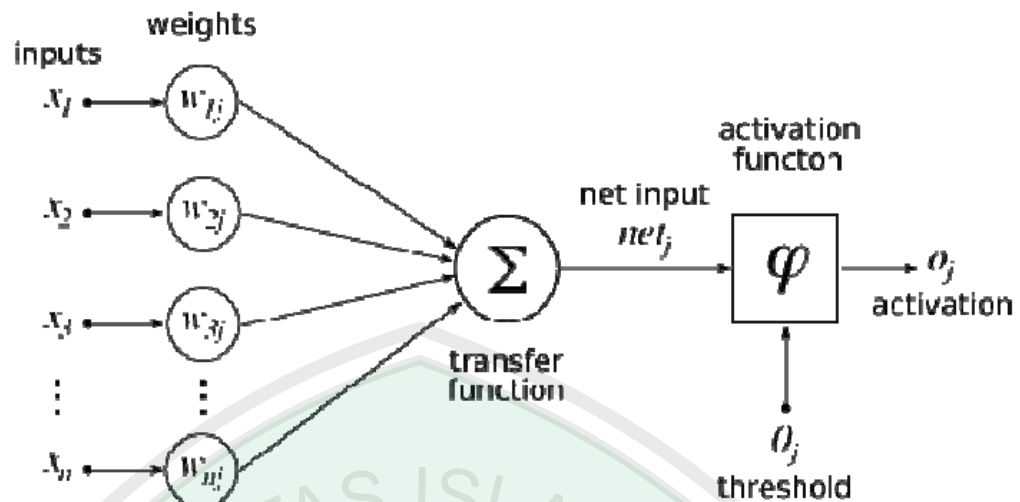
- Hecht-Nielsen (1988) mendefinisikan sistem saraf buatan sebagai berikut.
“Suatu *neural network* (NN), adalah suatu struktur pemroses informasi yang terdistribusi dan bekerja secara paralel, yang terdiri atas elemen pemroses (yang memiliki memori lokal dan beroperasi dengan informasi lokal) yang diinterkoneksi bersama dengan alur sinyal searah yang disebut koneksi. Setiap elemen pemroses memiliki koneksi keluaran tunggal yang bercabang (*fan out*) ke sejumlah koneksi kolateral yang diinginkan (setiap koneksi membawa sinyal yang sama dari keluaran elemen pemroses tersebut). Keluaran dari elemen pemroses tersebut dapat merupakan sembarang jenis persamaan matematis yang diinginkan. Seluruh proses yang berlangsung pada setiap elemen pemroses harus benar-benar dilakukan secara lokal, yaitu keluaran hanya bergantung pada nilai masukan pada saat itu yang diperoleh melalui koneksi dan nilai yang tersimpan pada memori lokal”.
- Menurut Zurada, J.M. (1992), *Introduction To Artificial neural Systems*, Boston: PWS Publishing Company, mendefinisikan sebagai berikut. “Sistem saraf tiruan atau jaringan saraf tiruan adalah sistem selular fisik yang dapat memperoleh, menyimpan dan menggunakan pengetahuan yang didapatkan dari pengalaman”.
- Menurut Haykin, S. (1994), *Neural Networks: A Comprehensive Foundation*, NY. Macmillan, mendefinisikan jaringan saraf sebagai berikut: “Sebuah jaringan saraf adalah sebuah prosesor yang terdistribusi paralel dan mempunyai kecenderungan untuk menyimpan pengetahuan yang didapkannya dari pengalaman dan membuatnya tetap tersedia untuk digunakan. Hal ini menyerupai kerja otak dalam dua hal yaitu pengetahuan diperoleh oleh jaringan

melalui suatu proses belajar dan kekuatan hubungan antar sel saraf yang dikenal dengan bobot sinapsis digunakan untuk menyimpan pengetahuan.

- *DARPA Neural Network Study* (1988, *AFCEA International Press*, halaman 60) mendefinisikan jaringan syaraf buatan sebagai berikut. “Sebuah jaringan syaraf adalah sebuah sistem yang dibentuk dari sejumlah elemen pemroses sederhana yang bekerja secara parallel, yang fungsinya ditentukan oleh struktur jaringan, kekuatan hubungan, dan pengolahan dilakukan pada komputasi elemen atau *nodes*”.

Biasanya, sebuah model neuron untuk ANN terdiri atas tiga bagian berikut.

- 1) *Synapsis* (Jalur penghubung) antara neuron yang masing-masing memiliki weight (bobot). Tiap Synapsis memiliki indeks tertentu untuk menunjukkan input mana yang akan diproses menjadi output.
- 2) *Summing Unit* untuk menghitung total input.
- 3) *Activation Function* (Fungsi Aktivasi) untuk membatasi output.



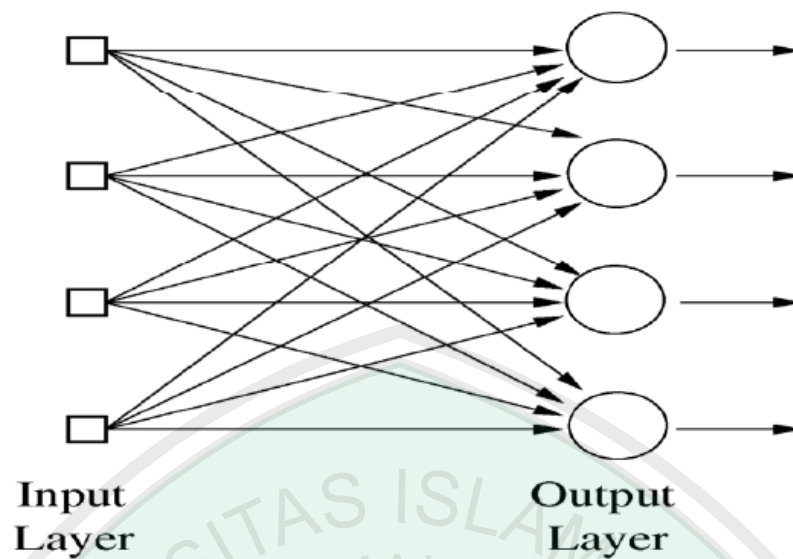
Gambar 2.2 Jaringan Syaraf Sederhana

Sumber: *Fundamentals of Neural Networks*.1994

Arsitektur ANN dapat dibedakan menjadi 3 macam sebagai berikut.

1) Jaringan syaraf dengan lapisan tunggal (*single layer net*).

Jaringan dengan lapisan tunggal hanya memiliki satu lapisan dengan bobot-bobot terhubung. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi. Dengan kata lain, ciri-ciri dari arsitektur syaraf dengan lapisan tunggal adalah hanya terdiri dari satu lapisan input dan lapisan output, tanpa lapisan tersembunyi.

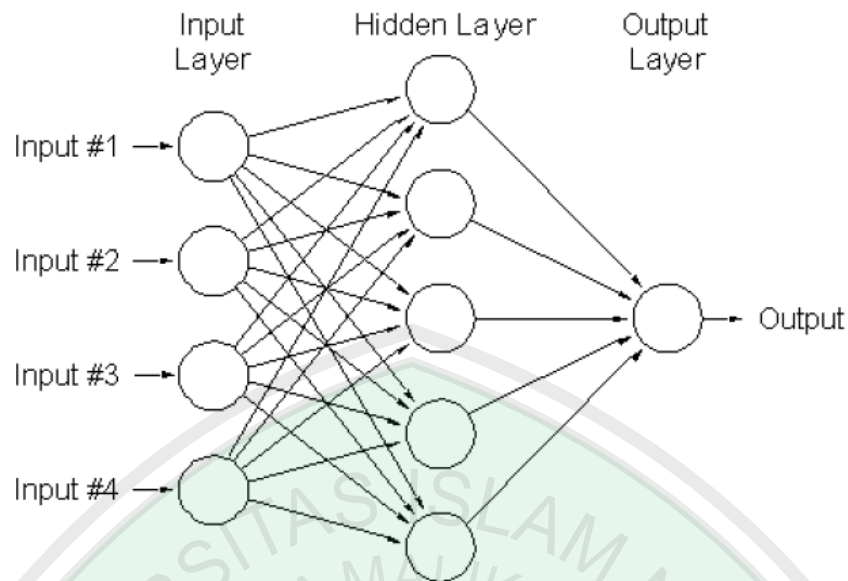


Gambar 2.3 Jaringan Syaraf Tiruan Satu Layer

Sumber: *Neuro-Fuzzy Integrasi Sistem Fuzzy & Jaringan Syaraf*, 2010

2) Jaringan syaraf dengan banyak lapisan (*multi layer net*).

Jaringan dengan banyak lapisan memiliki satu atau lebih lapisan yang terletak di antara lapisan input dan lapisan output (memiliki satu atau lebih lapisan tersembunyi). Umumnya ada lapisan bobot-bobot yang terletak di antara 2 lapisan yang bersebelahan. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit daripada jaringan dengan lapisan tunggal, tentu saja dengan pembelajaran yang lebih rumit.

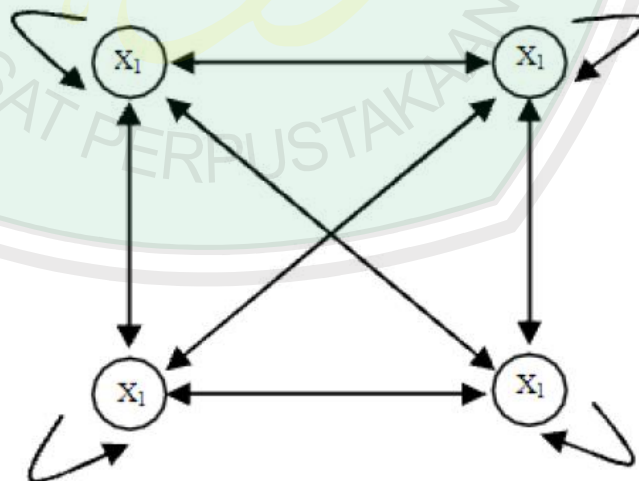


Gambar 2.4 Jaringan Syaraf Tiruan Multi Layer

Sumber: *Neuro-Fuzzy Integrasi Sistem Fuzzy & Jaringan Syaraf*,2010

3) Jaringan syaraf dengan lapisan kompetitif (*competitive layer*)

Arsitektur ini memiliki bentuk berbeda dari kedua arsitektur lainnya, dimana antar neuron saling dihubungkan. Jaringan ini sering disebut *feedback loop* karena unit *output* ada yang memberikan informasi terhadap unit masukan.



Gambar 2.5 Jaringan Syaraf Tiruan *Competitive Layer*

Sumber: *Neuro-Fuzzy Integrasi Sistem Fuzzy & Jaringan Syaraf*,2010

Struktur arsitektur dari neural network pada umumnya terdiri dari 3 layer yang menjadi ciri khas dari sistem ini yaitu

- Input Layer

Merupakan data yang kita masukkan sebagai data training pada sistem ANN. Banyaknya jumlah node pada input layer tergantung pada jumlah data input yang telah kita masukkan ke dalam sistem.

- Hidden Layer

Jumlah node pada hidden layer sangat bervariasi tergantung penggunaannya. Semakin banyak hidden layer yang digunakan semakin baik hasilnya, tetapi training yang dibutuhkan juga semakin lama. Jumlah node yang terlalu sedikit menyebabkan jaringan tidak cukup fleksibel untuk mempelajari data dan jika jumlah hidden node terlalu besar akan terjadi overfitting. Dan menurut Saludin Muis, tingkat ketelitian yang dihasilkan cukup memadai jika memakai 1 lapis hidden layer, 1 input layer dan 1 output layer.

- Output Layer

Banyaknya node pada output layer tergantung pada sistem ANN itu sendiri. Data output juga termasuk dari data training yang sebelumnya telah diberikan.

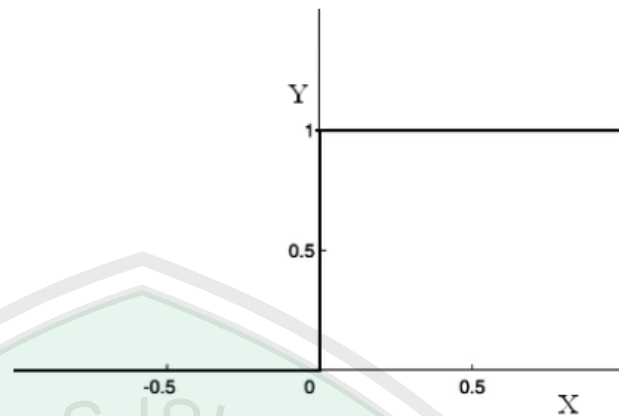
Beberapa fungsi aktivasi yang sering digunakan dalam ANN adalah.

- 1) Fungsi undak biner (*hard limit*)

Jaringan dengan lapisan tunggal sering menggunakan fungsi undak (*step function*) untuk mengkonversikan input dari suatu variabel yang bernilai kontinu ke suatu output biner (0 atau 1). Fungsi undak biner (*hard limit*) dirumuskan sebagai (Demuth, 1998):

$$y = 0, \text{ jika } x < 0$$

$y = 1$, jika $x \geq 0$



Gambar 2.6 Jaringan Syaraf Tiruan *Hard Limit*

Sumber: *Neuro-Fuzzy Integrasi Sistem Fuzzy & Jaringan*

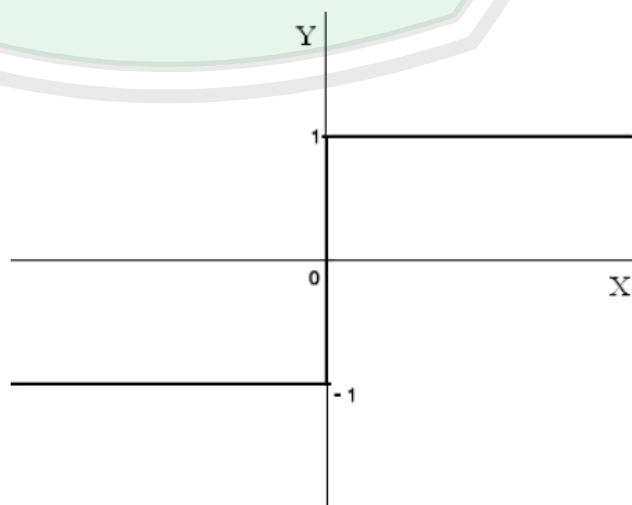
Syaraf, 2010

2) Fungsi bipolar (*symmetric hard limit*)

Fungsi bipolar hampir sama dengan fungsi undak biner, hanya saja output yang dihasilkan berupa 1, 0, atau -1. Fungsi *Symmetric Hard Limit* dirumuskan sebagai berikut (Demuth, 1998):

$y = 1$, jika $x \geq 0$

$y = -1$, jika $x < 0$



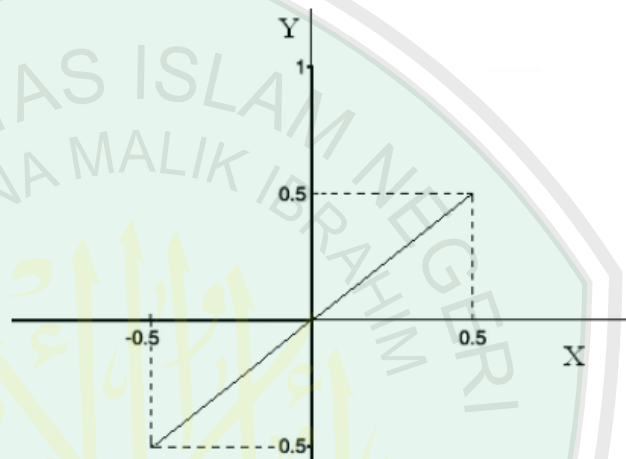
Gambar 2.7 Jaringan Syaraf Tiruan *Symmetric Hard Limit*

Sumber: *Neuro-Fuzzy Integrasi Sistem Fuzzy & Jaringan Syaraf*,2010

3) Fungsi linear (identitas)

Fungsi linear memiliki nilai output yang sama dengan nilai inputnya. Fungsi linear dirumuskan sebagai (Demuth, 1998):

$$y = x$$



Gambar 2.8 Jaringan Syaraf Tiruan Linear

Sumber: *Neuro-Fuzzy Integrasi Sistem Fuzzy & Jaringan Syaraf*,2010

4) Fungsi saturating linear

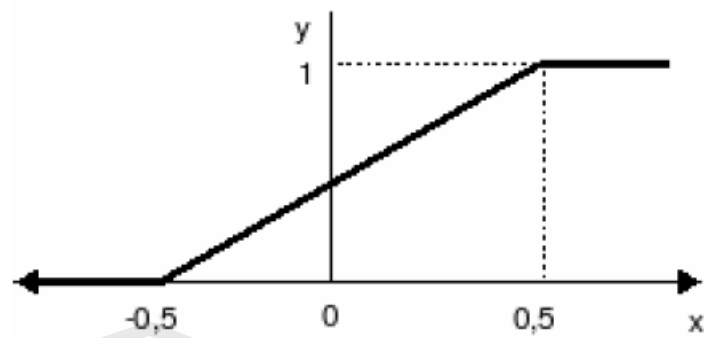
Fungsi ini akan bernilai 0 jika inputnya kurang dari -0,5 dan akan bernilai 1 jika inputnya lebih dari 0,5. Sedangkan jika nilai input terletak diantara -0,5 dan 0,5, maka outputnya akan bernilai sama dengan nilai input ditambah 0,5.

Fungsi *saturating linear* dirumuskan sebagai (Demuth, 1998):

$$y = 1, \text{ jika } x \geq 0,5$$

$$y = x + 0,5, \text{ jika } -0,5 \leq x \leq 0,5$$

$$y = 0, \text{ jika } x \leq -0,5$$



Gambar 2.9 Jaringan Syaraf Tiruan *Saturating Linear*

Sumber: *Neuro-Fuzzy Integrasi Sistem Fuzzy & Jaringan*

Syaraf, 2010

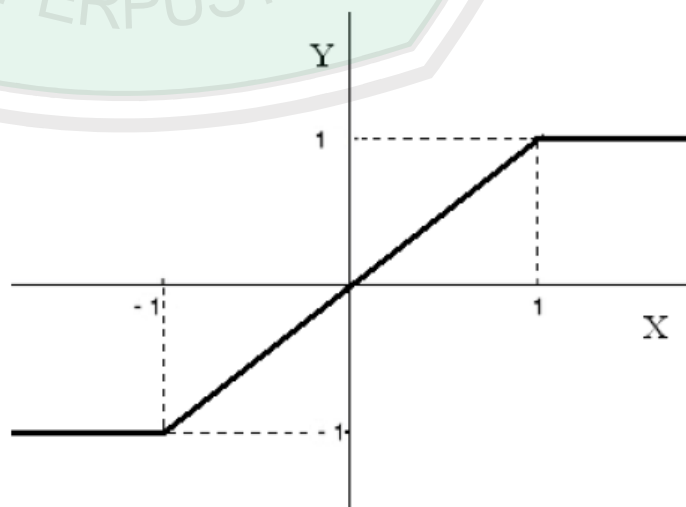
5) Fungsi *symmetric saturating linear*

Fungsi ini akan bernilai -1 jika inputnya kurang dari -1 dan akan bernilai 1 jika inputnya lebih dari 1. Sedangkan jika nilai input terletak antara -1 dan 1, maka outputnya akan bernilai sama dengan nilai inputnya. Fungsi *symmetric saturating linear* dirumuskan sebagai (Demuth, 1998):

$$y = 1, \text{ jika } x \geq 1$$

$$y = x, \text{ jika } -1 \leq x \leq 1$$

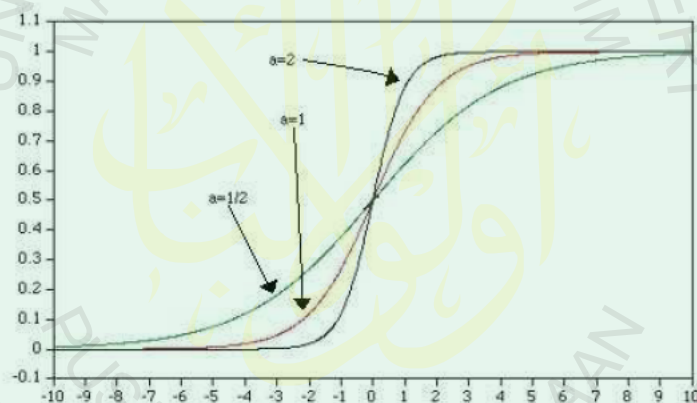
$$u = -1, \text{ jika } x \leq -1$$



Gambar 2.10 Jaringan Syaraf Tiruan *Symmetric Saturating Linear*Sumber: *Neuro-Fuzzy Integrasi Sistem Fuzzy & Jaringan Syaraf*, 20106) Fungsi *Sigmoid biner*

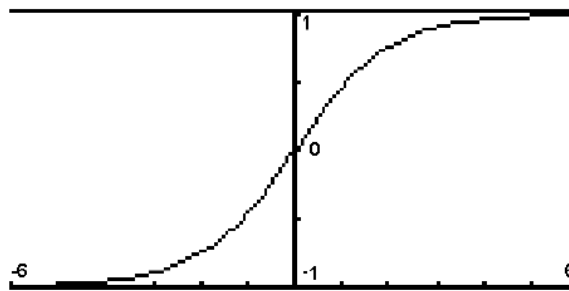
Fungsi ini digunakan untuk jaringan syaraf yang dilatih dengan menggunakan metode *backpropagation* . fungsi *sigmoid biner* memiliki nilai pada range 0 sampai 1. Oleh karena itu fungsi ini sering digunakan untuk jaringan syaraf yang membutuhkan nilai output yang terletak pada interval 0 sampai 1 . Namun, fungsi ini bias juga digunakan oleh jaringan syaraf yang nilai outputnya 0 atau 1. Fungsi *sigmoid biner* dirumuskan sebagai (Demuth, 1998):

$$y = f(x) = \frac{1}{1+e^{-\sigma x}} \text{ dengan } f'(x) = \sigma f(x)[1 - f(x)]$$

**Gambar 2.11** Jaringan Syaraf Tiruan *Sigmoid Biner*Sumber: *Neuro-Fuzzy Integrasi Sistem Fuzzy & Jaringan Syaraf*, 20107) Fungsi *sigmoid bipolar*

Fungsi *sigmoid bipolar* hamper sama dengan fungsi *sigmoid biner* , hanya saja output dari fungsi ini memiliki range antara -1 sampai 1. Fungsi *sigmoid bipolar* dirumuskan sebagai (Demuth, 1998):

$$y = f(x) = \frac{1-e^{-\sigma x}}{1+e^{-\sigma x}} \text{ dengan } f'(x) = \frac{\sigma}{2} [1 + f(x)][1 - f(x)]$$



Gambar 2.12 Jaringan Syaraf Tiruan *Sigmoid Bipolar*
 Sumber: *Neuro-Fuzzy Integrasi Sistem Fuzzy & Jaringan Syaraf*, 2010

Algoritma Pembelajaran

Salah satu bagian terpenting dari konsep jaringan syaraf tiruan adalah proses pembelajaran. Tujuan utama dari proses ini adalah melakukan pengaturan terhadap bobot-bobot yang ada pada jaringan syaraf, hingga didapat bobot akhir yang tepat sesuai dengan pola data yang dilatih. Selama proses pembelajaran, akan terjadi perbaikan bobot-bobot berdasarkan algoritma tertentu. Nilai bobot akan bertambah jika informasi yang diberikan ke neuron yang bersangkutan tersampaikan. Begitu pula sebaliknya, jika informasi tidak disampaikan oleh suatu neuron ke neuron yang lain, maka nilai bobot yang menghubungkan keduanya akan dikurangi. Pada saat pembelajaran dilakukan pada input yang berbeda, nilai bobot akan diubah secara dinamis hingga mencapai nilai yang cukup seimbang. Apabila nilai ini telah tercapai, berarti tiap-tiap input telah berhubungan dengan output yang diharapkan.

Pada dasarnya ada dua metode pembelajaran, yaitu metode pembelajaran terawasi (*supervised learning*) dan metode pembelajaran yang tidak terawasi (*unsupervised learning*).

1) Pembelajaran terawasi (*supervised learning*)

Metode pembelajaran pada jaringan syaraf disebut terawasi jika output yang diharapkan telah diketahui sebelumnya. Misalkan jaringan syaraf akan digunakan untuk mengenali pasangan pola, misalkan pada operasi logika matematika AND. Pada proses pembelajaran, satu pola input akan diberikan ke satu neuron pada lapisan input. Pola ini akan dirambatkan ke sepanjang jaringan syaraf hingga sampai ke neuron pada lapisan output. Lapisan output ini akan membangkitkan pola output yang nantinya akan dicocokkan dengan pola output targetnya. Apabila terjadi perbedaan antara pola output hasil pembelajaran dengan pola target, maka akan muncul *error*. Apabila nilai *error* ini masih cukup besar, berarti perlu dilakukan lebih banyak pembelajaran lagi. Model-model pembelajaran yang menggunakan *supervised learning* di antaranya adalah *Backpropagation*, *Bidirective Associative Memory (BAM)*, dan *Hopfield*.

2) Pembelajaran tak terawasi (*unsupervised learning*)

Metode pembelajaran yang tak terawasi tidak memerlukan target output. Pada metode ini, tidak dapat ditentukan hasil yang seperti apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu range tertentu tergantung pada nilai input yang diberikan. Tujuan pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pembelajaran ini biasanya cocok untuk pengelompokan (klasifikasi) pola. Model-model pembelajaran yang menggunakan *unsupervised learning* diantaranya adalah *Adaptive Resonance Theory (ART)* dan *Competitive*.

1.8 Backpropagation

2.3.1 Definisi

Backpropagation merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya

Algoritma *backpropagation* menggunakan *error* output untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan nilai *error* ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi yang dapat dideferensikan, seperti *sigmoid* (Fausett, 1994).

Algoritma Backpropagation (Fausett, 1994)

- 1) Inisialisasi bobot (ambil bobot awal dengan nilai random yang cukup kecil).
- 2) Tetapkan: maksimum *Epoch*, *Target Error*, dan *Learning Rate* (σ).
- 3) Inisialisasi: *Epoch* = 0.
- 4) Selama (*Epoch* < Maksimum *Epoch*) dan ($MSE(Mean Squared Error) < Target Error$), kerjakan langkah-langkah berikut.
 - a) *Epoch* = *Epoch* + 1
 - b) Untuk tiap-tiap pasangan elemen yang akan dilakukan pembelajaran, kerjakan:

Feedforward

1. Tiap-tiap unit input ($X_i, i = 1, 2, 3, \dots, n$) menerimanya x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).

2. Tiap-tiap unit pada suatu lapisan tersembunyi ($Z_i, j = 1, 2, 3, \dots, p$)

menjumlahkan sinyal-sinyal input terbobot:

$$z_{in_j} = b_{ij} \sum_{i=1}^n x_i v_{ij}$$

Gunakan fungsi aktivasi untuk menghitung sinyal outputnya: $z_j = f(z_{in_j})$ dan

kiriman sinyal tersebut ke semua unit dilapisan atasnya (unit-unit output).

Langkah ini dilakukan sebanyak jumlah lapisan tersembunyi.

3. Tiap – tiap unit output ($Y_k, k = 1, 2, 3, \dots, m$) menjumlahkan sinyal-sinyal input terbobot.

$$y_{in_k} = b_{2k} \sum_{i=1}^p z_i w_{ij}$$

Gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$$y_k = f(y_{in_k})$$

Dan kiriman sinyal tersebut ke semua unit lapisan atasnya (unit-unit output).

Backpropagation

4. Tiap-tiap unit output ($Y_k, k = 1, 2, 3, \dots, m$) menerima target pola yang berhubungan dengan pola input pembelajaran, hitung informasi *error* nya :

$$\delta_{2k} = (t_k - y_k) f'(y_{in_k})$$

$$\varphi_{2jk} = \delta_k - z_j$$

$$\beta_{2k} = \delta_k$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai w_{jk})

$$\Delta w_{jk} = \alpha \varphi_{jk}$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai w_{jk})

$$\Delta b_{2k} = \alpha \beta_k$$

Langkah ke 4 juga dilakukan sebanyak jumlah lapisan tersembunyi, yaitu menghitung informasi *error* dari suatu lapisan tersembunyi ke lapisan tersembunyi sebelumnya.

5. Tiap-tiap unit tersembunyi ($Z_j, j = 1, 2, 3, \dots, p$) menjumlahkan delta inputnya (dari unit-unit yang berada pada lapisan di atasnya):

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk}$$

Kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi *error*:

$$\delta_{1j} = \delta_{in_j} f'(z_{in_j})$$

$$\varphi_{1ij} = \delta_j x_j$$

$$\beta_{1j} = \delta_{1j}$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai v_{ij}):

$$\Delta v_{ij} = \alpha \varphi_{1ij}$$

Hitung juga koreksi bias (yang akan digunakan untuk memperbaiki nilai b_{1j}):

$$\Delta b_{1j} = \alpha \varphi_{1j}$$

6. Tiap-tiap unit output ($Y_k, k = 1, 2, 3, \dots, m$) memperbaiki bias dan bobotnya ($j = 0, 1, 2, 3, \dots, p$):

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}$$

$$b_{2k}(\text{baru}) = b_{2k}(\text{lama}) + \Delta b_{2k}$$

Tiap-tiap unit tersembunyi (Z_j , $j = 1, 2, 3, \dots, p$) memperbaiki bias dan bobotnya

($i = 0, 1, 2, 3, \dots, n$):

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}$$

$$b_{ij}(\text{baru}) = b_{ij}(\text{lama}) + \Delta b_{ij}$$

c) Hitung *MSE* (*Mean Squared Error*)

Fungsi ini mengambil rata-rata kuadrat error yang terjadi antara output jaringan dan target. MSE digunakan satu epoch pelatihan, dengan menghitung seluruh pola output aktual dari setiap pola input untuk dihitung selisihnya dengan pola output yang diinginkan.

Jika nilai $MSE \leq$ error yang diinginkan maka aringan dianggap sudah mengenali pola tetapi jika belum jaringan terus dilatih dan dihitung kembali setiap epochnya sampai nilai $MSE \leq$ error yang diinginkan. Rumus menghitung nilai MSE adalah :

$$MSE = \frac{1}{n} + \sum_{t=1}^n (y_t - \hat{y}_t)^2$$

Dimana y_t = nilai target keluaran yang diinginkan

$\hat{y}_t$ = nilai keluaran aktual

n = banyak output

1) Inisialisasi bobot secara random

Pemilihan bobot awal sangat mempengaruhi jaringan syaraf dalam mencapai minimum global (atau mungkin hanya lokal saja) terhadap nilai *error*, serta cepat tidaknya proses pelatihan menuju kekonvergenan. Apabila nilai bobot awal terlalu besar, maka input ke setiap lapisan tersembunyi atau lapisan output akan jatuh pada daerah dimana turunan fungsi *sigmoid*nya akan sangat besar. Sebaliknya, apabila nilai bobot awal terlalu kecil, maka input ke setiap lapisan tersembunyi atau lapisan output akan sangat kecil, yang akan menyebabkan proses pelatihan akan berjalan sangat lambat. Biasanya bobot awal diinisialisasi secara random dengan nilai antara -0,5 sampai 0,5 (atau -1 sampai 1, atau interval yang lainnya) (Fausett, 1994).

2) Inisialisasi bobot awal dengan metode *Nguyen-Widraw*

Metode *Nguyen-Widraw* akan menginisialisasi bobot-bobot lapisan dengan nilai antara -0,5 sampai 0,5. Sedangkan bobot-bobot dari lapisan input ke lapisan tersembunyi dirancang sedemikian rupa sehingga dapat meningkatkan proses pembelajaran.

Parameter α merupakan laju pembelajaran yang menentukan kecepatan iterasi. Nilai α terletak antara 0 dan 1 ($0 \leq \alpha \leq 1$). Semakin kecil harga α , semakin sedikit iterasi yang dipakai. Tetapi, jika harga α terlalu besar, maka akan merusak pola yang sudah benar sehingga pemahaman menjadi lambat. Setelah pelatihan selesai dilakukan, jaringan dapat digunakan untuk pengenalan pola. Dalam hal ini, hanya Propagasi Maju yang dipakai untuk menentukan keluaran jaringan. Apabila fungsi aktivasi yang dipakai *sigmoid biner*, maka langkah-langkahnya harus disesuaikan.

2.3 PAD (Pendapatan Asli Daerah)

2.3.1 Pengertian Pendapatan

Pendapatan merupakan pos yang penting dari laporan keuangan dan mempunyai penggunaan yang bermacam-macam untuk berbagai tujuan. Penggunaan informasi pendapatan yang paling utama adalah untuk tujuan pengambilan keputusan, dan biasanya sebagai tolok ukur berhasilnya suatu organisasi atau instansi dalam mengelol sumber daya yang dimilikinya. Gade (2000:100) menyatakan “Pendapatan merupakan penambahan kas pemerintah pusat yang berasal dari antara lain mencakup penerimaan pajak , cukai, penerimaan minyak, pendapatan yang berasal dari investasi, penerimaan bantuan luar negeri dan pinjaman dalam negeri serta hibah”.

Pendapatan daerah yang berasal dari semua penerimaan kas daerah dalam periode anggaran menjadi hak daerah diakui dan dicatat berdasarkan jumlah uang yang diterima dan merupakan hak daerah. Pendapatan Asli Daerah adalah pendapatan yang diperoleh dari sumber sumber pendapatan daerah dan dikelola sendiri oleh pemerintah daerah. Pendapatan Asli Daerah merupakan pendapatan pendapatan yang diperoleh dan digali dari potensi pendapatan yang ada di daerah.

NO. URUT	URAIAN	ANGGARAN	REALISASI	LEBIH / (KURANG)
1	PENDAPATAN	1,451,232,506,000.00	1,463,130,392,441.00	11,897,886,441.00
1 . 1	PENDAPATAN ASLI DAERAH	160,268,287,000.00	160,752,449,651.00	484,162,651.00
1 . 1 . 1	Pendapatan Pajak Daerah	38,718,812,000.00	43,593,346,327.00	4,874,534,327.00
1 . 1 . 2	Hasil Retribusi Daerah	45,787,987,000.00	41,482,304,590.00	(4,305,682,410.00)
1 . 1 . 3	Hasil Pengelolaan Kekayaan Daerah yang Dipisahkan	6,425,251,000.00	6,425,251,926.00	926.00
1 . 1 . 4	Lain-lain Pendapatan Asli Daerah yang Sah	69,336,237,000.00	69,251,546,808.00	(84,690,192.00)

Gambar 2.15 Data Pendapatan Asli Daerah

2.3.2 Pendapatan Asli Daerah

Pendapatan asli daerah merupakan pendapatan yang diperoleh dari sumber sumber pendapatan daerah dan dikelola sendiri oleh pemerintahan daerah. Pendapatan daerah juga merupakan pendapatan yang diperoleh oleh pemerintah

daerah dan digali dari potensi pendapatan yang ada di daerah. Dengan kata lain pendapatan asli daerah merupakan pendapatan yang diterima oleh pemerintah daerah atas segala sumber-sumber atau potensi yang ada pada daerah yang harus diolah oleh pemerintah daerah di dalam memperoleh pendapatan daerah.

2.3.3 Jenis-jenis Pendapatan Asli Daerah

Menurut Undang-undang No.34 2004 (RI,2004) tentang perimbangan keuangan Negara antara Pemerintah Pusat dan Pemerintah Daerah.

Pajak Asli Daerah bersumber dari :

- a. Pajak Daerah
- b. Retribusi Daerah
- c. Hasil pengelolaan kekayaan daerah yang dipisahkan
- d. Lain-lain PAD yang sah

a. Pajak Daerah

Pajak Daerah merupakan salah satu pendapatan yang member kontribusi terhadap Pendapatan Asli Daerah. Menurut Siahaan (2005:7) “Pajak adalah pembayaran wajib yang dikenakan berdasarkan undang-undang yang tidak dapat dihindari bagi yang berkewajiban dan bagi mereka yang tidak mau membayar pajak dapat dilakukan paksaan. Sedangkan menurut pendapat ahli yang lain yaitu mengenai pajak daerah menurut Sunarto (2005:15) beliau menyatakan bahwa pajak daerah merupakan pajak yang dikelola oleh pemerintah daerah, baik provinsi maupun kabupaten/kota yang berguna untuk menunjang penerimaan pendapatan asli daerah dan hasil penerimaan tersebut masuk di dalam APBD”. Didalam segi kewenangan pemungutan pajak atas objek di daerah, dibagi atas dua hal, yaitu :

1. Pajak daerah yang dipungut oleh provinsi
2. Pajak daerah yang dipungut oleh kabupaten/kota

1. Pajak daerah yang dipungut oleh provinsi

Pajak provinsi didalam kewenangan pungutannya terdapat pada pemerintah daerah provinsi. Didalam pajak provinsi jenis pajak tersebut ada beberapa jenis berdasarkan Undang-undang No. 34 tahun 2000, (RI,2000) tentang Pajak Daerah adalah

- (a) Pajak kendaraan bermotor dan kendaraan di atas air
- (b) Bea balik nama kendaraan bermotor dan kendaraan di atas air
- (c) Pajak bahan bakar kendaraan bermotor
- (d) Pajak pengambilan dan pemanfaatan air bawah tanah dan air permukaan

2. Pajak daerah yang dipungut oleh kabupaten/kota

Pajak kabupaten/kota kewenangan pemungutan ada pada pemerintah daerah kabupaten atau kota. Jenis pajak kabupaten atau kota berdasarkan Undang-undang No. 34 tahun 34 (RI, 2000) tentang Pajak daerah ditetapkan sebanyak tujuh, yaitu :

- (a) Pajak hotel
- (b) Pajak Restoran
- (c) Pajak Hiburan
- (d) Pajak Reklame
- (e) Pajak Penerangan Jalan
- (f) Pajak Pengambilan Bahan Galian Golongan C
- (g) Pajak Parkir

b. Retribusi Daerah

Penerimaan pemerintah daerah selain dari pajak daerah dan bagi hasil pajak pusat yang diperuntukkan ke pemerintah daerah berasal dari retribusi daerah. Akan tetapi, untuk retribusi tiap daerah memiliki potensi yang berbeda satu sama lain, untuk itu pemerintah daerah harus dapat melihat peluang apa saja yang dapat dilakukan dalam menggali penerimaan dari retribusi untuk menunjang penerimaan. Di dalam jenis pungutannya pajak dan retribusi tidaklah sama, perbedaannya ialah pada Take and Give. Pajak merupakan iuran wajib yang dibayarkan wajib pajak ke kas Negara tanpa ada kontra prestasi langsung dan yang dapat dipaksakan serta memiliki sanksi yang tegas yang ditetapkan sesuai dengan Undang-undang. Sedangkan retribusi menurut Siahaan (2005:5) “Retribusi adalah pembayaran wajib dari penduduk kepada negara adanya jasa tertentu yang diberikan oleh Negara bagi penduduknya secara perorangan”. Namun tidak semua jasa yang diberikan oleh pemerintah daerah dapat dipungut retribusinya. Tetapi, hanya jenis-jenis jasa tertentu yang menurut pertimbangan social-ekonomi layak dijadikan sebagai objek retribusi.

Menurut Undang-undang No. 18 Tahun 1997 (RI. 1997) menyebutkan bahwa retribusi daerah. Yang selanjutnya disebut retribusi adalah pungutan daerah sebagai pembayaran atas jasa atau pemberian izin tertentu yang khusus disediakan atau diberikan oleh Pemerintah Daerah untuk kepentingan pribadi atau badan. Jasa tertentu atau jasa khusus tertentu dikelompokkan ke dalam empat bagian yakni:

1. Retribusi jasa umum, yaitu retribusi atas jasa yang diberikan oleh pemerintah daerah untuk tujuan kepentingan umum serta dapat dinikmati oleh orang pribadi. jasa umum, antara lain pelayanan kesehatan, pelayanan kebersihan, retribusi

parker di tepi jalan umum, retribusi pelayanan pemakaman, penggantian biaya cetak KTP dan Akta pencatatan sipil. Yang tidak termasuk jasa umum yakni jasa umum pemerintahan.

2. Retribusi jasa usaha, yaitu retribusi atas jasa yang disediakan oleh pemda dengan menganut prinsip komersial karena pada dasarnya disediakan oleh sector swasta. Jasa usaha, antara lain penyewaan asset yang dimiliki / diakui oleh pemerintah daerah, penyediaan tempat penginapan, usaha bengkel kendaraan, tempat pencucian mobil, penjualan bibit, retribusi pasar grosir, retribusi penginapan.
3. Retribusi perizinan tertentu, yaitu retribusi atas kegiatan tertentu pemerintah daerah dalam rangka pemberian izin kepada orang pribadi atau badan yang dimaksudkan untuk pembinaan, pengaturan, penggunaan sumber daya alam, barang, prasarana, sarana atau fasilitas tertentu guna melindungi kepentingan umum dan menjaga kelestarian lingkungan. Mengingat bahwa fungsi perizinan dimaksudkan untuk mengadakan pembinaan pengaturan, pengadilan, dan pengawasan, maka pada dasarnya pemberian izin oleh pemerintah daerah tidak harus dipungut retribusi. Akan tetapi, untuk melaksanakan fungsi tersebut , pemerintah daerah mungkin masih mengalami kekurangan biaya yang tidak selaludapat dicukupi dari sumber-sumber penerimaan daerah. Perizinan tertentu yang dapat dipungut retribusi, antara lain izin mendirikan bangunan, izin pengunaantanah, retribusi izin trayek, retribusi iin tempat penjualan minuman beralkohol.
4. Retribusi lain-lain, sesuai dengan Undang-undang No. 34 tahun 2000 telah ditetapkan retribusi jasa umum, jasa usaha, dan juga retribusi perizinan tertentu. Sesuai dengan undang- undang tersebut daerah juga diberikankewenangan untuk

menetapkan jenis retribusi daerah lainnya yang dipandang sesuai dengan daerahnya, apakah ada potensi yang lainnya yang dipandang sesuai dengan daerahnya, apakah ada potensi yang lain yang dapat dijadikan oleh pemerintah daerah sebagai retribusi.

c. Hasil pengelolaan Kekayaan Daerah yang dipisahkan

Penerimaan pendapatan daerah selain pajak dan retribusi ialah pengelolaan kekayaan daerah yang dipisahkan, dimana didalam hal ini yang termasuk di dalamnya ialah laba BUMD dan hasil kerja sama pemerintah daerah dengan pihak ketiga.

d. Lain-lain pendapatan daerah yang sah

Penerimaan pendapatan daerah yang terakhir ialah melalui pendapatan daerah yang sah, dimanapendapatan tersebut meliputi:

1. Hasil penjualan kekayaan daerah yang tidak dipisahkan
2. Jasa giro
3. Pendapatan bunga
4. Keuntungan selisih nilai tukar rupiah terhadap mata uang asing
5. Komisi, potongan, ataupun bentuk lain sebagai akibat dari penjualan dan / atau jasa oleh daerah.

2.3.4 Dasar Hukum Pendapatan Asli Daerah

- a. Undang-undang No. 7 Tahun 1956 tentang pembentukan daerah otonom kabupaten-kabupaten dalam lingkungan daerah.

- b. Undang-undang No. 18 Tahun 1997 tentang pajak daerah dan retribusi daerah, yang diundangkan di Jakarta dan mulai berlaku sejak tanggal diundangkan yaitu 23 mei 1997.
- c. Peraturan Pemerintah No. 20 Tahun 1997 tentang Retribusi Daerah.
- d. Undang-Undang No. 22 Tahun 1999 tentang Pemerintah Daerah.
- e. Undang-Undang No. 34 Tahun 2000 tentang Perubahan Undang-undang No. 18 Tahun 1997 tentang Pajak Daerah dan Retribusi daerah. Undang-undang ini berlaku sejak tanggal 20 Desember 2000
- f. Peraturan Pemerintah No. 66 Tahun 2001 tentang retribusi Daerah, yang diundangkan di Jakarta dan mulai berlaku sejak 13 September 2001.



BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas desain dan perancangan system jaringan saraf tiruan metode backpropagation dalam prediksi jumlah pendapatan asli daerah sebagai sumber pendapatan daerah kabupaten Boyolali . Desain dan perancangan ini meliputi analisis system, perancangan system, rancangan basis data dan desain antarmuka.

3.1 Analisis Sistem

Pada sub bab analisis akan dibahas berbagai hasil analisis terhadap system dan elemen-elemen yang terkait, seperti user, evaluator, dan semua yang diperlukan dalam proses prediksi jumlah pendapatan daerah.

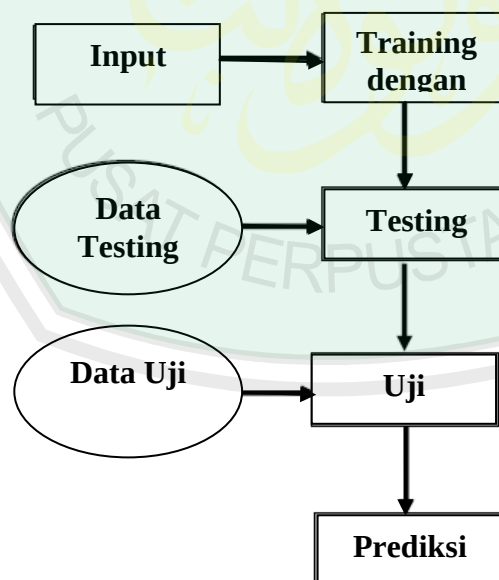
3.1.1 Deskripsi Umum Sistem

Sistem ini merupakan sistem yang dikembangkan untuk melakukan analisa terhadap pola data perhitungan pendapatan daerah. Parameter yang digunakan dalam analisa adalah jumlah masing-masing pendapatan dari tahun 2009 sampai 2013. Dari pola yang dihasilkan kita dapat mengetahui dan mengantisipasi naik turunnya jumlah pendapatan daerah. Selain dari pola yang dihasilkan tersebut dapat memberikan informasi yang berharga dari pemerintah daerah kab. Boyolali untuk mengetahui berapa jumlah pendapatan daerah sebagai sumber pendapatan asli daerah kab. Boyolali pada setiap tahunnya.

Dalam sistem ini digunakan jaringan saraf tiruan metode backpropagation untuk melakukan prediksi terhadap pola data jumlah pendapatan daerah. Jaringan saraf tiruan metode backpropagation terdiri dari lapisan unit untuk unit input, lapisan untuk unit pola dan lapisan untuk unit output (Limin Fu, 1994). Untuk unit input akan dimasukkan data jumlah pendapatan daerah tahun 2009 sampai jumlah

pendapatan daerah tahun 2013. Pada tahap-tahap dalam jaringan saraf tiruan metode backpropagation

1. Dimulai dengan lapisan masukan, hitung keluaran dari setiap elemen pemroses melalui lapisan luar.
2. Hitung kesalahan pada lapisan luar yang merupakan selisih antara data actual dan target.
3. Transformasikan kesalahan tersebut pada kesalahan yang sesuai di sisi masukan elemen pemroses.
4. Propagasi balik kesalahan-kesalahan ini pada keluaran setiap elemen pemroses ke kesalahan yang terdapat pada masukan. Ulangi proses ini sampai masukan tercapai.
5. Ubah seluruh bobot dengan menggunakan kesalahan pada sisi masukan elemen dan luaran elemen pemroses yang terhubung.



Gambar 3.1 Blok Diagram Proses Umum Sistem

Penjelasan dari blok diagram diatas, sebagai berikut :

- a. Input : masukan yang diperlukan adalah data emiten yang akan dilatih meliputi data jumlah pendapatan pada tahun 2009 sampai jumlah pendapatan tahun 2013, bobot pada lapisan tersembunyi ,bobot pada lapisan keluaran.
- b. Periode Training : pelatihan suatu jaringan dengan algoritma backpropagation meliputi dua tahap : perambatan maju dan perambatan mundur. Selama perambatan maju, tiap unit masukan (x_1) menerima sebuah masukan sinyal ini ke tiap-tiap lapisan tersembunyi $z_1, \dots, z_p$. Tiap unit tersembunyi ini kemudian menghitung aktivasinya dan mengirimkan sinyalnya (z_j) ke tiap unit keluaran. Tiap unit keluaran (y_k) menghitung aktivasinya (y_k) untuk membentuk respon pada jaringan untuk memberikan pola masukan. Selama pelatihan, tiap unit keluaran membandingkan perhitungan aktivasinya y_k dengan nilai targetnya t_k untuk menentukan kesalahan pola tersebut dengan unit itu. Berdasarkan kesalahan ini, faktor δ_k ($k = 1, \dots, m$) dihitung. δ_k digunakan untuk menyebarkan kesalahan pada unit keluaran y_k kembali ke semua unit pada lapisan sebelumnya (unit-unit tersembunyi yang dihubungkan ke y_k). Juga digunakan (nantinya) untuk mengupdate bobot-bobot antara keluaran dan lapisan tersembunyi. Dengan cara yang sama, faktor ($j = 1, \dots, p$) dihitung tiap unit tersembunyi z_j . Tidak perlu untuk menyebarkan kesalahan kembali ke lapisan masukan, tetapi δ_j digunakan untuk mengupdate bobot-bobot antara lapisan tersembunyi dan lapisan masukan. Setelah seluruh faktor δ ditemukan, bobot untuk semua lapisan diatur secara serentak. Pengaturan bobot w_{jk} (dari unit tersembunyi z_j ke unit keluaran y_k) didasarkan pada faktor δ_k dan aktivasinya z_j dari unit tersembunyi z_j . Didasarkan pada faktor δ_j dan aktivasinya x_i unit masukan.

- c. Periode Testing dan Uji : proses testing dan uji yaitu tahap perambatan maju dari algoritma pelatihan

3.1.2 Batasan Sistem

1. Sistem ini menghitung pendapatan daerah kabupaten boyolali di bidang pendapatan asli daerah saja.
2. Bilangan yang dibutuhkan dalam sistem adalah bilangan inputan yang telah diperkecil ,begitu juga hasil yang didapat adalah bilangan yang masih diperkecil.

3.1.3 Analisis Kebutuhan Perangkat Lunak

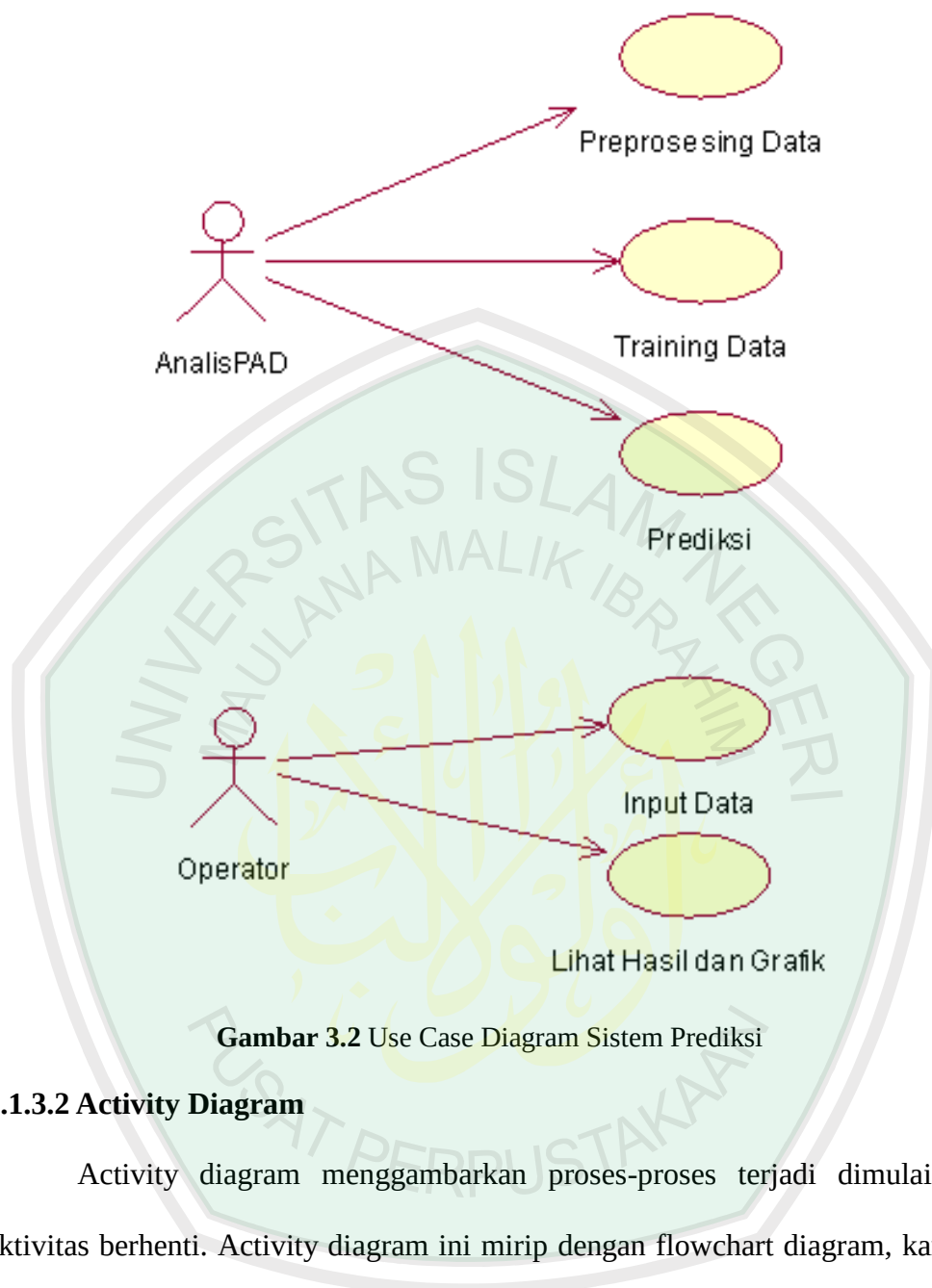
Analisis kebutuhan ini didasarkan pada sistem prediksi pendapatan daerah yang dapat memuaskan user. Kebutuhan perangkat ini meliputi hal pokok yang harus ada dan tambahan.

Hal pokok tersebut meliputi : Sistem dapat menghasilkan keputusan sesuai dengan algoritma yang telah ditentukan.

Hal tambahannya : Interaksi pengguna dengan sistem akan melalui GUI (Graphical User Interface).

3.1.3.1 Analisis Use Case

Dalam sistem prediksi user akan mendapatkan prediksi hasil pendapatan daerah, dengan cara menginputkan data historical pendapatan daerah yang telah disediakan, selanjutnya dilakukan proses training. Dimana proses training akan menentukan nilai hidden layer, MSE, konstanta belajar dan maksimum epoch , lalu pencarian nilai bobot v dan w yang sesuai setelah dilakukan koreksi. Setelah proses training selesai, selanjutnya proses prediksi. Dimana proses akan melakukan perhitungan jumlah delta ,perhitungan error di hidden layer ,koreksi nilai dari input delta ke hidden layer

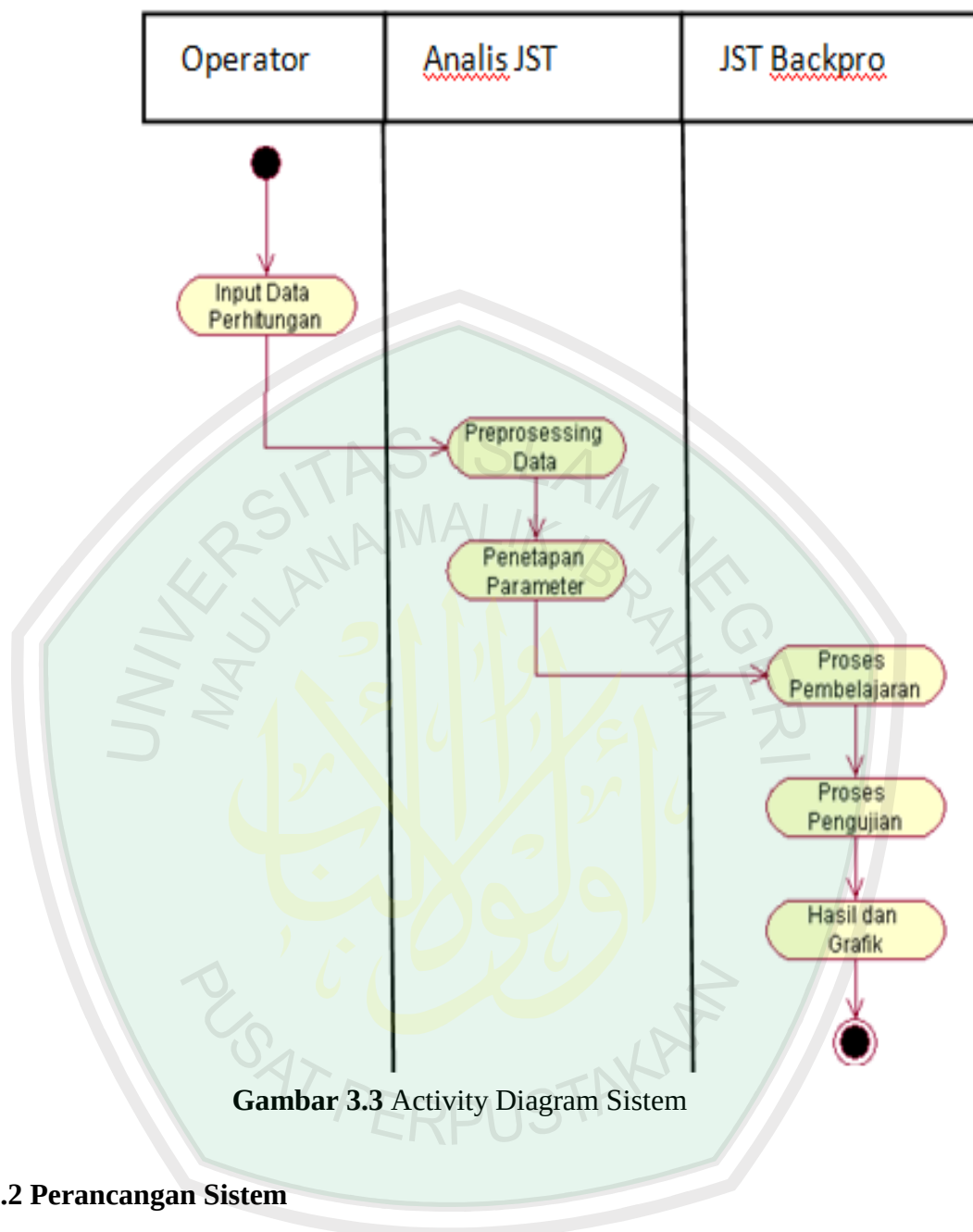


Gambar 3.2 Use Case Diagram Sistem Prediksi

3.1.3.2 Activity Diagram

Activity diagram menggambarkan proses-proses terjadi dimulai sampai aktivitas berhenti. Activity diagram ini mirip dengan flowchart diagram, karena kita dapat memodelkan satu alur kerja dari suatu aktifitas lainnya atau dari suatu aktifitas kedalam keadaan sesaat.

Activity diagram sistem berjalan proses prediksi yaitu :



Gambar 3.3 Activity Diagram Sistem

3.2 Perancangan Sistem

Sistem ini menggunakan perangkat lunak Netbeans 6.5. Perangkat lunak pemrograman ini dipilih karena menyediakan fungsi fungsi yang membantu dalam perhitungan matematik. Perangkat keras yang digunakan dalam mengimplementasikan program dari JST ini adalah Intel Dual Core 2.1GHz , memori memori DDR 1GB, VGA Intel GMA 4500M dan Harddisk 160GB

3.2.1 Prediksi PAD

Pola penerimaan pajak tiap tahunnya selalu mengalami kenaikan atau penurunan, kondisi ini tergantung dari banyak tidaknya penerimaan Pendapatan Asli Daerah yang berasal dari Pendapatan Pajak Daerah, Hasil Retribusi Daerah, Hasil Pengelolaan Kekayaan Daerah, dan Pendapatan Daerah yang lain yang sah.

Berdasarkan pola penerimaan PAD menunjukkan bahwa penerimaan PAD dipengaruhi oleh penerimaan PAD tahun tahun sebelumnya. Di dalam program ini digunakan hasil penerimaan PAD sebelum sebelumnya sebagai pembanding dan variabel penghitung untuk mendapatkan hasil dari proses JST Backpropagation.

3.2.2 Variabel Data Masukan

Variabel data masukan merupakan sekumpulan data serial atau berkala pada suatu jangka waktu. Di dalam sistem prediksi jaringan syaraf tiruan ini, data masukan yang digunakan yaitu :

- a. Data berkala selama 5 tahun periode penerimaan Pendapatan Asli Daerah
- b. Penerimaan PAD tertinggi pada data inputan
- c. Penerimaan PAD terendah pada data inputan
- d. Nilai normalisasi dari data inputan

3.2.3 Inisialisasi Parameter

Parameter yang diset untuk pembelajaran adalah training. Training merupakan fungsi pembelajaran untuk bobot-bobot dengan menggunakan nilai momentum. Selain itu, terdapat beberapa nilai parameter yang harus diset untuk pelatihan. Parameter-parameter tersebut yaitu :

a. Epoch

Epoch adalah perulangan atau iterasi dari proses yang dilakukan untuk mencapai target yang telah ditentukan. Maksimum epoch adalah jumlah epoch maksimum yang boleh dilakukan selama proses pelatihan. Iterasi akan dihentikan apabila nilai epoch melebihi maksimum epoch.

b. Learning Rate

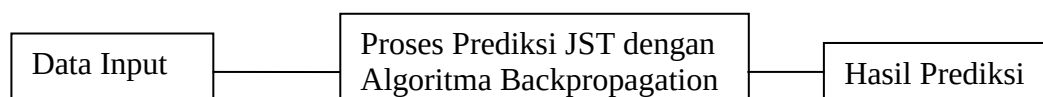
Learning rate adalah laju pembelajaran yang berupa perkalian negative gradient untuk menentukan perubahan pada nilai bobot dan bias. Semakin besar nilai learning rate akan berimplikasi pada semakin besarnya langkah pelatihan. Jika learning rate diset terlalu besar, maka algoritma akan menjadidi tidak stabil, jika sebaliknya maka algoritma akan mencapai target dalam jangka waktu yang lama.

c. Target Error

Target Error adalah batas toleransi error yang diijinkan.

d. Inisiasi bobot dengan mengambil bobot awal menggunakan nilai random yang terkecil.

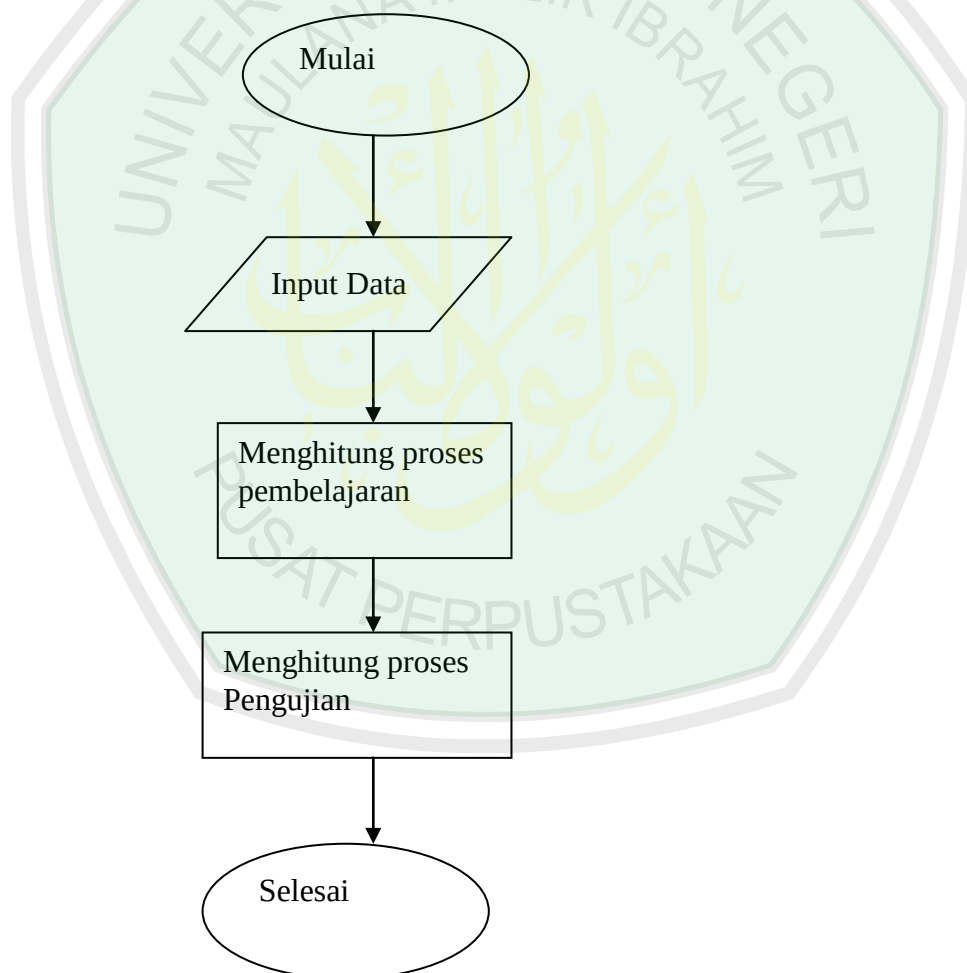
Sistem prediksi secara global dapat dilihat pada skema sederhana Gambar 3.2. Pada skema tersebut, terdapat tiga elemen utama. Pertama data masukan , menunjukan awal dari proses prediksi . kedua proses prediksi itu sendiri, khususnya menggunakan Jaringan Syaraf Tiruan Backpropagation dan yang ketiga adalah hasil keluaran dari proses prediksinya.



Gambar 3.4 Skema Global Sistem

Komponen selanjutnya dari skema global prediksi pada Gambar 3.2 adalah proses dalam sistem prediksi. Metode pendekatan yang dipilih untuk mendeskripsikan alur dan informasi di dalam prosesnya menggunakan flowchart. Secara lebih rinci akan dijelaskan melalui beberapa representasi gambar dibawah. Pada flowchart yang akan dibuat, entity eksternalnya adalah user. User sebagai entity eksternal berperan sebagai pemberi masukan dalam sistem. Selain sebagai pemberi masukan user berperan sebagai penerima keluaran dari sistem.

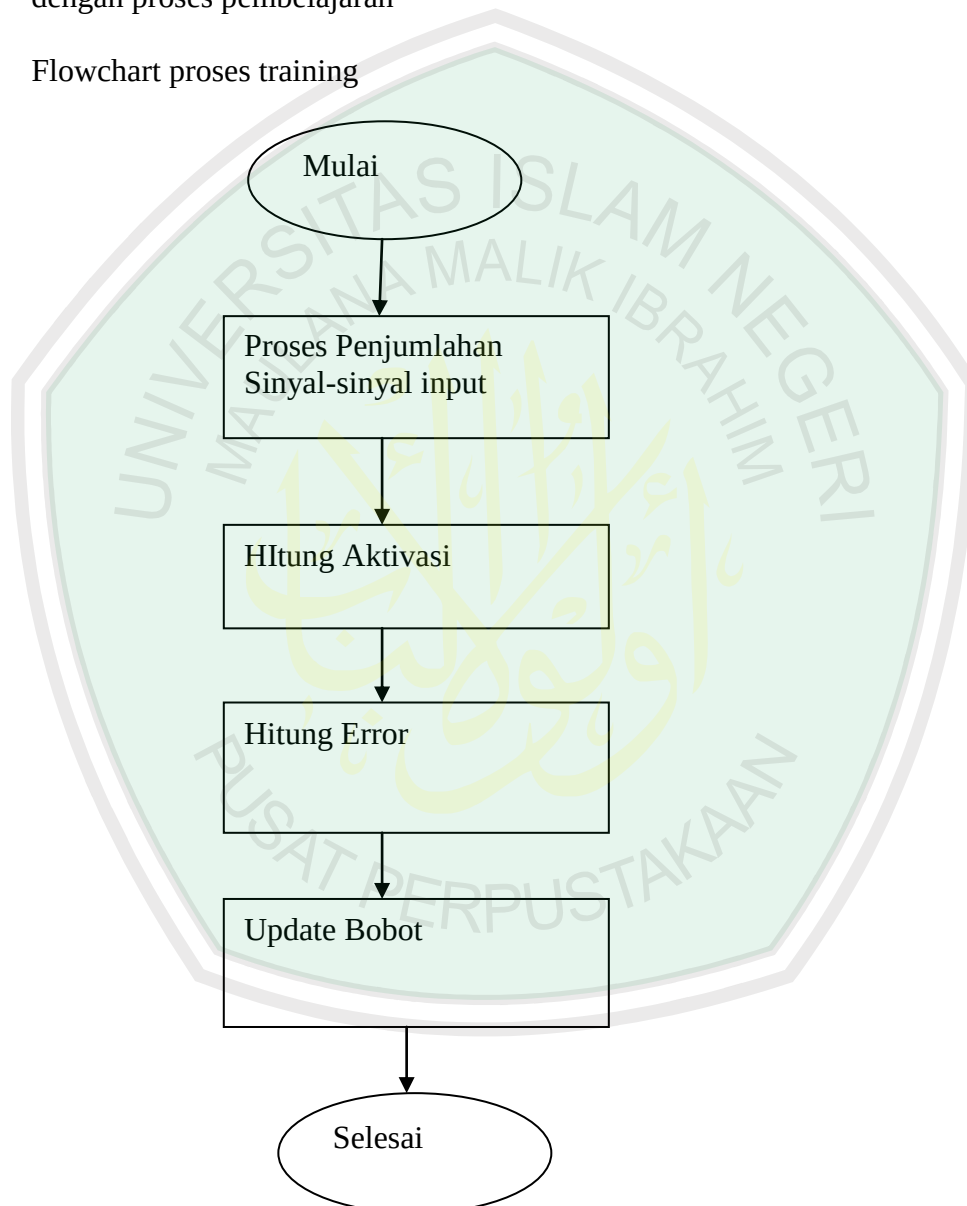
Flowchart Sistem



Gambar 3.5 Flowchart Sistem

Ada tiga proses utama yang terdapat pada Flowchart 1 yang terdiri dari proses penetapan parameter JST, proses pembelajaran, proses pengujian. Setelah user memasukkan nilai dari beberapa parameter jaringan maka parameter tersebut akan masuk kedalam proses penetapan variabel yang kemudiandilanjutkan dengan proses pembelajaran

Flowchart proses training



Gambar 3.6 Flowchart proses training

Pada Flowchart 2 ini, file pelatihan wk2 yang merupakan bagian dari entitas data penelitian yaitu dari historis digunakan pada proses penjumlahan sinyal sinyal input.

3.2.4 Pembahasan Perhitungan Data Training

Adapun data *training* yang digunakan terdiri atas empat pasang masukan dan keluaran yakni :

Parameter	Inputan Pertama	Inputan Kedua	Bias/Keluaran
PPD	12	14	19
HRD	43	25	20
HPKDD	3	4	4
LLPAD	9	42	52

Tabel 1 Tabel Parameter PAD Data

Sebuah jaringan terdiri atas dua input, satu unit tersembunyi, dan satu unit keluaran. Fungsi aktivasi yang digunakan adalah sigmoid biner, learning rate / alpha = 0,01. toleransi error yang diperkenankan adalah 0,41.

Data tersebut selanjutnya dinormalisasikan dengan rumus

$$x' = \frac{0.8(x - \min)}{\max - \min} + 0.1$$

Parameter	2009	2010	Bias
PPD	0.1	0.3	0.9
HRD	0.9	0.2	0.1
HPKDD	0.1	0.9	0.9
LLPAD	0.1	0.7	0.9

Tabel 2 Tabel Hasil Normalisasi Data

Langkah 0 : Inisialisasi acak bobot dan pada bias, misalnya

$$v_{01} = 1,718946$$

$$v_{11} = -1,263178$$

$$v_{21} = -1,083092$$

$$w_{01} = -0,541180$$

$$w_{11} = 0,543960$$

Langkah 1 : Dengan bobot acak tersebut, tentukan nilai *error* untuk data *training* secara keseluruhan dengan rumus sebagai berikut :

$$z_{in_j} = b_{ij} \sum_{i=1}^n x_i v_{ij}$$

Gunakan fungsi aktivasi untuk menghitung sinyal outputnya

$$z_j = f(z_{in_j})$$

$$\begin{aligned} z_{in_{11}} &= 1,718946 + \{(0,1x - 1,263178) + (0,3x - 1,083092)\} \\ &= 1,718946 \end{aligned}$$

$$z_{11} = f(z_{in_{11}}) = 0,847993$$

$$\begin{aligned} z_{in_{12}} &= 1,718946 + \{(0,9x - 1,263178) + (0,2x - 1,083092)\} \\ &= 0,635854 \end{aligned}$$

$$z_{12} = f(z_{in_{12}}) = 0,653816$$

$$\begin{aligned} z_{in_{13}} &= 1,718946 + \{(0,1x - 1,263178) + (0,9x - 1,083092)\} \\ &= 0,455768 \end{aligned}$$

$$z_{13} = f(z_{in_{13}}) = 0,612009$$

$$z_{in_{14}} = 1,718946 + \{(0,1x - 1,263178) + (0,7x - 1,083092)\}$$

$$= -0,627324$$

$$z_{14} = f(z_{in_{11}}) = 0,348118$$

Dimana indeks z_{jn} berarti bobot untuk unit tersembunyi ke- j dan data *training* ke- n

$$y_{in} = b_k + \sum_{i=1}^p z_i w_{ij}$$

Gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$$y_k = f(y_{in_k})$$

$$y_{in_{11}} = -0,541180 + (0,847993 \times 0,543960) = 0,079906$$

$$y_{11} = f(y_{in_{11}}) = 0,480034$$

$$y_{in_{12}} = -0,541180 + (0,653816 \times 0,543960) = -0,185530$$

$$y_{12} = f(y_{in_{12}}) = 0,4453750$$

$$y_{in_{13}} = -0,541180 + (0,612009 \times 0,543960) = 0,208271$$

$$y_{13} = f(y_{in_{13}}) = 0,448119$$

$$y_{in_{14}} = -0,541180 + (0,348118 \times 0,543960) = -0,351818$$

$$y_{14} = f(y_{in_{14}}) = 0,412941$$

Sehingga,

$$E = 0,5 \times \{(0,9 - 0,480034)^2 + (0,1 - 0,453750)^2 + (0,9 - 0,448119)^2 + (0,9 - 0,412941)^2\}$$

$$= 0,501957$$

Langkah 2 : Karena, data *error training* masih lebih besar dari toleransi yakni 0,41.

Maka, pelatihan dilanjutkan pada langkah 3-8

Langkah 3: $x_1=0,1$, $x_2=0,3$ (Training untuk data pertama)

Langkah 4 : $z_{in_{11}} = 1,718946 + \{(0,1x - 1,263178) + (0,3x - 1,083092)\}$

$$= 1,718946$$

$$z_{11} = f(z_{in_{11}}) = 0,847993$$

Langkah 5 : $y_{in_{11}} = -0,541180 + (0,847993 \times 0,543960) = 0,079906$

$$y_{11} = f(y_{in_{11}}) = 0,480034$$

Langkah 6 : $\delta_{2k} = (t_k - y_k) f'(y_{in_k})$

$$\Delta W_{kj} = \alpha \delta_k z_j$$

$$\delta_1 = (0,9 - 0,480034) f'(0,079906) = -0,119817$$

$$\Delta W_{11} = 0,01 \times -0,119817 \times 0,847993 = -0,001016$$

$$\Delta W_{01} = 0,01 \times -0,119817 = -0,00119817$$

Langkah 7 : $\delta_{in_1} = -0,119817 \times 0,543960 = -0,065176$

$$\delta_1 = -0,065176 f'(1,718946) = -0,008401$$

$$\Delta v_{11} = 0,01 \times -0,008401 \times 0,1 = -0,000008401$$

$$\Delta v_{21} = 0,01 \times -0,008401 \times 0,3 = -0,000025203$$

$$\Delta v_{11} = 0,01 \times -0,008401 = -0,00008401$$

Langkah 8 : $\Delta W_{01(\text{baru})} = -0,541180 + (-0,00119817) = -0,542378$

$$\Delta W_{11}(\text{baru}) = -0,543960 + (-0,001016) = -0,542944$$

$$V_{01}(\text{baru}) = 1,718946 + (-0,00008401) = 1,718862$$

$$V_{11}(\text{baru}) = -1,263178 + -0,000008401 = -1,26319$$

$$V_{21}(\text{baru}) = -1,083092 + -0,000025203 = -1,0831$$

Setelah langkah 3-8 untuk data *training* pertama dikerjakan, ulangi kembali langkah 3-8 untuk data *training* ke-2,3 dan 4. Setelah seluruh data *training* dikerjakan itu berarti satu iterasi telah diproses. Bobot yang dihasilkan pada itersai pertama untuk *training* ke-2,3,dan 4 adalah:

Data *training* ke 2

$$w_{01} = -0,541023$$

$$w_{11} = 0,543830$$

$$V_{01} = 1,718862$$

$$V_{11} = -1,263178$$

$$V_{21} = -1,083092$$

Data *training* ke 3

$$w_{01} = -0,539659$$

$$w_{11} = 0,544665$$

$$V_{01} = 1,719205$$

$$V_{11} = -1,263002$$

$$V_{21} = -1,082925$$

Data *training* ke 4

$$w_{01} = -0,540661$$

$$w_{11} = 0,544316$$

$$v_{01} = 1,719081$$

$$v_{11} = -1,263126$$

$$v_{21} = -1,083049$$

Setelah sampai pada data *training* ke-4, maka iterasi pertama selesai dikerjakan.

Proses *training* dilanjutkan pada langkah ke-9 yaitu memeriksa kondisi STOP dan kembali pada langkah ke-2. Demikian seterusnya sampai *error* yang dihasilkan memenuhi toleransi *error* yang ditentukan. Setelah proses *training* selesai, bobot akhir yang diperoleh untuk XOR adalah sebagai berikut:

$$w_{01} = -5,018457$$

$$w_{11} = 5,719889$$

$$v_{01} = 12,719601$$

$$v_{11} = -6,779127$$

$$v_{21} = -6,779127$$

Jika terdapat masukan baru, misalnya $x_1=0,1$ dan $x_2=0,7$ maka keluarannya dapat dicari dengan menggunakan langkah-langkah umpan maju berikut ini :

Langkah 0 : Bobot yang digunakan adalah bobot akhir hasil pelatihan diatas

Langkah 1 : Dalam contoh ini, bilangan digunakan telah berada dalam interval 0 dan

1. Jadi tidak perlu diskalakan lagi.

$$\text{Langkah 3 : } z_{in_1} = 12,719601 + \{(0,1x - 6,779127) + (0,7x - 6,779127)\}$$

$$= 5,262561$$

$$z_1 = f(z_{in_1}) = 0,994845$$

$$\text{Langkah 4 : } y_{in_1} = -5,018457 + (5,719889 \times 0,994845) = 0,671944$$

$$y_{11} = f(y_{in_{11}}) = 0,661938$$

Jadi, jika input data adalah $x_1=0,1$ dan $x_2=0,7$ output jaringan yang dihasilkan adalah 0,661938



BAB IV

IMPLEMENTASI DAN PEMBAHASAN

Dalam bab ini dibahas mengenai hasil uji coba program yang telah dirancang dan dibuat. Uji coba dilakukan untuk mengetahui apakah program dapat berjalan sebagaimana mestinya dengan lingkungan uji coba yang telah ditentukan sesuai dengan skenario uji coba.

Untuk mendapatkan suatu metode yang baik dalam sistem peramalan digunakan tingkat ketelitian sebagai ukuran. Semakin tinggi tingkat ketelitian yang didapat maka semakin baik metode tersebut.

Setelah dilakukan pelatihan dan pengujian didapat hasil-hasil penelitian dengan klasifikasi terhadap error yang dapat tercapai. Besarnya nilai error yang dicapai tersebut menentukan seberapa besar nilai ketepatan yang dapat dilakukan dalam prediksi. Untuk serangkaian data yang akan ditentukan nilainya, berbeda tingkat penyimpangannya yang secara terakumulasi membentuk hasil prediksi tersebut. Untuk memudahkan analisa, maka hasil-hasilnya ditampilkan dalam bentuk grafik. Pembentukan pola prediksi yang terbentuk secara implisit oleh Jaringan

4.1 Lingkungan Uji Coba

Pada subbab ini dijelaskan mengenai lingkungan uji coba yang meliputi perangkat keras yang digunakan. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam uji coba dan dalam tahap tes terhadap data training maupun data uji, yang antara lain adalah :

Perangkat Keras	Prosesor : Intel Pentium 2.1GHz Memori : 1GB Piranti Masukan : -Mouse -Keyboard
Perangkat Lunak	Sistem Operasi : Microsoft Windows 7 Perangkat pengembang :-Netbeans IDE 6.5 -Power Designer -Rational Rose Enterprise Edition

4.2 Data Uji Coba

Pada uji coba yang diproses, digunakan data jumlah PAD yang meliputi jumlah PAD selama beberapa tahun sesuai yang diperlukan pada program. Sebelum data diproses di jaringan saraf tiruan data diprosesing terlebih dahulu, setelah itu digunakan sebagai input pada jaringan saraf tiruan backpropagation.

4.3 Penjelasan Program

Di dalam penjelasan program ini dijelaskan tentang alur pembuatan dan kegunaan program yang dibuat beserta tampilan desainnya. Berikut ini tampilan halaman yang ada dalam program yang dibuat:

4.3.1 Proses menampilkan halaman login

Halaman login merupakan halaman pertama kali diakses oleh user. Dalam halaman ini user harus memasukkan user name dan password untuk masuk ke halaman selanjutnya. Tampilan halaman login ditunjukkan pada gambar 4.1 berikut:



4.3.2 Proses Prosesing Data (Normalisasi)

Halaman ini menjelaskan proses preprosesing data. Preprosesing data ini menggunakan normalisasi. Dengan merubah bilangan mentah menjadi bilangan desimal untuk perhitungan. Setelah data normalisasi selanjutnya data diproses

pada jaringan saraf tiruan. Listing program untuk preprosesing data menggunakan normalisasi sebagai berikut.

```

nilai_tertinggi=nilai[i];
nilai_terendah=nilai[i];

for (i=0; i<=4; i++){
    if (nilai_tertinggi<nilai[i])
    {
        nilai_tertinggi=nilai[i];
    }
}

for (i=0; i<=4; i++){
    if (nilai_terendah>nilai[i]){
        nilai_terendah=nilai[i];
    }
}

x_absen_a=((a_input-
nilai_terendah)*0.8)/(nilai_tertinggi-
nilai_terendah))+0.1;

x_absen_b=((0.8*(b_input-
nilai_terendah))/(nilai_tertinggi-nilai_terendah))+0.1;

```

```

        x_absen_c=((0.8*(c_input-
nilai_terendah))/(nilai_tertinggi-nilai_terendah))+0.1;
        x_absen_d=((0.8*(d_input-
nilai_terendah))/(nilai_tertinggi-nilai_terendah))+0.1;
        x_absen_e=((0.8*(e_input-
nilai_terendah))/(nilai_tertinggi-nilai_terendah))+0.1;

```

```

//Denormalisasi

nilai_tertinggi=nilai[0];
nilai_terendah=nilai[0];

for (i=0; i<=4; i++){
    if (nilai_tertinggi<nilai[i])
    {
        nilai_tertinggi=nilai[i];
    }
}

for (i=0; i<=4; i++){
    if (nilai_terendah>nilai[i]){
        nilai_terendah=nilai[i];
    }
}

hasilD = y* (nilai_tertinggi-
nilai_terendah)+nilai_terendah;
}

```

4.3.3 Proses Pembelajaran JST (Learning)

Setelah proses preprosesing normalisasi selesai selanjutnya data akan di learning atau dilakukan tahap pembelajaran oleh jaringan saraf tiruan. Proses learning terdiri atas:

- a. penghitungan sigma jumlah input ,hidden dan jumlah bias hidden
- b. penghitungan output dan error output

- c. penghitungan delta bias , delta bobot dan delta bias dan delta bias hidden
- d. penghitungan error hidden ,delta bias dan delta bobot
- e. penghitungan update bobot ,bias dan bias hidden

Listing program proses learning atau proses pembelajaran jaringan saraf tiruan yaitu:

```
void learn_static()
{
    double data[][] = this.x;

//    int data[][] = this.x;
    double target[] = this.t;

    int jumlah_data = data.length;
    int jumlah_input = this.unit_input;
    int jumlah_hidden = this.unit_hidden;
    int jumlah_output = this.unit_output;
    //int dt_pbdg= this.dt_pmbdg;

    //double tmp;

    // do it for learn
    int loop = 0;
```

```

this.maxloop = 100000;

do{
    // for all data
    for(int h=0; h<jumlah_data; h++){

        //hitung z_in dan z
        double z[] = new double[jumlah_hidden];
        for(int j=0; j<jumlah_hidden; j++){
            //itung sigma xi vij

            double z_in[] = new
double[jumlah_hidden];
            double jum_xv = 0;

            for(int i=0; i<jumlah_input; i++){
                double tmp_=x[h][i]*v[i][j];
                jum_xv=jum_xv+tmp_;
            }

            z_in[j] = v0[j]+jum_xv;
            //z[j] = 1/(1+(double)Math.exp(-
z_in[j]));
            z[j] = aktivasi(z_in[j]);
        }

        // itung y_in dan y [ output ]
        double y[] = new double[jumlah_output];
        for(int k=0; k<jumlah_output; k++){
            double y_in[] = new double[y.length];
            double jum_zw=0;

            for(int j=0; j<jumlah_hidden; j++){
                double tmp2=z[j]*w[k][j];
                jum_zw=jum_zw+tmp2;
            }

            y_in[k]=w0[k]+jum_zw;
            y[k]=1/(1+(double)Math.exp(-
y_in[k]));
        }
    }
}

```



```

        // ngitung error output dan delta bias
dan delta bobot atau disebut hitung MSE
        double sum_e = 0; //anyar
        double Err_y[] = new
double[jumlah_output];
        double Aw[][] = new
double[this.w.length][this.w[0].length];
        double Aw0[] = new double[this.w0.length];

        for(int k=0; k<jumlah_output; k++){
            //error output
            //Err_y[k]= (t[h]-y[k])*y[k]*(1-
y[k]);
            Err_y[k] = t[h]-y[k];
            sum_e += Math.pow(Err_y[k], 2);
        }

        this.MSE = 0.5*sum_e;

        for (int k=0; k<jumlah_output; k++){
            for(int j=0; j<jumlah_hidden; j++){

//delta bobot h0
                Aw[k][j] =
                alfa*Err_y[k]*y[k]*z[j];

                //delta bias h0
                Aw0[k] = alfa*Err_y[k]*y[k];
            }
        }

        //ngitung error hidden dan delta bias dan
delta bobot
        double Err_in[] = new
double[jumlah_hidden];
        double Err_z[] = new
double[jumlah_hidden];
        double Av[][] = new
double[this.v.length][this.v[0].length];
        double Av0[] = new double[this.v0.length];

```

```

        for(int j=0; j<jumlah_hidden; j++){
            double tmp3=0;
            for(int k=0; k<jumlah_output; k++){
                tmp3 = tmp3 +
                (Err_y[k]*this.w[k][j]);
            }

            //error sebelum output / setelah
hidden
            Err_in[j]=tmp3;

            //error hidden < t[< h > - y < k > >
            * y < k > * < 1 - y < k > > ;
            Err_z[j]=Err_in[j]*(z[j])*(1-z[j]);
            for(int i=0; i<jumlah_input; i++){
                //delta bobot iH
                Av[i][j]=this.alfa*Err_z[j]*this.x[h][i];
            }

            //delta bias hidden
            Av0[j]=this.alfa*Err_z[j];
        }

        //update bobot dan bias
        //update bobot bias output
        for(int j=0; j<jumlah_hidden; j++){
            for(int k=0; k<jumlah_output; k++){
                this.w[k][j]=this.w[k][j]+Aw[k][j];
            }
        }

        for(int k=0; k<jumlah_output; k++){
            this.w0[k]=this.w0[k]+Aw0[k];
        }

```

```

//update bobot bias hidden
for(int i=0; i<jumlah_input; i++){
    for(int j<jumlah_hidden; j++){

```

```

this.v[i][j]=this.v[i][j]+Av[i][j];
    }
}

for(int j=0; j<jumlah_hidden; j++){
    this.v0[j]=this.v0[j]+Av0[j];
}

}

loop++;

} while(is_stop()<this.ERR && loop<this.maxloop);

System.out.println("err : " +ERR);
System.out.println("loop : "+loop);
}

```

Code sebagai parameter pemberhentian perhitungan bila sudah terpenuhi syaratnya

```

double is_stop(){
    int jumlah_input = this.unit_input;
    int jumlah_hidden = this.unit_hidden;
    int jumlah_output = this.unit_output;
    int jumlah_data = this.x.length;
    double akumY=0;

    //~ itung z_in dan z
    for(int h=0; h<jumlah_data; h++){
        double z[] = new double[jumlah_hidden];

        for(int j=0; j<jumlah_hidden; j++){

            //itung sigma xi vij
            double z_in[] = new double[z.length];
            double jum_xv=0;

            for(int i=0; i<jumlah_input; i++){
                double
tmp=this.x[h][i]*this.v[i][j];
                jum_xv=jum_xv+tmp;
            }

```

```

        z_in[j]=this.v0[j]+jum_xv;
        z[j]= aktivasi(-z_in[j]);
        //z[j]=1/(1+(double)Math.exp(-
z_in[j]));

    }

    //~ itung y_in dan y (output)
    double y[] = new double[jumlah_output];

    for(int k=0; k<jumlah_output; k++){
        double y_in[] = new double[y.length];
        double jum_zw=0;

        for(int j=0; j<jumlah_hidden; j++){
            double tmp=z[j]*this.w[k][j];
            jum_zw=jum_zw+tmp;
        }

        y_in[k]=this.w0[k]+jum_zw;
        y[k]= aktivasi(-y_in[k]);
        //y[k]=1/(1+(double)Math.exp(-
y_in[k]));

        akumY += Math.pow((t[h]-y[k]),2);
    }
}

this.MSE = akumY/this.x[0].length;

//double E = akumY/this.x[0].length;
ERX = akumY;

//return E;
return this.MSE;
}

```

4.3.4 Proses uji test JST

Pada tahap ini adalah proses uji atau test jaringan saraf tiruan terhadap data yang sudah disiapkan . Setelah proses learning atau proses pembelajaran selesai diproses selanjutnya data akan di uji atau dilakukan tahap uji atau test oleh jaringan saraf tiruan *backpropagation*. Proses uji , atau test listing programnya sebagai berikut:



```
//pada hidden
double z[] = new double[jumlah_hidden];

for(int j=0; j<jumlah_hidden; j++){

    double z_in[] = new double[z.length];
    double tmp = 0;
    for(int i=0; i<data.length; i++){

        tmp = tmp + (data[i] * this.v[i][j]);
    }

    z_in[j] = this.v0[j] + tmp;
    z[j] = aktivasi(z_in[j]);
    //z[j] = 1/(1+(double)Math.exp(-z_in[j]));
}

//pada outprt
```

```

        double y[] = new double[jumlah_output];

        for(int k=0; k<jumlah_output; k++){

            double y_in[] = new double[y.length];
            double tmp = 0;

            for(int j=0; j<jumlah_hidden; j++){

                tmp = tmp + z[j] * this.w[k][j];
            }

            y_in[k] = this.w0[k] + tmp;
            y_in[k] = aktivasi(y_in[k]);
            //y_in[k] = 1/(1+(double)Math.exp(-
y_in[k]));

            /*
                if(y[k]>0.5)
                    y[k]=1;
                else
                    y[k]=0;
            */

            double err_pad = this.MSE*this.x[0].length;
            double hazil = 0;
            if(y[k] < err_pad){
                hazil = 0;
            }else if(y[k]>(1-err_pad)){
                hazil = 1;
            }else{
                hazil = 0.5;
            }
            this.hasil_mentah = y[k];
            this.hasil_akhir = y_in[k];
            //System.out.println("Output
"+(double)y[k]);
            //System.out.println("target error "+y[k]+"
"+hazil);
            System.out.println("Output :
"+(double)y_in[k]);

            hasilD=(double)y_in[k];

            double xx = 0;
            double xxx;
            double nilaipengurangD=nilai_tertinggi-
nilai_terendah;

```

```

double hasilakhir=0;

xxx=(e_input*25/100);
xx=((hasilD*(nilaipengurangD))+
((0.8*nilai_terendah)+0.1))/0.8;

//
hasilakhir = (xx+xxx);
String hasilfinal= String.valueOf(hasilakhir);
//System.out.println("nilai profit xxx :"+xxx);
System.out.println("hasil+profit xxx+xx
:"+hasilfinal);
//

//double hasilxx = xx*1000000;
System.out.println();
System.out.println("Hasil prediksi untuk tahun
selanjutnya adalah xx "+ xx);
System.out.println();
//System.out.println("Hasil prediksi untuk tahun
selanjutnya adalah "+ hasilxx);
System.out.println();
System.out.println("nilai tertinggi "+
nilai_tertinggi);
System.out.println("nilai terendah "+
nilai_terendah);
System.out.println("nilai pengurang D "+
nilaipengurangD);
}
}

```

4.4 Hasil Uji Coba

Adapun hasil uji coba digunakan terdiri a at pasang masukkan dan keluaran yakni :

$\alpha = 1$

toleransi err = 0,01

max loop = 1000

pembanding = 149

Tabel 3 Parameter Data Uji

Parameter	Inputan Pertama	Inputan Kedua	Bias
X1 / PPD	19	23	43
X2 / HRD	20	36	41
X3 / HPKDD	4	5	5
X4 / LLPAD	52	62	60

Langkah-langkah pada proses learning adalah sebagai berikut :

Langkah 0 : Inisialisai bobot dan pada bias

$$V_{01} = 0,2872$$

$$V_{11} = 0,4555$$

$$V_{21} = 0,2994$$

$$V_{02} = 0,1024$$

$$V_{12} = 0,1777$$

$$V_{22} = 0,3211$$

$$V_{03} = 0,4876$$

$$V_{13} = 0,2453$$

$$V_{23} = 0,1556$$

$$V_{04} = 0,4288$$

$$V_{14} = 0,3336$$

$$V_{24} = 0,2519$$

$$W_{01} = 0,4133$$

$$W_{11} = 0,4864$$

Langkah 1 : Lakukan Preprocessing / Normalisasi untuk data ujicoba secara

keseluruhan dengan rumus (Siang, 2005) :

$$x' = \frac{0.8(x - a)}{b - a} + 0.1$$

Dihasilkan nilai parameter

Parameter 1 (X1)

$$X1 = 0.1$$

$$X2 = 0.2$$

$$X3 = 0.9$$

Parameter 2 (X2)

$$X1 = 0.1$$

$$X2 = 0.7$$

$$X3 = 0.9$$

Parameter 3 (X3)

$$X1 = 0.1$$

$$X2 = 0.9$$

$$X3 = 0.9$$

Parameter 4 (X4)

$$X1 = 0.1$$

$$X2 = 0.9$$

$$X3 = 0.7$$

Langkah 2 : Dengan bobot tersebut , tentukan error untuk data ujicoba secara keseluruhan dengan rumus :

$$z_{in_j} = b_{ij} \sum_{i=1}^n x_i v_{ij}$$

$$z_{in_{01}} = 0.6031973838173046$$

$$z_{in_{11}} = 0.4989344712886769$$

$$z_{in_{21}} = 0.5826848079578912$$

$$z_{in_{02}} = 0.4197803044740819$$

$$z_{in_{12}} = 0.5029005914779161$$

$$z_{in_{22}} = 0.40534652643464464$$

$$z_{in_{03}} = 0.500007871849157$$

$$z_{in_{13}} = 0.41273726642109465$$

$$z_{in_{23}} = 0.44321592637583657$$

$$z_{in_{04}} = 0.4599406891683959$$

$$z_{in_{14}} = 0.4820885271194423$$

$$z_{in_{24}} = 0.4407390564438487$$

Dimana indeks z_{jn} berarti bobot untuk unit tersembunyi ke- j dan data *training* ke- n

$$y_{in} = b_2 k \sum_{i=1}^p z_i w_{ij}$$

Gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$$y_k = f(y_{in_k})$$

$$y_{in_{11}} = y_{11} = f(y_{in_{11}}) = 0,91071707$$

$$y_{in_{12}} = y_{12} = f(y_{in_{12}}) = 0,4453750$$

$$y_{in_{13}} = y_{13} = f(y_{in_{13}}) = 0,844857$$

$$y_{in_{14}} = y_{14} = f(y_{in_{14}}) = 0,84784$$

Sehingga,

Error dihitung berdasarkan nilai parameter

Nilai E untuk parameter 1 $= 0,5 \times \{(t_{k1}-y_{k1})^2 + (t_{kn}-y_{kn})^2\} = 0.022269452$

Setelah dilakukan iterasi ke-1000

Nilai E untuk parameter 2 $= 0,5 \times \{(t_{k1}-y_{k1})^2 + (t_{kn}-y_{kn})^2\} = 0.009994824$

Setelah dilakukan iterasi ke-994

Nilai E untuk parameter 3 $= 0,5 \times \{(t_{k1}-y_{k1})^2 + (t_{kn}-y_{kn})^2\} = 0.009996301$

Setelah dilakukan iterasi ke-987

Nilai E untuk parameter 4 $= 0,5 \times \{(t_{k1}-y_{k1})^2 + (t_{kn}-y_{kn})^2\} = 0.0010812$

Setelah dilakukan iterasi ke-1000

Setelah dilakukan iterasi untuk menentukan nilai *error* yang memenuhi syarat didapatkan hasil perhitungan

Hasil y untuk parameter 1 : 0,91071707

Hasil y untuk parameter 2 : 0,4453750

Hasil y untuk parameter 3 : 0,844857

Hasil y untuk parameter 4 : 0,84784

Lakukan Denormalisasi data , untuk merubahnya ke bilangan sebenarnya

$y_{1_hasil} = y_1 \cdot (\text{nilai tertinggi} - \text{nilai terendah}) + \text{nilai terendah}$

$= 0,91071707 \cdot (43 - 19) + 19$

$$= 40,857209$$

$$y_2_hasil = y_2 * (\text{nilai tertinggi} - \text{nilai terendah}) + \text{nilai terendah}$$

$$= 0,4453750 * (41 - 20) + 20$$

$$= 37,677120$$

$$y_1_hasil = y_1 * (\text{nilai tertinggi} - \text{nilai terendah}) + \text{nilai terendah}$$

$$= 0,844857 * (5 - 4) + 4$$

$$= 4,844857$$

$$y_2_hasil = y_2 * (\text{nilai tertinggi} - \text{nilai terendah}) + \text{nilai terendah}$$

$$= 0,84784 * (60 - 52) + 52$$

$$= 60,478406$$

Hasil prediksi untuk parameter Pendapatan Asli Daerah antara lain :

X1 / PPD	40,857209
X2 / HRD	37,677120
X3 / HPKDD	4,844857
X4 / LLPAD	60,478406

Tabel 4 Tabel Hasil Prediksi

Untuk hasil akhir perhitungan prediksi , dijumlahkan semua hasil tersebut :

$$y = \sum_{y=1}^n$$

$$= (40,857209 + 37,677120 + 4,844857 + 60,478406)$$

$$= 143.857592$$

Hasil dari prediksi adalah 143.857592 dengan hasil pembandingan sebenarnya 149

4.4 Integrasi Dalil dengan Prediksi Pendapatan Asli Daerah

Bahwasanya yang sudah disebutkan pada bab 1 tentang dalil yang mendasari adalah Surah Al-Lukman ayat 34 *“Sesungguhnya Allah, hanya pada sisi-Nya sajalah pengetahuan tentang hari Kiamat; dan Dia-lah yang menurunkan hujan, dan mengetahui apa yang ada dalam rahim dan tiada seorangpun yang dapat mengetahui (dengan pasti) apa yang akan diusahakannya besok dan tiada seorangpun yang dapat mengetahui di bumi mana Dia akan mati. Sesungguhnya Allah Maha mengetahui lagi Maha mengenal”*. Integrasi antara ayat tersebut dengan prediksi yaitu ayat ini menjelaskan bahwasanya segala sesuatu didasarkan pada prediksi dan manusia hanya berusaha melakukan ikhtiar termasuk menganalisa pendapatan asli daerah dimasa mendatang , dengan melakukan pembuatan program prediksi dimaksudkan untuk mendapatkan informasi berapa jumlah pendapatan asli daerah di masa mendatang.

Oleh karena itu , dibutuhkan perhitungan dan metode yang baik dalam melakukan prediksi yang terintegrasi dengan komputer dimana perhitungan tersebut dapat memunculkan hasil prediksi yang bisa menjadi informasi berapa jumlah pendapatan daerah di periode selanjutnya.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan uraian pada bab-bab sebelumnya dan berdasarkan hasil simulasi pada bab sebelumnya, maka pada penelitian ini dapat ditarik kesimpulan sebagai berikut : Dengan melihat rumusan masalah dan tujuan penelitian , bahwa penulis dapat merancang dan membuat perangkat lunak prediksi dengan penerapan Jaringan Syaraf Tiruan *Backpropagation* ,serta mendapatkan hasil prediksi berdasarkan error yang ditentukan pada perhitungan Jaringan Syaraf Tiruan *Backpropagation* pada studi kasus Pendapatan Asli Daerah Kabupaten Boyolali. Hasil yang didapatkan dari perhitungan dengan menggunakan Jaringan Syaraf Tiruan *Backpropagation* dari data parameter tahun 2011 sampai 2012 dan target data 2013 menghasilkan prediksi 143,857592 dengan data target 149.

5.2 Saran

Pada penelitian ini sebaiknya dilanjutkan pada tahap proses pengetesan atas hasil pembentukan pola peramalan oleh Jaringan Syaraf Tiruan dimana dimaksudkan untuk melihat kemampuan model yang dihasilkan oleh Jaringan Syaraf Tiruan pada tahap meramalkan kejadian berikutnya dengan berdasarkan pola-pola peramalan yang telah terbentuk sebelumnya.

Metode jaringan syaraf tiruan propagasi balik selain dapat diaplikasikan untuk meramalkan harga saham di pasar modal, metode ini juga dapat diaplikasikan untuk keperluan peramalan di bidang lain seperti tingkat pencemaran polusi, tingkat kerusakan ozon, jalur jaringan komunikasi, kedokteran, dan lain-lain. Hal ini

mengingat banyaknya manfaat dari pemanfaatan jaringan syaraf tiruan untuk beberapa bidang.



DAFTAR PUSTAKA

1. Dayhoff, Judith E. *Neural Network Architectures (An Introduction)*. USA : Van Nostrand Reinhold. 1990.
2. Patterson, Dan W. *Artificial Neural Networks (Theory And Applications)*. Prentice Hall. 1996.
3. Trippi, Robert R. And Turban, Efraim. *Neural Networks In Finance and Investing*, Probus Publishing Company. 1993.
4. Haykin, Simon. *Neural Networks (A Comprehensive Foundation)*. New York : Macmillan College Publishing Company. 1998
5. Kusumadewi, Sri. *Artificial Intelligence (Teknik Dan Aplikasinya)*. Jogjakarta : Graha Ilmu. 2003.
6. Kusumadewi, Sri. *Membangun Jaringan Syaraf Tiruan (Menggunakan MATLAB dan Excel Link)*. Jogjakarta : Graha Ilmu. 2004
7. Gade, Muhammad, 2000. *Akuntansi Pemerintahan*, Cetakan keenam, Lembaga Penerbit Fakultas Ekonomi Universitas Indonesia, Jakarta.
8. _____, Undang-Undang No. 33 Tahun 2004 *Tentang Perimbangan Keuangan antara Pemerintah Pusat dengan Daerah*
9. _____, Undang-Undang No. 34 Tahun 2004 *Tentang Pajak Daerah*
10. Siahaan, Marhot P, 2005. *Pajak Daerah dan Retribusi Daerah*, Edisi 1, PT Raja Grafindo Persada, Jakarta.
11. _____, Undang-Undang No. 18 Tahun 1997 *tentang Pajak Daerah dan Retribusi Daerah*.