

**AUTOENCODER UNTUK SISTEM PREDIKSI
BERAT LAHIR BAYI**



TESIS

Diajukan sebagai salah satu syarat untuk memperoleh gelar
Magister Ilmu Komputer (M.Kom)

FITRA SEPTIA NUGRAHA

14002225

Program Studi Ilmu Komputer (S2)
Sekolah Tinggi Manajemen Informatika Dan Komputer
Nusa Mandiri
2020

SURAT PERNYATAAN ORISINALITAS DAN BEBAS PLAGIARISME

Yang bertanda tangan di bawah ini :

Nama : Fitra Septia Nugraha
NIM : 14002225
Program Studi : Ilmu Komputer
Jenjang : Strata Dua (S2)
Konsentrasi : *Data Mining*

Dengan ini menyatakan bahwa tesis yang telah saya buat dengan judul: “Autoencoder Untuk Sistem Prediksi Berat Lahir Bayi” adalah hasil karya sendiri, dan semua sumber baik yang kutip maupun yang dirujuk telah saya nyatakan dengan benar dan tesis belum pernah diterbitkan atau dipublikasikan dimanapun dan dalam bentuk apapun.

Demikianlah surat pernyataan ini saya buat dengan sebenar-benarnya. Apabila dikemudian hari ternyata saya memberikan keterangan palsu dan atau ada pihak lain yang mengklaim bahwa tesis yang telah saya buat adalah hasil karya milik seseorang atau badan tertentu, saya bersedia diproses baik secara pidana maupun perdata dan kelulusan saya dari Program Studi Ilmu Komputer (S2) Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri dicabut/dibatalkan.

Jakarta, 05 Agustus 2020

Yang menyatakan



Fitra Septia Nugraha

HALAMAN PERSETUJUAN DAN PENGESAHAN TESIS

Tesis ini diajukan oleh:

Nama : Fitra Septia Nugraha
NIM : 14002225
Program Studi : Ilmu Komputer
Jenjang : Strata Dua (S2)
Konsentrasi : *Data Mining*
Judul Tesis : Autoencoder Untuk Sistem Prediksi Berat Lahir Bayi

Telah dipertahankan pada periode 2020-1 dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh Magister Ilmu Komputer (M.Kom) pada Program Studi Ilmu Komputer (S2) Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri (STMIK Nusa Mandiri).

Jakarta, 14 Agustus 2020

PEMBIMBING TESIS

Pembimbing I : Dr. Hilman Ferdinandus Pardede,
S.T, M.EICT



DEWAN PENGUJI

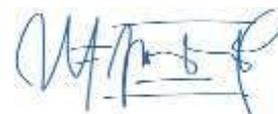
Penguji I : Dr. Agus Subekti, M.T



Penguji II : Dr. Didi Rosiyadi, M.Kom



Penguji III/
Pembimbing I : Dr. Hilman Ferdinandus Pardede,
S.T, M.EICT



	LEMBAR BIMBINGAN TESIS
	STMIK NUSA MANDIRI

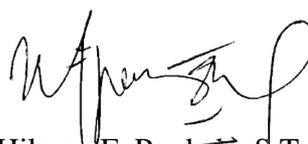
Nama : Fitra Septia Nugraha
 NIM : 14002225
 Dosen Pembimbing I : Dr. Hilman F. Pardede, S.T, M.EICT
 Judul Tesis : “Autoencoder Untuk Sistem Prediksi Berat Lahir Bayi”

No	Tanggal Bimbingan	Pokok Bahasan	Paraf Dosen Pembimbing
1	02 April 2020	Pengajuan Objek Penelitian	
2	04 April 2020	Pengajuan Metode Penelitian	
3	27 April 2020	Progress Eksperimen	
4	20 Mei 2020	Progress Eksperimen	
5	27 Juni 2020	BAB 3 dan BAB 4	
6	04 Juli 2020	Revisi BAB 3 dan BAB 4	
7	23 Juli 2020	BAB 1 dan BAB 2	
8	29 Juli 2020	All Laporan BAB 1 – BAB 5	

Catatan untuk dosen pembimbing Bimbingan Tesis

- Dimulai pada tanggal :
- Diakhiri pada tanggal :
- Jumlah pertemuan bimbingan : 8 (Delapan) kali bimbingan

Disetujui oleh,
Dosen Pembimbing



(Dr. Hilman F. Pardede, S.T, M.EICT)

KATA PENGANTAR

Alhamdulillah puji syukur kehadirat penulis panjatkan kehadirat Allah, SWT, yang telah melimpahkan rahmat dan karunia-Nya, sehingga pada akhirnya penulis dapat menyelesaikan laporan tesis ini tepat pada waktunya. Dimana laporan tesis ini penulis sajikan dalam bentuk buku yang sederhana.

Adapun judul tesis, yang penulis ambil sebagai berikut “Machine Learning Methods for Fetal Weight Forecasting with Artificial Neural Network”.

Tujuan penulisan laporan tesis ini dibuat sebagai salah satu untuk mendapatkan gelar Magister Ilmu Komputer (M.Kom) pada Program Studi Ilmu Komputer (S2) STMIK Nusa Mandiri.

Laporan Tesis ini diambil berdasarkan hasil penelitian atau riset mengenai berat bayi lahir. Penulis juga lakukan mencari dan menganalisa berbagai macam sumber referensi, baik dalam bentuk jurnal ilmiah, buku-buku literatur, internet, dll yang terkait dengan pembahasan pada laporan tesis ini.

Penulis menyadari bahwa tanpa bimbingan dan dukungan dari semua pihak dalam pembuatan laporan tesis ini, maka penulis tidak dapat menyelesaikan laporan tesis ini tepat pada waktunya. Untuk itu ijinkanlah penulis kesempatan ini untuk mengucapkan terimakasih kepada:

1. Ketua STMIK Nusa Mandiri Ibu Dr. Dwiza Riana, S.Si, MM, M.Kom
2. Wakil Ketua STMIK Nusa Mandiri Ibu Nita Merlina, M.Kom
3. Ketua Program Studi Ilmu Komputer Bapak Dr. Hilman F Pardede, ST, M.EICT bersama Sek.Prodi Ilmu Komputer Ibu Eni Heni Hermaliani, MM, M.Kom.
4. Bapak Dr. Hilman F Pardede, ST, M.EICT selaku pembimbing I yang selalu mengarahkan dan membimbing dalam melakukan penelitian dan menulis laporan tesis.
5. Bapak/Ibu dosen yang telah memberikan ilmu dan wawasan selama perkuliahan.
6. Seluruh staf, karyawan dan bagian administrasi yang telah memberikan informasi.

7. Orang tua tercinta serta seluruh keluarga yang telah memberikan dukungan material dan moral serta do'a yang tak pernah terhingga.

Serta semua pihak yang terlalu banyak untuk penulis sebutkan satu persatu sehingga terwujudnya penulisan laporan tesis ini. Penulis menyadari bahwa penulisan laporan tesis ini masih jauh sekali dari sempurna, untuk itu penulis mohon kritik dan saran yang bersifat membangun demi kesempurnaan penulisan karya ilmiah yang penulis hasilkan untuk yang akan datang.

Akhir kata semoga laporan tesis ini dapat bermanfaat bagi penulis khususnya dan bagi para pembaca yang berminat pada umumnya.

Jakarta, 05 Agustus 2020

Yang menyatakan

A handwritten signature in black ink, appearing to read 'Fitra Septia Nugraha', with a long horizontal stroke extending to the right. The signature is enclosed in a faint rectangular box.

Fitra Septia Nugraha

Penulis

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Nama : Fitra Septia Nugraha
NIM : 14002225
Program Studi : Ilmu Komputer
Jenjang : Strata Dua (S2)
Konsentrasi : *Data Mining*
Jenis Karya : Tesis

Demi pengembangan ilmu pengetahuan, dengan ini menyetujui untuk memberikan ijin kepada pihak Program Studi Ilmu Komputer (S2) Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri (STMIK Nusa Mandiri) Hak Bebas Royalti Non-Eksklusif (*Non-exclusive Royalti-Free Right*) atas karya ilmiah kami yang berjudul : “Autoencoder Untuk Sistem Prediksi Berat Lahir Bayi” beserta perangkat yang diperlukan (apabila ada).

Dengan Hak Bebas Royalti Non-Eksklusif ini pihak STMIK Nusa Mandiri berhak menyimpan, mengalih-media atau bentuk-kan, mengelolanya dalam pangkalan data (*database*), mendistribusikannya dan menampilkan atau mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari kami selama tetap mencantumkan nama kami sebagai penulis/pencipta karya ilmiah tersebut.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak STMIK Nusa Mandiri, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini .

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 05 Agustus 2020
Yang menyatakan,



Fitra Septia Nugraha

ABSTRAK

Nama : Fitra Septia Nugraha
NIM : 14002225
Program Studi : Ilmu Komputer
Jenjang : Strata Dua (S2)
Konsentrasi : *Data Mining*
Judul : “Autoencoder Untuk Sistem Prediksi Berat Lahir Bayi”

Salah satu ukuran terpenting saat awal persalinan adalah keakuratan prediksi berat lahir. Dengan menggunakan metode prediksi yang tepat, perkiraan ekstrim berat lahir bayi dapat dideteksi lebih atau kurang sehingga beberapa tindakan pencegahan dapat dilakukan sebelum persalinan. Di sisi lain, untuk meningkatkan akurasi suatu proses pembelajaran, dibutuhkan suatu prediksi yang akurat untuk masalah yang dihadapi atau dengan menggunakan gabungan beberapa metode. Penelitian bertujuan untuk prediksi berat lahir bayi menggunakan metode *Deep Learning autoencoder* untuk memprediksi berat lahir bayi. Salah satu tantangan dalam pengembangan sistem prediksi berat lahir bayi adalah datanya yang berdimensi tinggi. Teknik konvensional untuk *feature reduction* seperti *principal component analysis* (PCA), mengasumsikan *orthogonality* atau independensi antar komponen prinsipalnya. Dengan *autoencoder*, asumsi tersebut tidak ada. Sehingga *autoencoder* dapat memodelkan korelasi antar fitur. Dengan melakukan variasi parameter pada *autoencoder*, performa terbaik diperoleh adalah MSE 0.123, MAE 0.232, R^2 0.86 dengan *autoencoder* dengan 1 lapisan *hidden layer encoder* dan *decoder*. Ini lebih baik dibandingkan PCA.

Kata kunci: Algoritma *Deep Learning*, *Autoencoder*, Normalisasi, *Feature Reduction*, Berat lahir bayi

ABSTRACT

Nama : Fitra Septia Nugraha
NIM : 14002225
Program Studi : Ilmu Komputer
Jenjang : Strata Dua (S2)
Konsentrasi : *Data Mining*
Judul : *“Autoencoder For Infant Birth Weight Prediction System”*

One of the most important measurements at the onset of labor is the accuracy of the prediction of birth weight. By using precise prediction methods extreme estimates of baby birth weight can be detected more or less so that some precautions can be taken before delivery. On the other hand, to improve the accuracy of a learning process, an accurate prediction is needed for the problem at hand or by using a combination of several methods. This study aims to predict baby birth weight using the Deep Learning autoencoder method to predict baby birth weight. One of the challenges in developing a predictive system for infant birth weight is the high dimensional data. Conventional techniques for feature reduction, such as principal component analysis (PCA), assume orthogonality or independence between the principal components. With an autoencoder, that assumption doesn't exist. So that the autoencoder can model the correlation between features. By varying the parameters of the autoencoder, the best performance is MSE 0.123, MAE 0.232, R2 0.86 with an autoencoder with 1 hidden layer encoder and decoder layers. This is better than PCA.

Keywords: Deep Learning Algorithm, Autoencoder, Normalization, Infant birth weight

DAFTAR ISI

	Halaman
HALAMAN SAMPUL	i
HALAMAN JUDUL	ii
SURAT PERNYATAAN ORISINALITAS DAN BEBAS PLAGIARISM	iii
PERSETUJUAN DAN PENGESAHAN	iv
LEMBAR BIMBINGAN TESIS	v
KATA PENGANTAR	vi
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS	viii
ABSTRAK	ix
ABSTRACT	x
DAFTAR ISI	xi
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang Penelitian	1
1.2 Identifikasi Masalah.....	6
1.3 Tujuan Penelitian.....	6
1.4 Ruang Lingkup Penelitian.....	6
1.5 Hipotesis.....	7
BAB 2 LANDASAN/KERANGKA PEMIKIRAN	8
2.1 Tinjauan Pustaka	8
2.2 Tinjauan Studi	32
2.3 Tinjauan Organisasi/Obyek Penelitian	36
BAB 3 METODE PENELITIAN	37
BAB 4 HASIL PENELITIAN DAN PEMBAHASAN	49
BAB 5 PENUTUP	56
5.1 Kesimpulan	56
5.2 Saran	57
DAFTAR REFERENSI	56
DAFTAR RIWAYAT HIDUP.....	65

DAFTAR TABEL

	Halaman
Tabel 3.1 Penjelasan Atribut Pada Dataset Baby-data.....	38
Tabel 3.2 10 Record Dataset Baby	43
Tabel 3.3 Arsitektur Deep Autoencoder Yang Diujicobakan	47
Tabel 3.4 Arsitektur Deep Learning PCA Yang Diujicobakan	47
Tabel 3.5 Konfigurasi Parameter	47
Tabel 3.6 Parameter Yang Diujicobakan	48
Tabel 4.1 Evaluasi Hasil	53
Tabel 4.2 Perbandingan Performa Variasi Learning Rate	54

DAFTAR GAMBAR

	Halaman
Gambar 2.1 <i>Deep Neural Network</i>	16
Gambar 2.2 <i>Arsitektur Neural Network</i>	17
Gambar 2.3 <i>Model Single Perceptron</i>	18
Gambar 2.4 <i>Single Layer Neural Network</i>	19
Gambar 2.5 <i>Multi Layer Neural Network</i>	19
Gambar 2.6 Fungsi Aktivasi <i>Linear</i>	21
Gambar 2.7 Fungsi Aktivasi <i>Tanh</i> dan <i>Sigmoid</i>	22
Gambar 2.8 Fungsi Aktivasi <i>ReLU</i>	22
Gambar 2.9 Fungsi Aktivasi <i>GELU</i>	23
Gambar 2.10 <i>Arsitektur Jaringan Backpropagation</i>	24
Gambar 2.11 <i>Arsitektur Autoencoder</i>	31
Gambar 3.1 <i>10 Fold Cross Validation</i>	40
Gambar 3.2 <i>Metodologi Penelitian</i>	41
Gambar 3.3 <i>Tampilan Correlation Matrix</i>	43
Gambar 3.4 <i>Cek Missing Values Sesudah Replace</i>	44
Gambar 3.5 <i>Arsitektur Autoencoder</i>	46
Gambar 4.1 <i>Perbandingan Akurasi validasi Untuk Beberapa Lapisan Tersembunyi Normalisasi MinMaxScaler</i>	50
Gambar 4.2 <i>Perbandingan Akurasi validasi Untuk Beberapa Lapisan Tersembunyi Normalisasi StandarScaler</i>	51
Gambar 4.3 <i>Perbandingan Akurasi Validasi Autoencoder Metode Normalisasi</i>	52
Gambar 4.4 <i>Grafik Test Error Model</i>	55

BAB I PENDAHULUAN

1.1. Latar Belakang Penulisan

Berat Bayi Lahir Rendah (BBLR) adalah salah satu faktor utama yang berpengaruh terhadap kematian neonatal. *The Fifty Sixth Session Of Regional Commite WHO for South-East Asia* mengungkapkan, pada tahun 2015, kematian bayi terjadi pada usia neonatal dengan penyebab infeksi 33%, asfiksia atau trauma 28%, BBLR 24%, kelainan bawaan 10% dan lain-lain 5%. Resiko kematian BBLR 4 kali lebih besar dibandingkan bayi lahir dengan berat badan lebih dari 2500 gram Yulifah dan Yuswanto dalam[1]. Dan iliki risiko penyakit jantung dan diabetes di masa yang akan datang Trihardiani dalam[2]. Masalah yang sering terjadi pada BBLR adalah gangguan pada pernapasan, susunan saraf pusat, kardiovaskular, hematologi, gastro intestinal, ginjal dan termoregulasi [1][3]. BBLR dapat disebabkan oleh persalinan premature (persalinan pada usia kehamilan kurang dari 37 Minggu) ataupun karena *Intrauterin Growth Restriction* (IUG) [4].

Oleh karena itu, Ketepatan prediksi dari berat lahir bayi adalah salah satu pengukuran yang paling penting pada awal persalinan. Perawatan pasien sedini mungkin termasuk rencana perawatan, pencegahan komplikasi dan perawatan yang tepat. Dengan penggunaan metode prediksi yang akurat, bayi dengan perkiraan berat yang ekstrim lebih atau kurang dapat diketahui dan beberapa pencegahan dapat dilakukan untuk mengatasi masalah tersebut sebelum persalinan [5].

Penelitian prediksi berat lahir bayi ini sebelumnya sudah pernah dilakukan oleh para peneliti, H. Yasin and D. Ispriyanti tahun 2017 meneliti klasifikasi berat bayi lahir menggunakan *Weighted Probabilistic Neural Network* (WPNN) hasil penelitian menunjukkan bahwa pemodelan klasifikasi risiko BBLR dengan menggunakan model WPNN mempunyai tingkat akurasi yang sangat tinggi dengan akurasi sebesar 98,75% atribut yang digunakan adalah usia ibu, lama kehamilan, tinggi badan ibu, berat badan ibu, tensi darah ibu, *hemoglobin* ibu,

paritas dan berat bayi lahir sebagai label[1]. Penelitian lain A. Puspita and M. Wahyudi pada tahun 2015 meneliti klasifikasi kelahiran bayi premature menggunakan algoritma C4.5 dari hasil prediksi memberikan nilai akurasi sebesar 93.60% atribut yang digunakan dalam penelitian ini dengan melihat usia, sistol, diastol, riwayat darah tinggi, riwayat keguguran, riwayat premature, trauma, konsumsi rokok, konsumsi ganda, keputihan dan hasil[6]. Penelitian lain Jefa pada tahun 2019 meneliti prediksi klasifikasi bayi prematur menggunakan metode C4.5 berbasis *Particle Swarm Optimization* (PSO) terdapat beberapa faktor dan masalah kesehatan yang dapat memicu persalinan premature yaitu ibu yang tidak sehat, merokok, riwayat kehamilan, kondisi janin dan kondisi psikologis hasil penelitian algoritma C4.5 dengan menggunakan PSO diperoleh akurasi sebesar 97,91% [7]. Penelitian lain S. Kuhle *et al.* pada tahun 2018 meneliti perbandingan algoritma *Logistic Regression* dengan algoritma klasifikasi mesin learning lainnya untuk memprediksi kelainan pertumbuhan janin pada wanita *primipara* dan *multipara* dengan fitur-fitur yang digunakan dalam penelitian ini dilihat dari beberapa faktor yaitu faktor *socio demographics* yaitu usia ibu, status pernikahan, pendapatan dan tempat kemudian faktor resiko kehamilan yaitu merokok, BMI sebelum kehamilan, hipertensi, diabetes, penggunaan zat dan gangguan kejiwaan hasil penelitian menunjukkan bahwa metode pembelajaran mesin yang digunakan dalam penelitian ini tidak memberikan kelebihan dibandingkan dengan *Logistic Regression* dengan akurasi 60 – 75 % [3]. Selanjutnya penelitian Yueh-Chin Cheng, dkk pada tahun 2012 meneliti klasifikasi ukuran janin yang efisien dikombinasikan dengan jaringan saraf tiruan untuk estimasi berat janin, memanfaatkan algoritma K-means untuk mengklasifikasi ukuran janin yang selanjutnya digunakan untuk pelatihan JST untuk meningkatkan akurasi prediksi berat janin, fitur-fitur yang dipakai dalam penelitian ini diambil dari data pemeriksaan USG prenatal yaitu lingkaran perut (AC), *diameter biparietal* (BPD), berat lahir (BW), panjang femur (FL), presentasi janin (FP), usia kehamilan (GA), lingkaran kepala (HC), *diameter oksipitofrontal* OFD) dan jenis kelamin hasil penelitian diperoleh MAPE 5,26 4,14 % dan MAE 157,91 119,90 g[8].

Salah satu tantangan dalam penerapan sistem prediksi berat lahir adalah besarnya dimensi data. Teknik konvensional untuk *feature reduction* seperti

principal component analysis (PCA)[9]. PCA cukup unggul sebagai metode reduksi dimensi karena sederhana untuk diimplementasikan dan mampu mengeliminasi korelasi antar variabel masukan. Akan tetapi, PCA hanya mereduksi data secara linear, tidak memperhatikan hubungan antar data secara spasial, sedangkan kita ketahui sering terdapat hubungan nonlinear di dalam data.

Artificial neural network (ANN) adalah salah satu metode machine learning yang mempunyai struktur tersebar paralel yang sangat besar dan mempunyai kemampuan belajar, sehingga bisa melakukan generalisasi, yaitu bisa menghasilkan *output* yang benar untuk *input* yang belum pernah dilatihkan. Dengan kedua kemampuan pemrosesan informasi ini, ANN dapat menyelesaikan masalah-masalah yang sangat kompleks, mulai dari klasifikasi, optimasi, kompresi, peramalan (*forecasting*), sistem kontrol, sistem pendeteksian kecurangan (*intrusion detection system*) dan sebagainya. Pada umumnya, ANN sangat sesuai untuk permasalahan yang bernilai kontinyu seperti pengenalan tulisan tangan, pengenalan wajah, peramalan kurs tengah mata uang dan sebagainya[10]. ANN juga dapat melakukan pembelajaran data tidak hanya untuk data linear saja tetapi dapat digunakan untuk data non-linear[10].

ANN memiliki kemampuan untuk diterapkan sebagai metode *feature reduction*. Karena ANN menggunakan fungsi nonlinear. ANN dapat dilihat sebagai metode nonlinear PCA yang tidak menggunakan asumsi independensi antar data. Salah satu arsitektur ANN yang sering digunakan untuk itu adalah *Autoencoder*. *Autoencoder* yaitu *neural network* yang dapat merepresentasikan data kemudian merekonstruksinya kembali. *Autoencoder* mampu melakukan kompresi dengan baik karena ia mampu menemukan *hidden structure* dari data dan dapat juga mempelajari fitur non linear[11]. *Autoencoder* ini dilatih dengan metode *supervised learning*, tetapi tidak membutuhkan kelas dari data sebagai target. *Autoencoder* merupakan sebuah metode yang memiliki dua bagian utama yaitu *Encoder* dan *Decoder*, *Autoencoder* memiliki jumlah masukan dan keluaran yang sama dan selalu berbentuk simetris atau jam pasir. Selain itu juga memiliki layer code, dan jumlah neuron pada layer code merupakan jumlah dimensi yang mengurangi dimensi yang terdapat pada data[12][9]. Kelebihan dari metode ini adalah dapat melakukan *hierarchical feature learning* dari data yang digunakan ,

sehingga dapat menciptakan representasi bertingkat dimana representasi tersebut semakin abstrak dari tingkatan sebelumnya sehingga diharapkan menghasilkan representasi yang baik[13]. Autoencoder juga dapat digunakan untuk *denoising* atau menghilangkan derau pada suatu data masukan yang disebut dengan *Denoising Autoencoder*[9]. *Denoising Autoencoder* memiliki ciri yang sama dengan *Autoencoder* hanya saja didalam *Denoising Autoencoder* data yang diproses adalah data yang telah diberi noise sehingga pelatihan atau proses training akan menghasilkan data yang abstrak dibandingkan dengan *Autoencoder* biasa[13].

Preproses data merupakan salah satu tahap dalam *machine learning*. Algoritma machine learning belajar dari data, sehingga penting menyiapkan data secara tepat untuk menyelesaikan suatu masalah. Masalahnya data biasanya tidak tersedia dalam bentuk siap digunakan. Selain itu, algoritma yang berbeda mungkin memerlukan transformasi data yang berbeda. Meski memiliki data yang baik, perlu dipastikan lagi bahwa data tersebut ada dalam skala yang sama. Teknik *scaling data* memiliki peran penting dalam praproses data dan biasanya digunakan untuk menyamakan skala data agar bernilai antara 0 sampai 1. Penggunaan metode *data scaling* yang tepat dapat mengoptimalkan kinerja dari algoritma *machine learning*[14]. Kombinasi antara teknik normalisasi data dengan algoritma machine learning dapat dipastikan memiliki pengaruh terhadap performa yang dihasilkan. Li dan Liu 2011 dalam penelitiannya menyatakan, bahwa normalisasi min-max memiliki performa yang bagus dalam hal kecepatan, akurasi dan kuantitas suport vector pada SVM. Disisi lain Tang dan Sutskever 2011 menyatakan pada banyak algoritme machine learning langkah standar sebelum pelatihan adalah menghapus rata-rata dari data yang dikenal dengan standarisasi[14].

Set data yang diolah dalam sistem seperti *data mining* menggunakan set data sebagai basis data utama dalam menyelesaikan kasus yang ditangani, bisa memiliki ukuran yang sangat besar dalam hal jumlah fitur yang digunakan. Jumlah fitur yang besar diharapkan akan semakin banyak karakter data yang dapat di capture menjadi model sistem. Tetapi jumlah fitur yang semakin besar juga berpengaruh pada waktu komputasi yang juga lama. Jumlah fitur yang besar dapat

mengakibatkan terjadinya *curse of dimensionality*[15], dimana masalah ini mengakibatkan banyak jenis analisis data menjadi secara signifikan lebih berat atau susah dengan semakin meningkatnya jumlah dimensi. Secara khusus, dengan semakin meningkatnya jumlah dimensi, maka data menjadi semakin meningkat sebaran tempat yang dibutuhkan. Untuk klasifikasi atau prediksi, hal ini berarti bahwa tidak ada obyek data yang cukup untuk dapat membuat model yang handal dalam memberikan label kelas pada semua obyek yang ada. Hasilnya, banyak algoritma klasifikasi dan prediksi menjadi bermasalah pada data dengan dimensi tinggi berupa menurunkan akurasi dan kualitas yang jelek.

Berbagai faktor yang mempengaruhi berat lahir bayi antara lain meliputi jenis kelamin bayi, ras, keadaan plasenta, umur ibu, aktivitas ibu, kebiasaan merokok, paritas, jarak kelahiran, tinggi badan dan berat badan ibu sebelum kehamilan, status sosial ekonomi, gizi. Memperhatikan faktor-faktor tersebut mendukung intervensi dalam memantau kesehatan ibu hamil yang ditandai penambahan berat badan ibu yang akan mempengaruhi berat bayi lahir. Setiap dkk dalam [16].

Pada kesempatan ini penelitian yang akan dilakukan adalah penerapan algoritma *Deep learning* dengan metode *Autoencoder* dalam memprediksi berat badan bayi lahir, dan sebagai pembandingan digunakan metode PCA. Selain itu penelitian ini bertujuan untuk menganalisa performa algoritma machine learning terhadap teknik data scaling dalam prediksi berat badan bayi lahir diterapkan teknik normalisasi *MinMaxScaler* dan *StandardScaler*.

Atribut yang digunakan dalam penelitian ini adalah jumlah janin, outcome, tanggal lahir, hari kehamilan, jenis kelamin bayi, paritas, ras ibu, usia ibu, pendidikan ibu, tinggi ibu, berat ibu pra kehamilan, ras ayah, usia ayah, pendidikan ayah, tinggi ayah, berat ayah, status perkawinan, total pendapatan, ibu merokok, frekuensi merokok, quantity merokok sehari dan berat lahir bayi sebagai label. Hasil penelitian ini diharapkan dapat memberikan manfaat kepada masyarakat Indonesia khususnya kepada ibu hamil yang sedang mempersiapkan untuk proses persalinan yang baik dan tidak merugikan luaran bayi.

1.2. Identifikasi Permasalahan

Berdasarkan latar belakang tersebut di atas maka permasalahan dalam penelitian ini adalah:

1. Berat Bayi Lahir Rendah (BBLR) adalah salah satu faktor utama yang berpengaruh terhadap kematian neonatal.
2. Ketepatan prediksi dari berat lahir bayi adalah salah satu pengukuran yang paling penting pada awal persalinan.
3. Jumlah fitur yang besar dapat mengakibatkan terjadinya *curse of dimensionality*.
4. Model deep learning Autoencoder seperti apa yang memberikan hasil prediksi paling akurat dalam memprediksi berat badan bayi lahir.
5. Metode normalisasi apa yang paling efektif untuk prediksi berat badan lahir bayi.
6. Bagaimana hasil akurasi masing-masing eksperimen berdasarkan nilai *Mean Squared Error*, *Mean Absolute Error* dan R^2 score.

1.3. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah:

1. Mendapatkan model deep learning Autoencoder yang memberikan hasil prediksi paling akurat dalam memprediksi berat badan lahir bayi.
2. Mendapatkan metode normalisasi yang paling efektif untuk prediksi berat bayi lahir.
3. Menganalisa hubungan antar atribut terhadap hasil prediksi berat lahir bayi.

1.4. Ruang Lingkup Penelitian

Kasus yang diangkat dalam penelitian ini dibatasi pada memprediksi berat badan lahir bayi dengan 23 *feature* pada data baby yang di dapat dari data publik *kaggle* dengan metode *Deep learning Autoencoder* dan sebagai pembanding menggunakan metode PCA. Selain itu juga menganalisa performa algoritma *machine learning* terhadap teknik data *scaling* dalam prediksi berat badan bayi lahir diterapkan teknik normalisasi *MinMaxScaler* dan *StandarScaler*.

1.5. Hipotesis

Hipotesis dari penelitian ini adalah sebagai berikut:

1. Algoritma *Deep Learning Autoencoder* dan metode normalisasi yang tepat dapat menangani masalah prediksi berat bayi lahir dan memberikan hasil akurasi yang baik.
2. Algoritma *Deep Learning Autoencoder* dan metode normalisasi yang tepat tidak dapat menangani masalah prediksi berat bayi lahir dan tidak memberikan hasil akurasi yang baik.

BAB II

LANDASAN/KERANGKA PEMIKIRAN

2.1. Tinjauan Pustaka

Untuk mendapatkan kesamaan persepsi mengenai pembahasan pada penelitian ini, maka perlu penjelasan secara teoritis berkaitan dengan definisi atau pengertian objek kajian. Referensi diambil dari buku, jurnal nasional dan internasional yang berhubungan dengan judul penelitian.

2.1.1. Berat Bayi Lahir

a. Pengertian

Berat bayi lahir adalah berat badan bayi yang di timbang dalam waktu 1 jam pertama setelah lahir [17]. Hubungan antara berat lahir dengan umur kehamilan, berat bayi lahir dapat dikelompokan : bayi kurang bulan (BKB), yaitu bayi yang dilahirkan dengan masa gestasi < 37 minggu (259 hari). Bayi cukup bulan (BCB), bayi yang dilahirkan dengan masa gestasi antara 37- 42 minggu (259 - 293 hari), dan Bayi lebih bulan (BLB), bayi yang dilahirkan dengan masa gestasi > 42 minggu (294 hari) [18].

b. Klasifikasi Berat Bayi Lahir

1. Bayi Berat Lahir Rendah (BBLR)

Berat yang dilahirkan dengan berat lahir <2500 gram tanpa memandang usia gestasi. BBLR atau disebut *low birth weight infant* adalah *neonatus* dengan berat badan lahir pada saat kelahiran kurang dari 2500 gram (sampai 2499 gram). BBLR tidak selamanya prematur atau kurang bulan tetapi dapat cukup bulan maupun lebih bulan. Penelitian Gruenwald menunjukkan bahwa sepertiga bayi lahir dengan berat lahir rendah sebenarnya adalah bayi cukup bulan [19].

Bayi berat lahir rendah merupakan masalah penting dalam pengelolaannya karena mempunyai kecenderungan ke arah peningkatan terjadinya infeksi, kesukaran mengatur nafas tubuh sehingga mudah untuk menderita hipotermia. Selain itu bayi dengan Berat Bayi Lahir Rendah (BBLR) mudah terserang komplikasi tertentu seperti *ikterus*, *hipoglikomia*,

hiperbilirubinemia, polisitemia, trombositopenia dan *nekrotikans* yang dapat menyebabkan kematian [3]. Kelompok bayi berat lahir rendah yang dapat diistilahkan dengan kelompok resiko tinggi, karena pada bayi berat lahir rendah menunjukkan angka kematian dan kesehatan yang lebih tinggi dengan berat bayi lahir cukup [7].

2. Bayi Berat Lahir Normal

Bayi baru lahir normal adalah bayi yang lahir dari kehamilan sampai 42 minggu dan berat badan lahir > 2500 - 4000 gram [20].

3. Bayi Berat Lahir Lebih

Bayi berat lahir lebih adalah Bayi yang dilahirkan dengan berat lahir lebih > 4000 gram. Menurut dr.Yusfa ada beberapa faktor yang menyebabkan bayi besar, mulai dari faktor keturunan, kenaikan berat badan ibu yang berlebihan serta yang terseing karena diabetes gestational, atau diabetes dalam kehamilan. Dr. Yusfa menyatakan bahwa bayi besar dalam kandungan bisa sangat beresiko. Bisa berisiko adanya kelainan bawaan. Satu hal yang paling berat adalah sudden death, janin tiba-tiba meninggal dalam kandungan [21]. Berat badan janin >3500 gram, beresiko trauma partus melalui vagina seperti distosia bahu dan kerusakan jaringan lunak pada ibu[20]. Saat persalinan pada bayi yang beratnya kurang dari 4 kg, lingkaran kepala lebih besar dari lingkaran bahu. Sehingga kalau kepala sudah lahir, pasti bahu bayi dapat dilahirkan dengan risiko cedera pada bayi dan ibu yang kecil. Pada janin yang lebih 4 kg, lingkaran bahunya lebih besar dari kepala, jadi begitu kepala dilahirkan normal bisa bisa bahu bayi nyangkut, kepala geleng-geleng saja di luar. Oleh karena itu untuk mengeluarkan bagian bahu, ada beberapa manufer dilakukan dengan episiotomi luas (cedera pada ibu besar) atau memutar dan mematahkan tulang bahu si bayi, karena tidak ada jalan lain bayi harus lahir dari bawah karena kepala sudah lahir, tidak bisa kemudian diputuskan sesar. Akibatnya cedera pada bayi juga amat besar dan bisa berakhir dengan kematian janin, pendarahan ibu, atau lepasnya tulang simpisis [21].

c. Faktor yang Mempengaruhi Berat Bayi Lahir

Berat lahir merupakan hasil interaksi dari berbagai faktor melalui suatu proses yang berlangsung selama berada dalam kandungan. Menurut Jaya dalam [1] faktor-faktor yang dapat mempengaruhi berat bayi lahir adalah sebagai berikut:

Faktor Maternal mempengaruhi berat bayi lahir antara lain sebagai berikut :

1. Umur Ibu hamil

Umur ibu erat kaitannya dengan berat bayi lahir, kehamilan dibawah umur 20 tahun merupakan kehamilan berisiko tinggi, 2-4 kali lebih tinggi di bandingkan dengan kehamilan pada wanita yang cukup umur. Pada umur yang masih muda, perkembangan organ-organ reproduksi dan fungsi fisiologinya belum optimal. Selain itu emosi dan kejiwaannya belum cukup matang, sehingga pada saat kehamilan ibu tersebut belum dapat menanggapi kehamilannya secara sempurna dan sering terjadi komplikasi. Selain itu semakin muda usia ibu hamil, maka anak yang dilahirkan akan semakin ringan. Meski kehamilan dibawah umur sangat berisiko tetapi kehamilan diatas usia 35 tahun juga tidak dianjurkan, sangat berbahaya. Mengingat mulai usia ini sering muncul penyakit seperti hipertensi, tumor jinak peranakan, atau penyakit degeneratif pada persendian tulang belakang dan panggul. Kesulitan lain kehamilan diatas usia 35 tahun ini yakni bila ibu ternyata mengidap penyakit seperti diatas yang ditakutkan bayi lahir dengan membawa kelainan. Dalam proses persalinan sendiri, kehamilan di usia lebih ini akan menghadapi kesulitan akibat lemahnya kontraksi rahim serta sering timbul kelainan pada tulang panggul tengah. Mengingat bahwa faktor umur memegang peranan penting terhadap derajat kesehatan dan kesejahteraan ibu hamil serta bayi, maka sebaiknya merencanakan kehamilan pada usia antara 20-35 tahun [22]. Reproduksi sehat dikenal dengan usia aman untuk kehamilan yaitu usia ibu hamil pada umur 20 – 35 tahun, karena masa tersebut merupakan masa yang aman untuk hamil alasannya, mulai umur 20 tahun rahim dan bagian – bagian lainnya sudah benar – benar siap untuk menerima kehamilan. Pada umur tersebut biasanya wanita sudah merasa siap untuk menjadi ibu [23]. Dan sebaiknya

tidak hamil pada usia >35 tahun, karena kesehatan tubuh ibu sudah tidak sebaik pada umur 20 – 35 tahun, biasanya ibu sudah mempunyai dua anak atau lebih, kemungkinan memperoleh anak cacat lebih besar[22]. Menurut Depkes RI menyatakan bahwa kehamilan pada umur dibawah 20 tahun rahim dan panggul ibu belum berkembang dengan baik, hingga perlu diwaspadai kemungkinan mengalami persalinan yang sulit dan keracunan hamil, sedangkan kehamilan pada usia > 35 tahun kesehatan dan keadaan rahim tidak sebaik seperti pada umur 20 – 35 tahun sebelumnya, hingga perlu diwaspadai kemungkinan terjadinya persalinan lama, perdarahan dan risiko cacat bawaan. Selain itu semakin muda dan semakin tua umur seorang ibu yang sedang hamil, akan berpengaruh terhadap kebutuhan gizi yang diperlukan[23].

2. Jarak Kehamilan/Kelahiran

Jarak kelahiran yang ideal adalah 2 tahun atau lebih, karena jarak kelahiran yang pendek akan menyebabkan seorang ibu belum cukup untuk memulihkan kondisi tubuhnya setelah melahirkan sebelumnya [23]. Semakin pendek (< 2 tahun), ibu berisiko tinggi untuk mengalami pre-eklampsia dan komplikasi kehamilan lain yang sangat berbahaya dan juga bagi bayinya bisa lahir terlalu cepat, terlalu kecil atau dengan BBLR [24]. Ini merupakan salah satu faktor penyebab kelemahan dan kematian ibu serta bayi yang dilahirkan.

3. Paritas

Paritas secara luas mencakup gravida/jumlah kehamilan, prematur/jumlah kelahiran, dan abortus/jumlah keguguran. Paritas adalah jumlah anak yang pernah dilahirkan hidup yaitu kondisi yang menggambarkan kelahiran sekelompok wanita selama masa reproduksi. Klasifikasi paritas *Primipara* wanita yang telah melahirkan seorang anak, yang cukup besar untuk hidup di dunia luar. *Multipara* adalah wanita yang pernah melahirkan bayi beberapa kali (sampai 5 kali). *Grandemultipara* adalah wanita yang pernah melahirkan bayi 6 kali atau lebih hidup atau mati. Parita yang paling aman di tinjau dari sudut kematian maternal dan perinatal adalah paritas 2-3. Paritas 1 dan ≥ 4 mempunyai angka kematian maternal lebih tinggi [23].

4. Kadar *Hemoglobin* (Hb)

Kadar *hemoglobin* (Hb) ibu hamil sangat mempengaruhi berat bayi yang dilahirkan. Selain 3 faktor diatas yang mempengaruhi terjadinya BBLR yaitu anemia. Kebanyakan anemia dalam kehamilan disebabkan oleh defisiensi besi dan pendarahan akut, bahkan tidak jarang keduanya saling berinteraksi. Pada kehamilan kebutuhan oksigen lebih tinggi, sehingga memicu peningkatan produksi *eritropoietin*. Akibatnya, volume plasma bertambah dan sel darah (*eritrosit*) meningkat. Namun peningkatan volume plasma terjadi dalam proporsi yang lebih besar jika dibandingkan dengan peningkatan eritrosit sehingga terjadi penurunan konsentrasi hemoglobin (Hb) akibat hemodilusi. Anemia dapat menyebabkan pengangkutan oksigen menjadi terganggu sehingga suplai darah nutrisi akan oksigen pada placenta, yang akan berpengaruh pada fungsi plasenta terhadap janin sehingga nutrisi ke janin berkurang [24].

5. Status Gizi Ibu Hamil

Status gizi ibu pada waktu pembuahan dan selama hamil dapat mempengaruhi pertumbuhan janin yang sedang dikandung. Selain itu gizi ibu hamil menentukan berat bayi yang dilahirkan, maka pemantauan gizi ibu hamil sangatlah penting dilakukan. Ada beberapa cara yang dapat digunakan untuk mengetahui status gizi ibu hamil antara lain memantau pertambahan berat badan selama hamil, mengukur lingkaran lengan atas (LILA) dan mengukur kadar hemoglobin. Sebagai ukuran sekaligus pengawasan bagi kecukupan gizi ibu hamil bisa di lihat dari kenaikan berat badannya. Ibu yang kurus dan selama kehamilan disertai penambahan berat badan yang rendah atau turun, mempunyai risiko paling tinggi untuk melahirkan bayi dengan BBLR dengan dampak yang ditimbulkannya [16].

6. Penyakit Saat Kehamilan

Penyakit pada saat kehamilan yang dapat mempengaruhi berat bayi lahir diantaranya adalah *Diabetes Melitus Gestasional* (DMG), hipertensi, komplikasi *tromboemboli*, persalinan *premature* dan *makrosomia* [25]. Komplikasi kehamilan merupakan faktor resiko terjadinya BBLR. Ibu

hamil yang mengalami komplikasi kehamilan berisiko 2,54 kali lebih besar mengalami BBLR dibandingkan dengan ibu hamil yang tidak mengalami komplikasi kehamilan. Ibu yang mengalami penyakit selama kehamilan mempunyai peluang melahirkan BBLR 2,91 kali dibandingkan ibu yang tidak mengalami penyakit selama kehamilan [2].

Faktor-faktor yang mempengaruhi berat bayi lahir secara tidak langsung/eksternal dapat dijelaskan sebagai berikut :

1. Faktor lingkungan eksternal yang meliputi kondisi lingkungan, asupan zat gizi ibu hamil, kebersihan dan kesehatan lingkungan dan gaya hidup (perokok/alkohol). Faktor kebersihan dan kesehatan lingkungan berkaitan dengan cacing tambang, Seseorang yang asupan zat besinya cukup tetapi jika sering terinfeksi cacing tambang dapat menderita anemia. Demikian juga jika seorang yang asupan zat besi rendah maka daya tahan tubuhnya berkurang sehingga mudah sering mudah terserang penyakit dan akhirnya akan mengalami penurunan kadar Hb [24].

2. Faktor Sosial ekonomi

Tingkat sosial ekonomi terbukti sangat berpengaruh terhadap kondisi kesehatan fisik dan psikologis ibu hamil. Pada ibu hamil dengan tingkat sosial ibu hamil yang baik otomatis akan mendapatkan kesejahteraan fisik dan psikologis yang baik pula. Status gizi pun akan meningkat karena nutrisi yang didapatkan berkualitas, selain itu ibu tidak akan terbebani secara psikologis mengenai biaya persalinan dan pemenuhan kebutuhan sehari-hari setelah bayinya lahir. Menurut buku ajaran gizi untuk kebidanan bahwa faktor-faktor yang mempengaruhi gizi ibu hamil terutama dengan anemia salah satunya status ekonomi, karena ekonomi seseorang mempengaruhi dalam pemilihan makanan yang akan dikonsumsi sehari-hari. Kurangnya pendapatan keluarga menyebabkan lokasi dan untuk pembelian makanan sehari-hari sehingga mengurangi jumlah dan kualitas makanan ibu perhari yang berdampak pada penurunan status gizi yang mempengaruhi berat janin [26]. Status sosial ekonomi yang rendah juga akan mempengaruhi Faktor penggunaan sarana kesehatan yang berhubungan frekuensi pemeriksaan kehamilan / ANC

dikarenakan saat hamil seseorang dengan sosial ekonomi rendah tidak mampu untuk melakukan pemeriksaan. Pemeriksaan kehamilan bertujuan untuk mengenal dan mengidentifikasi masalah yang timbul selama kehamilan, sehingga kesehatan selama ibu hamil dapat terpelihara dan yang terpenting ibu dan bayi dalam kandungan akan baik dan sehat sampai saat persalinan[26].

3. Faktor Pendidikan

Tingkat pendidikan sangat mempengaruhi bagaimana seseorang untuk bertindak dan mencari penyebab serta solusi dalam hidupnya. Orang yang berpendidikan tinggi biasanya akan bertindak lebih rasional. Oleh karena itu orang yang berpendidikan akan lebih mudah menerima gagasan baru. Demikian halnya dengan ibu yang berpendidikan tinggi akan memeriksakan kehamilannya secara teratur demi menjaga keadaan kesehatan dirinya dan anak dalam kandungannya. Pendidikan sangat mempengaruhi kemampuan seseorang dalam informasi gizi. Semakin tinggi tingkat pendidikan (lama sekolah) seseorang, semakin mudah menerima hidup sehat secara mandiri, kreatif dan berkesinambungan. Oleh karena itu tingkat pendidikan mempunyai hubungan yang eksponensial terhadap status gizi dan kesehatan. Makin tinggi pendidikan makin tinggi pula kesadaran ibu untuk mendapatkan gizi yang baik sehingga tidak menimbulkan anemia pada kehamilan yang akan berdampak juga terhadap janin dalam kandungan [26].

2.1.2. Machine Learning (ML)

Machine Learning merupakan program komputer yang belajar dari data yang di berikan. Istilah *machine learning* pertama kali didefinisikan oleh Arthur Samuel ditahun 1959. Menurut Arthur Samuel, *machine learning* adalah salah satu bidang ilmu komputer yang memberikan kemampuan pembelajaran kepada komputer untuk mengetahui sesuatu tanpa pemrogram yang jelas. Sementara menurut Tom Mitchell – 1997 suatu program komputer dikatakan belajar dari pengalaman E sehubungan dengan beberapa tugas T dan beberapa ukuran kinerja

P, jika kinerjanya pada T, yang diukur dengan P, meningkat dengan pengalaman E [27].

Secara garis besar machine learning terbagi menjadi 3 bagian yaitu:

a. Supervised Learning

Supervised learning adalah memprediksi kelas berdasarkan *feature vector* yang merepresentasikan suatu instans (data/observasi). *Feature vector* bisa diibaratkan sebagai sifat-sifat atau keadaan yang diasosiasikan dengan kelas. Pada *supervised learning*, setiap *feature vector* berkorespondensi dengan kelas tertentu. Dengan kata lain *supervised learning* adalah pembelajaran menggunakan masukan data pembelajaran yang telah diberi label [11].

b. Unsupervised Learning

Unsupervised learning adalah algoritma pembelajaram untuk mengidentifikasi pola-pola tersembunyi dalam data input tidak berlabel. Unsupervised learning dapat mengolah informasi data yang tidak berlabel melalui pola data tersebut. Pada *unsupervised learning* tidak ada kelas yang berkorespondensi akan tetapi mengelompokan data dengan sifat-sifat yang mirip, disebut *clustering* [11].

c. Reinforcement Learning

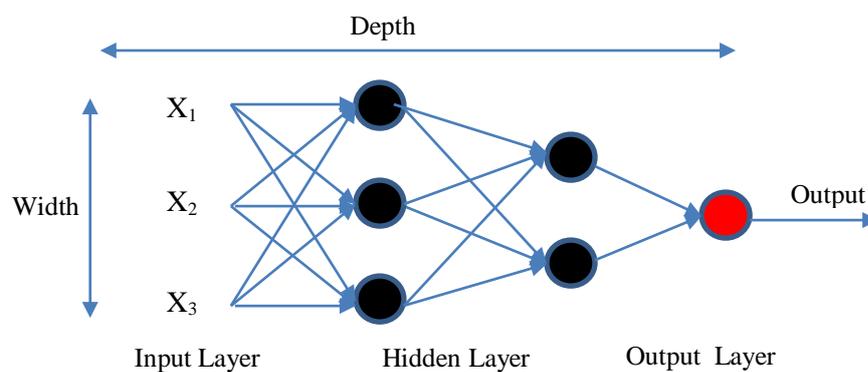
Di dalam *reinforcement learning*, algoritma berinteraksi dengan lingkungan dengan menghasilkan urutan tindakan a_1, a_2, a_3, \dots , an seiring waktu. Tindakan-tindakan ini mempengaruhi lingkungan, yang menghasilkan hadiah atau hukuman dalam setiapbatas waktu t . Tujuan dari algortima ini adalah untuk belajar bertindak dengan cara yang kemungkinan akan memaksimalkan beberapa ukuran utilitas di masa depan. Penguatan tidak harus berupa hadiah atau hukuman, melainkan dapat berupa tanggapan balik yang berguna untuk menentukan tindakan di masa mendatang [28]

Dengan kata lain reinforcemen learning adalah algoritma pembelajaran yang dapat belajar dari kesalahan dan mengandalkan tanggapan balik maupun itu positif atau negatif untuk berkembang.

2.1.3. Artificial Neural Network (ANN)

Artificial Neural Network (ANN) atau jaringan saraf tiruan merupakan sebuah sistem pemrosesan data yang terinspirasi dari bagaimana jaringan otak bekerja baik manusia maupun hewan. Algoritma ANN lahir dari gagasan psikolog, Warren McCulloch dan Walter Pitts pada tahun 1943 yang menjelaskan cara kerja jaringan saraf dengan perangkat jaringan elektronik. ANN adalah sebuah sistem yang terdiri atas banyak elemen pemrosesan sederhana yang terhubung secara paralel [29].

Artificial Neural Network, menghasilkan model yang sulit dibaca dan dimengerti oleh manusia karena memiliki banyak layer (kecuali *single perceptron*) dan sifat *non-linear* (merujuk pada fungsi aktivasi). ANN disebut agnostik – kita percaya, tetapi sulit membuktikan kenapa konfigurasi parameter yang dihasilkan training bisa benar. Konsep matematis ANN itu sendiri cukup solid, tetapi *interpretability* model rendah menyebabkan kita tidak dapat menganalisa proses inferensi yang terjadi pada model ANN. Secara matematis, ANN ibarat semua graf. ANN memiliki neuron/node (*vertex*), dan sinapsis (*edge*). Karena memiliki struktur seperti graf, operasi pada ANN mudah dijelaskan dalam notasi aljabar linear. Sebagai gambaran ANN berbentuk seperti gambar berikut [11].



Gambar 2. 1 Deep Neural Network

Pada gambar 2.1 terdapat *Depth* dan *Width*, *depth* adalah kedalaman ANN yang mengacu pada jumlah layer sementara *width* adalah lebar ANN yang mengacu pada jumlah unit pada layer. Lapisan-lapisan pada ANN adalah sebagai berikut [29]:

1. *Input Layer* (Lapisan Masukan)

merupakan lapisan yang menghubungkan sumber data ke jaringan pemrosesan. Dalam artian, setiap masukan akan merepresentasikan variabel-variabel bebas yang berpengaruh terhadap keluaran (*output*)

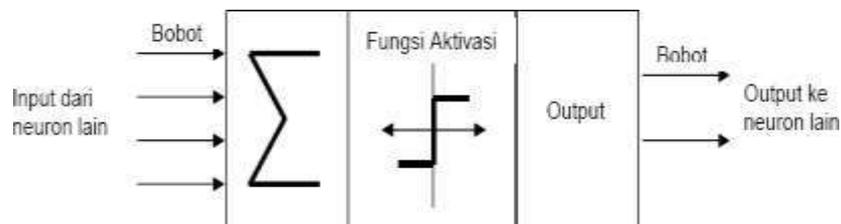
2. *Hidden Layer* (Lapisan Tersembunyi)

Merupakan lapisan perambat variabel-variabel input untuk mendapatkan hasil output yang lebih mendekati keinginan. Suatu ANN Multi Layer dapat memiliki satu atau lebih *hidden layer*.

3. *Output Layer* (Lapisan Keluaran)

merupakan hasil keluaran dari pemrosesan data ANN. Keluaran yang didapatkan bergantung pada bobot, jumlah lapisan tersembunyi (*hidden layer*), dan fungsi aktivasi yang

Seperti halnya jaringan syaraf pada otak manusia, neural network juga terdiri dari beberapa neuron yang saling berhubungan. Masing-masing neuron akan menerima informasi dan mengakumulasiannya dari neuron yang tersambung sebelumnya. Hubungan ini disebut dengan sebutan bobot (*Weight*). Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tertentu. Berikut adalah struktur Neuron pada *neural network* [30]:



Gambar 2. 2 Arsitektur *Neural Network*

Karakteristik dari ANN dilihat dari pola hubungan antar neuron, metode penentuan bobot dari tiap koneksi, dan fungsi aktivasinya. Gambar di atas menjelaskan struktur ANN secara mendasar, yang dalam kenyataannya tidak hanya sederhana seperti itu.

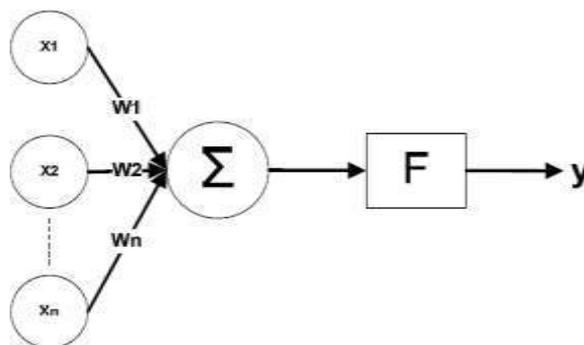
1. *Input*, berfungsi seperti dendrite
2. *Output*, berfungsi seperti akson
3. Fungsi aktivasi, berfungsi seperti sinapsis

Neural network dibangun dari banyak node/unit yang dihubungkan oleh *link* secara langsung. *Link* dari unit yang satu ke unit yang lainnya digunakan untuk melakukan propagasi aktivasi dari unit pertama ke unit selanjutnya. Setiap *link* memiliki bobot numerik. Bobot ini menentukan kekuatan serta penanda dari sebuah konektivitas.

Proses pada ANN dimulai dari *input* yang diterima oleh neuron beserta dengan nilai bobot dari tiap-tiap *input* yang ada. Setelah masuk ke dalam neuron, nilai *input* yang ada akan dijumlahkan oleh suatu fungsi perambatan (*summing function*), yang bisa dilihat seperti pada di gambar dengan lambang sigma (Σ). Hasil penjumlahan akan diproses oleh fungsi aktivasi setiap neuron, disini akan dibandingkan hasil penjumlahan dengan *threshold* (nilai ambang) tertentu. Jika nilai melebihi *threshold*, maka aktivasi neuron akan dibatalkan, sebaliknya, jika masih dibawah nilai *threshold*, neuron akan diaktifkan. Setelah aktif, neuron akan mengirimkan nilai *output* melalui bobot-bobot outputnya ke semua neuron yang berhubungan dengannya. Proses ini akan terus berulang pada input-input selanjutnya [30].

Pada gambar 3 sebuah *neuron* akan mengolah N masukan ($X_1, X_2, X_3, \dots, X_n$) yang masing-masing memiliki bobot $W_1, W_2, W_3, \dots, W_n$ dengan rumus :

$$y_{in} = \sum_{i=1}^n x_i w_i \quad (1)$$



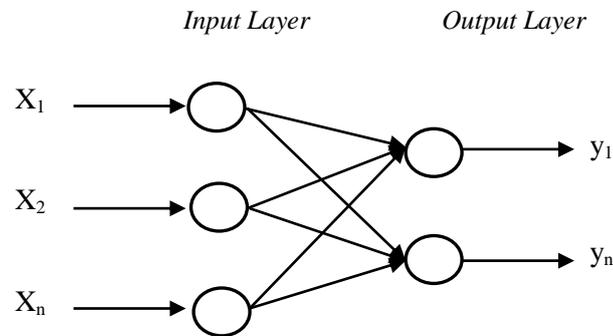
Gambar 2. 3 Model Single Perceptron

2.1.4. Arsitektur *Artificial Neural Network*

Arsitektur *neural network* dapat dibagi berdasarkan jumlah lapisannya diantaranya [31][11]:

a. *Single Layer Neural Network*

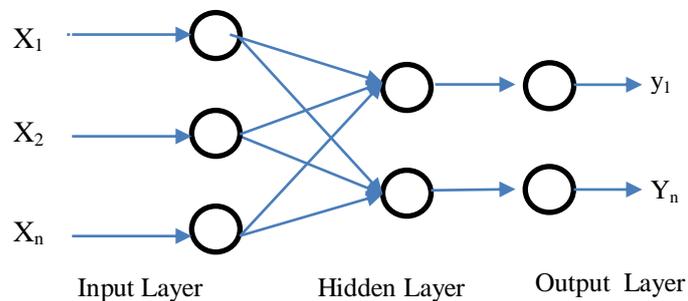
Pada *single layer neural network*, *input layer* terhubung langsung ke *output layer*. Kelemahan dari jenis ini adalah hanya bisa digunakan pada kasus sederhana.



Gambar 2.4 *Single Layer Neural Network*

b. *Multi Layers Neural Network*

Pada *multi layer neural network*, terdapat *hidden layer* yang terletak diantara *input layer* dan *output layer*.



Gambar 2. 5 *Multi Layer Neural Network*

2.1.5. *Deep Learning*

Setelah AI menjadi dominan pada 1950-an, jaringan saraf popularitasnya menurun. Namun, algoritma baru untuk pelatihan jaringan saraf dan peningkatan kecepatan pemrosesan komputer secara dramatis mengakibatkan munculnya kembali penggunaan jaringan saraf di bidang yang disebut *deep learning*. *Deep Learning* (DL) merupakan salah satu bidang dari *Machine Learning* yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset yang besar. Arsitektur DL neural network berbeda dari jaringan neural

terdahulu karena mereka sering memiliki lapisan tersembunyi. Selain itu, jaringan *deep learning* dapat dilatih dengan metode *unsupervised* ataupun *supervised learning*. *Deep learning* telah digunakan untuk menyelesaikan tugas-tugas seperti *computer vision*, *speech voice recognition* dan *natural language processing* yang sulit dengan pendekatan lainnya. *Deep learning* menerapkan arsitektur yang sangat kuat karena memanfaatkan jaringan syaraf tiruan yang sangat cocok untuk *supervised learning*, dan dapat menyelesaikan masalah kompleks. Dengan memanfaatkan hidden layer yang banyak *deep learning* bisa membuat model yang lebih bagus dan menghasilkan akurasi yang lebih tinggi dari neural network biasa.

Sumber kesulitan utama dalam banyak aplikasi kecerdasan buatan adalah banyak faktor variasi mempengaruhi setiap bagian data yang dapat kita amati. Sebagian besar aplikasi mengharuskan kita untuk memisahkan faktor-faktor variasi dan membuang faktor-faktor yang tidak kita pedulikan. Tentu saja, bisa sangat sulit untuk mengekstrak fitur abstrak tingkat tinggi seperti itu dari data mentah. Banyak dari faktor variasi ini, seperti aksent bicara, dapat diidentifikasi hanya dengan menggunakan pemahaman data tingkat manusia yang manusia yang cangguh dan nyaris manusiawi. *Deep learning* dapat memecahkan masalah sentral ini dalam pembelajaran representasi dengan memperkenalkan representasi yang diekspresikan dalam bentuk representasi lain yang lebih sederhana. Pembelajaran yang mendalam memungkinkan komputer untuk membangun konsep yang kompleks dari konsep yang lebih sederhana seperti sudut dan kontur yang dinamakan *Feature Engineering* [32].

Feature Engineering adalah salah satu fitur utama dari *Deep Learning* untuk mengekstrak pola yang berguna dari data yang akan memudahkan model untuk membedakan kelas. *Feature engineering* juga merupakan teknik yang paling penting untuk mencapai hasil yang baik pada tugas prediksi. Namun, sulit untuk dipelajari dan dikuasai karena kumpulan data dan jenis data yang berbeda memerlukan pendekatan teknik yang berbeda juga [32].

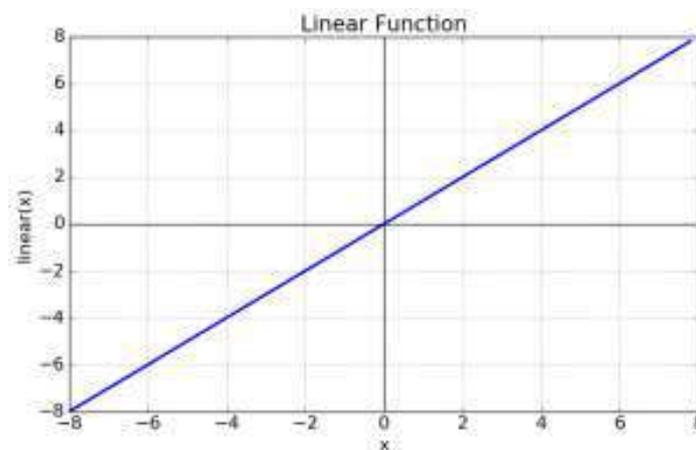
2.1.6. Activation Function

Activation function memiliki fungsi untuk menentukan apakah neuron tersebut harus “aktif” atau tidak berdasarkan weighted sum dari input. Artian lain

activation function merupakan suatu fungsi perambat pada metode ANN yang digunakan untuk mentransformasikan suatu masukan menjadi keluaran bernilai tertentu[29]. Secara umum terdapat 2 jenis *activation function*, *Linear dan Non-Linear Activation function* [29].

a. *Linear Function*

Fungsi ini umumnya digunakan untuk menghasilkan nilai output yang mendekati target yang diinginkan dengan formula $x=y$, yaitu nilai output sama dengan nilai input. Jika sebuah neuron menggunakan *linear function*, maka keluaran dari neuron tersebut adalah *weighted sum* dari input + bias.



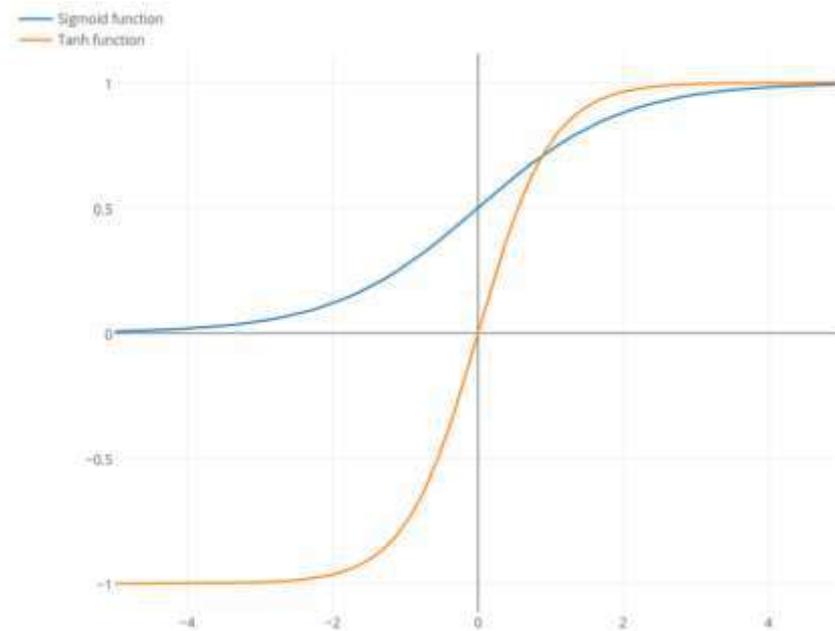
Gambar 2.6 Fungsi Aktivasi Linear

b. *Tanh dan Sigmoid Function (Non-Linear)*

Sigmoid function mempunyai rentang antara 0 hingga 1, sedangkan rentang *Tanh* adalah -1 hingga 1. Kedua fungsi ini biasanya digunakan untuk klasifikasi 2 class atau kelompok data.

$$\text{Sigmoid : } f(x) = \frac{1}{1+e^{-z}} \quad (2)$$

$$\text{Tanh : } f(x) = \tanh x = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (3)$$

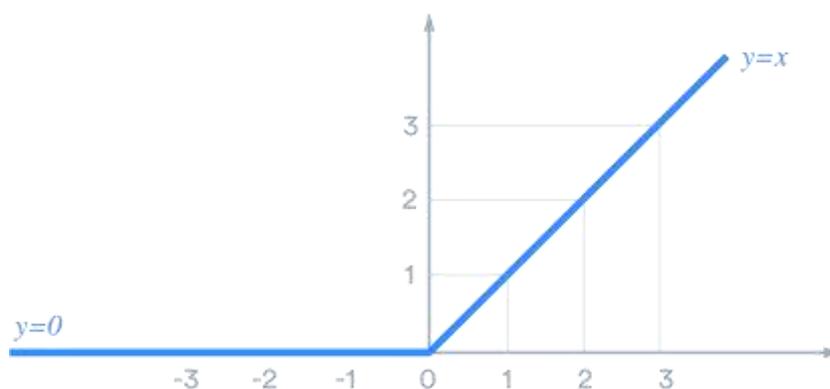


Gambar 2.7 Fungsi Aktivasi *Tanh* dan *Sigmoid*

c. ReLU (*Non-Linear*)

Pada dasarnya Relu melakukan “*reshold*” dari 0 hingga infinity. ReLU juga dapat menutupi kelemahan yang dimiliki oleh *Sigmoid* dan *Tanh*. Jika dibandingkan dengan sigmoid / tanh yang memiliki operasi-operasi yang “*expensive*”. ReLU bisa di implementasikan hanya dengan membuat pembatas (*threshold*) pada bilangan 0, artinya apabila $x \leq 0$ maka $x=0$ dan apabila $x > 0$ maka $x = x$ [33][34].

$$\text{ReLU} : f(x) = \max(0, x) \quad (4)$$

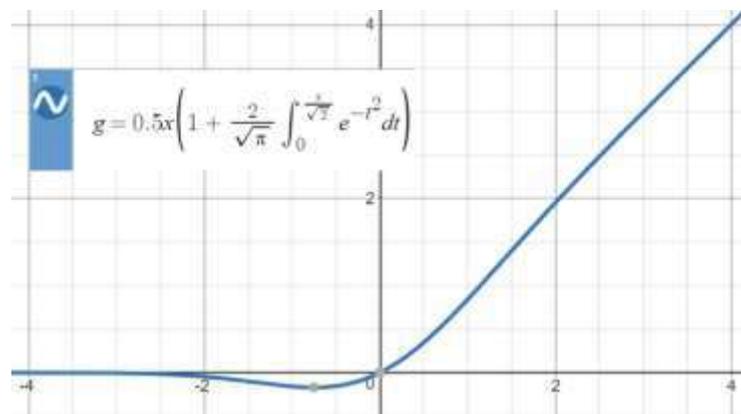


Gambar 2. 8 Fungsi Aktivasi ReLU

d. GELUs (*Gaussian Error Linear Units*)

Untuk menghasilkan kinerja yang lebih baik dari ReLU, GELUs diusulkan untuk memasukan distribusi stokastik yang tergantung pada input. Fungsi GELUs adalah merepresentasikan nonlinieritas menggunakan pengatur stokastik pada input, yang merupakan kumulatif fungsi distribusi yang berasal dari fungsi galat *Gaussian*[35]. Ketidak linieran GELU menimbang input berdasarkan nilainya. GELU secara acak menerapkan identitas atau nilai 0 ke peta input neuron dengan fungsi Gaussian Hendricks dan Gimpel[34][36].

$$\text{GeLU} : \text{GeLU}(x) = 0.5x(1 + \tanh[\sqrt{2/\pi}(x + 0.044715x^3)]) \quad (5)$$



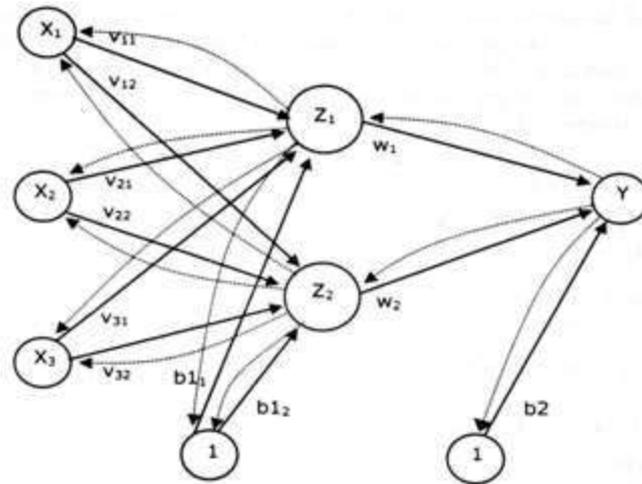
Gambar 2. 9 Fungsi Aktivasi GELU

2.1.7. Pembelajaran *Backpropagation*

Salah satu sifat *neural network* yang menyerupai dengan otak manusia adalah bahwa neural network membutuhkan proses pembelajaran. Pembelajaran dilakukan untuk menentukan nilai bobot yang tepat untuk masing-masing input. Bobot bertambah jika informasi yang diberikan oleh neuron yang bersangkutan tersampaikan. Sebaliknya jika informasi tidak disampaikan maka nilai bobot berubah secara dinamis sehingga dicapai suatu nilai yang seimbang. Apabila nilai ini telah mampu mengindikasikan hubungan yang diharapkan antara input dan output, proses pembelajaran bisa dihentikan.

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

tersembunyi. Algoritma ini menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan error ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi yang dapat diturunkan, seperti fungsi *sigmoid* Kusumadewi dalam [31][11].



Gambar 2. 10 Arsitektur Jaringan *Backpropagation*

Arsitektur jaringan *backpropagation* seperti ditunjukkan pada Gambar 2.10 menunjukkan neural network yang terdiri dari tiga unit neuron pada lapisan input (x_1 , x_2 , dan x_3), dua neuron pada lapisan tersembunyi (Z_1 dan Z_2), dan 1 unit neuron pada lapisan output (Y). Bobot yang menghubungkan x_1 , x_2 , dan x_3 dengan neuron pertama pada lapisan tersembunyi adalah V_{11} , V_{21} , dan V_{31} . b_{11} dan b_{12} adalah bobot bias yang menuju neuron pertama dan kedua pada lapisan tersembunyi. Bobot yang menghubungkan Z_1 dan Z_2 dengan neuron pada lapisan output adalah w_1 dan w_2 . Bobot bias b_2 menghubungkan lapisan tersembunyi dengan lapisan *output*.

Algoritma *backpropagation* untuk jaringan dengan satu lapisan tersembunyi sebagaimana pada Gambar 3.8 bekerja sebagai berikut [11]:

Langkah 0: Inisialisasi bobot awal, konstanta laju pelatihan (α), dan nilai bobot.

Langkah 1: lakukan tahap *forward propagation* untuk mendapatkan nilai *error*, tahap *forward propagation* adalah sebagai berikut:

- a. Tiap nilai input (X_i) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan tersembunyi (*hidden layer*).

- b. Unit tersembunyi (Z_j) menjumlahkan sinyal-sinyal input berbobot:

$$Z_{in j} = B_{1j} + \sum_{i=1}^n X_i V_{ij} \quad (6)$$

Dimana:

B_{1j} : Nilai bias yang menghubungkan bias *node* 1 dengan *hidden layer* 1

X_i : Nilai di input node i

V_{ij} : Nilai bobot yang menghubungkan *input node* i dengan *hidden node* j

- c. Setiap nilai dari unit *output* (Y_k) menjumlahkan sinyal input terbobot (*output* dari *hidden layer* 1) kemudian dikalikan dengan nilai bobot dan ditambahkan dengan bias dari *output layer*.

$$Y_{in} = b_{2k} + \sum_{j=1}^p Z_j W_{jk} \quad (7)$$

Dimana:

Y_{in} : Nilai fungsi aktivasi untuk menghitung signal output

B_{2k} : Nilai bias yang menghubungkan bias node 2 dengan hidden layer 2

Z_j : Nilai dari hasil output hidden layer

W_{jk} : Nilai bobot yang menghubungkan input node 2 dengan output layer 2

- d. Signal output dari output node k dengan memberikan fungsi aktivasi linear dengan formula $x=y$.

Langkah 2: Lakukan tahap backward propagation sebagai berikut:

- a. Hitung nilai error dari output node pada output layer (Y_k):

$$\delta_k = (t_k - y_k) y_k (1 - y_k) \quad (8)$$

Dimana :

δ_k : Nilai error output layer (Y_k)

t_k : Nilai output yang diharapkan

y_k : Nilai fungsi aktivasi pada output layer (y_k)

- b. Hitung nilai koreksi bobot yang akan digunakan untuk memperbaiki nilai W_{jk} .

$$\Delta W_{jk} = \alpha \cdot \delta_k Z_j \quad (9)$$

- c. Hitung koreksi bias yang akan digunakan untuk memperbaiki W_{0k} .

$$\Delta W_{0k} = \alpha \cdot \delta_k \quad (10)$$

Setelah itu kirim nilai δ_k ke unit yang ada di lapisan bawahnya.

- d. Tiap unit tersembunyi (Z_j) menjumlahkan delta input nya (dari unit-unit yang berada pada lapisan atasnya).

$$\delta_{inj} = \sum_{k=1}^m \delta_k W_{jk} \quad (11)$$

- e. Kalikan nilai tersebut dengan turunan dari fungsi aktivasi untuk menghitung nilai error.

$$\delta_j = \delta_{inj} f'(Z_{inj}) \quad (12)$$

- f. Hitung nilai koreksi bobot yang akan digunakan untuk memperbaiki nilai V_{ij} .

$$\Delta V_{ij} = \alpha \cdot \delta_j x_i \quad (13)$$

- g. Hitung koreksi bias yang akan digunakan untuk memperbaiki V_{oj} .

$$\Delta V_{oj} = \alpha \cdot \delta_j \quad (14)$$

- h. Tiap unit output (Y_k) memperbaiki bias dan bobotnya.

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \quad (15)$$

- i. Tiap unit tersembunyi memperbaiki bias dan bobotnya.

$$V_{ij}(\text{baru}) = V_{ij}(\text{lama}) + \Delta V_{ij} \quad (16)$$

Dari bobot-bobot inilah yang akan digunakan sebagai bobot awal pada proses *epoch* selanjutnya, proses ini dilakukan sampai target error sekecil mungkin (maksimum *epoch*).

2.1.8. Bias

Bias dapat ditambahkan sebagai salah satu komponen dengan nilai bobot yang selalu bernilai 1. Jika melibatkan bias, maka fungsi aktivasi menjadi:

$$f(x) = \begin{cases} 1 & \text{jika } net \geq 0 \\ -1 & \text{jika } net < 0 \end{cases} \quad (17)$$

Dimana :

$$x = b + \sum_i X_i W_i \quad (18)$$

2.1.9. Parameter

Parameter-parameter yang dapat digunakan untuk membantu peramalan adalah sebagai berikut[29][11]:

a. Learning Rate

Penggunaan parameter *learning rate* memiliki pengaruh penting terhadap waktu yang dibutuhkan untuk tercapainya target yang diinginkan. Secara perlahan

akan mengoptimalkan nilai perubahan bobot dan menghasilkan *error* yang lebih kecil. Variabel *learning rate* menyatakan suatu konstanta yang bernilai antara 0 - 1. Nilai tersebut menunjukkan kecepatan belajar dari jaringannya. Jika nilai *learning rate* yang digunakan terlalu kecil maka terlalu banyak *epoch* yang dibutuhkan untuk mencapai nilai target yang diinginkan, sehingga menyebabkan proses *training* membutuhkan waktu yang lama. Semakin besar nilai *learning rate* yang digunakan maka proses pelatihan jaringan akan semakin cepat, namun jika terlalu besar justru akan mengakibatkan jaringan menjadi tidak stabil dan menyebabkan nilai *error* berulang bolak-balik diantara nilai tertentu, sehingga mencegah *error* mencapai target yang diharapkan. Oleh karena itu pemilihan nilai *learning rate* harus seoptimal mungkin agar didapatkan proses *training* yang cepat.

b. *Epoch*

Epoch merupakan pendefinisian dari banyaknya jumlah iterasi yang digunakan. *Epoch* adalah ketika seluruh dataset sudah menjalani proses training pada neural network sampai dikembalikan ke awal dalam satu putaran. Dalam neural network satu epoch itu terlalu besar dalam proses pelatihan karena seluruh data diikutkan kedalam proses training sehingga akan membutuhkan waktu cukup lama. Untuk mempermudah dan mempercepat proses training, data rate dibagi per batch (Batch size)[37].

c. *Momentum*

Parameter *momentum* berfungsi untuk mendefinisikan kemiringan batas kesalahan maksimal yang boleh dihasilkan oleh model. Skala *momentum* yang dapat digunakan antara 0 hingga 0.9.

d. *Batch Size*

Batch size merupakan jumlah sampel data yang akan disebar dalam sebuah neural network. *Batch size* efisien secara komputasi ketika berhadapan dengan dataset yang besar. Sebagai contoh jika kita mempunyai 100 dataset dan *batch size* sebesar 5, maka algoritma ini akan menggunakan 5 sampel data pertama dari 100 data yang kita miliki (ke-1, ke-2, ke-3, ke-4, ke-5) lalu disebar atau ditraining oleh neural network sampai selesai kemudian

mengambil kembali 5 sampel data kedua dari 100 data (ke-6, ke-7, ke-8, ke-9, ke-10) dan begitu seterusnya sampai 5 sampel data ke 20 ($100/5=20$)[37].

e. Optimasi Adam

Optimasi merupakan algoritma yang digunakan untuk meminimalisir ataupun memaksimalkan sebuah fungsi objective (disebut juga *error function*), yang merupakan sebuah fungsi matematika yang bergantung pada parameter internal yang dapat dipelajari oleh model learning. Parameter internal tersebut digunakan dalam menghitung target value (y) dari sekumpulan predictor (x) yang digunakan dalam model. Sebagai contoh, nilai bobot dan bias dari neural network disebut sebagai parameter internal yang dapat dipelajari dan digunakan dalam menghitung nilai output. Ketika nilai tersebut dipelajari dan diperbarui ke arah solusi optimal dengan meminimalisir nilai loss dari proses training jaringan.

Adaptive Moment Estimation (Adam) adalah metode untuk optimasi stokastik efisien yang hanya memerlukan gradien orde pertama dengan kebutuhan memori yang sedikit. Metode ini menghitung tingkat pembelajaran adaptif individu untuk parameter yang berbeda dari perkiraan momen pertama dan kedua dari gradient. Beberapa kelebihan dari Adam adalah bahwa ukuran parameter dari parameter tunggal tidak berubah untuk penskalaan gradien, ukurannya kira-kira dibatasi oleh ukuran langkah *hyperparameter*, itu tidak memerlukan tujuan stasioner, ia bekerja dengan gradien yang jarang, dan ia secara alami melakukan bentuk ukuran step annealing [38].

Pada adam, *learning rate* yang beradaptasi berdasarkan parameter θ . Perubahan *learning rate* dihitung dari estimasi *momentum*. *Momentum* adalah nilai yang dihitung berdasarkan arah dari pelatihan sebelumnya. Dalam Adam, kuadrat gradien disimpan dan disimpan juga rata-rata gradien sebelumnya untuk perhitungan *momentum* [39].

2.1.10. Loss Function Mean Square Error

Loss function adalah fungsi yang menghitung perbedaan antara output aktual dan output yang diprediksi. *Loss function* merupakan fungsi yang menggambarkan kerugian yang terkait dengan semua kemungkinan yang dihasilkan oleh sebuah model. *Loss function* ini bekerja ketika model

pembelajaran memberikan kesalahan yang harus diperhatikan. Loss function yang baik memberikan nilai error yang rendah[40]. Pada penelitian ini akan menggunakan Loss Function yang disebut *Mean Square Error* (MSE). Nilai *Mean Square Error* dinyatakan sebagai berikut[41]:

$$MSE = \sqrt{\frac{\sum_{i=1}^n (x_i - E(i))^2}{n}} \quad (19)$$

Keterangan :

MSE : *Mean Square Error*

n : Jumlah data tahap pelatihan

x_i : Parameter input

$E(i)$: target output

Mean Square Error adalah indikator ukuran secara keseluruhan untuk melihat hasil running dari pelatihan telah berhasil atau tidak Al-Tabtabai dalam [41]. Pada tahapan ini mempunyai fungsi untuk mengevaluasi kinerja model selama proses pelatihan. Pada umumnya, tidak ada manfaat untuk melatih model ANN di luar titik dimana kinerjanya berhenti membaik pada saat rangkaian pengamatan uji. Suatu tahapan pelatihan dalam satu set proses pelatihan disebut *epoch*. Kesalahan error diukur untuk setiap running dari jumlah *epoch* yang dipilih dan hasilnya ditunjukkan sebagai kurva kinerja. Pelatihan harus dihentikan saat *mean square error* tidak berubah selama periode tertentu sesuai dengan *epoch* target. Hal ini dilakukan untuk menghindari *overfitting*, dalam hal ini jaringan merekam nilai pelatihan dan tidak dapat membuat prediksi saat ada contoh yang tidak diketahui[41]

2.1.11. Performa Peramalan

a. *Mean Absolute Error* (MAE)

Mean Absolute Error (MAE) adalah salah satu metode yang digunakan untuk mengukur tingkat keakuratan model peramalan. Nilai MAE menunjukkan rata-rata kesalahan (*error*) absolut antara hasil peramalan/prediksi dengan nilai riil[42]. Berikut adalah rumus *Mean Absolute Error* (MAE).

$$MAE = \frac{1}{n} \sum_{i=1, n} |y - y'| \quad (20)$$

Keterangan:

MAE : *Mean Absolute Error* (MAE)

N : Jumlah Data

y' : Nilai sebenarnya

y : Nilai Hasil Peramalan

b. R^2 Score

Uji *koefisien determinasi* (R^2) digunakan untuk melihat seberapa besar variabel-variabel independen secara bersama mampu memberikan penjelasan mengenai variabel dependen dimana nilai R^2 berkisar antara 0 sampai 1 ($0 \leq R^2 \leq 1$). R^2 menunjukkan proporsi dari total variability yang dapat dijelaskan oleh model yang diperoleh. Jika model yang diperoleh sangat sesuai dengan data yang ada (artinya variabilitas totalnya kecil), maka R^2 akan mendekati 1 ini menunjukkan bahwa kontribusi variabel bebas terhadap variabel terikat secara simultan semakin kuat. Sebaliknya bila model memiliki kecocokan yang rendah dengan data maka R^2 akan mendekati 0 ini menunjukkan bahwa kontribusi variabel bebas terhadap variabel terikat secara simultan semakin lemah. Secara umum nilai $R^2 > 0.9$ mengindikasikan model yang bagus [43][44].

2.1.12. *Fully- Connected Layer*

Lapisan Fully-Connected adalah lapisan dimana semua neuron aktivasi dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya. Setiap aktivasi dari lapisan sebelumnya perlu diubah menjadi data satu dimensi sebelum dapat dihubungkan ke semua neuron di lapisan.

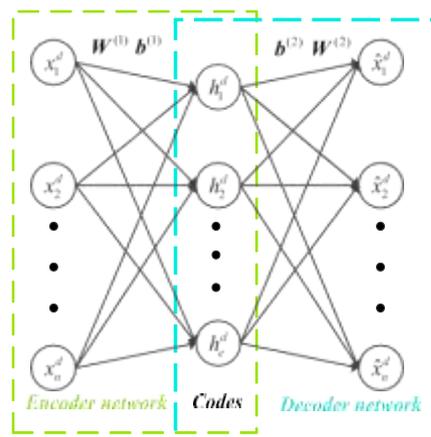
Lapisan *Fully-Connected* biasanya digunakan pada metode *Multi* lapisan *Perceptron* dan bertujuan untuk mengolah data sehingga bisa diklasifikasikan. Perbedaan antara lapisan *Fully-Connected* dan lapisan *Convolutional* biasa adalah neuron di lapisan convolutional terhubung hanya ke daerah tertentu pada input, sementara lapisan *Fully-Connected* memiliki neuron yang secara keseluruhan terhubung[45].

2.1.13. *Representation Learning Autoencoder*

Representation learning adalah metode untuk melakukan kompresi *feature vector* menggunakan neural network. proses melakukan kompresi disebut

encoding, hasil *feature vector* dalam bentuk terkompres disebut *coding*, proses mengembalikan hasil kompresi bentuk awal disebut *decoding*. *Neural network* yang mampu melakukan proses *encoding* disebut *encoder*, sedangkan *decoder* untuk proses *decoding* [11].

Contoh *representation learning* paling sederhana adalah *autoencoder* yaitu neural network yang dapat merepresentasikan data kemudian merekonstruksinya kembali. Jaringan autoencoder ini dilatih dengan metode *supervised learning*, tetapi tidak membutuhkan kelas dari data sebagai target. *Autoencoder* adalah model *neural network* yang memiliki masukan dan keluaran yang sama yang ilustrasinya ditunjukkan pada Gambar 2.11. *Autoencoder* memiliki bobot (*weight*) yang sama antara *encoder* dan *decoder*-nya, yang biasa disebut dengan *tied weight*. *Autoencoder* berusaha untuk mempelajari data masukan dan berusaha untuk merekonstruksi kembali masukan tersebut, tetapi dengan informasi yang lebih sedikit. Informasi ini biasa disebut sebagai *latent feature* [9][46].



Gambar 2. 11 Arsitektur Autoencoder

Fungsi *autoencoder* adalah untuk mengurangi dimensi dari fitur yang didapat dari suatu masukan atau bisa disebut dengan *dimensionality reduction*. Cara memanfaatkan *autoencoder* adalah dengan memotong jaringan pada bagian encodernya, kemudian *latent feature* yang dihasilkan diumpungkan ke classifier. *Autoencoder* juga dapat digunakan untuk *denoising* atau menghilangkan derau pada suatu data masukan. *Autoencoder* yang menghilangkan derau disebut juga dengan *denoising autoencoder* [9].

Metode paling umum yang digunakan untuk masalah *Dimensionality Reduction* adalah *Principal Component Analysis* (PCA). PCA cukup unggul

sebagai metode reduksi dimensi karena sederhana untuk diimplementasikan dan mampu mengeliminasi korelasi antar variabel masukan. Akan tetapi, PCA hanya mereduksi data secara linear, dan tidak memperhatikan hubungan antar data secara spasial [9].

2.1.14. Tools Python

Sering kali *Machine Learning* sulit untuk di pelajari karena sulitnya dalam memahami pengkodean yang dilakukan untuk mengenali pola pada dunia nyata, maka dari itu munculah banyak *library* atau pustaka yang memudahkan pengembang aplikasi dalam membuat program *machine learning*. *Library machine learning* berkualitas produksi biasanya ditulis untuk pengembangan dengan bahasa python dan C++. Pada penelitian ini dibutuhkan Jupyter Notebook, Tensorflow, Keras dan Python 3.7. Python versi 3.7 yang digunakan sebagai bahasa pemrograman dengan versi 3.7 yang cocok dengan Tensorflow dan Keras, Jupyter Notebook sebagai alat untuk mengolah data dan program, Tensorflow dan Keras sebagai *library Machine Learning* yang memudahkan dalam membuat model [47].

2.2. Tinjauan Studi

Beberapa penelitian yang sudah ada terkait dengan berat badan bayi lahir, adalah sebagai berikut:

1. Penelitian yang dilakukan oleh Hasbi Yasin dan Dwi Ispriyanti [1], dengan judul “Klasifikasi Data Berat Bayi Lahir Menggunakan *Weighted Probabilistic Neural Network* (WPNN)”. Pada penelitian ini di implementasikan algoritma *Probabilistic Neural Network* (PNN) yang merupakan algoritma klasifikasi dalam algoritma NN yang menggunakan fungsi probabilitik.yang memiliki kelebihan dapat mengatasi waktu pelatihan yang lama dan tidak membutuhkan dataset yang besar dalam tahap pembelajarannya. Data yang digunakan dalam penelitian ini merupakan data sekunder yang berasal buku status pasien ibu melahirkan di rumah sakit islam sultan agung semarang sebanyak 98 data. Fitur-fitur yang dipakai yang mempengaruhi berat lahir yaitu umur ibu, lama gestasi,

tinggi badan, berat badan, tensi darah, hb, dan paritas. Kesimpulan Hasil penelitian menunjukkan bahwa pemodelan klasifikasi risiko BBLR dengan model WPNN mempunyai akurasi yang tinggi ditunjukkan dengan akurasi training sebesar 98,75% dan akurasi testing sebesar 94,44%.

2. Penelitian yang dilakukan oleh Jefa [7], dengan judul “ Prediksi Bayi Lahir secara Prematur dengan Menggunakan Metode C.45 Berbasis *Particle Swarm Optimization* Pada Klinik Umi”. Pada penelitian ini di implementasikan algoritma C.45 dan C.45 berbasis *particle swarm optimization* untuk mendapatkan rule dalam memprediksi kelahiran bayi prematur yang memberika nilai akurasi yang lebih akurat. Data yang digunakan dalam penelitian ini merupakan data primer klinik persalinan di daerah tangerang sebanyak 157 data. Fitur-fitur yang dipakai dalam penelitian ini adalah ibu meroko, sering mengalami keputihan, riwayat pernah keguguran, riwayat darah tinggi, dan usia ibu. Kesimpulan Hasil penelitian menunjukkan bahwa pemodelan klasifikasi kelahiran bayi premature dengan dua model yaitu algoritma c.45 menghasilkan nilai akurasi sebesar 94,30% dan AUC 0,986 sedangkan model kedua algoritma C.45 berbasis PSO nilai akurasi sebesar 97,91% dan nilai AUC sebesar 0,997.
3. Penelitian yang dilakukan oleh Stefan Kuhle,dkk [3], dengan judul “ *Comparison of logistic regression with machine learning methods for the prediction of fetal growth abnormalities: a retrospective cohort study*”. Pada penelitian ini di implementasikan perbandingan algoritma *Logistic regression* dengan algoritma klasifikasi mesin learning lainnya untuk memprediksi kelainan pertumbuhan janin yaitu berat bayi lahir rendah (BBLR) dan berat bayi lahir berlebih (BBLB) pada wanita *primipara* dan *multipara*. Data yang digunakan dalam penelitian ini adalah data bayi tunggal yang lahir yang diperoleh dari basis data perinata nova scotia atlee sebanyak 30.705 data. Fitur-fitur yang dipakai dalam penelitian ini dilihat dari beberapa faktor yaitu faktor *socio demographics* yaitu usia ibu, status pernikahan, pendapatan, tempat. Faktor resiko kehamilan yaitu merokok sebelum hamil, BMI sebelum kehamilan, *Hipertensi* dan

diabetes . Faktor riwayat kehamilan yaitu, diabetes *gestasional*, bayi premature <2500 g, bayi berlebih >4080 g, sesar dan kematian neonatus. Faktor kehamilan saat ini yaitu jenis kelamin janin, berat badan, meroko selama kehamilan, penggunaan zat , diabetes, hipertensi dan gangguan kejiwaan. Kesimpulan hasil penelitian ini prediksi kelainan pertumbuhan janin berdasarkan faktor *sosiodemografi* dan informasi klinis memiliki nilai terbatas untuk wanita *primipara*, tetapi akurasi prediksi cukup untuk wanita *multipara* sebelum kehamilan dan pada usia kehamilan 26 minggu. Metode pembelajaran mesin yang digunakan dalam penelitian ini tidak menawarkan kelebihan dibandingkan logistik regresi dalam prediksi kelainan pertumbuhan janin. Akurasi prediksi untuk BBLR dan BBLB berdasarkan maternal di peroleh akurasi buruk untuk wanita primipara dan cukup untuk wanita multipara diperoleh nilai AUC sebesar 60 – 75 %. Meroko, bayi BBLR sebelumnya dan berat kehamilan diidentifikasi sebagai prediktor utama yang mempengaruhi BBLR, sementara BMI sebelum kehamilan, berat kehamilan dan berat bayi sebelumnya >4080 g adalah prediktor utama yang mempengaruhi LGA.

4. Penelitian yang dilakukan oleh Yueh-Chin Cheng, dkk [8], dengan judul “*Efficient Fetal Size Classification Combined with Artificial Neural Network for Estimation of Fetal Weight*”. Tujuan dari penelitian ini adalah satu untuk menggunakan analisis korelasi untuk memvalidasi signifikansi di antara parameter *Ultrasonografi* (USP). Kedua memanfaatkan algoritma K-means untuk mengklasifikasikan ukuran janin berkaitan dengan kelas *diskriminatif* untuk pelatihan JST untuk meningkatkan akurasi prediksi berat janin. Data yang digunakan adalah data pemeriksaan USG prenatal pada studi *retrospektif* yang dilakukan di national cheng rumah sakit universitas kung sebanyak 2127 data. Fitur-fitur yang dipakai dalam penelitian ini yaitu lingkar perut (AC), *diameter biparietal* (BPD), berat lahir (BW), panjang femur (FL), presentasi janin (FP), usia kehamilan (GA), lingkar kepala (HC), *diameter oksipitofrontal* (OFD) dan jenis kelamin. Kesimpulan Hasil penelitian menunjukkan bahwa prediksi berat janin dari model baru menunjukkan MAPE sebesar 5,26 4,14 % dan MAE

157,91 119,90 g. Perbandingan akurasi prediksi berat janin menunjukkan bahwa model baru secara signifikan mengungguli yang biasa digunakan.

5. Penelitian yang dilakukan oleh Ashley I.Naimi, dkk [48], dengan judul “*Machine Learning for Fetal Growth Prediction*”. Tujuan dari penelitian ini untuk memprediksi estimasi berat janin dan mengevaluasi bagaimana pengaruh status merokok terhadap seluruh distribusi berat janin lahir dengan menggunakan beberapa algoritma yaitu *generalized linear model* (GLM), *quantile regression* (QR), *random forests* (RF), *bayesian additive regression trees* (BRT), dan *generalized boosted models* (GBM). Dataset yang digunakan adalah data kehamilan dari kunjungan *ultrasonografi* pada *Magee Women’s Obstetric Maternal* sebanyak 31.948 dan di ambil sampel sebanyak 18.517. Serta dataset terpisah sebagai data testing dari *Obstetrical Determinants of Neonatal Survival* yaitu data bayi sebanyak 240 kehamilan yang berisiko tinggi. Fitur-fitur yang dipakai dalam penelitian ini terdiri dari beberapa faktor yaitu demografi , klinis ibu, janin dan neonatal diantaranya usia kehamilan saat lahir, berat lahir, ras, usia ibu, Apgar skor satu dan lima menit, jeni kelamin bayi, status ibu merokok dan tahun kelahiran. Kesimpulan hasil penelitian menunjukkan bahwa algoritma *quantile regression* memberikan hasil prediksi lebih baik dibandingkan dengan algoritma lain yang diujikan, model digeneralisasi model terbaik secara keseluruhan menggunakan standar berat lahir dengan nilai MAD (95% CI: 2.70, 5.47) dan hasil penelitian menunjukkan merokok selama kehamilan meningkatkan 3.84 kali lipat berat bayi lahir rendah. Kemudian dengan menggunakan standar berat janin yang benar rasio ini turun dan diperoleh nilai MAD (95% CI: 1.50, 1.81) dengan resiko merokok selama kehamilan meningkatkan 1.65 kali lipat berat bayi lahir rendah.

2.3. Tinjauan Obyek Penelitian

Pada penelitian ini objek yang dijadikan bahan penelitian adalah berat bayi lahir dengan dataset yang di ambil dari *Kaggle* pada link <https://www.kaggle.com/doyonghoon/baby-data>. Data yang digunakan adalah data dari *Doyonghoon* yaitu baby-data, dengan jumlah data 1236 *record* dan 23 atribut.

BAB III

METODOLOGI PENELITIAN

3.1. Jenis Penelitian

Terdapat dua jenis pendekatan penelitian yang secara umum dipakai dalam proses penelitian yaitu pendekatan kualitatif dan pendekatan kuantitatif. Dalam penelitian ini metode yang digunakan yaitu metode penelitian kuantitatif. Penelitian kuantitatif adalah penelitian yang menggunakan angket dan data-data yang berupa angka, tabulasi, perhitungan-perhitungan menggunakan sejumlah metode analisis matematik/statistik yang hasilnya menjadi dasar pijakan untuk mengambil keputusan atau kesimpulan. Data dari pendekatan kuantitatif lebih banyak berbentuk angka dan tabel. Sementara itu, tahap analisis hanya dapat dilakukan jika data telah terkumpul dengan lengkap dan tersaji dalam tabulasi yang siap diolah secara statistik Slevitch dalam [49].

Penelitian ini dilakukan dengan cara menguji kebenaran sebuah hipotesis dengan statistik yang melibatkan penyelidikan beberapa variabel dengan menggunakan tes tertentu dan menghubungkannya dengan masalah penelitian. Metode penelitian kuantitatif disebut juga sebagai metode *positivistic* karena berlandaskan pada *filsafat positivisme*. Metode ini sebagai metode ilmiah/*scientific* karena telah memenuhi kaidah-kaidah ilmiah yaitu konkrit/empiris, obyektif, terukur, rasional dan sistematis [50].

Tujuan dari penelitian ini adalah untuk mendapatkan model *Deep learning Autoencoder* yang memberikan hasil prediksi paling akurat dalam memprediksi berat badan lahir bayi dan sebagai pembanding digunakan metode PCA. Selain itu penelitian ini bertujuan mendapatkan metode normalisasi yang paling efektif untuk prediksi berat bayi lahir. Evaluasi model ditinjau dengan nilai Mean Squared Error (MSE), Mean Absolute Error (MAE) dan R^2 score yang memiliki nilai mendekati 1.

3.2. Metode Pengumpulan Data

Metode pengumpulan data untuk mendapatkan sumber data dibagi menjadi dua, yaitu data primer dan data sekunder. Data primer yaitu data yang

dikumpulkan pertama kali, dan untuk melihat apa yang sesungguhnya terjadi melalui observasi, *interview*, *kuesioner*, dll. Data sekunder adalah data yang telah dikumpulkan dan dianalisis oleh orang lain baik yang telah dipublikasikan maupun yang belum dipublikasikan, misalnya dari dokumentasi, literatur, buku, jurnal, dan informasi lainnya yang ada hubungannya dengan masalah yang diteliti.

Dalam penelitian ini metode pengumpulan data untuk mendapatkan sumber data yang digunakan adalah metode pengumpulan data sekunder. Data penelitian ini diambil dari data publik Kaggle pada link <https://www.kaggle.com/doyonghoon/baby-data>. Data yang digunakan adalah data dari *Doyonghoon* yaitu *baby-data*, dengan jumlah data 1236 *record* dan 23 atribut. Spesifikasi dataset yang digunakan dalam penelitian ini sebagai berikut.

Tabel 3.1
Penjelasan Atribut pada *dataset Baby-data*

Atribut	Keterangan
<i>Id</i>	Id responden
<i>Plurality</i>	Banyaknya janin
<i>Outcome</i>	Kelahiran hidup yang bertahan setidaknya 28 hari
<i>Date</i>	Tanggal lahir
<i>Gestation</i>	Jumlah hari kehamilan
<i>Sex</i>	Jenis kelamin
<i>Parity</i>	Anak ke-
<i>Race</i>	Ras Ibu
<i>Age</i>	Usia Ibu
<i>Ed</i>	Pendidikan Ibu
<i>Ht</i>	Tinggi badan Ibu
<i>Wt.1</i>	Berat badan Ibu sebelum kehamilan
<i>Drace</i>	Ras Ayah
<i>Dage</i>	Usia Ayah
<i>Ded</i>	Pendidikan Ayah

<i>Dht</i>	Tinggi badan Ayah
<i>Dwt</i>	Berat Badan Ayah
<i>Marital</i>	Status Pernikahan
<i>Inc</i>	Total pendapatan dalam peningkatan 2500
<i>Smoke</i>	Ibu meroko
<i>Time</i>	Berapa Lama berhenti meroko
<i>Number</i>	Jumlah kuantitas meroko dalam sehari
<i>Wt</i>	Berat badan bayi lahir

3.3. Instrumen Penelitian

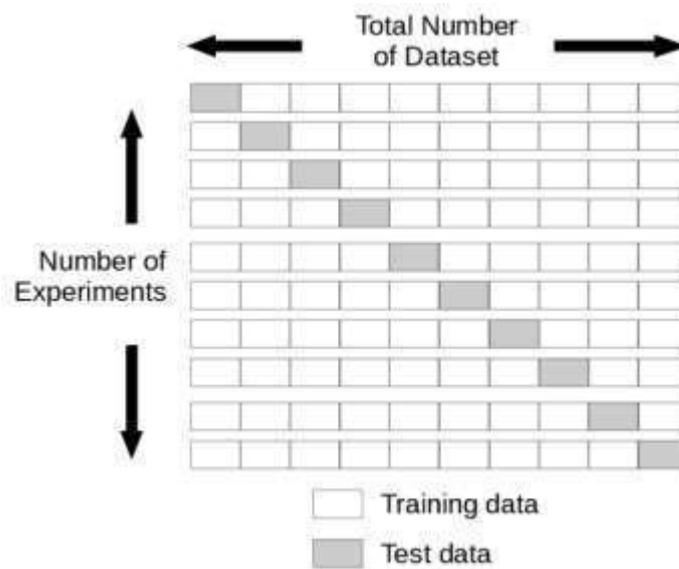
Beberapa hal penting yang menjadi instrumen pada penelitian ini sebagai berikut:

1. Penelitian ini menggunakan data sekunder berupa data kelahiran bayi dari data publik Kaggle.
2. Data disajikan dalam bentuk tabulasi sebanyak 1236 record dan 23 atribut. Data yang diambil adalah data kelahiran bayi pada tahun 2018 yang di peroleh dari data kaggle.
3. Algoritma yang digunakan pada penelitian ini adalah *Deep Learning Autoencoder*.
4. Pada penelitian ini dilakukan *feature selection* dengan *matrix correlation*. *Matrix correlation* berguna untuk menginvestigasi seluruh hubungan antar variabel numerik dalam dataset.
5. Pada penelitian ini dilakukan 2 metode normalisasi data yaitu *MinMaxScaler* dan *StandarScaler*.
6. Pada penelitian ini dilakukan 2 metode dimentionality reduction yang akan dibandingkan yaitu *feature reduction* PCA dan *feature extraction autoencoder*.
7. Perangkat lunak yang digunakan untuk menganalisis adalah *Python 3.7*.

3.4. Pemisahan Dataset Training dan Data Testing

Suatu cara yang handal untuk memvalidasi kinerja model *Machine Learning* (ML) adalah melatih model dengan data yang tersedia dan menilai kinerja klasifikasi atau prediksinya menggunakan data yang baru dikumpulkan atau dataset terpisah.

Pada penelitian ini data training dan testing akan dipisah dengan menggunakan *Cross validation 10 fold*. *10 fold CV* adalah salah satu K fold yang direkomendasikan untuk pemilihan model terbaik karena cenderung memberikan estimasi akurasi yang kurang bias di bandingkan dengan CV biasa, *leave-one-out cv* dan *bootstrap*. Dalam 10 fold cv, data dibagi menjadi 10 fold berukuran kira-kira sama, sehingga kita memiliki 10 subset data untuk mengevaluasi kinerja model atau algoritma. Untuk masing-masing dari 10 subset data tersebut, CV akan menggunakan 9 fold untuk pelatihan dan 1 fold untuk pengujian[51]. Seperti di ilustrasikan pada Gambar 3.1.

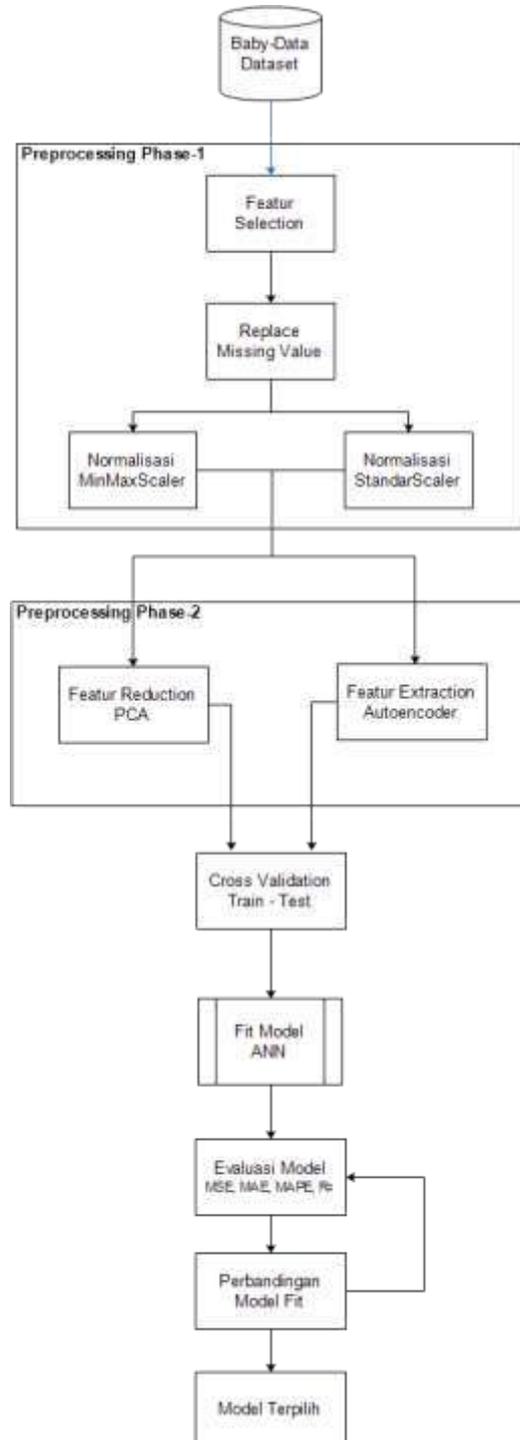


Gambar 3. 1 10 Fold *Cross Validation*

Sumber : www.researchgate.net

3.5. Langkah-langkah Penelitian

Penelitian ini dilakukan dalam beberapa tahapan penelitian. Langkah penelitian yang dilakukan dapat di gambarkan dalam kerangka pemikiran berikut.



Gambar 3. 2 Metodologi Penelitian

Sumber: (Penelitian, 2020)

Berdasarkan gambar langkah-langkah pemikiran diatas, dapat dijelaskan sebagai berikut:

a. *Data Preprocessing*

Pada penelitian ini dilakukan beberapa *preprocessing* data, yaitu dengan menggunakan *Feature Selection*, *Replace Missing*, Normalisasi dan *preprocessing data transformation* yaitu mengubah suatu data agar diperoleh data yang lebih berkualitas menggunakan *dimensionality reduction* PCA dan Autoencoder. Tahap *preprocessing* dilakukan untuk mempermudah dalam melakukan eksperimen terhadap model yang tepat. Selain itu tahap *preprocessing* dilakukan untuk mendapatkan *Mean Square Error* (MSE) yang lebih kecil. Berikut metode *preprocessing* yang digunakan dalam penelitian ini.

1. *Feature Selection*

Feature Selection adalah sebuah proses yang biasa digunakan pada *Machine Learning* dimana sekumpulan dari fitur yang dimiliki oleh data digunakan untuk pembelajaran algoritma. Seleksi fitur adalah salah satu faktor yang paling penting yang dapat mempengaruhi tingkat akurasi karena jika dataset berisi sejumlah fitur, dimensi dataset akan menjadi besar hal ini membuat rendahnya nilai akurasi [52]. Teknik seleksi fitur dilakukan untuk *decreasing the learning cost* atau penurunan biaya pembelajaran, *increasing the learning performance* atau meningkatkan kinerja pembelajaran dan *reducing irrelevant dimensions* atau mengurangi dimensi yang berlebihan Maimon [52].

Feature Selection memilih subset dari fitur yang ada tanpa transformasi karena tidak semua fitur/atribut *relevan* dengan masalah. Bahkan dari fitur tersebut mengganggu dan mengurangi akurasi. Fitur yang tidak terpakai tersebut harus dihapus untuk meningkatkan akurasi. *Feature Selection* dalam penelitian ini menggunakan metode *matrix correlation*. *Matrix correlation* berguna untuk menginvestigasi seluruh hubungan antar variabel numerik dalam dataset.

Preprocessing Feature Selection dalam penelitian ini adalah sebagai berikut:

1) Import dataset yang akan diuji.

Upload file dataset → cari file dataset → upload → read dataset

Tabel 3. 2
10 Record Dataset Baby

date	Plurality	Outcome	Gestation	Sex	Parity	...	Inc	Smoke	Time	Number	Wt
2003-11-11	5	1	284	1	1	...	1	0	0	0	120
2004-07-02	5	1	282	1	2	...	4	0	0	0	113
2004-04-24	5	1	279	1	1	...	2	1	0	1	128
2004-12-02	5	1	NaN	1	2	...	8	2	1	3	123
2003-11-25	5	1	282	1	1	...	1	1	0	3	108
2004-07-30	5	1	286	1	4	...	4	2	1	2	136
2003-12-19	5	1	244	1	4	...	NaN	0	0	0	138
2004-10-04	5	1	245	1	2	...	2	0	0	0	132
2003-08-11	5	1	289	1	3	...	2	0	0	0	120
2004-04-16	5	1	299	1	3	...	2	1	0	3	143

10 rows x 21 columns

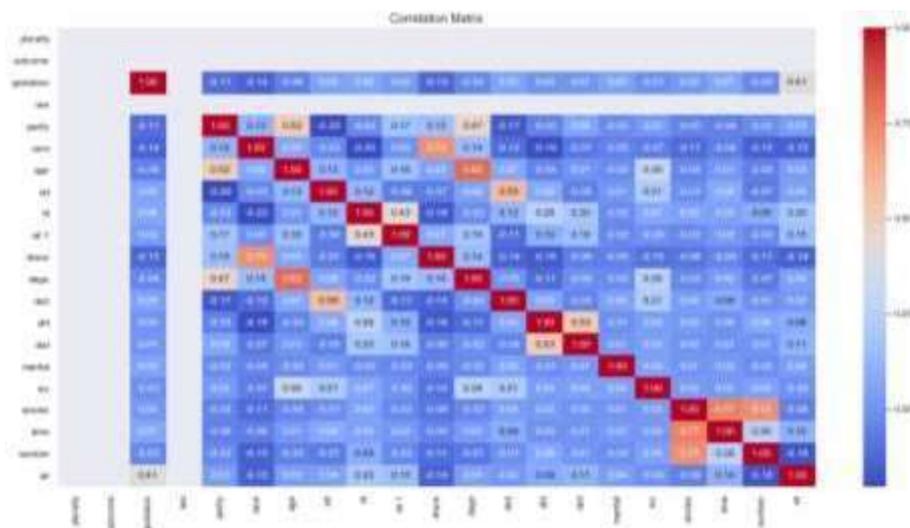
Pada Tabel 3.2 tampil isi 10 record dari dataset, dari record tersebut kita bisa mengetahui jumlah *attribute* terdapat 21 *attribute* dan 1 *attribute* sebagai id. Selain itu juga kita bisa mengetahui terdapat beberapa atribut yang berisi NAN yang menandakan adanya *missing value* (nilai yang hilang pada setiap attribute).

- 2) Merubah format wt (berat badan) dari ons ke kg.

Merubah format attribute berat bayi lahir (wt) dari satuan ons ke kilogram.

1 ons = 0.0283495 kg.

- 3) Memilih attribute berdasarkan *Matrix Correlation*



Gambar 3. 3 Tampilan *Correlation Matrix*

Sumber: (Penelitian, 2020)

Dari matriks korelasi ini dapat dilihat bahwa pada bagian diagonal, nilainya pasti =1 sebab variabel tersebut berkorelasi sempurna positif dengan dirinya sendiri. Nilai korelasi antara pasangan variabel pada dataset menunjukkan nilai dalam range -1 dan 1 (walaupun dengan satuan dan range berbeda). Dari Gambar 3.3 dapat diketahui bahwa attribute *plurality*, *outcome* dan *sex* tidak memiliki korelasi dengan attribute manapun sehingga attribute tersebut tidak *relevan* dan mengganggu nilai akurasi sehingga harus di hapus.

2. Replace Missing Value

Pada Tabel 3.2 Diketahui terdapat beberapa attribute yang mengandung nilai NaN ini menunjukkan adanya data yang hilang (*missing value*). *Missing value* di *replace* dengan metode mean untuk data continous dan metode median untuk data diskrit. Yakni dengan me *replace* semua nilai yang hilang dengan nilai rata-rata data tersebut dan juga nilai tengah data tersebut.

plurality	0
outcome	0
gestation	0
sex	0
parity	0
race	0
age	0
ed	0
ht	0
wt.1	0
drace	0
dage	0
ded	0
dht	0
dwt	0
marital	0
inc	0
smoke	0
time	0
number	0
wt	0

Gambar 3. 4 Cek *Missing Values* Sesudah *Replace*

Sumber: (Penelitian, 2020)

Pada Gambar 3.4 dapat terlihat bahwa *missing value* pada tiap-tiap atribut telah teratasi dengan baik dengan nilai missing value =0.

3. Normalisasi

Metode normalisasi yang digunakan dalam penelitian ini adalah *mentransformasi* data ke dalam range 0 dan 1. Metode – metode yang digunakan adalah *StandarScaler* dan *MinMaxScaler*.

1) *StandarScaler*

StandarScaler adalah teknik untuk mentransformasikan tiap fitur memiliki nilai rata-rata 0 dan variansi 1.

2) *MinMaxScaler*

MinMaxScaler berfungsi untuk mengubah data berada dalam rentang 0 sampai 1 [53].

4. Reduksi data

Setelah data bersih dari *missing value*, data terduplikasi dan data yang tidak konsisten dilakukan *preprocessing* phase 2 dengan penambahan metode *feature reduction* dan *feature extraction*. *Feature reduction* disini dengan penambahan metode *Principal Component Analysis* (PCA) untuk mereduksi variabel sehingga dari variabel yang ada pada dataset akan diperoleh variabel baru yang mewakili variabel asal. Sedangkan *feature extraction* disini dengan menambahkan metode *Autoencoder* untuk mengekstraksi fitur penting dari data lalu merekonstruksinya kembali.

b. Fit Model

Dataset diuji menggunakan algoritma *Deep learning Autoencoder* dengan beberapa model menggunakan *cross validation*

c. Evaluasi Model

Berdasarkan evaluasi pengujian akan diperoleh nilai akurasi, dimana evaluasi yang digunakan menggunakan empat *measurement* yaitu berdasar evaluasi MSE, MAE dan R^2 score.

d. Perbandingan Model Fit

Langkah selanjutnya yaitu perbandingan kinerja model deep learning Autoencoder dengan PCA berdasarkan teknik normalisasi data.

e. Model Terpilih

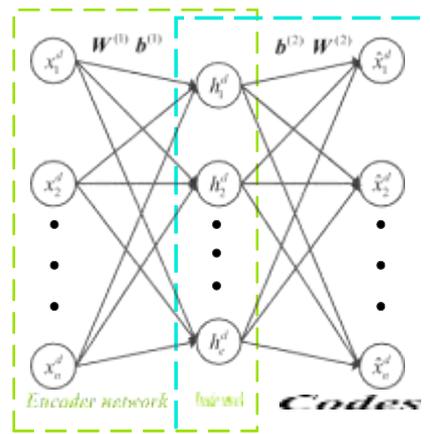
Setelah melakukan perbandingan maka didapat hasil dan model terbaik dengan nilai akurasi tertinggi untuk prediksi berat bayi lahir.

3.6. Evaluasi dan Validasi Hasil

Pada tahap ini akan dilakukan proses pengujian metode dan model yang diusulkan dengan mengevaluasi model *Deep Learning Autoencoder* dan sebagai pembandingan menggunakan PCA, serta metode normalisasi yang menghasilkan model prediksi yang dianggap paling optimal dalam prediksi berat lahir bayi berdasarkan nilai MSE, MAE terkecil dan R^2 score tertinggi.

3.7. Desain Eksperimen

Tahap eksperimen terdiri atas dua bagian yaitu, pertama membandingkan hasil prediksi berat bayi lahir antara model autoencoder dan PCA berdasarkan teknik scaling data *MinMaxScaler* dan *StandardScaler*. Kedua, variasi parameter untuk mencari hasil prediksi yang lebih baik.



Gambar 3. 5 Arsitektur *Autoencoder*

Pada Gambar 3.5 menunjukkan arsitektur yang di pakai dalam penelitian ini yaitu arsitektur *Deep learning Autoencoder*. Secara teori, pada metode ANN tidak ada aturan secara baku untuk menentukan arsitektur ANN yang optimal untuk diterapkan ke dalam sistem Kardan et al. dalam [54]. Sehingga pencarian arsitektur dan parameter pelatihan ANN harus dilakukan secara *trial* dan *error*. Ringkasan variasi arsitektur *Deep learning Autoencoder* jaringan pada eksperimen ini ditunjukkan pada Tabel 3.3 dan sebagai pembandingan PCA ditunjukkan pada Tabel 3.4.

Tabel 3. 3
Arsitektur Deep *Autoencoder* Yang Diujicobakan

Model	Normalisasi	Input Neuron	Encoder		Encoding	Decoder		Output Neuron
			Jumlah Hidden Layer	Jumlah Hidden Neuron		Jumlah Hidden Layer	Jumlah Hidden Neuron	
Model 1	MinMaxScaler	18	1	32	8	1	32	18
Model 2		18	2	32-16	8	2	16-32	18
Model 3		18	3	64-32-16	8	3	16-32-64	18
Model 4		18	4	128-64-32-16	8	4	16-32-64-128	18
Model 1	StandarScaler	18	1	32	8	1	32	18
Model 2		18	2	32-16	8	2	16-32	18
Model 3		18	3	64-32-16	8	3	16-32-64	18
Model 4		18	4	128-64-32-16	8	4	16-32-64-128	18

Tabel 3. 4
Arsitektur Deep *Learning Principal Componen Analysis* yang Diujicobakan

Model	Normalisasi	Feature Reduction	Jumlah		
			Input neuron	Hidden Layer	Hidden Neuron
Model 1	MinMaxScaler	PCA	17	1	16
Model 2			17	2	16-32
Model 3			17	3	16-32-8
Model 4			17	4	16-32-8-32
Model 1	Standar Scaler	PCA	17	1	16
Model 2			17	2	16-32
Model 3			17	3	16-32-8
Model 4			17	4	16-32-8-32

Untuk melakukan ujicoba pelatihan deep learning ANN langkah selanjutnya adalah menentukan parameter pelatihan. Pada penelitian ini, konfigurasi parameter yang di ujikan ditampilkan pada Tabel 3.5.

Tabel 3.5
Konfigurasi Parameter

Parameter	Nilai Parameter Yang Diujicobakan
<i>Optimizer</i>	Adam lr 0.001
<i>Momentum</i>	0.9
<i>Epoch</i>	45

<i>Batch Size</i>	12
<i>Aktivasi Hidden Layer</i>	<i>Relu</i>
<i>Aktivasi Hidden Layer Encoder</i>	<i>Custom_gelu</i>
<i>Aktivasi Hidden Layer Decoder</i>	<i>Relu</i>
<i>Aktivasi Output</i>	<i>Linear</i>

Parameter-parameter tersebut diujicobakan pada tiap arsitektur (model 1, model 2 , model 3 dan model 4) dengan ketentuan kombinasi seperti yang ditampilkan pada Tabel 3.3 dan Tabel 3.4.

Untuk mendapatkan output model terbaik semua model pelatihan trial yang diuji coba dalam penelitian ini dilatih dalam model yang dilandasi dengan algoritma pembelajaran *backpropagation*. Algoritma *backpropagation* melibatkan pengurangan kesalahan secara bertahap antara output model dan output target Rumelhart[41]. Oleh karena itu, perlu pengembangan input ke pemetaan output dengan meminimalkan fungsi kesalahan biaya (*loss function*) kuadrat rata-rata yang diukur dari serangkaian tahap pelatihan. Pada tahapan ini mempunyai fungsi untuk mengevaluasi kinerja model selama proses pelatihan apakah telah berhasil atau tidak. Pada penelitian ini nilai *loss function* yang digunakan adalah MSE (*Mean Square Error*) sedangkan pengukuran hasil performa peramalan pada penelitian ini menggunakan perhitungan MAE (*Mean Absolute Error*) dan R² Score.

Setelah diperoleh model dan metode yang menghasilkan prediksi berat lahir bayi optimal, untuk memastikan hasil dari pengujian model terpilih adalah yang terbaik diperlukan suatu perbandingan parameter yang lainnya. Dalam perbandingan parameter ini akan dibandingkan adalah nilai *learning rate*.

Tabel 3. 6

Parameter Yang Diujicobakan

Nilai Parameter				
Learning rate	0.001	0.002	0.003	0.004

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Eksperimen Dan Pengujian Model

Suatu metode untuk merepresentasikan data ke dalam bentuk yang membutuhkan memori lebih sedikit tanpa adanya kehilangan informasi. Selain metode *principal component analysis* dalam neural network terdapat metode *representation learning autoencoder*. *Autoencoder* yaitu neural network yang dapat merepresentasikan data kemudian merekonstruksi kembali.

Langkah-langkah pengujian model *Autoencoder* adalah sebagai berikut:

1. Pembuatan Model *Encoder*

Implementasi Arsitektur model pada Tabel 3.3 model encoder untuk input layer menggunakan input shape 18. Kemudian untuk variasi *hidden layer* menggunakan variasi dari 1 *hidden layer* sampai 4 *hidden layer* dengan activation *custom_gelu*. Semua layer merupakan *fully –connected layer*.

2. Pembuatan Model *Decoder*

Selanjutnya, buat sebuah model baru sebagai *decoder* yang bercabang dari model sebelumnya. Model ini akan mengambil input yang kemudian menggabungkan semua lapisan tengah dari encoder dan meneruskannya melalui bagian *encoding* dari *autoencoder* yang selanjutnya akan digunakan untuk melatih prediksi berdasarkan fitur output *encoder*.

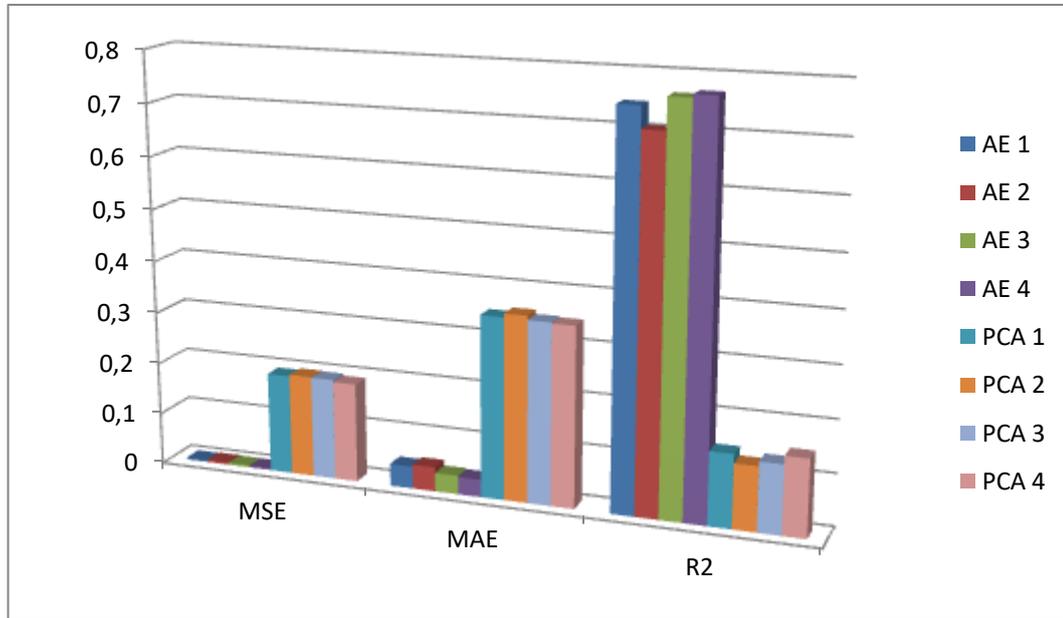
3. Pelatihan *autoencoder*

Selanjutnya yaitu melakukan proses pelatihan model menggunakan model yang telah dibuat. Dalam proses pelatihan model ini ditambahkan fungsi *callbacks* berupa *ModelCheckpoint()* untuk menyimpan model hasil pelatihan pada saat model memperoleh nilai validasi yang lebih bagus (*validation loss* lebih kecil).

4. Evaluasi Model

Untuk mengevaluasi kinerja model selama proses pelatihan apakah telah berhasil atau tidak dengan melihat nilai *loss function* MSE (*Mean Square Error*) dan pengukuran hasil performa peramalan pada penelitian menggunakan nilai MAE (*Mean Absolute Error*) dan R^2 Score.

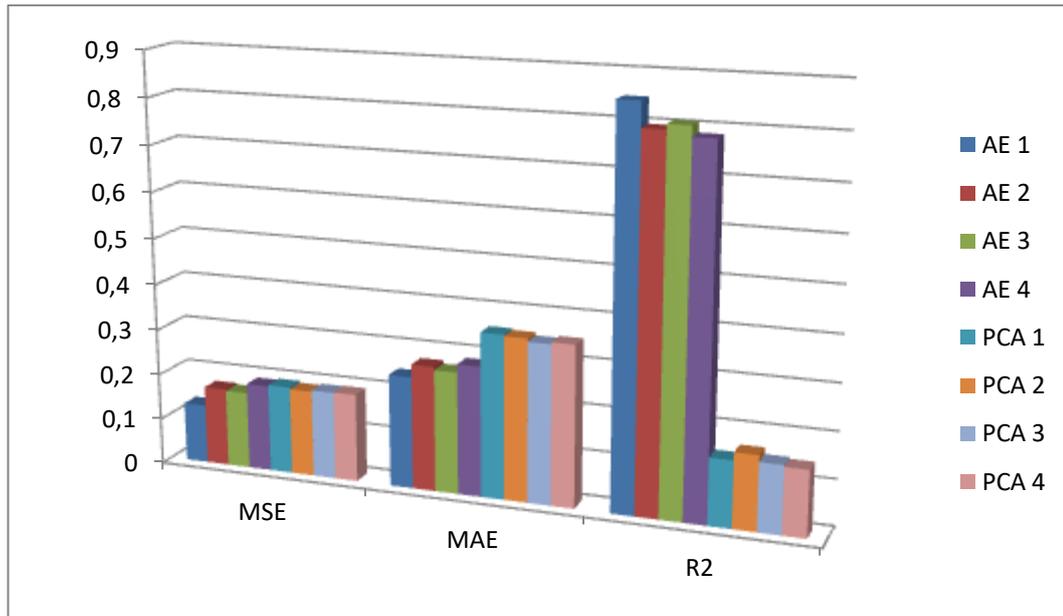
Berikut ini merupakan hasil pemodelan semua percobaan yang dilakukan pada arsitektur *Autoencoder* dengan mengambil best model dan juga PCA.



Gambar 4. 1 Perbandingan Akurasi validasi Untuk Beberapa Lapisan Tersembunyi Normalisasi *MinMaxScaler*

Sumber: (Penelitian, 2020)

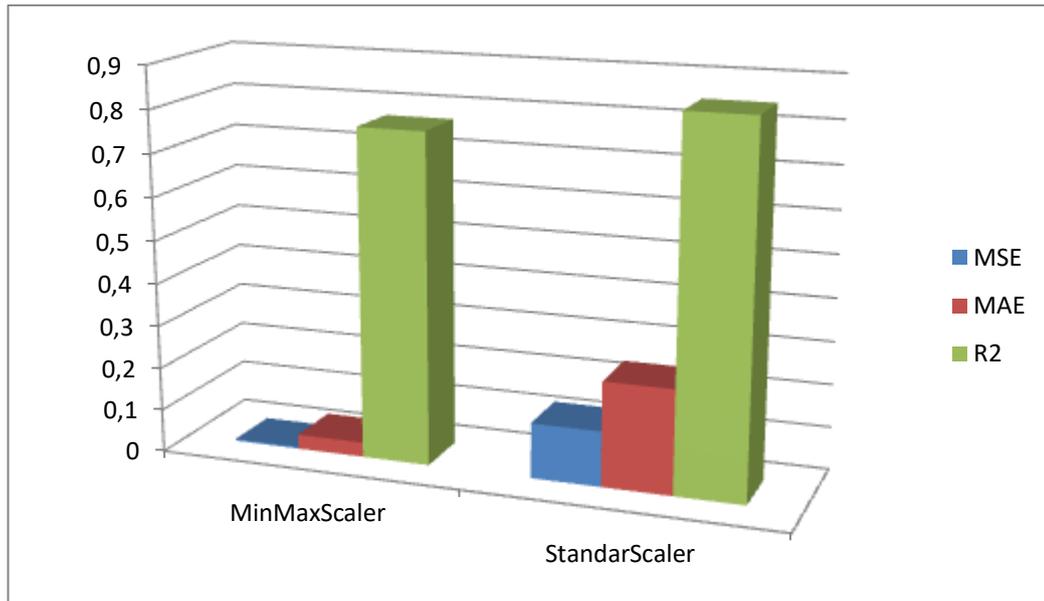
Pada eksperimen dengan variasi parameter jumlah hidden layer dan normalisasi *MinMaxScaler*, terlihat pada Gambar 4.1 bahwa dengan *Autoencoder*, jumlah hidden layer 4 sudah mendapatkan hasil akurasi terbaik, sementara dengan PCA hasil terbaik untuk jumlah hidden layer 4 juga. Meskipun demikian, hasil akurasi terbaik dengan *Autoencoder*, yaitu dengan MSE 0.004, MAE 0.034 dan R^2 0.769 masih jauh lebih tinggi dari akurasi terbaik dengan PCA, yaitu MSE 0.191, MAE 0.346 dan R^2 0.149. Berdasarkan Gambar 4.1 Dari hasil pengujian ke 4 model *Autoencoder* dan PCA dengan metode normalisasi *MinMaxScaler* menunjukkan bahwa variasi *hidden layer* sangat mempengaruhi performa hasil akurasi, dengan variasi *hidden layer* yang tepat akan memberikan hasil akurasi yang baik. Dari hasil evaluasi menunjukkan bahwa dengan menambah jumlah *hidden layer*, dapat meningkatkan performa hasil ANN.



Gambar 4. 2 Perbandingan Akurasi validasi Untuk Beberapa Lapisan Tersembunyi Normalisasi *StandarScaler*

Sumber: (Penelitian, 2020)

Pada eksperimen dengan variasi parameter jumlah *hidden layer* dan normalisasi *StandarScaler*, terlihat pada Gambar 4.2 bahwa dengan *Autoencoder*, jumlah *hidden layer* 1 sudah mendapatkan hasil akurasi terbaik, sementara dengan PCA hasil terbaik untuk jumlah *hidden layer* 2. Meskipun demikian, hasil akurasi terbaik dengan *Autoencoder*, yaitu dengan MSE 0.129, MAE 0.244 dan R^2 0.848 masih jauh lebih tinggi dari akurasi terbaik dengan PCA, yaitu MSE 0.188, MAE 0.353 dan R^2 0.162. Berdasarkan Gambar 4.2 Dari hasil pengujian ke 4 model *Autoencoder* dengan metode normalisasi *StandarScaler* menunjukkan bahwa variasi *hidden layer* sangat mempengaruhi performa hasil akurasi, dengan variasi *hidden layer* yang tepat akan memberikan hasil akurasi yang baik.



Gambar 4. 3 Perbandingan Akurasi Validasi *Autoencoder* Untuk Metode Normalisasi

Sumber: (Penelitian, 2020)

Perbandingan akurasi validasi *Autoencoder* dengan variasi normalisasi, yaitu pada Gambar 4.3 menunjukkan bahwa nilai terbaik adalah MSE 0.129, MAE 0.244 dan R^2 0.848, yaitu pada penggunaan metode normalisasi *StandarScaler*. Sementara untuk penggunaan normalisasi *MinMaxScaler*, nilai akurasi terbaik adalah MSE 0.004, MAE 0.034 dan R^2 0.769, berbeda sedikit dari akurasi untuk penggunaan *Autoencoder StandarScaler*.

Pada penelitian ini, terbukti bahwa metode normalisasi yang tepat dapat cukup efektif dalam memperbaiki akurasi validasi. Hasil penelitian menunjukkan bahwa metode normalisasi *StandarScaler* lebih baik dalam memprediksi berat bayi lahir.

4.2. Evaluasi dan Validasi Hasil

Dalam evaluasi hasil penelitian, dilakukan perbandingan nilai MSE, MAE dan R^2 berdasarkan jumlah *hidden layer* dan metode normalisasi yang digunakan. Setelah dilakukan pengujian pada setiap model yang diusulkan, langkah selanjutnya adalah melakukan perbandingan pada kinerja model regresi yang diujikan.

Tabel 4. 1
Evaluasi Hasil

<i>Feature Reduction</i>	<i>Hidden Layer</i>	<i>MinMaxScaler</i>			<i>StandarScaler</i>		
		MSE	MAE	R ²	MSE	MAE	R ²
<i>Autoencoder</i>	1	0.004	0,043	0,745	0,129	0,244	0,848
	2	0.005	0,047	0,704	0,171	0,272	0,795
	3	0.004	0,036	0,763	0,168	0,265	0,806
	4	0.004	0,034	0,769	0,189	0,283	0,784
<i>Principal Component Analysis</i>	1	0.193	0.349	0.141	0.192	0.356	0.145
	2	0.196	0.356	0.124	0.188	0.353	0.162
	3	0.195	0.348	0.133	0.191	0.346	0.148
	4	0.191	0.346	0.149	0.192	0.350	0.144

Sumber: (Penelitian, 2020)

Dari hasil-hasil eksperimen perbandingan penggunaan *Autoencoder* dan PCA serta metode normalisasi yang digunakan, secara umum terlihat bahwa penggunaan *Deep Learning Autoencoder* selalu mendapatkan akurasi yang lebih baik dibanding pada penggunaan PCA baik menggunakan normalisasi *MinMaxScaler* maupun *StandarScaler*. Meskipun telah dilakukan pengkondisian dengan beberapa variasi parameter, hasil *deep learning autoencoder* lebih baik. Hal ini dikarenakan *Deep Learning Autoencoder* dapat mampu mengatasi data yang bersifat *nonlinear*.

Berdasarkan hasil eksperimen pada Tabel 4.1, terlihat bahwa banyaknya lapisan *hidden layer* mempengaruhi terhadap hasil akurasi. Hal ini dilihat dari semua eksperimen, pada eksperimen dengan normalisasi *MinMaxScaler* semakin banyak jumlah *hidden layer* maka hasil akurasi semakin baik sedangkan pada eksperimen dengan normalisasi *StandardScaler* dengan jumlah layer sedikit memberikan hasil yang lebih baik. Sementara itu hasil akurasi terbaik yang diperoleh adalah dengan MSE 0.129, MAE 0.244 dan R² 0.848, yaitu untuk penggunaan 1 lapisan *hidden layer* dengan variasi 32 *hidden neuron encoder*, encoding 8 dan 32 *hidden neuron decoder* menggunakan teknik scaling data yaitu normalisasi *StandarScaler*.

4.3. Perbandingan Parameter ANN

Untuk memastikan hasil dari parameter ANN yang terpilih adalah yang terbaik diperlukan suatu perbandingan parameter yang lainnya. Dalam perbandingan parameter ini akan dibandingkan adalah nilai *learning rate*. Perbandingan kinerja model berdasarkan *learning rate* disajikan dalam tabel berikut ini.

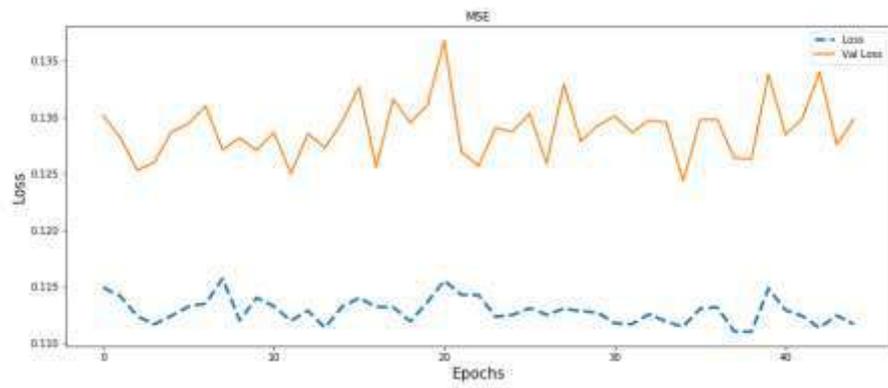
Tabel 4. 2
Perbandingan Performa Variasi *Learning Rate*

ANN Autoencoder			
Learning Rate	MSE	MAE	R ²
0.001	0.129	0.244	0.848
0.002	0.145	0.25	0.831
0.003	0.141	0.25	0.84
0.004	0.123	0.232	0.86

Sumber: (Penelitian, 2020)

Dilihat pada Tabel 4.2, penggunaan variasi *learning rate* yang digunakan dikatakan cukup berpengaruh terhadap hasil akhir peramalan, yaitu dilihat dari hasil test error yang cenderung stabil. Berdasarkan hasil implementasi dan hasil analisis pada penelitian ini, maka didapatkan model yang paling optimal adalah Deep Learning *Autoencoder* dngan *learning rate* 0.004 menghasilkan nilai MSE 0.123, MAE 0.232.

Untuk mengetahui erat atau tidaknya hubungan, arah hubungan dan berarti tidaknya hubungan antar variabel input independen dengan variabel output atau dependen dapat dilihat dari nilai R². Berdasarkan nilai R² 0.86 dapat dikatakan semua fitur secara bersama-sama sangat mempengaruhi dengan baik terhadap hasil akhir peramalan (R² score mendekati nilai 1). Berikut ini grafik hasil test error pada model optimal:



Gambar 4. 4 Grafik Test Error Model

Sumber: (Penelitian, 2020)

BAB V

PENUTUP

5.1. Kesimpulan

Penelitian ini dibuat untuk mencari model terbaik dalam *Deep learning Autoencoder* yang menghasilkan akurasi terbaik dalam memprediksi berat bayi lahir. Selain itu juga bertujuan untuk mengetahui metode normalisasi apa yang paling efektif untuk prediksi berat lahir bayi serta menganalisa hubungan antar atribut terhadap hasil prediksi.

Berdasarkan penelitian yang dilakukan penulis, maka dapat diambil kesimpulan sebagai berikut:

1. Dataset yang digunakan adalah data baby yang di ambil dari *kaggle*. Agar data siap digunakan untuk proses modeling dilakukan *preprocessing* data terlebih dahulu, yaitu dengan 2 tahapan *preprocessing* data. Tahap *preprocessing* pertama melakukan *feature selection* menggunakan metode *matrix correlation*, *replace missing value* dengan metode *mean* dan *median*, dan normalisasi data dengan metode *MinMaxScaler* dan *StandarScaler*. Tahap kedua melakukan *feature extraxtion* dengan metode *Autoencoder* dan dibandingkan dengan *feature reduction* dengan metode *Principal Component Analysis* (PCA).
2. Dalam memprediksi berat bayi lahir algoritma yang digunakan pada penelitian ini adalah *Deep Learning Autoencoder* dan dibandingkan dengan metode reduction yaitu PCA dengan memvariasikan jumlah *hidden layer*, *hidden neuron* dan *hyperparameternya* serta metode normalisasi.
3. Hasil evaluasi dan validasi yang dilakukan dengan mencari nilai *Mean Squared Error* (MSE), *Mean Absolute Error* (MAE) terkecil dan nilai R^2 score yang mendekati 1, menunjukkan dari seluruh pengujian model yang di ujikan didapatkan bahwa model *deep learning autoencoder* dapat memberikan hasil akurasi yang lebih baik dibandingkan dengan PCA untuk penggunaan 1 lapisan *hidden layer* dengan variasi 32 *hidden neuron encoder* , encoding 8 dan 32 *hidden neuron decoder* menggunakan teknik scaling data

4. yaitu normalisasi *StandarScaler*. Menghasilkan nilai MSE 0.129, MAE 0.244 dan R^2 0.848.
5. Untuk memastikan hasil dari parameter *Deep Learning Autoencoder* yang terpilih adalah yang terbaik diperlukan suatu perbandingan parameter yang lainnya. Model terpilih hasil evaluasi dan validasi model kemudian di uji berdasarkan perbandingan nilai learning rate. Hasil implementasi dan hasil analisis pada pengujian ini maka didapatkan model yang paling optimal menggunakan *learning rate* 0.004 dengan menghasilkan MSE 0.123, MAE 0.232 dan R^2 0.86.
6. Berdasarkan penelitian yang dilakukan maka hipotesis yang dapat diterima adalah H_0 , Algoritma *Deep Learning Autoencoder* dan metode normalisasi yang tepat dapat menangani masalah prediksi berat bayi lahir dan memberikan hasil akurasi yang baik.
7. Berdasarkan nilai R^2 score 0.86 dapat dikatakan bahwa 86 % semua fitur secara bersama-sama sangat mempengaruhi dengan baik terhadap hasil akhir peramalan dan sebesar 14 % dipengaruhi oleh faktor lain di luar penelitian ini.

5.2. Saran

Hasil penelitian ini menunjukkan bahwa algoritma *Deep Learning Autoencoder* dengan arsitektur dan parameter yang tepat sangat baik untuk memprediksi berat lahir bayi, namun untuk penelitian selanjutnya agar penelitian ini dapat ditingkatkan dan dikembangkan, berikut adalah saran-saran yang diusulkan:

1. Menambahkan jumlah data yang lebih besar dan atribut di luar penelitian ini yang dapat mempengaruhi berat bayi lahir, sehingga hasil pengukuran yang akan didapatkan lebih baik lagi.
2. Menggunakan *preprocessing* data dengan normalisasi data yang lain seperti *RobustScaler*.
3. Melakukan pengembangan dengan *featur selection* lainnya seperti *genetic algorithm*, *chi square* dan metode *feature selection* lainnya. Untuk menyeleksi atribut yang berpengaruh kuat, sehingga atribut yang dipakai

hanya sedikit namun tidak mengurangi akurasi dari algoritma yang digunakan.

4. Melakukan variasi-variasi lain baik dalam *hidden layer*, *hidden neuron* ataupun *hyperparameter* lainnya. Untuk mendapatkan model paling optimal.

DAFTAR REFERENSI

- [1] H. Yasin and D. Ispriyansti, “Klasifikasi Data Berat Bayi Lahir Menggunakan Weighted Probabilistic Neural Network (WPNN) (Studi Kasus di Rumah Sakit Islam Sultan Agung Semarang),” *Media Stat.*, vol. 10, no. 1, p. 61, 2017.
- [2] D. D. Kusumawati, R. Septiyaningsih, and Kania, “Faktor-Faktor Ibu yang Mempengaruhi Kejadian Bayi Berat Lahir Rendah (BBLR),” *J. Kesehat. Al-Irsyad*, vol. 9, no. 2, pp. 8–16, 2016.
- [3] S. Kuhle *et al.*, “Comparison of logistic regression with machine learning methods for the prediction of fetal growth abnormalities: A retrospective cohort study,” *BMC Pregnancy Childbirth*, vol. 18, no. 1, pp. 1–9, 2018.
- [4] D. I. Anggraini and S. Septira, “Nutrisi bagi Bayi Berat Badan Lahir Rendah (BBLR) untuk Mengoptimalkan Tumbuh Kembang Nutrition for Low Birth Weight Infant to Optimize Infant Growth and Development,” vol. 5, no. September, pp. 151–155, 2016.
- [5] R. D. Nindrea, “Perbedaan Taksiran Berat Badan Janin Menurut Perhitungan Formula Berat Badan Lahir Bayi,” *J. Ipteks Terap.*, vol. 11, no. 1, p. 36, 2017.
- [6] A. Puspita and M. Wahyudi, “Algoritma C4.5 Berbasis Decision Tree untuk Prediksi Kelahiran Bayi Prematur,” *Konf. Nasional Ilmu Pengetah. dan Teknol.*, vol. 1, no. 1, pp. 97–102, 2015.
- [7] Jefi, “Prediksi Bayi Lahir Secara Prematur Dengan Menggunakan Metode C . 45 Berbasis Particle Swarm Optimization Pada Klinik Umi,” vol. 8, no. 2, pp. 1–7, 2019.
- [8] Y. C. Cheng, G. L. Yan, Y. H. Chiu, F. M. Chang, C. H. Chang, and K. C. Chung, “Efficient fetal size classification combined with artificial neural network for estimation of fetal weight,” *Taiwan. J. Obstet. Gynecol.*, vol. 51, no. 4, pp. 545–553, 2012.
- [9] Y. Kristian, I. K. E. Purnama, E. H. Sutanto, L. Zaman, E. I. Setiawan, and M. H. Purnomo, “Klasifikasi Nyeri pada Video Ekspresi Wajah Bayi Menggunakan DCNN Autoencoder dan LSTM,” *J. Nas. Tek. Elektro dan*

- Teknol. Inf.*, vol. 7, no. 3, pp. 308–316, 2018.
- [10] J. T. Informasi *et al.*, “OPTIMASI PARAMETER ARTIFICIAL NEURAL NETWORK DENGAN,” vol. 11, pp. 196–209, 2015.
- [11] J. W. G. Putra, “Pengenalan Konsep Pembelajaran Mesin dan Deep Learning,” pp. 1–235, 2019.
- [12] M. R. Fauzi *et al.*, “MODEL PREDIKSI CANCER MENGGUNAKAN AUTOENCODER,” vol. 3, no. 1, 2019.
- [13] S. Prodi, T. Informatika, F. Informatika, and U. Telkom, “DENOISING SINYAL EKG MENGGUNAKAN DEEP NEURAL NETWORK DENGAN STACKED DENOISING AUTOENCODERS ECG SIGNAL DENOISING USING DEEP NEURAL NETWORK WITH STACKED DENOISING AUTOENCODERS,” vol. 4, no. 3, pp. 5024–5030, 2017.
- [14] A. Ambarwari, Q. J. Adrian, and Y. Herdiyen, “Analisis Pengaruh Data Scaling Terhadap Performa Algoritme Machine Learning untuk Identifikasi Tanaman,” vol. 1, no. 10, pp. 117–122, 2020.
- [15] E. Prasetyo, “REDUKSI DIMENSI SET DATA DENGAN DRC PADA METODE KLASIFIKASI SVM DENGAN UPAYA PENAMBAHAN KOMPONEN KETIGA,” pp. 293–300, 2014.
- [16] F. Finandakasih, S. A. Rosmah, and M. A. Tiro, “Model Prediksi Berat Lahir Bayi Berdasarkan Faktor Pengaruhnya Di Puskesmas Kaluku Bodoa,” *Pros. Semin. Nas. Variansi*, vol. 2018, pp. 162–174, 2018.
- [17] F. Wahyuni, R. Purwana, D. G. Matondang, D. D. Keperawatan, and I. K. Helvetia, “HUBUNGAN LAMA MENGIKUTI BABY SPA DENGAN KENAIKAN BERAT BADAN BAYI USIA 0-6 BULAN DI YAYASAN IBU KOMPLEKS MEDAN PERMAI KOTA MEDAN TAHUN 2018 Dosen S1 Kebidanan STIKES Senior Medan , Indonesia Alumni D4 Kebidanan , Institut Kesehatan Helvetia Background ; B,” vol. 2, no. 2, 2020.
- [18] D. A. WIBOWO, “Pengaruh Terapi Massage Terhadap Peningkatanberat Badan Bayi Prematur Di Ruang Perinatologi Rumah Sakit Umum Daerah Tasikmalaya,” *J. Kesehat. Bakti Tunas Husada J. Ilmu-ilmu Keperawatan, Anal. Kesehat. dan Farm.*, vol. 17, no. 1, p. 41, 2017.

- [19] N. Nurhudayaen, R. Majid, and A. ainurafiq, “Model Prediksi Berat Lahir Bayi Berdasarkan Berat Badan Ibu Sebelum Hamil Dan Pertambahan Berat Badan Pertrimester Di Wilayah Kerja Puskesmas Puuwatu Tahun 2015-2016,” *J. Ilm. Mhs. Kesehat. Masy. Unsyiah*, vol. 2, no. 6, p. 185487, 2017.
- [20] Y. Haryanti, E. Frelestanty, and R. Amartani, “Analisis Kejadian Rupture Perineum Persalinan Normal pada Ibu Primigravida di RSUD Ade Muhammad Djoen Sintang Tahun 2018,” *Pros. Semin. Nas. Unimus*, vol. 1, no. 0, pp. 250–255, 2018.
- [21] F. Afifah, “Bayi besar dalam kandungan sangat berisiko, bumil perlu tahu apa saja penyebabnya,” *theAsianparent Indonesia*, 2020. [Online]. Available: <https://id.theasianparent.com/bayi-besar-dalam-kandungan>. [Accessed: 18-Jul-2020].
- [22] S. Banu, “Batas usia ideal untuk hamil,” *theAsianparent Indonesia*, 2020. [Online]. Available: <https://id.theasianparent.com/batas-usia-ideal-hamil>. [Accessed: 18-Jul-2020].
- [23] V. Pinontoan and S. Tombokan, “Hubungan Umur Dan Paritas Ibu Dengan Kejadian Bayi Berat Lahir Rendah,” *J. Ilm. Bidan*, vol. 3, no. 1, p. 90765, 2015.
- [24] F. Monita, D. Suhaimi, and Y. Ernalia, “Hubungan usia, jarak kelahiran dan kadar hemoglobin ibu hamil dengan kejadian berat bayi lahir rendah di rsud arifin achmad provinsi riau,” *Jom FK*, vol. 3, no. No. 1, pp. 1–17, 2016.
- [25] N. O. Rosdianto, H. Herman, V. Murniati, D. Orthopedi, F. Kedokteran, and U. Padjadjaran, “HUBUNGAN ANTARA PENAMBAHAN BERAT BADAN IBU SELAMA HAMIL DENGAN,” vol. 5, no. 4, pp. 317–323, 2019.
- [26] I. B. U. Hamil, D. I. Bps, T. Y. Way, H. Bandar, L. Tahun, and A. Mariza, “HUBUNGAN PENDIDIKAN DAN SOSIAL EKONOMI DENGAN KEJADIAN ANEMIA PADA,” vol. 10, no. 1, pp. 5–8, 2016.
- [27] A. Geron, *Hands-On Machine Learning With Scikit-Learn & Tensor Flow*. 2017.
- [28] R. E. Neapolitan and X. Jiang, *Artificial Intelligence With an Introduction*

to *Machine Learning*. 2019.

- [29] M. Azhar, E. Riksakomara, and A. P. Terkait, “Peramalan Jumlah Produksi Ikan dengan Menggunakan,” vol. 6, no. 1, pp. 142–148, 2017.
- [30] A. Sudarsono, “Jaringan Syaraf Tiruan Untuk Memprediksi Laju Pertumbuhan Penduduk Menggunakan Metode,” *Media Infotama*, vol. 12, no. 1, pp. 61–69, 2016.
- [31] E. S. Novianto *et al.*, “Studi Penerapan ANN (Artificial Neural Network) Untuk Menghilangkan Harmonisa Pada Gedung Pusat Komputer Abstrack Gambar 2 Bentuk Gelombang Harmonisa (Dugan dan Mc Granaghan , 1996 dalam Sungkowo H , 2013) Sedangkan Untuk Batas distorsi tegangan yang ,” vol. 3, no. 2, pp. 1–6, 2016.
- [32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning (Adaptive Computation and Machine Learning series)*. 2016.
- [33] P. H. K, “Menilik Activation Function,” *Medium*, 2017. [Online]. Available: <https://medium.com/@opam22/menilik-activation-functions-7710177a54c9>. [Accessed: 22-Jul-2020].
- [34] Y. Yu, K. Adu, N. Tashi, P. Anokye, X. Wang, and M. A. Ayidzoe, “RMAF: Relu-Memristor-Like Activation Function for Deep Learning,” *IEEE Access*, vol. 8, pp. 72727–72741, 2020.
- [35] C. Yu, Z. Su, and J. Province, “S g e l u (sgelu),” 2019.
- [36] Z. Huang, X. Du, and L. Chen, “Convolutional Neural Network Based on Complex Networks for Brain Tumor Image Classification With a Modified Activation Function,” pp. 89281–89290, 2020.
- [37] A. Nur, A. Thohari, and G. B. Hertantyo, “Implementasi Convolutional Neural Network untuk Klasifikasi Pembalap MotoGP Berbasis GPU,” pp. 50–55, 2018.
- [38] D. P. Kingma and J. L. Ba, “A : a m s o,” pp. 1–15, 2015.
- [39] A. Santoso and G. Ariyanto, “IMPLEMENTASI DEEP LEARNING BERBASIS KERAS UNTUK PENGENALAN WAJAH,” *Emit. J. Tek. Elektro*, 2018.
- [40] K. Janocha and W. M. Czarnecki, “On loss functions for deep neural networks in classification,” *Schedae Informaticae*, 2016.

- [41] A. Y. Prathama, A. Aminullah, A. Saputra, D. Teknik, and U. G. Mada, "UNTUK PENENTUAN PROSENTASE BOBOT PEKERJAAN PADA RUMAH SAKIT PRATAMA," vol. 7, 2017.
- [42] J. T. Informatika and U. P. Ronggolawe, "PENERAPAN METODE MEAN ABSOLUTE ERROR (MEA) DALAM ALGORITMA REGRESI LINEAR UNTUK PREDIKSI PRODUKSI PADI," no. 1, pp. 78–83, 2019.
- [43] A. Rachman, "MODEL PERAMALAN KONSUMSI BAHAN BAKAR JENIS PREMIUM DI INDONESIA DENGAN," pp. 166–176, 2014.
- [44] N. Thaharah, "PENGARUH MEKANISME CORPORATE GOVERNANCE DAN KINERJA KEUANGAN TERHADAP NILAI PERUSAHAAN LQ 45," vol. 5, 2016.
- [45] K. P. Danakusumo, "IMPLEMENTASI DEEP LEARNING MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI CITRA CANDI BERBASIS GPU," 2017.
- [46] H. Shao, H. Jiang, H. Zhao, and F. Wang, "A novel deep autoencoder feature learning method for rotating machinery fault diagnosis," *Mech. Syst. Signal Process.*, vol. 95, pp. 187–204, 2017.
- [47] D. Sarkar, R. Bali, and T. Ghosh, "Transfer Learning Fundamentals," in *Hands-on transfer learning with Python: implement advanced deep learning and neural network models using TensorFlow and Keras*, 2018.
- [48] A. I. Naimi, R. W. Platt, and J. C. Larkin, "Machine Learning for Fetal Growth Prediction," *Epidemiology*, vol. 29, no. 2, pp. 290–298, 2018.
- [49] S. E. Zaluchu, "Strategi Penelitian Kualitatif dan Kuantitatif Di Dalam Penelitian Agama," *Evang. J. Teol. Injili dan Pembn. Warga Jemaat*, vol. 4, no. 1, p. 28, 2020.
- [50] Sugiyono, "Metode Penelitian Kuantitatif, Kualitatif dan R&D," in *ke-26*, 2018.
- [51] L. Nilawati, "Penilaian Apartemen Pada Perusahaan Konsultan Properti Menggunakan Metode Naïve Bayes," vol. 4, no. 2, pp. 114–123, 2020.
- [52] M. F. Nugroho and S. Wibowo, "Fitur Seleksi Forward Selection Untuk Menentukan Atribut Yang Berpengaruh Pada Klasifikasi Kelulusan Mahasiswa Fakultas Ilmu Komputer UNAKI Semarang Menggunakan

- Algoritma Naive Bayes,” *J. Inform. Upgris*, vol. 3, no. 1, pp. 63–70, 2017.
- [53] N. Chamidah, Wiharto, and U. Salamah, “Pengaruh Normalisasi Data pada Jaringan Syaraf Tiruan Backpropagasi Gradient Pengaruh Normalisasi Data pada Jaringan Syaraf Tiruan Backpropagasi Gradient Descent Adaptive Gain (BPGDAG) untuk Klasifikasi,” *ITSMART*, vol. Vol 1. No, no. September 2018, pp. 28–33, 2012.
- [54] B. Badieah, R. Gernowo, and B. Surarso, “Metode Jaringan Syaraf Tiruan Untuk Prediksi Performa Mahasiswa Pada Pembelajaran Berbasis Problem Based Learning (PBL),” *J. Sist. Inf. Bisnis*, vol. 6, no. 1, p. 46, 2016.

DAFTAR RIWAYAT HIDUP

A. Biodata Mahasiswa

NIM : 16170122
Nama Lengkap : Fitra Septia Nugraha
Tempat & Tanggal Lahir : Tasikmalaya, 12 Maret 1994
Jenis Kelamin : Laki-laki
Alamat : Kp. Cibuntiris RT/RW 01/02 Desa
Mangkonjaya Kec. Bojonggambir
Kab. Tasikmalaya

B. Riwayat Pendidikan Formal dan Non-Formal

1. SDN 2 Bojonggambir, Lulus tahun 2007
2. SMPN 1 Bojonggambir, Lulus tahun 2010
3. SMA AL-Mukrom Bojonggambir, Lulus tahun 2013
4. AMIK BSI Tasikmalaya (D3-MI), Lulus tahun 2017
5. Universitas BSI Bandung, Lulus Tahun 2018

C. Riwayat Pengalaman Berorganisasi / Pekerjaan

a. Pengalaman Berorganisasi

1. Tahun 2008–2009 : Pengurus OSIS SMPN 1 Bojonggambir (Anggota)
2. Tahun 2008–2009 : Pengurus Pramuka SMPN 1 Bojonggambir (Pratama)
3. Tahun 2011–2013 : Pengurus Pramuka SMA Al-Mukrom (Pradana)
4. Tahun 2011–2012 : Pengurus Osis SMA Al-Mukrom (Sekretaris)
5. Tahun 2012–2013 : Anggota Dewan Kerja Ranting Bojonggambir
6. Tahun 2011–2012 : Anggota Paskibra (Paska) Kab. Tasikmalaya
7. Tahun 2014–2016 : UKM English Club, Dapur seni, Broadcasting
8. Tahun 2015–2016 : Pengurus Senator Mahasiswa (Sie.Pendidikan)
9. Tahun 2016–2017 : Pengurus HMMI

b. Pengalaman Pekerjaan

1. Tahun 2013–2014 : Admin Retur dan Audit Distributor Hp Sinarjaya Cell Tasikmalaya
2. Tahun 2015 : Reseller Mie Lidi-Q
3. Tahun 2016–2017 : Asisten Dosen Bagian BSI Entrepreneur Center
4. Tahun 2018–2019 : Wirausaha Konter HP dan Accesories
5. Tahun 2019 : Staff Instruktur STMIK Nusa Mandiri



Jakarta, 05 Agustus 2020


Fitra Septia Nugraha