|  |  |
| --- | --- |
| jepin logoJEPIN (Jurnal Edukasi dan Penelitian Informatika)ISSN(e): 2548-9364 / ISSN(p) : 2460-0741 | Vol. xNo. ymm yy |

**Prediksi Kelulusan TOEFL Menggunakan Metode Resilient Backpropagation**Harly Okprana1, Muhammad Ridwan Lubis2, Jaya Tata Hadinata3
1, 3 STIKOM Tunas Bangsa, 2 AMIK Tunas Bangsa Pematangsiantar
1 harly@amiktunasbangsa.ac.id, 2 ridwanlubis@amiktunasbangsa.ac.id, 3 jayahardinata@gmail.com

**Abstrak**

Prediksi kelulusan TOEFL peserta didik Michigan Computer English Course diperlukan untuk meninjau sejauh mana tingkat pemahaman peserta didik. Backpropagation merupakan salah satu teknik yang baik digunakan untuk prediksi, akan tetapi jika backpropagation dalam training data dengan jumlah besar serta parameter-parameter yang digunakan kurang tepat, akan terjadi proses traning data lebih lambat. Maka diperlukan metode optimasi untuk mempercepat training Bacpropagation dalam memprediksi kelulusan dengan menggunakan metode Resilient Backpropagation. Data yang diolah sebanyak 182 data peserta didik tahun 2016-2018. Tingkat akurasi pengujian semakin baik yakni 100% dengan nilai MSE semakin kecil 0.00342 serta nilai Epoch juga semakin kecil menjadi 5. Sehingga penelitian ini menjadi indikator dalam pengembangan prediksi TOEFL dimasa yang akan datang.

Kata kunci : Backpropagation, Optimasi, Resilient Backpropagation, Prediksi TOEFL, Kelulusan.

# Pendahuluan

# Kecerdasan buatan atau disebut juga Artificial Intelegent (AI) merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia. Salah satu yang dipelajari pada kecerdasan buatan adalah Jaringan Saraf Tiruan (JST) dengan menggunakan metode backpropogation. Jaringan Saraf Tiruan (JST) merupakan salah satu representasi buatan otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia[1][2][3]. Teknik paling populer pada metode JST adalah algoritma backpropation, kinerjanya yang sangat baik membuat backpropagation banyak digunakan memecahkan masalah di dunia nyata [4]. Dari beberapa penelitian, backpropogation masih memiliki kelemahan diantaranya waktu training yang terlalu lama sehingga menghasilkan Epoch yang sangat banyak, Maka diperlukan algortima untuk optimasi training bacpropagation. Salah satu metode yang dapat digunakan untuk optimasi training pada backpropagation adalah metode resilient backpropagation dimana metode ini memiliki keunggulan untuk mencapai titik konvergen training lebih cepat dibandingkan dengan backpropagation standar[2].

Berdasarkan uraian diatas, penelitian ini akan menerapkan algoritma resilient backpropagation untuk mengoptimumkan akurasi dari prediksi kelulusan TOEFL peserta didik Michigan Computer English Course Pematangsiantar. Dengan dilakukannya prediksi terhadap kelulusan peserta didik dalam ujian TOEFL dapat dilakukan upaya-upaya persiapan serta pembekalan pada peserta didik sehingga diharapkan dapat lulus dengan baik pada ujian TOEFL sebenarnya.

# Metode Penelitian

Penelitian ini menggunakan data nilai ujian simulasi TOEFL Michigan Computer English Course tahun 2016-2018, data tersebut akan di uji untuk mendapatkan prediksi yang akurat menggunakan algoritma backpropagation, kemudian di obtimasi menggunakan algoritma resilient bacpropagation sehingga dapat menghasilkan rekomendasi keputusan untuk Michigan Computer English Course.

* 1. **Backpropagation**

Ada tiga tahap Pelatihan Backpropagation yaitu tahap maju dimana layer Input dan output dapat dihitung maju dengan ditentukan melalui fungsi aktivasi[5][6] . Kedua adalah tahap mundur, yaitu kondisi ketika target output nya yang dinginkan memiliki selisih dengan jaringan keluarannya merupakan kesalahan terjadi. Sehingga dalam kesalahan tersebut dipropagasikan mundur, dimulai dari layer output pada unit-unit garis yang berhubungan. Tahap ketiga adalah merubah nilai bobot agar memperkecil kesalahan yang terjadi [7][8].

Ketiga tahap tersebut terus-menerus dilakukan sampai kepada kondisi penghentiannya dicapai. Jumlah iterasi atau kesalahan merupakan suatu kondisi dimana penghentian yang sering digunakan. Iterasi atau kesalahan akan diberhentikan jika melewati batas maksimal yang telah ditentukan atau batas toleransi jika dibandikan lebih besar dari kesalahan yang terjadi. Langkah-langkah dalam algoritma backpropagation sebagai berikut [9]:

Langkah 0 : Menginisialisasi semua bobot (set bobot dengan bilangan random yang kecil)

Langkah 1 : Jika suatu kondisi penghentian belum terpenuhi, lakukan langkah 2-9.

Langkah 2 : Untuk setiap pasangan training, lakukan langkah 3-8.

Langkah 3 : Dari setiap unit input menerima dan unit tersembunyi diteruskan ke di atasnya.

Langkah 4 : Hitung semua keluaran di unit tersembunyi zj (j =1, 2, …, p).

z\_netj =  (1)

zj = f (z\_netj) =  (2)

Langkah 5 : Hitung semua keluaran jaringan di unit yk (k = 1, 2, …, m)

y\_netk =  (3)

 zk = f (z\_netk) =  (4)

Langkah 6 : Menghitung faktor δ pada unit keluaran didasarkan kesalahan pada unit keluaran

 yk (k = 1, 2, …, m) δk = (tk – yk) f’(y\_netk) = (tk – yk) yk (1- yk) (5)

δk merupakan unit kesalahan yang akan dipakai pada perubahan bobot layar dibawahnya [5].

(langkah 7) . Hitung suku perubahan bobot wkj ( digunakan untuk merubah bobot wkj ) dengan laju percepatan α.Δwkj = α δk zj ; k = 1, 2, …, m ; j = 0, 1, …, p

Langkah 7 : Hitung faktor δ untuk unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi zj ( j = 1, 2,…,p)

 **** (6)

 Faktor δ unit tersembunyi:

  (7)

 Hitung suku perubahan bobot vji (digunakan untuk merubah bobot vji)

Δvji = α δj xi ; j = 1, 2, …, p ; i = 0, 1, n.

Langkah 8: Hitung semua perubahan bobot garis yang menuju ke unit keluaran: *wkj*(*baru*) = *wkj* (*lama*) + Δ*wkj* (8)

 (k = 1, 2, …,m ; j = 0, 1, …, p)

Perubahan bobot garis yang menuju ke unit tersembunyi:

*vji* (*baru*) = *vji* (*lama*) + Δ*vji*(9)

(j = 1, 2, …,p ; i = 0, 1, …, n)

Langkah 9 : Menguji kondisi berhenti apakah sudah terpenuhi. Yaitu ketika kondisi sudah mencapai batas kesalahan yang diharapkan atau batas maksimalnya sudah terpenuhi.

* 1. **Resilient Backpropagation**

Resilient Backpropagation merupakan perkembangan dari backpropagation. Resilientyaitu algortima *neural network* yang bersifat *supervised* dan *adaptive learning*. Pada resilient backpropagationparameter sudah ditetapkan jadi tidak diperlukan penentuan *learning rate* lagi [6]. Tahap pelatihan *feedforward* pada algoritma resilientbackpropagation sama dengan pada algoritma backpropagation, yang membedakan algoritma tersebut yaitu pada waktu melakukan *updateweight* dengan *learningrate* pada pelatihan *backward* [3]. Perhitungan *learningrate* akan dijabarkan pada langkah-langkah sebagai berikut :

1. Perhitungan Nilai Δ*i*(*t*)

Untuk menentukan nilai Δ*i*(*t*)  ada beberapa aturan yang harus dipenuhi untuk mendapat nilai delta tersebut yaitu :

$Δ\_{ij}^{(t)}=\left\{\begin{array}{c}η^{+} \*Δ\_{ij}^{\left(t-1\right)} ,if \frac{∂E^{\left(t-1\right)}}{∂ωi}\*\frac{∂E^{\left(t\right)}}{∂ωi}>0\\η^{+} \*Δ\_{ij}^{\left(t-1\right)} ,if \frac{∂E^{\left(t-1\right)}}{∂ωi}\*\frac{∂E^{\left(t\right)}}{∂ωi}<0\\\end{array}\right.$

 $ $

$ Δ\_{ij}^{(t)}$*, else* (10)

Untuk mendapatkan Δ*i*(*t*) untuk pertama kali pembelajaran Δ*i*(*t*-1) =Δ0 . Nilai Δ0 = 0.1 nilai pada delta 0 dapat saja ditetapkan lebih besar atau lebih kecil dari 0.1, karena nilai delta 0 tidak memiliki pengaruh besar terhadap laju proses pembelajarannya. Untuk pembelajaran berikutnya maka Δ*i* yang terdahulu akan dikalikan dengan *learningrate*. Pada perkalian Δ*i*(*t*-1) dengan *learningrate* ada aturan yang harus dicapai yaitu error gradien pada hasil terdahulu $\frac{∂E^{(t-1)}}{∂ωi}$ dikalikan dengan errorgradien $\frac{∂E^{(t)}}{∂ωi}$. Dari hasil perkalian error tersebut maka didapatkan hasil jika hasil lebih besar dari 0 maka Δ*i*(*t*-1) dikalikan dengan η+, dan jika hasil lebih kecil dari 0 maka Δ*i*(*t*-1) dikalikan dengan η-. Nilai η+ dan η- merupakan learningrate yang membedakan adalah pada η+ memiliki nilai yang lebih besar daripada . Onggrono, et al. (2017) menetapkan nilai standar pada η+ = 1.2 sedangkan η- = 0.5 untuk algoritma resilient propagation.

1. Menentukan Nilai Δw*i*(*t*)

Jika Δ*i*(*t*) sudah ditentukan maka ankan dilanjutkan kedalam aturan untuk menentukan fungsi operator yang akan dipakai untuk melakukan *updateweight* dengan aturan sebagai berikut :

$Δw\_{i}^{(t)}=\left\{\begin{array}{c}-Δi^{(t)}, if \frac{∂E^{(t-1)}}{∂ωi}>0\\+Δi^{\left(t\right)}, \&if \frac{∂E^{\left(t-1\right)}}{∂ωi}<0\end{array}\right.$ (11)

$$w\_{i}^{(t+1)}=w\_{i}^{(t)}+Δw\_{i}^{(t)}$$

Dengan menggunakan *error* function maka diketahui Δ*i*(*t*) yang akan digunakan apakah akan dikurangi atau ditambah atau tidak terjadi perubahan bobot. Untuk menentukan fungsi operator yang dipakai pada Δ*i*(*t*) maka dibandingkan nilai *error* gradien terhadap nol. Jika nilai error gradient lebih besar daripada nol maka Δw*i*(*t*) akan menerimanilai -Δ*i*(*t*), dan jika nilai *error* gradien lebih kecil daripada 0 maka Δw*i*(*t*) akan menerima +Δ*i*(*t*). Setelah mendapatkan nilai delta w maka dilanjutkan ke *updateweight*nya secara langsung.

Pada aturan pertama untuk penentuan *learningrate* memiliki masalah pada aturan kedua yaitu terjadi peningkatan pembelajaran yang melebih batas minimum maka dilakukan pengurangan *weight* secara langsung dengan *weight* terdahulunya.

$$Δw\_{i}^{(t)}=-Δw\_{i}^{\left(t-1\right)},if \frac{∂E^{\left(t-1\right)}}{∂ωi}\*\frac{∂E^{\left(t\right)}}{∂ωi}<0$$

 Untuk menghindari terjadi keadaan tersebut untuk terjadi kedua kalinya maka pada$ \frac{∂E^{(t)}}{∂ωi}$ = 0.

# Implementasi Dan Hasil

Penelitian ini terfokus pada hasil akurasi dan jumlah epoch yang dihasilkan pada proses training dan testing yang disimulasikan dan dianalisis menggunakan 2 algoritma, yaitu algoritma backpropagation standar dan Resilient backpropagation. Untuk parameter pada jaringan backpropagation yang akan digunakan pada penelitian ini dijelaskan pada Tabel 1[10]:

**Tabel 1. Parameter *Backpropagation***

|  |  |
| --- | --- |
| **Karakteristik** | **Spesifikasi** |
| Fungsi Aktivasi | Sigmoid biner |
| Inisialisasi bobot | Random |
| Target *Error* | 0.005 |
| Maksimum *Epoch* | 100000 |
| *Learning Rate* | 0.1 |

Arsitektur jaringan backpropagation yang digunakan sebanyak 5 arsitektur. Setiap arsitektur mempunyai jumlah *hidden layer* yang berbeda, jumlah neuron setiap layer tersembunyi berbeda-beda. Untuk semua arsitektur jumlah neuron input pada *Input Layer* adalah 3 dan jumlah neuron output pada *Output Layer* adalah 1. Jumlah neuron pada *hidden layer* ditentukan secara random. Daftar arsitektur jaringan backpropagation dapat dilihat pada Tabel 2.

**Tabel 2. Arsitektur *Backpropagation***

|  |  |  |
| --- | --- | --- |
| **No** | **Arsitektur Jaringan** | **Keterangan** |
| 1. | 3-1-1 | Jumlah neuron input 3, Jumlah layer tersembunyi 1 neuron dan 1 neuron output |
| 2. | 3-2-1 | Jumlah neuron input 3, Jumlah layer tersembunyi 2 neuron dan 1 neuron output |
| 3. | 3-3-1 | Jumlah neuron input 3, Jumlah layer tersembunyi 3 neuron dan 1 neuron output |
| 4. | 3-4-1 | Jumlah neuron input 3, Jumlah layer tersembunyi 4 neuron dan 1 neuron output |
| 5. | 3-5-1 | Jumlah neuron input 3, Jumlah layer tersembunyi 5 neurondan 1 neuron output |

Pada table 1. dan 2. menunjukkan dalam melakukan sebuah pengujian implementasi sistem simulasi hal yang perlu diperhatikan yaitu menentukan parameter-parameter jaringan agar nantinya prediksi yang dilakukan sesuai dengan yang diharapkan. Dalam tahap ini proses pengujian dilakukan dengan 5 sampel arsitektur yang sama dilakukan pada backpropagation standar danResilientbackpropagation menggunakan aplikasi Matlab*.*

* 1. **Spesifikasi Implementasi Perangkat Keras Dan Lunak**

Spesifikasi implementasi perangkat keras dan lunak ini menjelaskan mengenai kebutuhan perangkat keras dan lunak yang dibutuhkan dalam implementasi sistem. Adapun spesifikasi implementasi perangkat keras yang digunakan dapat dilihat pada tabel 3. dibawah ini:

**Tabel 3. Spesifikasi Implementasi Perangkat Keras**

|  |  |  |
| --- | --- | --- |
| **No.** | **Perangkat Keras** | **Spesifikasi** |
| 1. | Processor | Intel Core i5-7200U CPU 2.50 GHz |
| 2. | Memori | 6064 MB RAM DDR 3 |
| 3. | Harddisk SSD | 250 GB |
| 4. | Sistem Operasi | Windows 10 Pro 64-Bit |
| 5. | Program | Microsoft Visio 2007, Matlab R2011b |

* 1. **Persiapan Data**

Hal yang pertama dilakukan sebelum melakukan proses pengujian dengan menggunakan aplikasi Matlab R2011b*.* dimana dalam hal ini mempersiapkan data yang telah dinormalisasi berdasarkan data *training* dan data *testing*. Pada proses pelatihan data dibagi menjadi dua yaitu data *training* yang berjumlah 91 dan data *testing* berjumlah 91. Adapun data training dan data *testing* dapat dilihat pada table 4. dibawah ini :

**Tabel 4. Data Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **POLA** | **X1** | **X2** | **X3** | **TARGET** |
| 1 | 0.7667 | 0.5308 | 0.7286 | 1 |
| 2 | 0.6333 | 0.5000 | 0.7571 | 1 |
| 3 | 0.6333 | 0.8385 | 0.7857 | 1 |
| 4 | 0.7667 | 0.4692 | 0.9000 | 1 |
| 5 | 0.5667 | 0.4692 | 0.7571 | 1 |
| … | … | … | … | … |
| 90 | 0.1000 | 0.1000 | 0.1000 | 0 |
| 91 | 0.7000 | 0.8385 | 0.9000 | 1 |

Sedangkan data training dapat dilihat pada table 5.:

**Tabel 5. Data Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **POLA** | **X1** | **X2** | **X3** | **TARGET** |
| 1 | 0.7667 | 0.5308 | 0.7571 | 1 |
| 2 | 0.1000 | 0.1308 | 0.1000 | 0 |
| 3 | 0.2333 | 0.1000 | 0.2714 | 0 |
| 4 | 0.6333 | 0.5923 | 0.7571 | 1 |
| 5 | 0.6333 | 0.5000 | 0.7857 | 1 |
| … | … | … | … | … |
| 90 | 0.7000 | 0.4692 | 0.7857 | 1 |
| 91 | 0.7667 | 0.5308 | 0.7286 | 1 |

Setelah dilakukan proses persiapan data maka selanjutnya dilakukan proses pengujian Backpropagation menggunakan GUI pada aplikasi Matlab R2011byang telah dirancang sebelumnya.

* 1. **Hasil Pelatihan Algoritma Backpropagation dan Resilient Backpropagation**

Pada pelatihan algoritma backpropagation dengan beberapa simulasi/percobaan arsitektur yang berbeda tetapi parameter backpropagation yang sama, proses pelatihan dan pengujian selesai pada iterasi yang berbeda-beda juga. Pada hasil pengujian dengan nilai TOEFL, jumlah epoch terkecil diperoleh pada arsitektur 3-5-1 yaitu 360 dengan MSE 0.00499 dan akurasi 98.90 %. Sedangkan jumlah epoch terbesar adalah 1.899 pada arsitektur 3-3-1. Akurasi tertinggi pada pengujian backpropagation adalah 100% dengan MSE 0.00490 dan 0.00500 sedangkan akurasi terendah adalah 98.90% dengan MSE 0.00499. Hasil rekapitulasi pengujian data nilai TOEFL dengan beberapa arsitektur pada *backpropagation* dapat dilihat pada Tabel 6.

**Tabel 6. Rekapitulasi Pengujian Backpropagation**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Arsitektur BP** | **Epoch** | **Training** |
| **MSE** | **Akurasi (%)** |
| 1 | 3-1-1 | 447 | 0.00498 | 100% |
| 2 | 3-2-1 | 549 | 0.00500 | 100% |
| 3 | 3-3-1 | 1899 | 0.00500 | 100% |
| 4 | 3-4-1 | 1259 | 0.00500 | 100% |
| 5 | 3-5-1 | 360 | 0.00499 | 98.90% |

Pada pelatihan dengan resilient backpropagation dengan beberapa arsitektur yang berbeda dengan parameter backpropagation yang sama, proses pelatihan dan pengujian selesai pada iterasi yang berbeda-beda juga. Parameter yang digunakan pada proses optimasi backpropagation dengan resilient backpropagation sama dengan parameter yang digunakan pada Backpropagation. Pada pengujian backpropagation dengan resilient backpropagation jumlah *epoch* pada jaringan backpropagation dengan resilient backpropagation mengalami penurunan yang sangat signifikan. Nilai epoch tertinggi adalah 38 dan nilai epoch terendah adalah 5. Sedangkan akurasi seluruh percobaan pada resilient backpropagation adalah 100%. Hasil pengujian data nilai TOEFL dengan beberapa arsitektur pada resilient backpropagation dapat dilihat pada Tabel 7.

**Tabel 7. Rekapitulasi Pengujian Resilient backpropagation**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Arsitektur TrainRp** | **Epoch** | **Training** |
| **MSE** | **Akurasi (%)** |
| 1 | 3-1-1 | 38 | 0.00421 | 100% |
| 2 | 3-2-1 | 38 | 0.00376 | 100% |
| 3 | 3-3-1 | 5 | 0.00342 | 100% |
| 4 | 3-4-1 | 8 | 0.00425 | 100% |
| 5 | 3-5-1 | 35 | 0.00498 | 100% |

Berdasarkan penelitian yang dilakukan pada algoritma Backpropagationstandar dengan arsitektur yang berbeda-beda, nilai *epoch* masih terlalu besar. Pada pengujian yang dilakukan terjadi penurunan yang signifikan pada nilai epoch. Penurunan nilai epoch terbesar terjadi pada arsitektur 3-3-1 dari 1.899 (Backpropagation Standar) menjadi 5 (Resilient backpropagation) dengan akurasi 100% (Backpropagation Standar) dan 100% (Resilient backpropagation), terjadi penurunan jumlah *epoch* sebesar 99.81%. Rata-rata penurunan jumlah epoch adalah sebesar 92.39%. Presentasi penurunan jumlah *epoch* dari Backpropagation Standar dan Resilient backpropagation dapat dilihat pada Tabel 8.

**Tabel 8. Perbandingan Jumlah Epoch pada Backpropagation dan Resilient backpropagation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Arsitektur Jaringan** | **Epoch Backpropagation** | **Epoch Resilient Backpropagation** | **Penurunan Jumlah Epoch** |
| 1 | 3-1-1 | 447 | 38 | 409 | 91.50% |
| 2 | 3-2-1 | 549 | 38 | 511 | 93.08% |
| 3 | 3-3-1 | 1899 | 5 | 1894 | 99.74% |
| 4 | 3-4-1 | 1259 | 8 | 1251 | 99.36% |
| 5 | 3-5-1 | 360 | 35 | 325 | 90.28% |
| Rata – Rata Penurunan Jumlah Epoch | 94.79% |

Hasil akurasi dari ke 5 arsitektur yang diuji, tidak terjadi terjadi perubahan akurasi pada 4 arsitektur sedangkan pada 1 arsitektur nilai akurasi mengalami perubahan. Arsitektur perubahan yang dimaksud adalah mengalami peningkatan akurasi. Peningkatan akurasi yang dimaksud adalah membandingkan akurasi yang dihasilkan oleh backpropagation dengan akurasi yang dihasilkan resilient backpropagation. Arsitektur jaringan yang tidak kenaikan terjadi pada arsitektur 3-1-1, 3-2-1, 3-3-1, 3-4-1, dengan akurasi 100 %. Sedangkan arsitektur yang mengalami peningkatan akurasi adalah arsitektur 3-5-1. Perbandingan nilai akurasi pada backpropagation dengan resilient backpropagation dapat dilihat pada tabel 9.:

**Tabel 9. Perbandingan Akurasi pada Backpropagation dan Resilient backpropagation**

| **No** | **Arsitektur Jaringan** | **Hasil Akurasi Backpropagation** | **Hasil Akurasi Resilient backpropagation** | **Peningkatan Akurasi** |
| --- | --- | --- | --- | --- |
| 1 | 3-1-1 | 100% | 100% | 0.00% |
| 2 | 3-2-1 | 100% | 100% | 0.00% |
| 3 | 3-3-1 | 100% | 100% | 0.00% |
| 4 | 3-4-1 | 100% | 100% | 0.00% |
| 5 | 3-5-1 | 98.90% | 100% | 1.10% |

Berdasarkan perbandingan hasil penurunan epoch dan kenaikan akurasi pada tabel 3.8 dan 9 dapat disimpulkan arsitektur terbaik serta yang menjadi rekomendasi adalah arsitektur 3-3-1 karena memiliki nilai epoch 5, nilai error (MSE) terkecil 0.00342 dan akurasi 100%.

**Tabel 10. Rekapitulasi Perbandingan Hasil Prediksi Bakpropagation dan Resilient Bakpropagation Terhadap Target**

|  |  |
| --- | --- |
| Algoritma | Arsitektur |
| 3-1-1 | 3-2-1 | 3-3-1 | 3-4-1 | 3-5-1 |
| Output yang Sesuai dengan Target menggunakan Algotirma Backpropagation | 182 | 182 | 182 | 182 | 181 |
| Output yang Sesuai dengan Target menggunakan Algotirma Resilient Backpropagation | 182 | 182 | 182 | 182 | 182 |

Berdasarkan tabel 10. dapat disimpulkan bahwasannya menggunakan algoritma resilient backpropagation dengan arsitektur 3-1-1, 3-2-1, 3-3-1, 3-4-1, 3-5-1 mendapatkan 182 output yang sesuai dengan target presiksi 182 (100%).

# Penutup

Hasil penelitian yang telah dilakukan dalam memprediksi kelulusan TOEFL menunjukkan bahwa perubahan parameter learning rate serta epoch mempengaruhi tingkat akurasi, sehinga pemilihan parameter yang tepat sangat berpengaruh terhadap hasil. Dari 5 percobaan hasil yang paling signifikan ditunjukkan pada percobaan dengan arsitektur 3-3-1 dengan perbandingan algoritma backpropagation

bernilai epoch 1.899 dengan akurasi 100% sedangkan algoritma resilient backpropagation nilai opoch menjadi 5 akurasi tetap 100%. Dengan melihat hasil pengujian tersebut dapat diambil sebuah kesimpulan yaitu pengujian yang dihasilkan algoritma backpropagation mengalami peningkatan setelah dioptimasi dengan algoritma resilient backpropagation. Metode backpropagation dan resilient backpropagationdapat diterapkan dalam memprediksi kelulusan TOEFL peserta didik Michigan Computer English Course.

# Daftar Pustaka

[1] M. R. Lubis, “Analisis jaringan saraf tiruan back propgation untuk peningkatan akurasi prediksi hasil pertandingan sepakbola,” *Jurnal Teknik Informatika*, vol. 10, no. 1, pp. 51–62, 2018.

[2] D. H. Widodo Saputra, Tulus, Muhammad Zarlis, RAhmat Widia Sembiring, “Analysis Resilient Algorithm on Artificial Neural Network Backpropagation,” *IOP Conf. Series: Journal of Physics: Conf. Series 930 (2017) 012035*, 2017.

[3] M. R. Lubis and I. Parlina, “Analisis Algoritma Backpropagation Dalam Memprediksi Tingkat Keberhasilan Pelatih Sepakbola Sebagai Salah Satu Kemajuan Olahraga Di Indonesia,” *Klik - Kumpulan Jurnal Ilmu Komputer*, vol. 06, no. 03, pp. 264–274, 2019.

[4] M. R. Lubis, “Model Jaringan Saraf Tiruan Backpropagation Untuk Meningkatkan Penguasaan Mahasiswa Pada Matakuliah Algoritma Dan Pemrograman,” *Jurnal Informatika dan Komputer*, vol. XXI, no. 1, pp. 91–94, 2019.

[5] M. R. Lubis, W. Saputra, A. Wanto, S. R. Andani, and P. Poningsih, “Analysis of Artificial Neural Networks Method Backpropagation to Improve the Understanding Student in Algorithm and Programming Analysis of Artificial Neural Networks Method Backpropagation to Improve the Understanding Student in Algorithm and Programming,” 2019.

[6] Agus Perdana Windarto, “Implementasi JST Dalam Menentukan Kelayakan Nasabah Pinjaman Kur Pada Bank Mandiri Mikro Serbelawan Dengan Metode Backpropogation,” *Jurnal Sains Komputer & Informatika (J-SAKTI)*, vol. 1, no. 1, pp. 12–23, 2017.

[7] A. P. Windarto, M. R. Lubis, and S. Solikhun, “Model Arsitektur Neural Network Dengan Backpropogation Pada Prediksi Total Laba Rugi Komprehensif Bank Umum Konvensional,” *Klik - Kumpulan Jurnal Ilmu Komputer*, vol. 5, no. 2, p. 147, 2018.

[8] A. P. Windarto, M. R. Lubis, and S. Solikhun, “Implementasi JST pada Prediksi Total Laba Rugi Komprehensif Bank Umum dan Konvensional dengan Backpropagation,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 4, p. 411, 2018.

[9] M. R. Lubis, “Metode Hybrid Particle Swarm Optimization - Neural Network Backpropagation Untuk Prediksi Hasil Pertandingan Sepak Bola,” *Jurnal Sains Komputer & Informatika (J-SAKTI)*, vol. 1, no. 1, pp. 71–83, 2017.

[10] K. Onggrono, Tulus, and E. B. Nababan, “Analisis Penggunaan Parallel Processing Multithreading Pada Resilient Backpropagation,” *Jurnal Nasional Informatika dan Teknologi Jaringan*, vol. 2, no. 1, pp. 33–40, 2017.