

NASKAH PUBLIKASI
IMPLEMENTASI BACKPROPAGATION UNTUK MENDIAGNOSA
PENYAKIT DIABETES

Program Studi Teknik Informatika



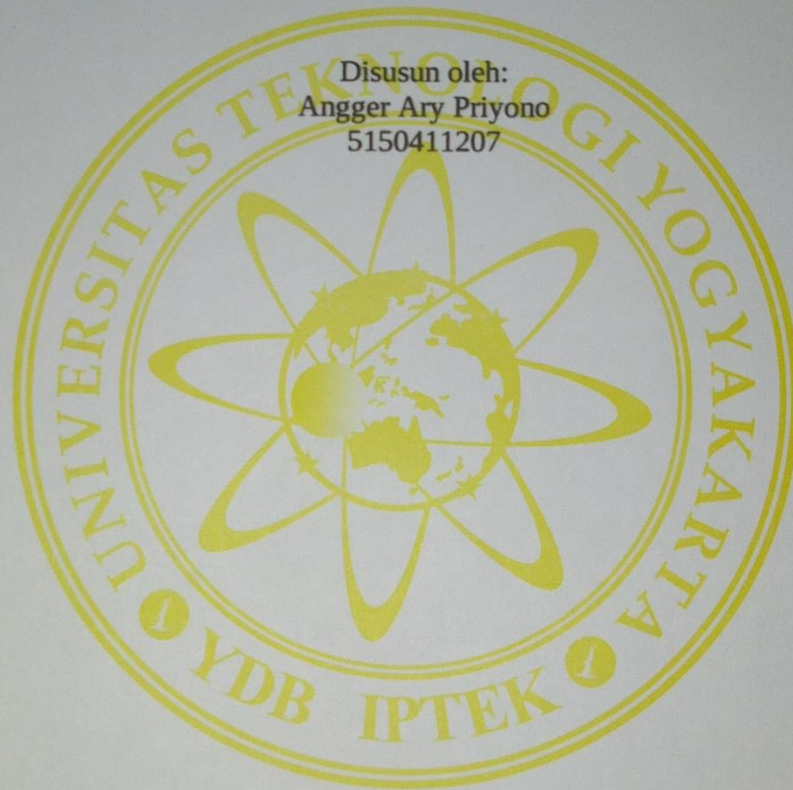
Disusun oleh:
Angger Ary Priyono
5150411207

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN ELEKTRO
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2019

NASKAH PUBLIKASI

IMPLEMENTASI BACKPROPAGATION UNTUK MENDIAGNOSA
PENYAKIT DIABETES

Disusun oleh:
Angger Ary Priyono
5150411207



Pembimbing,

A handwritten signature in black ink, appearing to read 'Dodi Hariadi', is written over a horizontal line.

Dodi Hariadi, ST., M.Eng.

Tanggal: 21/8/2019.....

Implementasi Backpropagation untuk Mendiagnosa Penyakit Diabetes

ANGGER ARY PRIYONO

Program Studi Teknik Informatika, Fakultas Teknologi Informasi dan Elektro

Universitas Teknologi Yogyakarta

Jl. Ringroad Utara Jombor Sleman Yogyakarta

E-mail : anggeraryp@gmail.com

ABSTRAK

Beberapa penyakit akan berdampak saat nanti usia mulai bertambah, salah satunya adalah penyakit diabetes. Untuk mengetahui potensi diabetes dilakukan pemeriksaan dengan beberapa identifikasi dan nantinya akan disimpulkan hasil dari pemeriksaan yang dilakukan. Diagnosa yang disimpulkan dari hasil pemeriksaan dapat dilakuakn oleh sistem dengan melakukan pembelajaran terhadap sistem atau machine learning dengan data yang telah ada. Pembelajaran yang dilakukan bertujuan untuk membuat sistem mampu mengenali pola masukan dan keluaran berdasarkan riwayat diagnosa yang telah ada. Penggunaan sistem dapat memakai banyak metode salah satunya backpropagation. Backpropagation adalah salah satu metode yang ada pada jaringan saraf tiruan yang digunakan untuk melakukan prediksi pada masalah-masalah yang rumit. Implementasi backpropagation telah dilakuakn untuk melakuakn prediksi dengan beberapa pengujian terhadap learning rate, hidden layer, epoch, MSE dan percobaan untuk menari bobot awal yang baik dengan hasil akurasi yang diperoleh sebesar 80% - 90%.

Kata kunci : Prediksi, Machine Learning, Jaringan Saraf Tiruan, Backpropagation

1. PENDAHULUAN

Kesehatan merupakan hal penting dan mahal harganya. Menjaga kesehatan harus dilakukan sejak sekarang karena tidak semua penyakit langsung dapat dirasakan dampaknya. Beberapa penyakit akan berdampak saat nanti usia mulai bertambah, salah satunya adalah penyakit diabetes. Diabetes adalah penyakit tertua di dunia. Diabetes berhubungan dengan metabolisme kadar glukosa dalam darah. Secara medis, pengertian diabetes melitus meluas pada suatu kumpulan aspek gejala yang timbul pada seseorang yang disebabkan oleh adanya peningkatan kadar gula darah (hiperglikemia) akibat kekurangan insulin.

Untuk mengetahui potensi diabetes dilakukan pemeriksaan dengan beberapa identifikasi dan nantinya akan disimpulkan hasil dari pemeriksaan yang dilakukan. Menyimpulkan hasil dari pemeriksaan memerlukan waktu, semakin banyak pasien yang melakukan pemeriksaan semakin banyak waktu yang dibutuhkan untuk mendapatkan hasil dari data pemeriksaan.

Sistem prediksi, peramalan atau diagnosa mulai digunakan untuk mendiagnosa hasil dari pemeriksaan yang dilakukan untuk mendapatkan hasil yang tepat dan cepat. Penggunaan sistem dilakukan untuk

menghemat waktu, biaya dan dapat mengatasi banyak data pasien pemeriksaan sekaligus dengan hasil yang konsisten. Pembuatan sistem dilakukan dengan menggunakan berbagai metode untuk mendapatkan hasil yang lebih akurat dan konsisten.

Salah satu metode yang dapat digunakan adalah jaringan saraf tiruan yang merupakan suatu model komputasi dengan mengadopsi sistem pembelajaran pada otak manusia. Jaringan saraf tiruan banyak diimplementasikan secara intensif pada sistem peramalan, prediksi atau diagnosa karena kelebihanannya pada kontrol area, prediksi dan pengenalan pola. Perancangan sistem dengan jaringan saraf tiruan diharapkan dapat memngoptimalkan hasil yang didapat.

Pada penelitian menggunakan dataset yang didapat dari Kaggle. Dataset yang ada berupa data dianosa wanita yang berjumlah 768 dan sudah terklasifikasi. Output yang dirancang dari sistem nantinya hanya akan menampilkan apah terkena diabetes atau tidak.

Penelitian ini bertujuan untuk mengimplementasikan backpropagation sebagai metode untuk mendiagnosa penyakit diabetes dan nantinya akan dilakuakn anlisa terhadap akurasi yang didapatkan.

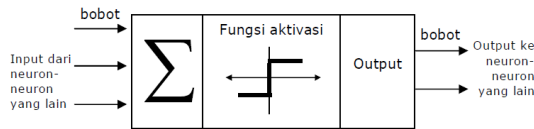
2. LANDASAN TEORI.

2.1. Dataset

Dataset adalah kumpulan data. Paling umum satu set data sesuai dengan isi tabel database tunggal, atau matriks data statistik tunggal, di mana setiap kolom tabel mewakili variabel tertentu, dan setiap baris sesuai dengan anggota tertentu dari set data yang dipertanyakan [1].

2.2. Jaringan Saraf Tiruan

Jaringan syaraf tiruan atau neural networks merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Neural networks berusaha meniru struktur atau arsitektur dan cara kerja otak manusia sehingga mampu menggantikan beberapa pekerjaan manusia. Pekerjaan seperti mengenali pola (pattern recognition), prediksi, klasifikasi, pendekatan fungsi adalah pekerjaan-pekerjaan yang diharapkan dapat diselesaikan dengan menggunakan neural networks [2].



Gambar 2. 1 Struktur Jaringan Saraf Tiruan

Jaringan syaraf terdiri dari beberapa *neuron* dan terdapat hubungan antara *neuron-neuron* tersebut (bobot). Informasi (*input*) akan dikirim ke *neuron* dengan bobot kedatangan tertentu. *Input* ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktvasi setiap neuron. Kemudian akan mengirimkan output melalui bobot-bobot *outputnya* ke semua *neuron* yang berhubungan dengannya. Jaringan syaraf tiruan memiliki 3 lapisan, yaitu lapisan *input*, lapisan tersembunyi dan lapisan *output*. Beberapa arsitektur jaringan syaraf yaitu lapisan tunggal, lapisan banyak dan lapisan kompetitif. Jaringan syaraf tiruan menggunakan fungsi aktivasi yang dipakai untuk membatasi keluaran dari *neuron* agar sesuai dengan batasan sinyal/nilai keluarannya. Beberapa fungsi aktivasi yang sering digunakan dalam jaringan syaraf tiruan antara lain:

- a. Fungsi Undak Biner (*Hard Limit*)

$$y = \begin{cases} 0; & \text{jika } x \leq 0 \\ 1; & \text{jika } x > 0 \end{cases} \quad (2.1)$$

- b. Fungsi Undak Biner (*Threshold*)

$$y = \begin{cases} 0; & \text{jika } x < 0 \\ 1; & \text{jika } x \geq 0 \end{cases} \quad (2.2)$$

- c. Fungsi *Bipolar* (*Symmetric Hard Limit*)

$$y = \begin{cases} 1; & \text{jika } x > 0 \\ 0; & \text{jika } x = 0 \\ -1; & \text{jika } x < 0 \end{cases} \quad (2.3)$$

- d. Fungsi *Bipolar* dengan *Threshold*

$$y = \begin{cases} 1; & \text{jika } x \geq 0 \\ -1; & \text{jika } x < 0 \end{cases} \quad (2.4)$$

- e. Fungsi *Linear* (identitas)

$$y = x \quad (2.5)$$

- f. Fungsi *Saturating Linear*

$$y = \begin{cases} 1; & \text{jika } x \geq 0,5 \\ x + 0,5; & \text{jika } -0,5 \leq x \leq 0,5 \\ 0; & \text{jika } x \leq -0,5 \end{cases} \quad (2.6)$$

- g. Fungsi *Symmetric Saturating Linear*

$$f(x) = \begin{cases} 1; & \text{jika } x \geq 1 \\ x; & \text{jika } -1 \leq x \leq 1 \\ -1; & \text{jika } x \leq -1 \end{cases} \quad (2.7)$$

- h. Fungsi *Sigmoid Biner*

$$y = f(x) = \frac{1}{1 + e^{-\delta x}} \quad (2.8)$$

Ditinjau dari jumlah *layer*, *Neural Network* bisa dibagi menjadi dua macam, yaitu *Neural Network layer* tunggal dan *Neural Network layer* jamak. *Neural Network layer* tunggal mempunyai satu lapis *neuron* pemroses. Satu lapis tersebut bisa berisi banyak *neuron*. Contoh algoritma *Neural Network layer* tunggal adalah *Perceptron*, *Delta* dan sebagainya. Sementara *Neural Network layer* jamak mempunyai sejumlah *neuron* perantara yang menghubungkan vektor masukan dengan *layer* jamak. *Layer* perantara ini disebut *layer* tersembunyi (*hidden layer*). Contoh algoritma *Neural Network layer* jamak adalah *Backpropagation*, *Constructive Backpropagation*, *Recurrent Neural Network* dan sebagainya.

2.2. Algoritma Backpropagation

Backpropagation merupakan algoritma pembelajaran yang terawasi dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyi. Algoritma backpropagation menggunakan error output untuk mengubah nilai bobot-bobotnya dalam arah mundur

(backward) [3]. Untuk mendapatkan error ini, tahap perambatan maju (forward propagation) harus dikerjakan terlebih dahulu. Pada proses pelatihan backpropagation ada beberapa parameter yang sangat berpengaruh, di antaranya adalah sebagai berikut:

1. Penentuan bobot awal. Bobot awal umumnya diambil secara acak dengan angka dalam jangkauan $[-1,+1]$ atau $[-0.5,+0.5]$ atau ditentukan dengan jangkauan $(-2.4/F_i, +2.4/F_i)$ dengan F_i adalah jumlah neuron masukan dalam neural network. Bahkan, ada juga algoritma Nguyen-Widrow (1990) untuk memberikan optimalisasi komposisi nilai bobot terbaik.
2. Laju pembelajaran (learning rate). Nilai parameter ini ada dalam jangkauan 0 sampai 1. Semakin besar nilainya, semakin cepat selesai proses pelatihannya, namun semakin lebih mudah terjebak pada daerah local optima (hasil yang didapatkan dapat berbeda pada setiap percobaan yang dilakukan).
3. Jumlah iterasi (epoch). Jika kriteria error hanya menggunakan SSE atau MSE, terkadang data yang sangat tidak linear sulit untuk bisa mencapai kriterianya. Pilihan lain dalam kriteria biasanya menggunakan jumlah maksimal iterasi. Jika jumlah maksimal tercapai, meskipun target error belum tercapai, proses pelatihan tetap akan dihentikan.
4. Target error. Target error merupakan akumulasi selisih nilai antara nilai keluaran yang diharapkan dengan nilai keluaran yang didapatkan. Kriteria yang umum digunakan adalah Sum of Square Error (SSE) atau Mean of Square Error (MSE). Nilai yang umum digunakan adalah 0,001 atau 0,0001.
5. Jumlah neuron dalam layer tersembunyi (hidden layer). Belum ada cara yang pasti untuk menentukan jumlah neuron dalam layer tersembunyi. Untuk mendapatkan komposisi jumlah neuron dalam layer tersembunyi yang tepat biasanya digunakan cara coba-coba sehingga dari beberapa kali percobaan akan diambil arsitektur yang memberikan hasil prediksi terbaik.

3. METODOLOGI PENELITIAN

3.1 Pengumpulan Data

Pengumpulan data dilakukan dengan melakukan penggalian data dan informasi dari data set. Data set diabetes yang digunakan pada penelitian ini adalah data set bersifat klasifikasi (classification) yang diambil dari UCI Repository atau Kaggle. Data set diabetes ini diberikan oleh Vincent Sigillito dari The Johns Hopkins University dan diberikan untuk National Institute of Diabetes and Digestive and Kidney Diseases pada 9 Mei 1990. Data set ini terdiri dari populasi wanita yang berumur paling muda 21

tahun dan tinggal di daerah Phoenix, Arizona. Data set diambil menurut kriteria World Health Organization (WHO), dimana ada total 768 orang yang diambil menjadi data set pada penelitian ini yang nantinya akan dibagi menjadi data training dan data testing.

3.2 Cleansing Data

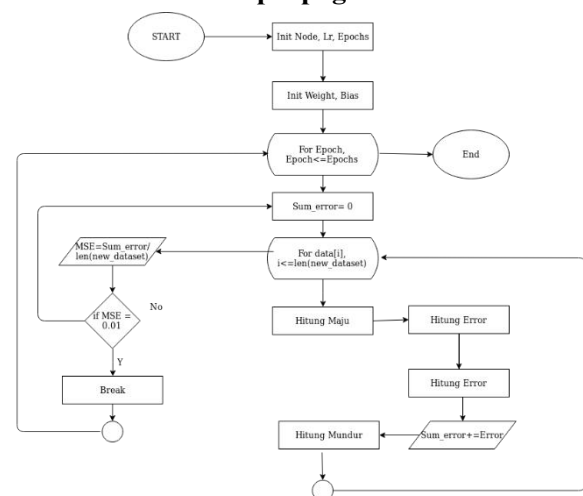
Proses dimana data set yang akan digunakan diolah dengan cara pembuangan baris data yang memiliki atribut null. Selanjutnya setelah dataset dibersihkan dilakukan normalisasi untuk mengurangi waktu proses dan menghasilkan output dengan rentang nilai 0 sampai dengan 1, dengan cara membagi dengan nilai minimal dan maksimal dalam dataset.

3.3 Perancangan Sistem

Perancangan sistem untuk membuat jaringan saraf tiruan dan mengimplmentasikan backpropagation membutuhkan beberapa langkah yaitu:

- Langkah 1: Membaca data set yang sudah di cleansing dan membagi sebanyak 75% sebagai data training dan 25% sebagai data testing.
- Langkah 2: Melakukan normalisasi pada data training dan data testing.
- Langkah 3: Membuat pola input dan target (output) berupa matrik berdasarkan data yang sudah dibaca.
- Langkah 4: Buat parameter jaringan saraf tiruan dengan menentukan selang tampilan maksimal, iterasi, MSE (Mean Square Error) dan konstanta belajar.
- Langkah 5: Membuat arsitektur jaringan saraf tiruan pada sistem dengan pembacaan parameter input, lapisan tersembunyi dan target.
- Langkah 6: Melatih jaringan dengan data training yang berbentuk matrik.
- Langkah 7: Menguji jaringan dengan data testing.

3.4 Flowchart Backpropagation



Untuk melatih jaringan pada backpropagation memerlukan inisialisasi terlebih dahulu agar jaringan terbentuk sehingga nantinya ajingan yang akan terbentuk akan dilatih hingga jaringan dapat mengenali pola dataset yang ada.

3.5 Pengujian

Pengujian sistem akan keseluruhan fungsi akan menggunakan blackbox testing dimana pengujian akan melakukan testing pada program sesuai dengan fungsi dan input yang disediakan. Pada pengujian juga dilakukan pengamatan sistem apakah hasil analisis telah sesuai yang diharapkan atau belum, serta dilakukan pengecekan kesalahan apa saja yang ada pada sistem.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Pengujian

Perancangan sistem yang dibuat dilakukan pengujian terhadap jaringan dengan berbagai kondisi untuk menemukan arsitektur jaringan yang paling sesuai agar mendapatkan hasil yang lebih optimal. Berikut beberapa singkatan yang akan digunakan pada tabel:

- uji = Pengujian
- lr = Learning Rate (Konstanta Belajar)
- hd = Hidden Layer
- acc = Akurasi Hasil Pengujian
- st_epc = Iterasi yang Ditentukan
- epc = Iterasi Berhenti
- st_sec = Waktu yang ditentukan (Detik)
- sec = Waktu Berhenti (Detik)
- st_MSE = MSE yang ditentukan
- MSE = MSE (toleransi error rata-rata) yang dicapai

4.1.1. Pengujian Learning Rate

Pengujian learning rate (Konstanta Belajar) dilakukan untuk mendapatkan nilai learning rate yang paling optimal dari berbagai pengujian yang dilakukan, untuk pengujian ini menggunakan nilai epochs = 50000, hidden layer = 16, MSE = 0.01 yang sama pada setiap pengujian seperti pada Tabel 4.1.

Tabel 4. 1 Pengujian Learning Rate

uji	lr	acc	Epc	sec	MSE
1	0.1	72.80%	50000	1096.25	0.01706
2	0.2	80.47%	50000	1013.45	0.02658
3	0.3	76.56%	50000	1002.09	0.03789
4	0.4	81.25%	50000	986.68	0.02275
5	0.5	81.25%	50000	1000.6	0.02653

6	0.6	77.35%	50000	983.43	0.01895
7	0.7	82.25%	50000	999.36	0.02652
8	0.8	76.56%	50000	960.99	0.04429

Kondisi training jaringan pada pengujian dikatakan berhenti apabila epochs (iterasi) mencapai 50000 atau MSE mencapai ≤ 0.01 . Pada Tabel 4.1 terlihat bahwa pengujian mendekati MSE yang ditentukan yaitu pada pengujian 1 dan pengujian 6 dengan nilai akurasi tertinggi pada pengujian 6.

4.1.2. Pengujian Hidden Layer

Setelah memilih nilai learning rate dari data pengujian yang dilakukan, maka selanjutnya kan dilakukan pengujian untuk mencari hidden layer yang optimal. Pengujian dilakukan menggunakan epochs = 50000, learning rate = 0.01, MSE ≤ 0.01 . Jika epoch atau MSE mencapai kondisi yang ditentukan maka pengujian akan selesai, sesuai dengan Tabel 4.2.

Tabel 4. 2 Pengujian Hidden Layer

uji	hd	acc	Epc	sec	MSE
1	4	77.34%	50000	897.75	0.08162
2	5	75.78%	50000	891.81	0.07871
3	6	76.56%	50000	920.48	0.04376
4	7	75.78%	50000	896.02	0.05562
5	8	78.90%	50000	935.54	0.06394
6	10	81.25%	50000	887.32	0.02839
7	12	78.12%	50000	927.23	0.02273
8	16	77.34%	50000	983.43	0.01895
9	20	74.22%	10613	226.24	0.01203
10	24	77.34%	50000	984.47	0.01894

Terlihat pada Tabel 4.2 bahwa pengujian yang mencapai akurasi tertinggi adalah pengujian 6 dengan akurasi 81.25% tetapi MSE yang dicapai belum memenuhi kondisi. MSE yang memenuhi kondisi adalah MSE pada pEngujian 8, 9, dan 10 dengan nilai MSE yang terkecil di pengujian 9 dan MSE pada pengujian 8, 10 adalah 0.018 dengan angka hanpir mendekati 0.0, maka hidden layer yang akan digunakan adalah hidden layer yang berasal dari pengujian 9.

4.1.3. Pengujian Bobot Awal

Selanjutnya adalah melakukan pencarian terhadap bobot awal yang nantinya akan digunakan sebagai bobot awal untuk melatih jaringan dengan learning rate = 0.6 dan hidden layer = 20 yang sudah ditetapkan berdasarkan percobaan yang telah dilakukan. Beberapa percobaan dilakukan untuk mencari bobot awal optimal seperti pada Tabel 4.3.

Tabel 4. 3 Pengujian Bobot Awal

uji	acc	epc	sec	MSE
-----	-----	-----	-----	-----

1	80.47 %	12000	261.48	0.022769
2	81.25 %	12000	257.22	0.045588
3	77.34 %	12000	263.41	0.083389
4	79.68 %	12000	266.26	0.075833
5	75.78 %	12000	243.04	0.056906
6	80.47 %	4740	86.99	0.013271

Pengujian untuk pencarian bobot pada Tabel 4.3 menggunakan kondisi training berhenti apabila epochs mencapai 12000 atau MSE mencapai 0.01. Hasil dari pengujian yang telah dilakukan menunjukkan pengujian 6 memiliki nilai terbaik dengan kondisi berhenti pada epochs 4740 dengan MSE 0.013271, maka bobot awal yang digunakan adalah bobot yang berasal dari pengujian 6.

4.1.4. Pengujian Epochs dan Penurunan MSE

Pengujian selanjutnya dilakukan berdasarkan percobaan sebelumnya dengan nilai learning rate = 0.6, hidden layer = 20, dan bobot awal yang tetap, maka dilakukan perlakuan berbeda terhadap jumlah iterasi (epochs):

Tabel 4. 4 Pengujian Epochs dan Penuruann MSE

uj i	st_ep c	st_MS E	acc	epc	sec	MSE
1	1100 0	0.01	80.47 %	4740	87.36	0.01327 1
2	1100 0	0.001	79.69 %	1100 0	197.8 6	0.01144 9
3	2000 0	0.001	78.90 %	2000 0	374.8 4	0.01138 6
4	5000 0	0.001	78.90 %	5000 0	926.8 8	0.01137 8

Tabel 4.4 menunjukkan bahwa pada pengujian 1 MSE mencapai kondisi 0.01 saat iterasi ke 4740 dengan MSE = 0.013271 dan akurasi sebanyak 80.47%. Pada pengujian selanjutnya di tambahkan tingkat MSE menjadi 0.001 untuk mencari kemampuan dari jaringan yang akan terbentuk dengan menggunakan MSE terkecil. Karena semakin kecil MSE jaringan akan dinilai semakin bagus dalam mengenali pola input pada hasil prediksi data [3].

4.2 Pembahasan

Didapatkan hasil dari pengujian yang dilakukan dengan nilai akurasi sebesar 80.47 % dengan tingkat MSE = 0.013271. Untuk memastikan ketepatan akurasi maka dilakukan lagi uji akurasi dengan bobot akhir yang disimpan dengan berbagai data acak yang diambil dari dataset. Pengujian 1 menggunakan 10 dataset secara acak seperti terlihat pada Tabel 4.5.

Dt = Data
Pg = Pregnancies
Gl = Glucose
Bp = Blood Pressure
St = Skin Thickness

Ins = Insulin
Dpf = Diabetes Pedegri Fuction
Ag = Age
Otc = Outcome
Jum_ack = Jumlah data acak yang digunakan

Tabel 4. 5 Sampel 10 Data Asli

D t	P g	Gl	B p	St	Ins	BM I	Dpf	A g	O t c
1	0	93	60	2 5	92	28	0.53 2	22	0
2	5	15 5	84	4 4	54 5	38.7	0.61 9	34	0
3	3	11 5	66	3 9	14 0	38	0.15	28	0
4	1	11 9	54	1 3	50	22.3	0.20 5	24	0
5	4	18 4	78	3 9	27 7	37	0.26 4	31	1
6	0	10 4	64	3 7	64	33	0.51	22	1
7	0	91	68	3 2	21 0	39.9	0.38 1	25	0
8	9	12 0	72	2 2	56	20.8	0.73 3	48	0
9	4	13 1	68	1 1	16 6	33.1	0.16	28	0
1 0	3	16 9	74	1 9	12 5	29.9	0.26 8	31	1

Dari Tabel 4.5 maka dilakukan preprocessing data dengan normalisasi sehingga data akan terlihat seperti Tabel 4.6. Setelah dilakukannya normalisasi data, selanjutnya data akan dihitung untuk menentukan akurasi yang didapatkan terlihat pada Tabel 4.7.

Tabel 4. 6 Sampel 10 Data Normalisasi

D t	Pg	Gl	Bp	St	Ins	BM I	Dpf	Ag	O t c
1	0	0.2 605 63	0.4 186 05	0.3 214 29	0.0 937 50	0.2 147 24	0.1 914 35	0.0 166 67	0
2	0.2	0.6 941 18	0.6 971 74	0.6 607 14	0.6 382 21	0.4 192 23	0.2 286 94	0.2 166 67	0
3	0.1	0.4 764 71	0.4 883 72	0.5 714 29	0.1 514 42	0.4 069 53	0.0 278 37	0.1 166 67	0
4	0.0	0.4 588 24	0.3 436 62	0.1 071 43.	0.0 432 69	0.0 838 45	0.0 513 92	0.0 500 00	0
5	0.2	0.9 352 94	0.6 014 08	0.5 714 29	0.3 161 06	0.3 844 58	0.0 766 60	0.1 666 67	1
6	0	0.3 380 28	0.4 651 16	0.5 357 14	0.0 600 96	0.3 149 28	0.1 820 13	0.0 166 67	1
7	0	0.2 464 79	0.5 116 28	0.4 464 29	0.2 355 77	0.4 437 63	0.1 267 67	0.0 666 67	0
8	0.5	0.4 294 12	0.5 507 04	0.2 678 54	0.0 504 81	0.0 531 70	0.2 775 16	0.4 500 00	0

9	0.2	0.5	0.5	0.2	0.1	0.3	0.0	0.1	0
	352	281	116	500	826	047	321	166	
	94	69	28	00	92	03	20	67	
1	0.1	0.7	0.5	0.2	0.1	0.2	0.0	0.1	1
0	764	957	813	142	334	392	783	666	
	71	75	95	86	13	634	373	67	

Tabel 4. 7 Pengujian Akurasi Pertama

Dt	Hasil	Pembulatan Hasil	Hasil Seharusnya	Keterangan
1	0.000051	0	0	Salah
2	0.999998	1	0	Benar
3	0.000004	0	0	Benar
4	0.000000	0	0	Benar
5	0.999998	1	1	Benar
6	0.980815	1	1	Benar
7	0.001622	0	0	Benar
8	0.000147	0	0	Benar
9	0.000002	0	0	Benar
10	0.998876	0	1	Benar
Akurasi			90.00 %	

Selanjutnya dilakukan berbagai pengujian dengan jumlah data berbeda dan data diambil secara acak dari dataset. Untuk cara pengujian yang dilakukan sama seperti pengujian sebelumnya yaitu pengujian 1 seperti terlihat pada Tabel 4.7.

Tabel 4. 8 Pengujian Akurasi

Pengujian	Jum ack	Akurasi
2	20	100.00 %
3	30	90.00 %
4	40	88.00 %
5	50	82.50 %
6	100	91.00 %
7	120	95.83 %
8	150	93.30 %
9	240	93.75 %
10	392	92.85 %

Pada Tabel 4.8 dilakukan pengujian yang bertujuan untuk mengetahui perubahan akurasi dengan data acak dari dataset berdasarkan jaringan yang sudah ditentukan. Berdasarkan pengujian yang telah dilakukan akurasi mencapai angka lebih tinggi dari 80.47 % menggunakan data secara acak, hal tersebut menunjukkan jaringan sudah dianggap mampu untuk mengenali pola input digunakan untuk memprediksi penyakit diabetes.

5. PENUTUP

5.1. Kesimpulan

Berdasarkan keseluruhan proses analisis, perancangan, dan implementasi, pada pembuatan sistem, diperoleh kesimpulan sebagai berikut:

a. Sistem yang dibangun berbasis web dengan menggunakan bobot yang sudah disimpan dan dilatih menggunakan backpropagation, proses pelatihan yang dilakukan berada diluar sistem dengan bantuan Jupiter Notebook untuk mencari bobot yang akan digunakan pada sistem.

b. Jaringan saraf tiruan yang dibuat dengan backpropagation mampu mengenali pola input yang ada pada dataset.

c. Hasil akurasi yang didapat pada dengan jaringan yang dibuat mencapai 80% hingga 90% dengan tingkat toleransi error sebanyak 0.013271.

5.2. Saran

Berdasarkan kesimpulan di atas, penulis mencantumkan beberapa saran, antara lain:

a. Untuk mempermudah penggunaan sistem sebaiknya pembangunan sistem dilakukan dengan membuat restful API.

b. Penambahan metode stochastic gradient descent diperlukan untuk melakuakn penurunan toleransi error (loss function)

c. Penggunaan Framework Machine Learning seperti Tensor Flow dan Theano diperlukan untuk mempercepat proses pelatihan.

d. Penggunaan dataset baru dapat memberikan hasil yang lebih baik dalam melakukan prediksi dengan penggolongan dewasa, remaja, dan anak-anak.

e. Penggunaan dataset hanya memberikan dua output yaitu 0 dan 1 (ya dan tidak), sebaiknya digunakan presentasi yang menunjukkan potensi besarnya diabetes.

DAFTAR PUSTAKA

- [1] Nelli Fabio, (2018), *Python Data Analytics With Pandas, NumPy, and Matplotlib*, Rome, Italy, Apress.
- [2] Suyanto (2018), *Machine Learning Tingkat Dasar dan Lanjut*, Informatika Bandung
- [3] Hermawan Arief (2006), *Jaringan Saraf Tiruan*, Yogyakarta: Andi.