

Sistem Evaluasi Kelayakan Mahasiswa Magang Menggunakan *Elman Recurrent Neural Network*

Agus Aan Jiwa Permana^{*1}, Widodo Prijodiprodjo²

¹Jurusan Manajemen Informatika, FTK, Undiksha, Singaraja

²Jurusan Ilmu Komputer dan Elektronika, FMIPA, UGM, Yogyakarta

e-mail: ^{*1}agus-aan@mail.ugm.ac.id, ²widodo@gmail.com

Abstrak

Jaringan Syaraf Tiruan (JST) dapat digunakan untuk memecahkan permasalahan tertentu seperti prediksi, klasifikasi, pengolahan data, dan robotik. Berdasarkan paparan tersebut, sehingga dalam penelitian ini mencoba menerapkan JST untuk menangani permasalahan dalam program magang yang sedang dihadapi dalam upaya untuk meningkatkan kompetensi, pengalaman, serta melatih softskill mahasiswa. Sistem yang dikembangkan dapat digunakan untuk mengevaluasi kelayakan mahasiswa dalam program magang ke luar daerah dengan menerapkan Elman Recurrent Neural Network (ERNN), sehingga dapat memberikan informasi yang akurat kepada pihak jurusan untuk menentukan keputusan yang tepat.

Struktur Elman dipilih karena dapat membuat iterasi jauh lebih cepat sehingga memudahkan proses konvergensi. Adapun metode pembelajaran yang digunakan adalah Backpropagation Through Time dengan model epochwise training mode. Sistem diimplementasikan dengan menggunakan bahasa pemrograman C# dengan basis data MySQL. Vektor input yang digunakan terdiri dari 11 variabel. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan akan cepat mengalami konvergen dan mampu mencapai nilai error paling optimal (minimum error) apabila menggunakan 1 hidden layer dengan jumlah neuron 20 unit. Akurasi terbaik dapat diperoleh dengan menggunakan LR sebesar 0.01 dan momentum 0.85 dimana akurasi rata-rata dalam pengujian mencapai 87.50%.

Kata kunci—Evaluasi, Kelayakan, Jaringan Syaraf Tiruan (JST), Elman Recurrent Neural Network, Magang

Abstract

Artificial Neural Network (ANN) can be used to solve specific problems such as prediction, classification, data processing, and robotics. Based on the exposure, so in this study tried to apply neural networks to handle problems in apprentice program facing in an effort to increase the competence, experience and soft skills training students. The system developed can be used to evaluate the students in the apprentice program to other regions by applying the Elman Recurrent Neural Network (ERNN), so it can provide accurate information to the department to determine appropriate decisions.

Elman structure was chosen because it can be create much more rapidly iterations so as to facilitate the convergence process. The learning method used is Backpropagation Through Time with model epochwise training mode. The system is implemented using the C # programming language with a MySQL database. Input vector used consists of 11 variables. The results showed that the developed system will rapidly converge and can reach optimal error value (minimum error) when using one hidden layer with 20 units number of neurons. Best accuracy can be obtained using the LR of 0.01 and momentum 0.85 which average accuracy reaches 87.50% in testing.

Keywords—Evaluation, Feasibility, Artificial Neural Network (ANN), Elman Recurrent Neural Network, Apprenticeship

1. PENDAHULUAN

Perkembangan *softcomputing* sebagai salah satu teknologi komputasi, telah banyak diaplikasikan dalam berbagai bidang. Salah satunya adalah jaringan syaraf tiruan (JST) yang memiliki kemampuan belajar terhadap informasi yang telah diterima dengan mensimulasikan proses pembelajaran seperti otak manusia. JST dapat digunakan untuk memecahkan permasalahan tertentu seperti prediksi, klasifikasi, pengolahan data, dan robotik, sehingga dicoba untuk diterapkan dalam permasalahan program magang yang sedang dihadapi.

Program magang dilakukan dalam upaya untuk meningkatkan kompetensi, pengalaman, serta melatih *softskill* mahasiswa. Selain itu, mahasiswa diharapkan juga dapat memiliki pengetahuan dalam memenuhi tuntutan dunia kerja. Jurusan Manajemen Informatika yang berada di bawah Fakultas Teknik dan Kejuruan (FTK), Universitas Pendidikan Ganesha (Undiksha) mengadakan magang dengan pesertanya adalah mahasiswa program diploma tiga yang sudah menginjak semester enam. Program yang memiliki bobot 3 SKS ini, memberikan kesempatan kepada mahasiswa untuk melakukan magang ke perusahaan atau tempat-tempat yang dalam aktivitasnya sudah memanfaatkan teknologi informasi (TI) dalam bidang kerjanya.

Kualitas sumber daya manusia di bidang TI yang diperlukan pasar juga semakin meningkat [1]. Namun, fakta di lapangan kompetensi mahasiswa semakin menurun disertai terjadinya beberapa kasus yang menyebabkan tempat magang menolak menerima mahasiswa dari universitas tertentu karena dianggap tidak kompeten dengan berbagai macam alasan. Persaingan yang semakin ketat dalam mencari tempat magang berkualitas untuk menghasilkan tenaga-tenaga kerja yang memiliki kemampuan praktis dan terampil dalam menggunakan dan menerapkan TI khususnya di Bali, membuat jurusan mengeluarkan kebijakan terkait dengan magang. Kebijakan yang dikeluarkan ini juga diharapkan mampu menjaga standar kompetensi mahasiswa terutama untuk yang mengambil magang keluar daerah.

Instansi tempat pelaksanaan magang justru merasa terganggu apabila mendapatkan mahasiswa yang tidak memiliki kemampuan yang dibutuhkan pada posisi dimana mahasiswa tersebut ditempatkan untuk magang [2]. Adapun kebijakan yang diterapkan sebelum mahasiswa melakukan magang adalah mengadakan tes uji ketrampilan (*skill*). Faktor penentu yang lain adalah indeks prestasi kumulatif (IPK) yang merupakan kemampuan kognitif dalam hal prestasi atau kemajuan belajar mahasiswa secara kumulatif. Selain itu, terdapat penilaian sikap dan perilaku menggunakan tes kepribadian yang sudah berstandar internasional yang dikembangkan oleh John Holland, meliputi unsur realistik, investigatif, artistik, sosial, *entrepreneur*, dan konvensional. Seseorang akan dikatakan kompeten pada suatu bidang, jika telah memenuhi 3 domain kompetensi pada bidang tersebut, yaitu domain *skill* (ketrampilan/psikomotorik), domain *knowledge* (pengetahuan/kognitif) serta domain *attitude* (sikap/afektif) [2].

Setelah program magang berakhir, ada penilaian yang dilakukan oleh jurusan dan supervisor terkait dengan produk dan aktivitas mahasiswa selama menjalani magang. Setelah program magang berakhir, ada penilaian yang dilakukan oleh jurusan dan supervisor terkait dengan produk dan aktivitas mahasiswa selama menjalani magang. Nilai akhir inilah yang sebenarnya mencerminkan kompetensi mahasiswa selama mengikuti program magang dimana nilai tersebut dapat dipengaruhi oleh berbagai macam faktor saat di lapangan disamping pengaruh tes uji keterampilan yang dilakukan sebelum magang.

Melihat kenyataan tersebut, dengan adanya beberapa kasus magang yang tersimpan di jurusan, dapat digunakan sebagai acuan untuk menentukan kelayakan mahasiswa dalam mengikuti program magang ke luar daerah di kemudian hari serta sebagai bahan pertimbangan untuk mengadakan evaluasi. Perlu dikembangkan sebuah sistem yang dapat membantu pihak jurusan dalam melakukan evaluasi terhadap kelayakan mahasiswa yang akan mengikuti magang berdasarkan data dari kasus yang sudah ada dibantu dengan JST. Struktur JST yang digunakan adalah *Elman Recurrent Neural Network* (ERNN) yang memiliki koneksi umpan balik (*feedback*) dari masukan sebelumnya, sehingga diharapkan dapat meningkatkan kinerja JST. Struktur tersebut membuat iterasi akan jauh lebih cepat dan konvergensi akan menjadi lebih cepat yang disebabkan karena Elman memiliki *feedback* yang menawarkan proses pembelajaran

yang lebih cepat pada jaringan [3]. Sistem yang dikembangkan diharapkan nantinya dapat membantu mengevaluasi dengan keakuratan hasil yang lebih baik serta dapat memberi rekomendasi lokasi magang yang sesuai sehingga dapat digunakan sebagai acuan dan membantu mempermudah dalam menentukan keputusan terkait dengan permasalahan.

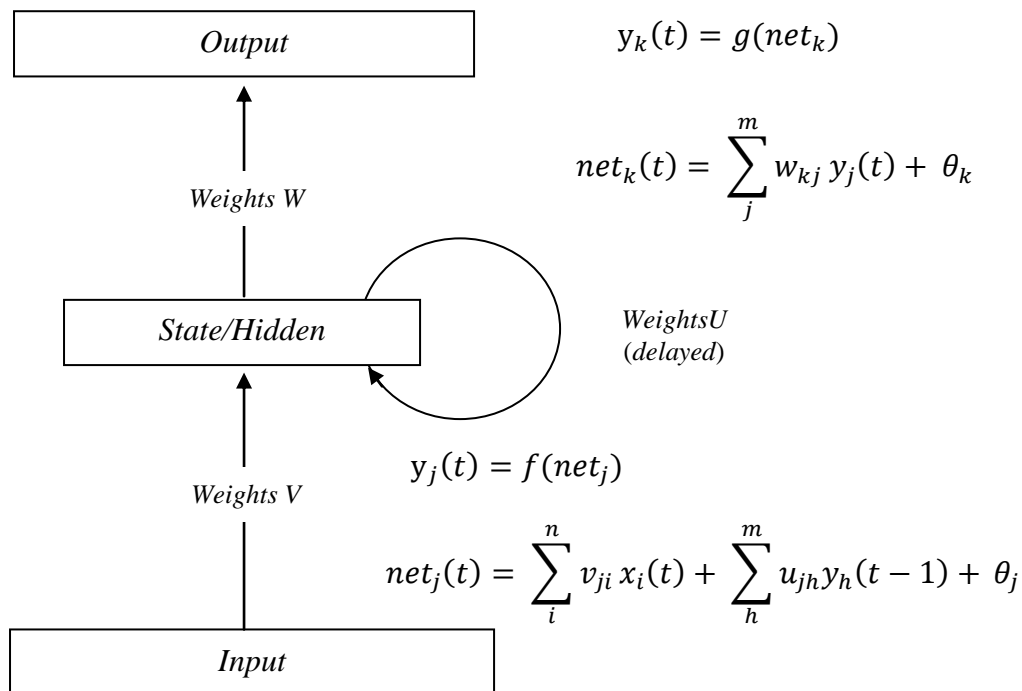
2. METODE PENELITIAN

2.1. Arsitektur ERNN (Simple Recurrent Network)

Elman Neural Network dianggap sebagai jaringan *recurrent* yang parsial karena pada umumnya koneksinya hanya berupa *feed forward* [4]. Elman [5], menjelaskan bahwa *simple recurrent network* memiliki arsitektur yang mirip dengan jaringan *recurrent state spacemodel* dengan layer *hidden* bersifat non linear, namun dengan *output* layer yang dimungkinkan bersifat non linear tanpa unit *delay*. Menurut [6], Elman mengatakan bahwa *Elman Neural Network* adalah modifikasi *feed forward* dengan perbedaan yang utama adalah pada tambahan layer *context neuron* yang menyediakan pola *hidden* unit untuk diumpun balik ke dirinya sendiri. Jaringan *recurrent* ini memiliki dua buah *input*, yaitu *input* sesungguhnya dan *contextual input*. Terdapat *feedback* dapat menyebabkan proses iterasi akan jauh lebih cepat, sehingga membuat kecepatan *update* parameter dan konvergensi menjadi lebih cepat [7].

2.2. Algoritma ERNN

Arsitektur *Elman Neural Network* hampir sama dengan arsitektur *feed forwardback propagation*, namun ditambah dengan layer *context* untuk menampung hasil *output* dari *hidden* layer seperti Gambar 1.



Gambar 1 Koneksi antar layer JST *recurrent* [8]

Adapun tahapan dalam algoritma JST *recurrent* Elman adalah :

- Sebuah jaringan *recurrent* sederhana, vektor masukannya juga sama disebarkan melalui lapisan berbobot, tetapi juga dikombinasikan dengan aktivasi keadaan sebelumnya melalui lapisan bobot tambahan yaitu " U ", dimana " U " adalah bobot *delay*.

- Setiap unit akan menghitung aktivasinya seperti pada jaringan *feed forward* Gambar 1 Pada setiap layer akan memiliki indeks masing-masing. Variabel k untuk node *output*, j node tersembunyi, h untuk node *context*, dan i untuk node *input*. Sebuah jaringan umpan maju memiliki vektor masukan x , yang disebarkan melalui lapisan berbobot v . Adapun keterangan dari Gambar 1 dapat disampaikan seperti pada Tabel 1.

Tabel 1 Keterangan variabel pada Gambar 1

Variabel	Keterangan	Variabel	Keterangan
k	<i>node output</i>	n	: jumlah <i>node</i> masukan
h	<i>node context</i>	m	: jumlah <i>node context / hidden</i>
j	<i>node hidden</i>	f, g	: fungsi <i>output</i>
i	<i>node input</i>	v	: bobot <i>input</i> dengan <i>hidden layer</i>
x	<i>input</i> (vektor masukan)	w	: bobot <i>hidden</i> dengan <i>output</i>
θ	bias	u	: bobot <i>context</i> dengan <i>hidden</i>

2.3. Analisa Data

Terdapat 110 data kasus yang akan dibagi lagi menjadi beberapa bagian yaitu :

- 80% untuk *training* (88 data).
- 20% untuk *testing* (22 data).

2.3.1. Vektor input

Merupakan parameter-parameter yang akan mempengaruhi hasil kompetensi mahasiswa selama magang di lapangan. Vektor ini, dapat dibagi menjadi tiga bagian utama seperti : nilai *skill*, IPK, dan sikap dan perilaku seperti dalam Tabel 2.

Tabel 2 Vektor *input* sistem detail

Simbol	Nama vektor	Kompetensi Dasar	Skor
A	Data uji ketrampilan bidang informatika atau komputer untuk mengetahui <i>skill</i> mahasiswa, antara lain mencakup :		0 - 100
A ₁	Programming	Dapat mengkoneksikan <i>database&query</i>	
A ₂	Web desain	Mendesain layout sebuah situs	
A ₃	Multimedia	Membuat suatu animasi <i>flash</i> sederhana	
A ₄	Jaringan	Dapat melakukan konfigurasi IP/ <i>Subnetting</i>	
B	Data tentang kemampuan kognitif mahasiswa. Data yang digunakan adalah nilai IPK mahasiswa terakhir sebelum magang yang berasal dari KDN.		0 - 4
C	Data tentang sikap dan perilaku mahasiswa yang ditunjukkan dengan hasil tes kepribadian, yang memiliki unsur-unsur sebagai berikut :		0 - 42
C ₁	Realistik	Minat dan kemampuan yang realistik	
C ₂	Investigatif	Minat dan kemampuan di bidang investigasi	
C ₃	Artistik	Minat dan kemampuan dalam bidang seni	
C ₄	Sosial	Minat dan kemampuan dalam bidang sosial	
C ₅	<i>Enterpreneur</i>	Minat dan kemampuan di pemasaran/bisnis	
C ₆	Konvensional	Menyukai tugas terstruktur dan aturan	

Nilai *skill* mahasiswa diperoleh dari hasil tes yang dilakukan oleh pihak jurusan. Adapun prosedur dalam tes ini, jurusan akan menunjuk beberapa orang pengujian yang kompeten

dalam bidang komputer sesuai dengan kompetensi yang diujikan. Nilai IPK menunjukkan kemampuan kognitif mahasiswa, yang diperoleh selama proses pembelajaran di bangku perkuliahan. Nilai IPK yang digunakan, berasal dari kutipan daftar nilai (KDN). Nilai sikap dan perilaku terkait dengan kompetensi mahasiswa diperoleh dari tes kepribadian dengan cara memberikan kuisioner yang bernama *Inventory Personal Survey* (IPS) kepada mahasiswa yang akan mengikuti magang. Vektor input akan disebarakan melalui sebuah lapisan berbobot, dimana bobotnya diset secara random menggunakan rumus pada persamaan 3 yang digunakan Haykin [9].

2.3.2. Target output

Penentuan target dari jaringan didasarkan pada hasil akhir nilai magang yang seperti Tabel 3. Nilai magang berasal dari kombinasi penilaian yang dilakukan penguji dan supervisor saat mahasiswa telah selesai melakukan magang, sehingga dipengaruhi oleh berbagai macam faktor. Nilai inilah yang mencerminkan bagaimana kompetensi mahasiswa sebenarnya di lapangan yang selanjutnya dapat digunakan sebagai acuan dalam memprediksi kompetensi mahasiswa yang akan mengikuti magang ke luar daerah.

Tabel 3 Klasifikasi nilai magang mahasiswa
(Modifikasi prosedur penilaian, [10])

Nilai Magang	Keterangan	Klasifikasi	Target Output (Biner)
85 - 100	Layak	3	1 1
70 - 84	Cukup layak	2	1 0
55 - 69	Tidak layak	1	0 1

Target dalam jaringan dikelompokkan ke dalam tiga kelas yaitu : layak, cukup layak, dan tidak layak. Setelah nilai dalam Tabel 3 dinormalisasi dengan persamaan 1 dan 2 sehingga rentangan nilai target menjadi sebagai berikut :

- 1) *Output* ($0.85 \leq n \leq 1.00$), berarti “layak”.
- 2) *Output* ($0.70 \leq n \leq 0.84$), berarti “cukup layak”.
- 3) *Output* ($0.55 \leq n \leq 0.69$), berarti “tidak layak”.

$$X_{norm} = X_{low} + \Delta * (X - X_{min}) \quad (1)$$

$$\Delta = \frac{X_{high} - X_{low}}{X_{max} - X_{min}} \quad (2)$$

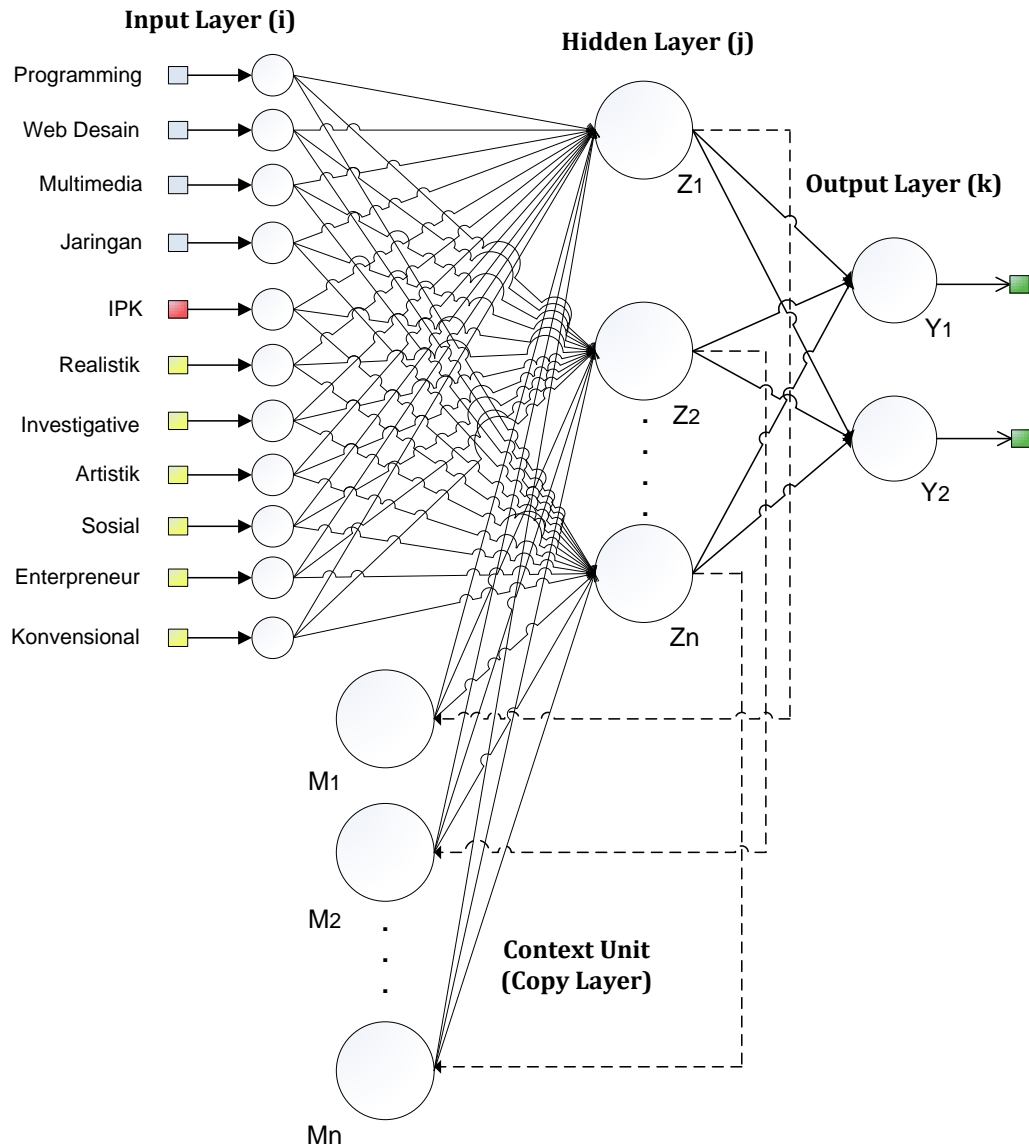
Dengan keterangan sebagai berikut :

X_{norm}	=	Data yang telah dinormalisasi.
X	=	Data yang belum dinormalisasi.
X_{low}	=	Rentangan terkecil (0).
X_{high}	=	Rentangan terbesar (1).
X_{max}	=	Nilai terbesar dari data.
X_{min}	=	Nilai terendah dari data.

$$\left(-\frac{2.4}{F_i}, +\frac{2.4}{F_i} \right) \quad (3)$$

2.3.3. Rancangan arsitektur

Gambar 2 menunjukkan sebuah arsitektur jaringan dari Sevakhm yang sedang dikembangkan dengan metode ERNN.



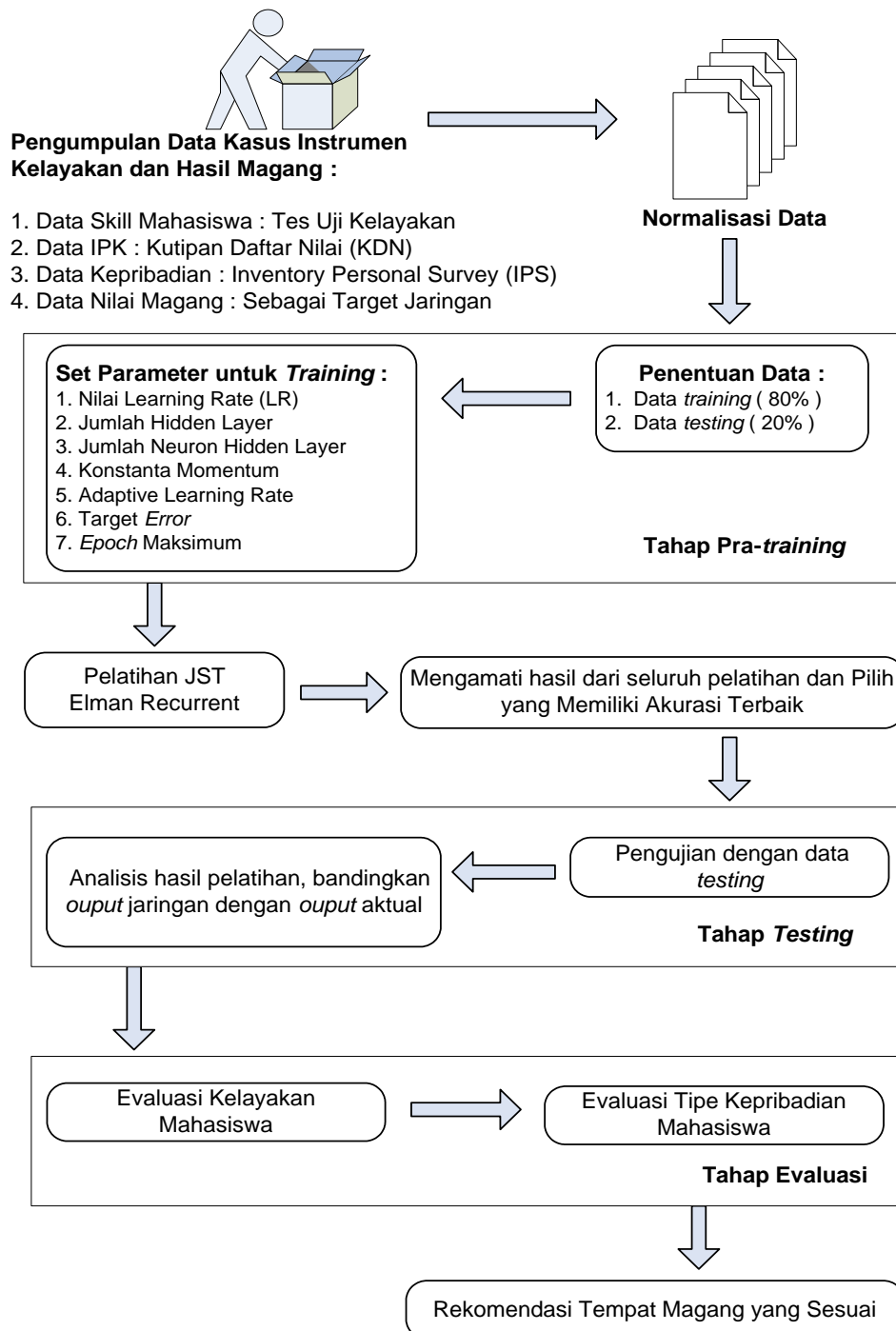
Gambar 2 Rancangan arsitektur ERNN dengan satu *hidden* layer

Arsitektur pada Gambar 2 memiliki tiga lapisan, antara lain : lapisan masukan (*input*), lapisan tersembunyi (*hidden*), dan lapisan keluaran (*output*). Pada lapisan masukan terdiri dari 11 unit sel syaraf (*neuron*), dengan sejumlah unit *neuron* pada lapisan *hidden*, dan 2 unit *neuron* lapisan keluaran.

2.4. Rancangan Sistem

Secara umum alur kerja sistem dapat digambarkan dalam sebuah diagram seperti pada Gambar 3. Pada Gambar 3 diawali dengan pengumpulan data kasus instrumen kelayakan yang meliputi : nilai *skill*, IPK, dan nilai unsur-unsur dalam tes kepribadian. Pengumpulan nilai hasil magang juga dilakukan untuk dipergunakan sebagai target keluaran dari JST. Setelah semua

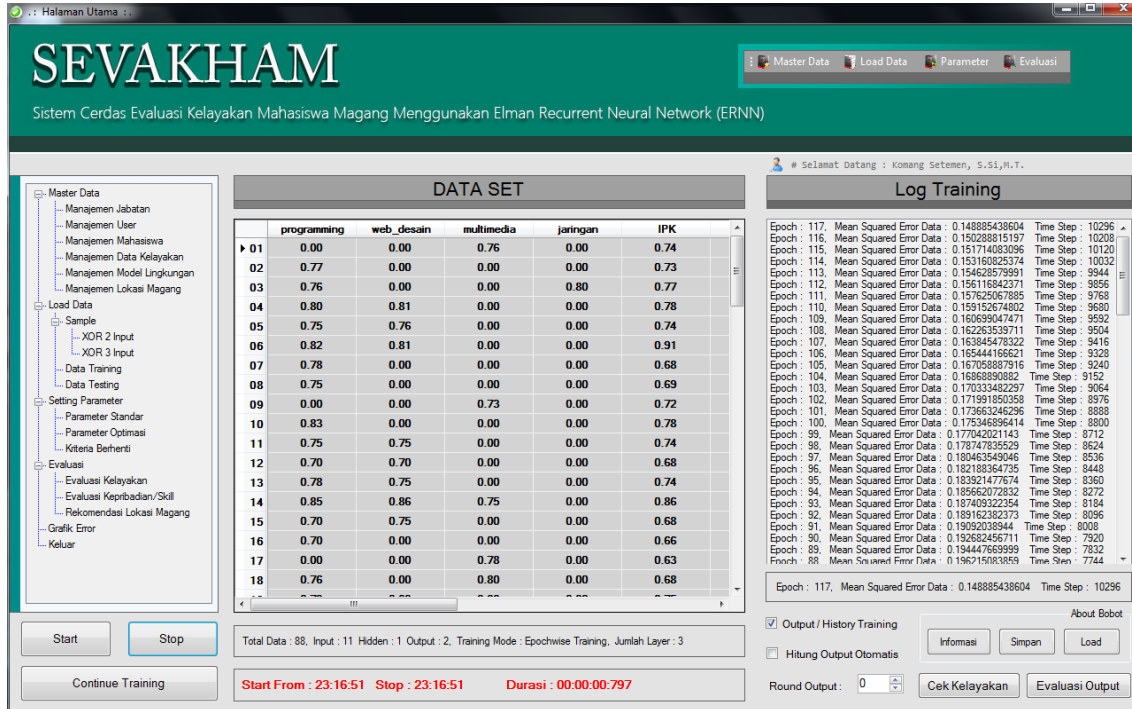
data terkumpul, selanjutnya dilakukan proses normalisasi terhadap data kasus untuk menyamakan rentangan dari data sebelum akhirnya diproses. Sebelum melakukan pelatihan terdapat beberapa hal perlu dilakukan, seperti menentukan jumlah data yang akan digunakan untuk *training* dan *testing*, dan melakukan setingan parameter JST. Setelah melakukan pelatihan dan mendapatkan arsitektur yang terbaik, selanjutnya masuk pada tahap *testing*. Proses evaluasi dilakukan setelah mendapatkan hasil *testing* dengan akurasi terbaik. Sistem juga diharapkan mampu memberikan rekomendasi tempat magang sesuai dengan kompetensi yang dimiliki oleh masing-masing mahasiswa.



Gambar 3 Diagram alur kerja sistem

2.5. Implementasi

Sesuai dengan rancangan sistem yang sudah dibuat, dapat diimplementasikan beberapa halaman yang merupakan bagian dari sistem yang dikembangkan. Adapun tampilan dari halaman utama sistem yang digunakan untuk *training* dan *testing* dapat dilihat pada Gambar 4.



Gambar 4 Halaman utama sistem

3. HASIL DAN PEMBAHASAN

Tahapan pengujian (*training*) merupakan suatu tahapan terpenting dalam sebuah JST untuk mengajarkan suatu pola sampai jaringan berhasil mengenali pola tersebut. Pada penelitian ini yang menggunakan arsitektur *Elman* ini, dalam penentuan arsitektur terbaiknya dilakukan dengan *trial and error*. Pada tahap ini, terdapat iterasi (*timestep*) akan mencapai hingga puluhan ribu atau lebih apabila pola yang dikenalkan memiliki jumlah yang banyak dan beragam.

Tabel 4 Hasil pengujian dengan target 0.001

No	Neuron	Epoch	MSE	Durasi	Runtime(detik)
1	2	*5000	0.011584308	00:00:07:514	7.51400
2	5	*5000	0.011461087	00:00:08:580	8.58000
3	10	580	0.000996874	00:00:02:642	2.64200
4	15	503	0.000998065	00:00:04:104	4.10400
5	20	667	0.000965526	00:00:04:250	4.25000
6	25	*5000	0.2746639	00:00:23:617	23.61700
7	30	*5000	0.40912463	00:00:33:025	33.02500
8	35	*5000	0.295221298	00:00:42:565	42.56500
9	40	*5000	0.289193365	00:00:44:389	44.38900
10	45	*5000	0.443181818	00:00:53:016	53.01600

Terdapat beberapa faktor yang dapat mempengaruhi kinerja sistem agar dapat menghasilkan keluaran yang baik, yaitu nilai *learning rate*, momentum, target *error*, *epoch* maksimal, serta jumlah *hidden layer* beserta *neuron-neuron*.

3.1. Pengujian dengan parameter jumlah neuron hidden layer

Beberapa percobaan dilakukan, dan hasil yang dijadikan sebagai acuan awal adalah dengan membandingkan target *error* dengan toleransi 0.001 dengan 0.0001. Proses akan dihentikan apabila nilai fungsi biaya/kinerja kurang dari atau sama dengan target *error*. Pada kedua percobaan ini, menggunakan setingan parameter seperti Tabel 5. Pada percobaan dengan target 0.001 diperoleh hasil seperti Tabel 4, sedangkan dengan target 0.0001 diperoleh hasil seperti Tabel 6.

Tabel 5 *Setting* parameter untuk pengujian

Parameter	Nilai	Keterangan
<i>Learning rate</i>	: 0.05	Kecepatan pembelajaran sistem
<i>Maxepoch</i>	: 5000	<i>Epoch</i> maksimal untuk berhenti
Momentum	: 0.85	Konstanta momentum
Jumlah <i>hidden</i>	: 1	Jumlah lapisan tersembunyi

Tabel 6 Hasil pengujian dengan target 0.0001

No	<i>Neuron</i>	<i>Epoch</i>	MSE	Durasi	<i>Runtime</i> (detik)
1	2	*5000	0.274664068	00:00:42:367	42.36700
2	5	*5000	0.011478904	00:00:11:627	11.62700
3	10	2837	1.00E-04	00:00:07:219	7.21900
4	15	2382	9.995819E-0.5	00:00:07:371	7.37100
5	20	2326	9.99E-05	00:00:12:212	12.21200
6	25	3453	1.00E-04	00:00:07:976	7.97600
7	30	2724	1.00E-04	00:00:13:615	13.61500
8	35	*5000	4.43E-01	00:00:36:972	36.97200
9	40	*5000	2.99E-01	00:00:37:457	37.45700
10	45	*5000	4.43E-01	00:00:35:556	35.55600

Keterangan :* = Belum mencapai konvergen

Berdasarkan Tabel 4 dan 6 menunjukkan bahwa dengan target 0.0001 proses *training* berjalan lebih lama dibandingkan dengan target 0.001. Jumlah *neuron* pada *hidden layer* dengan nilai *mean square error* (MSE) terkecil diperoleh pada jumlah 20 unit, baik pada percobaan pertama ataupun kedua. MSE mengambil nilai rata-rata kuadrat *error* yang terjadi antara *output* dengan target.

Percobaan lain juga dilakukan untuk mendapatkan arsitektur jaringan dengan jumlah layer yang tepat untuk menghasilkan evaluasi yang terbaik yaitu dengan 2 *hidden layer*, dimana pada lapisan pertama terdapat 20 *neuron* dan yang kedua terdapat 35 *neuron*. Arsitektur ini menghasilkan nilai MSE sebesar 0.000958507173 dengan *runtime* 117.984 detik. Walaupun nilai *error* kecil, namun memerlukan waktu yang sangat lama untuk mencapai konvergen. Pada kasus lain, percobaan juga dilakukan dengan menggunakan 3 *hidden layer* dengan arsitektur terbaik menggunakan 5 *neuron* di setiap lapisannya. Nilai MSE sebesar 0.000999675649 dengan *runtime* 4.112 detik, namun hasil pembelajaran dalam mengenali pola yang dilakukan tidak sebaik seperti pada percobaan pertama dan kedua.

3.2. Pengujian dengan parameter learning rate (LR)

Parameter *learning rate* berpengaruh pada keakuratan sebuah prediksi untuk mengevaluasi kelayakan mahasiswa dalam program magang ke luar daerah. Berapa nilai LR yang cocok digunakan dapat diperoleh melalui proses *training*. Nilai LR yang digunakan adalah 0.01, 0.025, 0.05, dan 0.075 dengan momentum sebesar 0.85. Pada setiap nilai LR yang digunakan dapat dihitung berapa persentase data dapat dikenali jaringan seperti Tabel 7 dengan target 0.001 dan dengan target 0.0001, dapat dihitung dengan menggunakan rumus persamaan 4. Akurasi terbaik saat pengujian dengan target 0.0001 dan target 0.001 bernilai sama yaitu 90.91%.

$$\frac{\text{Jumlah data yang dikenali}}{\text{Jumlah total seluruh data}} \times 100\% \quad (4)$$

Tabel 7 Persentase pengujian LR dengan target 0.001

Jumlah total data :22					
Nilai LR	Epoch	MSE	Jumlah Data		Prosentase
			Dikenali	Tidak	
0.01	2151	0.000999858	20	2	90.91%
0.025	1235	0.000997785	19	3	86.36%
0.05	874	0.000984879	19	3	86.36%
0.075	712	0.000995399	19	3	86.36%

Tabel 8 Persentase pengujian LR dengan target 0.001

Jumlah total data :22					
Nilai LR	Epoch	MSE	Jumlah Data		Prosentase
			Dikenali	Tidak	
0.01	6631	9.999E-05	19	3	86.36%
0.025	6449	9.998E-05	20	2	90.91%
0.05	1689	9.988E-05	19	3	86.36%
0.075	1504	9.995E-05	19	3	86.36%

3.3. Pengujian dengan parameter momentum

Nilai momentum digunakan pada jaringan untuk mengatur kecenderungan bagi bobot untuk bergerak ke arah tertentu dari pergerakan nilai bobot ke nilai optimalnya atau keluar dari *local minima*. Penggunaan momentum dalam proses *training* tidak boleh terlalu besar atau terlalu kecil. Adapun nilai momentum yang coba diujikan adalah : 0.75, 0.85, dan 0.95 seperti pada Tabel 9 dengan menggunakan target *error* 0.001 dan 0.0001.

Pada pengujian dengan target 0.001 menunjukkan bahwa dengan nilai momentum sebesar 0.85 memiliki nilai *epoch* paling kecil. Hal ini berarti bahwa konvergensi lebih cepat dicapai dan nilai ini cocok untuk digunakan pada setiap nilai LR. Sedangkan dengan target 0.0001 menunjukkan bahwa *epoch* dengan nilai MSE terkecil berada pada momentum dengan nilai 0.85 dengan nilai LR 0.05.

Tabel 9 Hasil pengujian momentum

LR	M	Target 0.001			Target 0.0001		
		Epoch	MSE	Durasi	Epoch	MSE	Durasi
0.01	0.75	4236	0.000997	00:00:34:854	7265	1.00E-04	00:01:48:909
	0.85	2037	0.000999	00:00:13:457	9254	1.00E-04	00:01:53:586
	0.95	3361	0.000990	00:00:27:635	3386	1.00E-04	00:00:41:848
0.025	0.75	2043	0.000999	00:00:24:665	11304	1.00E-04	00:02:16:111
	0.85	1066	0.000998	00:00:13:290	*10413	2.76E-01	00:01:17:835
	0.95	1499	0.000940	00:00:18:665	3055	1.00E-04	00:00:24:296
0.05	0.75	775	0.000998	00:00:09:422	5637	1.00E-04	00:01:07:592
	0.85	701	0.000991	00:00:08:554	1857	9.99E-05	00:00:23:933
	0.95	*5000	0.274664	00:00:43:766	*10110	1.14E-02	00:01:10:721
0.075	0.75	743	0.000996	00:00:09:130	*10103	2.75E-01	00:01:24:771
	0.85	384	0.000996	00:00:04:801	2135	1.00E-04	00:00:25:560
	0.95	*5000	0.443181	00:00:48:599	10104	4.42E-01	00:01:12:184

Keterangan : * = Belum mencapai konvergen

Pada pengujian momentum dengan target 0.0001 memerlukan waktu yang lebih lama untuk mencapai *konvergen*. Berdasarkan hasil pengujian seperti pada Tabel 9 dengan nilai momentum 0.85 dapat mempercepat proses konvergensi dengan nilai *error* yang sudah relatif kecil.

3.4. Analisa hasil training dan testing

Pada penelitian ini, proses *testing* diawali dengan memilih data sebanyak 22 dari 110 data secara manual. Data yang digunakan untuk *testing* adalah data baru yang tidak diikuti sertakan dalam *training*. Akurasi hasil *testing* sangat dipengaruhi oleh bobot hasil *training*, yang menunjukkan kemampuan jaringan dalam mengenali pola-pola yang dilatih. Berdasarkan hasil percobaan yang dilakukan, dengan target 0.001 dan 0.0001, diperoleh akurasi terbaik yang sama dalam mengenali data sebesar 90.91%. Namun dilihat dari jumlah *epoch* target 0.001 memiliki jumlah yang jauh lebih sedikit dibandingkan dengan menggunakan target 0.0001. Akurasi hasil *testing* terbaik adalah dengan menggunakan LR sebesar 0.01 untuk target 0.001 dan LR sebesar 0.005 untuk target 0.0001.

3.5. Analisa hasil rekomendasi lokasi magang

Setelah melakukan evaluasi kelayakan, maka sistem akan memberikan informasi tentang status kelayakan mahasiswa. Berdasarkan data klasifikasi tersebut, proses selanjutnya adalah melakukan evaluasi kepribadian. Setelah hasil dari proses evaluasi kepribadian dilakukan dan disimpan, nantinya akan dapat digunakan sistem untuk memberikan sebuah rekomendasi lokasi magang kepada mahasiswa yang bersangkutan berdasarkan tipe kepribadian dan status kelayakannya. Mahasiswa dengan status layak diberikan rekomendasi lokasi magang ke luar daerah sesuai dengan tipe kepribadian yang dimiliki. Mahasiswa dengan status cukup diberikan rekomendasi ke luar daerah dan dalam kota, namun penetapan lokasi akan dibicarakan lagi dalam rapat jurusan. Sedangkan untuk mahasiswa dengan status tidak layak hanya akan direkomendasikan lokasi magang di dalam kota atau sekitar kampus saja. Pada tahap ini, sistem telah mampu membuat rekomendasi lokasi sesuai dengan yang diharapkan.

4. KESIMPULAN

Berdasarkan penelitian yang sudah dilakukan dan berdasarkan hasil pengujian sistem, maka beberapa kesimpulan yang diperoleh antara lain seperti berikut :

1. Sistem dapat mengevaluasi kelayakan mahasiswa yang akan mengikuti magang berdasarkan kompetensi yang dimilikinya ke dalam tiga klasifikasi yaitu layak, cukup, dan tidak layak serta dapat merekomendasikan ke lokasi magang sesuai dengan kompetensi yang sedang dibutuhkan pada tempat magang.
2. Sistem akan cepat *konvergen* dan mampu mencapai *minimum error*, menggunakan sebuah *hidden layer* dengan 20 unit *neuron*. Akurasi terbaik sebesar 90.91% yang diperoleh dengan menggunakan momentum 0.85. Sedangkan LR yang digunakan sebesar 0.01 jika target *error* 0.001 dan 0.025 jika target *error* 0.0001. Pengujian sistem sudah dapat berhasil dengan baik, dengan akurasi 90.91%. Hasil ini memperlihatkan bahwa jaringan telah dapat mengenali dengan baik pola yang telah dilatih, walau terdapat beberapa yang belum mampu untuk dikenali.

5. SARAN

Apabila melakukan pengembangan sistem ke depannya, maka disarankan untuk memperhatikan beberapa hal seperti : perlu dilakukan pengujian dengan data yang lebih banyak lagi agar jaringan lebih banyak belajar dalam mengenal pola-pola baru. Selain itu perlu dikembangkan dengan menggunakan algoritma pembelajaran lain misalnya *Kalman filter*, *Levenberg Marquardt Algorithm* (LMA), atau dengan pembelajaran *unsupervised learning*.

DAFTAR PUSTAKA

- [1] Setemen, K., Suputra, P. H., Purnamawan, K., 2009, Studi Kelayakan Pembukaan Program Diploma III Teknologi Informasi Undiksha Singaraja, *JPTK*, Vol. 6, No. 1, Hal.1-16.
- [2] Sugiyanto, Suprapedi, Himawan, H., 2009, Penentuan Kompetensi Mahasiswa Berdasarkan Prestasi Akademik, Sertifikasi Kompetensi, Minat, dan Kegiatan Pendukung, *Jurnal Teknologi Informasi*, Vol. 5, No. 2, Hal. 766-774.
- [3] Pasila, F., Lesmana, T., Ferdinando, H., 2009, Elman Neural Network Application with accelerated LMA Training for East Java-Bali Electrical Load Time Series Data Forecasting, http://jportfolio.petra.ac.id/user_files/98-056/ENN_ICTS.pdf, (diakses: 30 Maret 2012, 10:11).
- [4] Fauset, L. 1994, *Fundamentals of Neural Network : Architectures, Algorithms, and Application*, Prentice-Hall : New Jersey.
- [5] Haykin, S., 1999, *Neural Network A Comprehensive Foundation*, Second Edition, Pearson Education, Singapore.
- [6] Habarulema, J. B., 2010, A Contribution to TEC Modelling Over Southern africa Using GPS Data, *Desertasi*, Rhodes University, South Africa.
- [7] Noorviani, Y., 2011, Penerapan Elman Recurrent Neural Network untuk Diagnosis Gangguan Autis Pada Anak, *Tesis*, Fakultas Matematika dan Ilmu Pengetahuan Alam, Program Studi S2 Ilmu Komputer, Universitas Gadjah Mada, Yogyakarta.
- [8] Boden, M., 2001, *A guide to recurrent neural network dan backpropagation*, Computer and Electrical Engineering, Halmstad University, 13 November 2001.
- [9] Negnevitsky, M., 2005, *Artificial Intelligence A Guide to Intelligent Systems*, Second Edition, Pearson Education, London.
- [10] Undiksha, 2011, *Pedoman Studi Program Sarjana dan Diploma Universitas Pendidikan Ganesha*, Universitas Pendidikan Ganesha, Singaraja.