

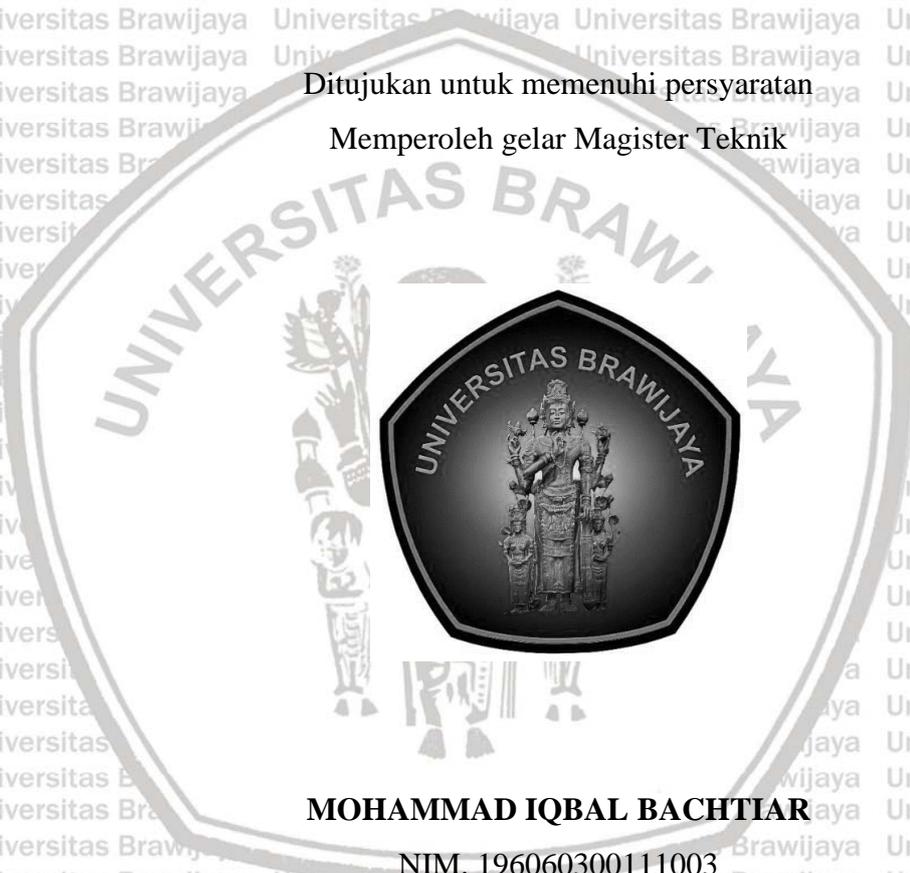
**DEEP LEARNING DAN ANALYTICAL HIERARCY PROCESS
DALAM SISTEM PENDUKUNG KEPUTUSAN
SELEKSI BEASISWA**

TESIS

**PROGRAM MAGISTER TEKNIK ELEKTRO
MINAT SISTEM KOMUNIKASI DAN INFORMASI (SKI)**

Ditujukan untuk memenuhi persyaratan

Memperoleh gelar Magister Teknik



MOHAMMAD IQBAL BACHTIAR

NIM. 196060300111003

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2021



TESIS

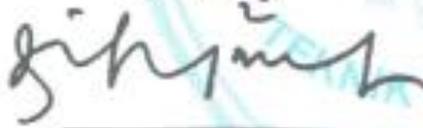
DEEP LEARNING DAN ANALYTICAL HIERARCY PROCESS DALAM SISTEM PENDUKUNG KEPUTUSAN SELEKSI BEASISWA

MOHAMMAD IQBAL BACHTIAR
NIM. 196060300111003

telah dipertahankan di depan penguji
pada Tanggal 21 Juli 2021
dinyatakan telah memenuhi syarat
untuk memperoleh gelar Magister Teknik

Komisi Pembimbing

Pembimbing I



Prof. Ir. Hadi Suyono, S.T., M.T., Ph.D., IPU., ASEAN Eng.

Pembimbing II



M. Fauzan Edy P., S.T., M.T., Ph.D.

Malang, 28 Juli 2021

Universitas Bruwijaya

Fakultas Teknik, Jurusan Teknik Elektro

Pd. Ketua Jurusan Teknik Elektro



M. Aziz Muband, S.T., M.T., Ph.D.

NIM. 197412032000121001

JUDUL TESIS:**DEEP LEARNING DAN ANALYTICAL HIERARCHY PROCESS DALAM SISTEM PENDUKUNG KEPUTUSAN SELEKSI BEASISWA**

Nama Mahasiswa : Mohammad Iqbal Bachtiar

NIM : 196060300111003

Program Studi : S2 Teknik Elektro

Minat (bila ada) : Sistem Komunikasi dan Informasi (SKI)

KOMISI PEMBIMBING :

Ketua : Prof. Ir. Hadi Suyono, ST., MT., Ph.D., IPU., ASEAN Eng.

Anggota : M. Fauzan Edy Purnomo, ST., MT., Ph.D

TIM DOSEN PENGUJI :

Dosen Penguji 1 : Muhammad Aziz Muslim, ST., MT., Ph.D

Dosen Penguji 2 : Dr. Eng. Panca Mudjirahardjo, ST., MT.

Dosen Penguji Saksi (Bila Ada) : -

Tanggal Ujian : 21 Juli 2021

SK Penguji : 1238 Tahun 2021





*Teriring Ucapan Terima Kasih kepada:
Ayahanda dan Ibunda tercinta*

PERNYATAAN ORISINALITAS TESIS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan Saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Tesis ini adalah asli dari pemikiran Saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Tesis ini dapat dibuktikan terdapat unsur-unsur jiplakan, Saya bersedia Tesis dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 tahun 2003, Pasal 25 Ayat 2 dan Pasal 70).

Malang, 28 Juli 2021

Mahasiswa,



Nama : MOHAMMAD IQBAL BACHTIAR

NIM : 196060300111003

PM : TEKNIK ELEKTRO

PROGRAM MAGISTER TEKNIK ELEKTRO

RINGKASAN

Mohammad Iqbal Bachtiar, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Januari 2021, *Deep Learning* dan *Analytical Hierarchy Process* Dalam Sistem Pendukung Keputusan Seleksi Beasiswa. Dosen Pembimbing: Prof. Ir. Hadi Suyono, ST., MT., Ph.D., IPU dan M. Fauzan Edy Purnomo, ST., MT., Ph.D.

Beasiswa merupakan salah satu bentuk dukungan finansial terhadap mahasiswa untuk mendukung proses studinya. Proses seleksi beasiswa yang ada sekarang umumnya memiliki target sasaran dan kriteria calon penerima beasiswa yang berbeda. Hal ini menyebabkan proses pengambilan keputusan seleksi beasiswa menjadi kompleks, padahal pada umumnya proses seleksi calon penerima beasiswa dibatasi waktu. Permasalahan ini dapat dipecahkan dengan menggunakan bantuan SPK. Beberapa alternatif algoritma untuk SPK yang telah tersedia antara lain menggunakan AHP, TOPSIS, SAW, maupun pendekatan berbasis *machine learning* yaitu *shallow learning*, dan *deep learning*. *Deep learning* merupakan salah satu pendekatan paling potensial karena teknik ini mempelajari pola yang terdapat didalam data, dan menerapkannya pada permasalahan selanjutnya. Selain itu dipilih pula teknik yang lebih tradisional, yaitu AHP dan TOPSIS untuk perbandingan kinerja. Didapatkan bahwa akurasi metode TOPSIS untuk seleksi beasiswa mandiri, bidikmisi, dan prestasi berturut-turut adalah 56,72%, 65,21%, dan 95,87%. Sementara akurasi dari metode *deep learning* berturut-turut 71,93%, 100%, 100%. Dari hasil ini disimpulkan bahwa pendekatan berbasis *deep learning* memiliki kinerja akurasi prediksi yang lebih baik.

Kata kunci: *Analytical Hierarchy Process, TOPSIS, Deep Learning, Seleksi Beasiswa, Sistem Pendukung Keputusan*

KATA PENGANTAR

Alhamdulillah Rabbil ‘Alamin, Puji syukur senantiasa kami ucapkan atas nikmat, karunia, rahmat, serta hidayah Allah SWT, sehingga penyusunan Tesis ini yang berjudul “*Deep Learning dan Analytical Hierarchy Process* Dalam Sistem Pendukung Keputusan Seleksi Beasiswa” ini dapat terselesaikan dengan baik.

Dalam penyusunan Tesis ini, semua tidak lepas dari peran dan dukungan dari berbagai pihak. Oleh karena itu penulis tidak lupa menyampaikan terima kasih kepada:

1. Bapak Prof. Ir. Hadi Suyono, ST., MT., Ph.D., IPU. selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya.
2. Bapak Dr. Eng. Panca Mudjirahardjo, ST., MT. selaku Ketua Program Studi S2 Teknik Elektro.
3. Bapak Prof. Ir. Hadi Suyono, ST., MT., Ph.D., IPU. dan M. Fauzan Edy Purnomo, ST., MT., Ph.D. selaku dosen pembimbing tesis penulis.
4. Orang tua dan keluarga penulis yang telah banyak memberi dukungan baik moral, materil, dan spiritual.
5. Teman 1 kelas S2 Teknik Elektro minat Sistem Komunikasi dan Informasi saya, Mas Arief, Mas Eko, Mas Satyo, Fauzie, Rizki, Anas dan Eva mahasiswa Teknik Elektro angkatan 2019.
6. Semua pihak yang telah membantu dalam proses penyelesaian tesis ini yang tidak bisa disebutkan satu per satu.

Akhir kata, penulis menyadari bahwa dalam penulisan laporan Tesis ini masih terdapat banyak kekurangan maupun adanya kesalahan di dalamnya, oleh karena itu penulis mengharap adanya saran dan kritik yang bersifat membangun dari pembaca demi membantu kesempurnaan laporan ini dan semoga Tesis ini dapat bermanfaat bagi pembaca dan kita semua. Amin.

Malang, 1 Januari 2021

Penulis

DAFTAR ISI

Halaman

LEMBAR PENGESAHAN..... ii

RINGKASAN..... vi

KATA PENGANTAR..... vii

DAFTAR ISI..... viii

DAFTAR TABEL..... x

DAFTAR GAMBAR..... xii

BAB I PENDAHULUAN..... 1

1.1 Latar Belakang 1

1.2 Rumusan Masalah 2

1.3 Batasan Masalah..... 3

1.4 Tujuan Penelitian..... 3

1.5 Manfaat Penelitian..... 3

BAB II TINJAUAN PUSTAKA..... 5

2.1 Penelitian Sebelumnya 5

2.2 Dasar Teori..... 7

2.2.1 *Analytical Hierarchy Process* 7

2.2.2 Contoh Pengambilan Keputusan dengan AHP..... 9

2.2.3 *Deep Learning* 12

2.2.4 *Technique For Others Reference by Similarity to Ideal Solution* 16

2.2.5 *Confusion Matrix*..... 18

2.3 Kedudukan Penelitian 20

2.4 Daftar Istilah..... 21

BAB III KERANGKA KONSEP PENELITIAN..... 23

3.1 Kerangka Berpikir..... 23

3.2 Hipotesis..... 24

3.3 Definisi Variabel Operasional..... 24

BAB IV METODE PENELITIAN..... 25

4.1 Jenis dan Cara Perolehan Data 25

4.2 Variabel dan Cara Analisis Data	25
4.3 Tahapan Penelitian	26
4.3.1 Validasi Data	28
4.3.2 Pembuatan SPK Metode AHP-TOPSIS	32
4.3.3 Melewatkan Data Kedalam SPK AHP-TOPSIS	36
4.3.4 Evaluasi SPK AHP-TOPSIS Menggunakan <i>Confusion Matrix</i>	36
4.3.5 Pembuatan SPK Metode <i>Deep Learning</i>	37
4.3.6 Perbandingan Kinerja Kedua SPK	41
4.3.7 Analisis Kelebihan dan Kekurangan Kedua SPK	41
4.4 Waktu Penelitian	41
BAB V HASIL DAN PEMBAHASAN	43
5.1 Persiapan Data	43
5.1.1 Hasil Proses Validasi Data	43
5.2 Pembuatan SPK Metode AHP-TOPSIS	47
5.2.1 Pembuatan Model Kalkulasi SPK AHP-TOPSIS	47
5.2.2 Melewatkan Data Kedalam SPK Metode AHP-TOPSIS	51
5.2.3 Evaluasi Kinerja SPK	52
5.3 Pembuatan SPK Metode <i>Deep Learning</i>	53
5.3.1 Pembuatan Model	54
5.3.2 Pelatihan Model	56
5.3.3 Evaluasi Kinerja SPK	60
5.4 Perbandingan dan Analisis Kinerja Kedua SPK	61
BAB VI KESIMPULAN DAN SARAN	63
DAFTAR PUSTAKA	65



DAFTAR TABEL

No	Judul	Halaman
Tabel 2.1	Penelitian oleh Puspitasari, et. al.....	5
Tabel 2.2	Penelitian oleh Schocken & Ariav	6
Tabel 2.3	Penelitian oleh Rahim, et. al	6
Tabel 2.4	Penelitian oleh Alhassan dan Lawal	7
Tabel 2.5	Skala Fundamental Nilai Absolut	8
Tabel 2.6	Contoh Perbandingan Berpasangan Konsumsi Minuman di AS	9
Tabel 2.7	Matriks Perbandingan Berpasangan Penimbang Tujuan Akhir	10
Tabel 2.8	Matriks Perbandingan Berpasangan Potensi Gaji	11
Tabel 2.9	Hasil Akhir Analisis Alternatif Keputusan dengan AHP.....	12
Tabel 2.10	Confusion Matrix Cost.....	Error! Bookmark not defined.
Tabel 2.11	Daftar Istilah Yang Digunakan	22
Tabel 4.1	Prosedur Penyelesaian Permasalahan Data	31
Tabel 4.2	Contoh Matriks Perbandingan Berpasangan	33
Tabel 4.3	Contoh Rekomendasi Beasiswa	36
Tabel 4.4	<i>Confusion Matrix</i> Untuk Atribusi <i>Cost</i>	37
Tabel 4.5	Jadwal Penelitian.....	42
Tabel 5.1	<i>Hasil Validasi Data Beasiswa Prestasi</i>	44
Tabel 5.2	<i>Hasil Validasi Data Beasiswa Bidikmisi</i>	45
Tabel 5.3	<i>Hasil Validasi Data Beasiswa Mandiri</i>	46
Tabel 5.4	<i>Sub-kriteria Penyusun Kriteria Komposit</i>	47
Tabel 5.5	<i>Bobot Masing-Masing Kriteria Komposit</i>	48
Tabel 5.6	<i>Bobot Subkriteria Administratif</i>	48
Tabel 5.7	<i>Bobot Subkriteria Prestasi Akademik</i>	49
Tabel 5.8	<i>Bobot Masing-Masing Kriteria Komposit Beasiswa Bidikmisi</i>	49
Tabel 5.9	<i>Bobot Subkriteria Kondisi Finansial</i>	50
Tabel 5.10	<i>Bobot Subkriteria Bantuan Terdahulu</i>	50
Tabel 5.11	<i>Bobot Subkriteria Beban</i>	50
Tabel 5.12	<i>Bobot Kriteria Beasiswa Mandiri</i>	51
Tabel 5.13	<i>Solusi Positif dan Negatif Ideal</i>	52
Tabel 5.14	<i>Jumlah Calon Penerima Beasiswa Yang Mendapatkan Beasiswa</i>	53
Tabel 5.15	<i>Akurasi Rekomendasi SPK Terhadap Hasil Sebenarnya</i>	53

Tabel 5.16 *Parameter Pelatihan Model* 56

Tabel 5.17 *Akurasi Rekomendasi SPK Terhadap Hasil Sebenarnya* 60

Tabel 5.18 *Perbandingan Kinerja Kedua Pendekatan* 61



DAFTAR GAMBAR

No	Judul	Halaman
	Gambar 2.1 Hierarki pengambilan keputusan pekerjaan terbaik.....	10
	Gambar 2.2 Model <i>perceptron</i>	13
	Gambar 2.3 Arsitektur <i>shallow learning</i>	14
	Gambar 2. 4 Arsitektur <i>deep learning</i>	15
	Gambar 3.1 Kerangka konsep kedudukan metode dalam ranah pengambilan keputusan ..	23
	Gambar 4.1 Bagan alir tahapan penelitian	26
	Gambar 4.2 Bagan alur proses validasi data	28
	Gambar 4. 3 Pratinjau data penerima beasiswa prestasi	29
	Gambar 4. 4 Pratinjau data penerima beasiswa bidikmisi	29
	Gambar 4. 5 Pratinjau data penerima beasiswa mandiri	30
	Gambar 4. 6 Diagram alir perhitungan pada algoritma AHP-TOPSIS	32
	Gambar 4.7 Arsitektur model <i>deep learning multilayer perceptron</i>	38
	Gambar 4. 8 Diagram alir algoritma berbasis <i>deep learning</i>	40
	Gambar 5.1 Ikhtisar model untuk beasiswa prestasi	54
	Gambar 5.2 Ikhtisar model untuk beasiswa bidikmisi	54
	Gambar 5.3 Ikhtisar model untuk beasiswa mandiri.....	55
	Gambar 5.4 Hasil pelatihan pada titik optimal <i>epoch</i> ke-29	56
	Gambar 5.5 Grafik nilai <i>loss</i> pelatihan terhadap <i>epochs</i> pelatihan	57
	Gambar 5.6 Hasil pelatihan telah konvergen pada <i>Epochs</i> ke 6	57
	Gambar 5.7 Grafik nilai <i>loss</i> pelatihan terhadap <i>epochs</i> pelatihan.....	58
	Gambar 5.8 Hasil pelatihan telah konvergen pada <i>Epochs</i> ke 14	58
	Gambar 5.9 Grafik nilai <i>loss</i> pelatihan terhadap <i>epochs</i> pelatihan.....	59

BAB I PENDAHULUAN

1.1 Latar Belakang

Beasiswa merupakan suatu bentuk dukungan finansial kepada mahasiswa dalam rangka mendukung mahasiswa untuk memperlancar penyelesaian kegiatan belajarnya. Dukungan dana untuk sebuah program beasiswa pendidikan yang diselenggarakan oleh sebuah perguruan tinggi bisa berasal dari berbagai sumber, antara lain bisa dari pemerintah, dari donor diluar perguruan tinggi, maupun dari dana mandiri perguruan tinggi. Program beasiswa ini, umumnya disediakan, dengan penetapan terhadap jenis beasiswanya, berdasarkan kriteria utama yang umumnya ditentukan oleh lembaga penyanggah dana beasiswa tersebut.

Pada umumnya kriteria utama yang ditetapkan oleh masing-masing sumber pendanaan ini akan berbeda, sehingga akan terdapat perbedaan sasaran potensial mahasiswa yang dapat diberikan beasiswa tersebut. Perbedaan kriteria utama dari masing-masing sumber pendanaan ini, umumnya terkait dengan jenis program beasiswa yang diselenggarakan. Beberapa beasiswa mendasarkan kriterianya terhadap prestasi dari mahasiswa. Beberapa beasiswa menyasar mahasiswa dengan kondisi ekonomi kurang baik. Beasiswa lain mendasarkan kriterianya dengan model bidikmisi, yaitu menyasar mahasiswa dari desa, daerah tertinggal, rawan, perbatasan, dan wilayah timur Indonesia.

Disamping kriteria utama yang telah ditetapkan tersebut, umumnya juga akan ditetapkan kriteria penunjang, yang akan menjadi pertimbangan dalam pengambilan keputusan seleksi beasiswa. Kriteria ini dapat berupa kriteria akademik, maupun kriteria non akademik. Permasalahan yang timbul dari kriteria-kriteria ini adalah bahwa terdapat perbedaan dalam bentuk data ini, sehingga diperlukan suatu metode untuk melakukan penilaian terhadap kriteria-kriteria ini. Selain permasalahan seleksi berdasarkan kriteria, permasalahan lain dalam proses seleksi ini adalah bahwa terdapat banyak data, yang perlu dianalisis dalam waktu singkat, karena umumnya terdapat tenggat waktu yang harus dipenuhi dalam pengusulan dan proses administratif beasiswa. Seorang calon penerima beasiswa pun hanya diperbolehkan untuk menerima satu beasiswa (tidak boleh terjadi rangkap beasiswa) walaupun penerima tersebut memenuhi kriteria pada lebih dari satu

program beasiswa. Hal tersebut diperparah juga dengan kondisi data yang seringkali tidak lengkap, yang menjadikan proses pengambilan keputusan biasa menjadi sangat kompleks. Untuk itu diperlukan suatu solusi untuk membantu pengambilan keputusan seleksi penerima beasiswa ini secara cepat dan akurat.

Pengambilan keputusan secara cepat, dapat dilakukan dengan dukungan sebuah Sistem Pendukung Keputusan (SPK). Dalam usulannya, (Keen, 1980) menunjukkan konsep pengambilan keputusan dengan bantuan sistem yang terkomputerisasi. Penggunaan ini memungkinkan pengambilan keputusan dengan lebih cepat. Terdapat beberapa alternatif metode yang tersedia untuk SPK. Yang pertama adalah metode AHP (*Analytical Hierarchy Process*) oleh (Puspitasari, Sari, Destarianto, & Riskiawan, 2018). Kemudian, metode SAW (*Simple Additive Weighting*) yang relatif lebih sederhana sebagaimana ditunjukkan oleh (Josaputri, Sugiharti, & Arifudin, 2016). Lalu, alternatif lain yaitu pendekatan TOPSIS yang diperkenalkan oleh (Rahim, Supiyandi, Siahaan, Listyorini, & Utomo, 2018). Pada penelitiannya, (Putra, Ariyanti, & Cholissodin, 2013) menunjukkan mengenai penggabungan beberapa metode tersebut dalam pengambilan keputusannya. Perkembangan teknik-teknik dalam bidang *machine learning*, juga mempengaruhi implementasi dalam SPK, sebagaimana ditunjukkan oleh (Aulck, Nambi, & West, 2020). Perkembangan teknik data mining dan implementasinya dalam SPK ditunjukkan pula dalam penelitian oleh (Alhassan & Lawal, 2015). Namun, pada penelitian-penelitian yang sebelumnya belum ada yang secara khusus membahas mengenai pendekatan *deep learning* yang memiliki potensi kinerja yang lebih baik, dalam penggunaannya dengan SPK.

Karena itu, pada penelitian ini akan diterapkan teknik *deep learning*, yaitu menggunakan *neuron* yang disusun dalam beberapa lapisan, untuk membentuk sebuah Sistem Pendukung Keputusan seleksi beasiswa. Sebagai acuan kinerjanya, akan dibuat pula sebuah SPK berbasis metode AHP-TOPSIS yang merupakan salah satu teknik tradisional agar dapat dibandingkan kinerjanya, dan melihat kekuatan serta kelemahan dari kedua metode ini dalam pembuatan sebuah SPK seleksi beasiswa. Atas dasar itu, judul yang diambil adalah “*Deep Learning dan Analytical Hierarchy Process Dalam Sistem Pendukung Keputusan Seleksi Beasiswa*”.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah tersebut, dari permasalahan yang ada dirumuskan pada penelitian ini yaitu sebagai berikut:

1. Bagaimana prosedur untuk validasi data beasiswa yang akan digunakan?

2. Bagaimana pengembangan SPK alokasi beasiswa menggunakan metode *deep learning*?
3. Bagaimana mengembangkan SPK alokasi beasiswa menggunakan metode AHP – TOPSIS?
4. Bagaimana melakukan evaluasi kinerja SPK yang telah dikembangkan?

1.3 Batasan Masalah

Karena cukup banyaknya permasalahan yang terkait dalam pengembangan SPK, perlu ditetapkan suatu batasan dalam penelitian disini. Batasan permasalahan didalam penelitian disini ditetapkan yaitu sebagai berikut:

1. Metode yang akan digunakan adalah metode AHP yang disusun dengan TOPSIS, dan metode *deep learning*.
2. Kriteria penilaian yang dipakai yaitu semester, Indeks Prestasi Kumulatif (IPK), pendapatan orang tua, keterangan status orang tua, jumlah yang ditanggung orang tua dan kriteria-kriteria lain.
3. Program beasiswa yang akan digunakan dibatasi menjadi tiga, yaitu beasiswa prestasi, beasiswa bidikmisi, dan beasiswa mandiri. Tiap mahasiswa hanya dapat memperoleh rekomendasi sistem untuk menerima satu beasiswa.

1.4 Tujuan Penelitian

Dari yang telah diuraikan di atas yaitu latar belakang dan rumusan masalah maka tujuan dari penelitian disini yaitu membuat prosedur validasi data, mengembangkan SPK dengan metode *deep learning*, dan AHP - TOPSIS, untuk kemudian membandingkan dan menganalisis kinerja sistem pendukung keputusan (SPK) tersebut.

1.5 Manfaat Penelitian

Dan manfaat dari penelitian disini yaitu:

1. Bagi peneliti, yaitu sebagai sumber referensi penerapan *deep learning* dalam aplikasi Sistem Pendukung Keputusan (SPK).
2. Bagi pengguna SPK, dapat menjadi referensi alternatif metode untuk SPK, serta analisis terhadap kelebihan dan kekurangannya.
3. Bagi penerima beasiswa, dapat menjadi salah satu alternatif SPK untuk mendukung proses seleksi penerima beasiswa.



[Halaman Ini Sengaja Dikosongkan]

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Pengembangan sistem pendukung keputusan untuk seleksi beasiswa telah dilakukan sebelumnya. Namun, penelitian yang ada sebelumnya hanya membahas mengenai alokasi dana dan alokasi penerima berdasarkan dana yang dimiliki. Belum ada penelitian yang secara khusus membahas ketika terdapat beberapa skema dengan kriteria berbeda dan alokasi berbeda dalam penerima beasiswa. Pada penelitiannya (Puspitasari, et. al, 2018) menerapkan pendekatan AHP dan mendapatkan akurasi 90%. Penelitian ini ditunjukkan pada Tabel 2.1.

Tabel 2.1 Penelitian oleh Puspitasari, et. al

<i>Penulis, Tahun</i>	Puspitasari, et. al, 2018
<i>Permasalahan</i>	Seleksi Beasiswa
<i>Lingkup</i>	Seleksi beasiswa dengan menggunakan metode AHP
<i>Masalah</i>	dengan rating yang disusun berdasarkan Saaty dan Matriks perbandingan berpasangan.
<i>Metode</i>	<ul style="list-style-type: none"> • <i>Saaty Rating Scale for Pairwise Comparison Matrix</i> • <i>Analutical Hierarchy Process (AHP)</i>
<i>Hasil</i>	Akurasi sistem SPK sebesar 90% dibandingkan hasil seleksi aktual.

Kemudian, (Schocken & Ariav, 1991), membahas mengenai peluang penggunaan *neural network* dalam pengembangan sebuah pendukung keputusan. Penggunaan *neural network* dalam pendukung keputusan, memerlukan sebuah perancangan yang didasarkan pada kebutuhan pendukung keputusan tersebut. Hal ini disebabkan karena teknik-teknik *neural network* merupakan teknik yang melakukan inferensi untuk mencari hubungan dari data yang telah ada. Sementara pada pengambilan keputusan, lebih banyak penekanan pada pembuatan sebuah model pengambilan keputusan, yang didasarkan pada kriteria-kriteria

ideal, sehingga dapat membuat implementasi ini secara inheren, memiliki *bias* yang berasal dari data yang digunakan. Penelitian ini ditunjukkan pada Tabel 2.2

Tabel 2.2 Penelitian oleh Schocken & Ariav

<i>Penulis, Tahun</i>	Schocken & Ariav, 1991
<i>Permasalahan</i>	Penggunaan <i>neural network</i> dalam SPK
<i>Lingkup Masalah</i>	<i>Neural network</i> merupakan metode yang berbasis <i>data driven</i> , yang mana dilakukan pemodelan berdasarkan data yang ada.
<i>Metode</i>	<ul style="list-style-type: none"> • <i>Neural Network</i>
<i>Hasil</i>	Perlu identifikasi masalah secara lebih tepat sebelum memutuskan penggunaan <i>neural network</i> dalam pengambilan keputusan.

Kemudian, (Rahim, et. al, 2018) melakukan implementasi TOPSIS dalam Sistem Pendukung Keputusan, dalam hal ini dalam pengambilan keputusan pemilihan karyawan terbaik, berdasarkan beberapa kriteria. Dalam penerapannya, metode TOPSIS mampu memberikan rekomendasi dan menghasilkan peringkat rekomendasi atas keputusan untuk diambil.

Tabel 2.3 Penelitian oleh Rahim, et. al

<i>Penulis, Tahun</i>	Rahim, et. al, 2018
<i>Permasalahan</i>	Penentuan Karyawan Terbaik
<i>Lingkup Masalah</i>	Pembuatan sebuah sistem pendukung keputusan untuk seleksi karyawan terbaik dalam suatu perusahaan.
<i>Metode</i>	<ul style="list-style-type: none"> • <i>Technique for Order of Preference by Similarity to Idea Solution (TOPSIS)</i>
<i>Hasil</i>	Sistem mampu memberikan suatu rekomendasi

Selain itu, menggunakan pendekatan *machine learning* dan teknik-teknik *data mining*, (Alhassan & Lawal, 2015) sebuah pendekatan baru dalam seleksi beasiswa, berhasil menghasilkan sebuah keputusan dalam seleksi penerima beasiswa.

Tabel 2.4 Penelitian oleh Alhassan dan Lawal

<i>Penulis, Tahun</i>	Alhassan dan Lawal, 2015
<i>Permasalahan</i>	Seleksi Alokasi Beasiswa
<i>Lingkup Masalah</i>	Pembuatan sebuah sistem pendukung keputusan untuk seleksi alokasi beasiswa, dengan menggunakan pendekatan <i>decision tree</i> .
<i>Metode</i>	<ul style="list-style-type: none"> • <i>Decision Tree</i> • <i>Decision support system</i>
<i>Hasil</i>	Efektif dan efisien dalam pelaksanaan seleksi mahasiswa penerima beasiswa.

2.2 Dasar Teori

2.2.1 Analytical Hierarchy Process

Sebuah keputusan harus diambil dengan melalui metode yang terstruktur dan dilakukan secara secara terorganisir (Saaty, 2008). Proses pengambilan keputusan ini memerlukan sebuah pengorganisasian yang membagi tahapannya kedalam beberapa langkah. Langkah-langkah tersebut adalah sebagai berikut:

1. Definisi permasalahan, dan merumuskan segala jenis informasi yang dibutuhkan dan diinginkan untuk didapatkan dalam proses pengambilan keputusan metode *analytical hierarchy process* ini.
2. Penyusunan hierarki keputusan dari tingkat paling atas, yaitu tujuan utama pengambilan keputusan, kemudian disusul tingkat dibawahnya, berisi sasaran dari perspektif yang lebih luas. Diantara tingkat paling atas dan paling bawah tersebut, dilalui tingkat-tingkat perantara dimana unsur-unsur terkait dengan sasaran tersebut dimiliki, hingga terjadi beberapa alternatif keputusan.
3. Penyusunan matriks perbandingan berpasangan, untuk membandingkan tingkat kepentingan antara faktor-faktor yang dinilai dalam pengambilan keputusan. Setiap unsur pada suatu tingkat diatas, digunakan untuk membandingkan unsur pada tingkat yang berada satu tingkat dibawahnya.
4. Gunakan prioritas yang dihasilkan suatu tingkat, untuk menjadi bobot prioritas bagi tingkat yang berada di bawahnya langsung. Hal ini dilakukan untuk semua unsur. Proses in dilakukan secara terus menerus hingga didapati bobot prioritas global dari tingkat paling bawah.

Untuk membuat perbandingan, dibutuhkan suatu skala angka yang mengindikasikan berapa kali lipat suatu kriteria memiliki nilai kepentingan, apabila dibandingkan dengan kriteria lainnya. Skala ini ditunjukkan sebagaimana pada ditunjukkan pada Tabel 2.5.

Ketika berada pada nilai 1, ini menunjukkan bahwa kriteria satu sama pentingnya dibandingkan kriteria lainnya.

Tabel 2. 5 Skala Fundamental Nilai Absolut

Intensitas Kepentingan	Definisi	Penjelasan
1	Sama Penting	Kedua unsur berkontribusi setara
2	Sedikit Lebih Penting	
3	Sedang	Pengalaman atau hasil penilaian
4	Sedang Lebih	sedikit menunjukkan suatu unsur lebih penting dari lainnya
5	Kuat	Pengalaman atau hasil penilaian cukup
6	Kuat Lebih	menunjukkan bahwa suatu unsur lebih penting dari lainnya
7	Sangat Kuat	Pengalaman atau hasil penilaian cukup
8	Sangat Kuat	menunjukkan bahwa suatu unsur jauh lebih penting dari lainnya, dan telah dibuktikan dalam praktik.
9	Paling Kuat	Bukti kuat dan nyata menunjukkan bahwa suatu unsur lebih penting dari unsur lainnya.

Kemudian, dicontohkan pada Tabel 2.6 yang dikutip dari (Saaty, 2008), tentang penggunaan skala, dicontohkan suatu permasalahan, dimana dilakukan survey terhadap preferensi dari orang-orang di Amerika Serikat dalam mengkonsumsi minuman. Tingkat prioritas tersebut merupakan perbandingan berpasangan antara satu pasang minuman (pada satu baris) terhadap minuman lainnya (pada kolom). Maka pada kolom *coffee* dan baris *wine*, didapatkan nilai 9, menunjukkan bahwa kopi 9 kali lebih diminati dibandingkan *wine*.

Tabel 2. 6 Contoh Perbandingan Berpasangan Berpasangan Konsumsi Minuman di AS

	<i>Coffee</i>	<i>Wine</i>	<i>Tea</i>	<i>Beer</i>	<i>Sodas</i>	<i>Milk</i>	<i>Water</i>
<i>Coffee</i>	1	9	5	2	1	1	$\frac{1}{2}$
<i>Wine</i>	$\frac{1}{9}$	1	$\frac{1}{3}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
<i>Tea</i>	$\frac{1}{5}$	2	1	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{9}$
<i>Beer</i>	$\frac{1}{2}$	9	3	1	$\frac{1}{2}$	1	$\frac{1}{3}$
<i>Sodas</i>	1	9	4	2	1	2	$\frac{1}{2}$
<i>Milk</i>	1	9	3	1	$\frac{1}{2}$	1	$\frac{1}{3}$
<i>Water</i>	2	9	9	3	2	3	1

Sumber: Saaty (2008)

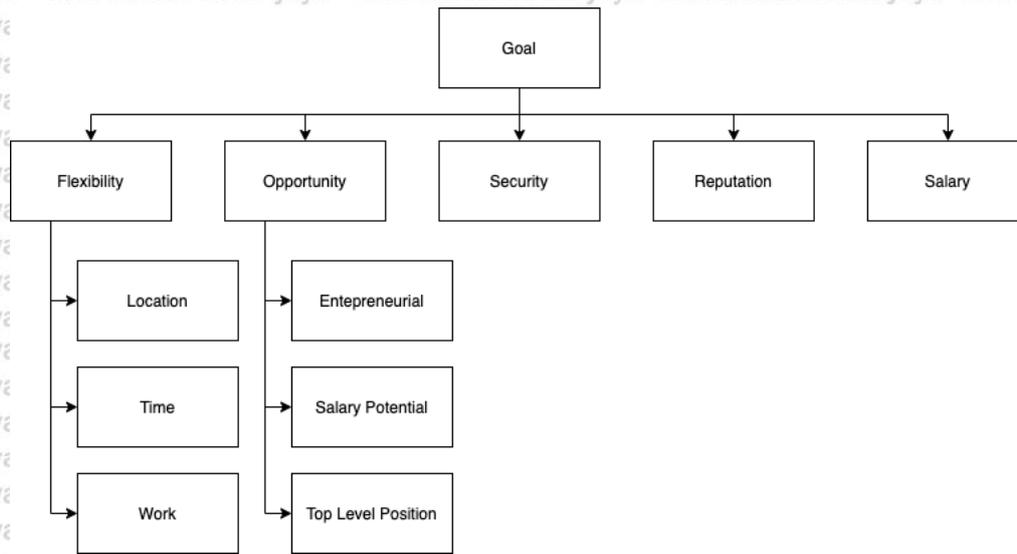
Pada kasus diatas, responden menjawab pertanyaan mengenai preferensi minumasin, dan hasil pertanyaan dimasukkan kedalam tabel dengan menggunakan nilai skala yang sesuai, yang telah ditunjukkan dari Tabel 2.5. Sebagai contoh, dimasukkannya angka 9 pada (baris, kolom) untuk (*coffee*, *wine*), menunjukkan bahwa konsumsi kopi adalah 9 kali lebih banyak atau lebih diminati dibandingkan konsumsi *wine*. Secara otomatis, juga diasumsikan pada (*wine*, *coffe*), bahwa *wine* diminati sebanyak $\frac{1}{9}$ kali dibandingkan kopi. Selain itu juga didapati bahwa air (*water*) dikonsumsi 2 kali lebih banyak dari kopi, maka diisikan nilai 2 pada posisi (*water*, *coffee*) dan diisikan nilai $\frac{1}{2}$ pada posisi (*coffee*, *water*). Nilai perbandingan terhadap diri sendiri secara otomatis akan menghasilkan nilai 1.

Prioritas dari masing-masing unsur yang ditunjukkan dalam tabel, didapatkan dari menjumlahkan setiap baris dari matriks, dan membagi dengan totalnya. Penggunaan metode ini, dapat menunjukkan tingkat preferensi, atau kepentingan dari tiap unsur, dibandingkan unsur lainnya dalam bentuk komparasi atau perbandingan. Pada penelitian tersebut, telah dilakukan divalidasi dengan menggunakan data dari 30 orang responden, dengan didapatkan nilai 0.22 dari rasio konsistensi jawaban. Hal ini dijadikan dasar bahwa pendekatan AHP dapat mengekstraksi informasi yang dibutuhkan untuk proses pengambilan keputusan nantinya.

2.2.2 Contoh Pengambilan Keputusan dengan AHP

Pada subbab ini akan dibahas mengenai contoh penggunaan AHP dalam suatu contoh pengambilan keputusan. Contoh penggunaan ini dimaksudkan untuk memberikan gambaran yang lebih jelas mengenai penggunaan AHP dalam proses pengambilan

keputusan. Contoh ini diambil dari tulisan mengenai AHP (Saaty, 2008). Pada contoh ini akan diberikan sebuah pengambilan keputusan sederhana, mengenai seseorang yang sedang mempertimbangkan pekerjaan apa yang terbaik setelah ia mendapatkan gelar Doktor, mengenai seleksi pekerjaan apa yang terbaik. Ada beberapa alternatif yaitu bekerja untuk perusahaan domestik, bekerja untuk perusahaan internasional, bekerja untuk sebuah perguruan tinggi, atau bekerja untuk universitas negara.



Gambar 2.1 Hierarki pengambilan keputusan pekerjaan terbaik
Sumber: Saaty (2008)

Secara keseluruhan, terdapat dua belas buah matriks perbandingan berpasangan. Masing-masing matriks berpasangan menghasilkan kriteria untuk pertimbangan dalam pemilihan prospek karir yang akan dijalani. Matriks berpasangan kriteria ini salah satunya ditunjukkan pada Tabel 2.7.

Tabel 2.7 Matriks Perbandingan Berpasangan Penimbang Tujuan Akhir

	<i>Flexibility</i>	<i>Opportunities</i>	<i>Security</i>	<i>Reputation</i>	<i>Salary</i>	<i>Priorities</i>
<i>Flexibility</i>	1	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{4}$	$\frac{1}{8}$	0.036
<i>Opportunities</i>	4	1	$\frac{1}{3}$	3	$\frac{1}{7}$	0.122
<i>Security</i>	6	3	1	4	$\frac{1}{2}$	0.262
<i>Reputation</i>	4	$\frac{1}{3}$	$\frac{1}{4}$	1	$\frac{1}{7}$	0.075
<i>Salary</i>	8	7	2	7	1	0.506

Sumber: Saaty (2008)

Kemudian, setelah dapat ditentukan masing-masing bobot prioritas pada tingkat pertama, perlu dilakukan juga proses perbandingan berpasangan ini terhadap tingkat kedua, yaitu dua unsur yang memiliki unsur pertimbangan lebih jauh, yaitu *flexibility* dan unsur *opportunity*. Unsur fleksibilitas pekerjaan yang dipertimbangkan terdiri dari faktor lokasi pekerjaan, waktu yang dibutuhkan untuk pekerjaan tersebut, dan beban dari pekerjaan tersebut. Sementara, unsur tingkat kedua *opportunity* atau kesempatan berkembang lebih lanjut, terdiri dari faktor *entrepreneurial* atau kesempatan untuk perkembangan ke arah kewirausahaan, kemudian potensi gaji, hingga tingkat paling atas dalam karir yang dapat diraih. Contoh matriks hasil perbandingan berpasangan ini ditunjukkan pada Tabel 2.8, yaitu matriks hasil perbandingan berpasangan dari faktor potensi gaji, terhadap alternatif keputusan.

Tabel 2. 8 Matriks Perbandingan Berpasangan Potensi Gaji

	Domestic Co	Int'l Co	College	State Univ.	Priorities
Domestic Co.	1	4	3	6	0.555
Int'l Co.	1/4	1	3	5	0.258
College	1/3	1/3	1	2	0.124
State Univ.	1/6	1/5	1/2	1	0.064

Sumber: Saaty (2008)

Pada Gambar 2.1 telah ditunjukkan kriteria yang menunjukkan mengenai faktor atau unsur yang digunakan untuk menentukan tujuan akhir, yaitu mengambil keputusan mengenai pekerjaan terbaik. Dari lima faktor tersebut, kemudian dibuat matriks perbandingan berpasangan yang ditunjukkan pada Tabel 2.7. Setelah didapatkan skala prioritas pada tingkat ini, faktor-faktor yang memiliki faktor turunan kemudian dipertimbangkan lebih jauh terhadap alternatif keputusan yang dapat diambil, dengan contohnya dituangkan pada Tabel 2.8 yang menggambarkan nilai masing-masing alternatif keputusan terhadap faktor Potensi Gaji.

Setelah didapatkan seluruh matriks perbandingan berpasangan yang dibutuhkan, langkah selanjutnya yang harus dilakukan adalah sintesis. Proses sintesis dalam konteks ini adalah proses dimana nilai prioritas yang dihasilkan oleh sebuah unsur dibawahnya, dikalikan dengan bobot nilai dari tingkat diatasnya. Proses sintesis ini akan menghasilkan nilai prioritas dari sebuah keputusan dan kontribusinya terhadap prioritas global (tujuan akhir pengambilan keputusan). Dari tabel tersebut, telah didapatkan nilai bobot dari masing-

masing kriteria dan sub-kriteria, dan nilai dari masing-masing alternatif keputusan yang akan diambil.

Kemudian, dari tabel tersebut, hasil akhir dituangkan kedalam prioritas yang dinormalisasi, dan yang diidealkan dalam Tabel 2.9. Dari tabel kemudian dapat dijadikan bahan untuk pengambilan keputusan pekerjaan yang akan diambil. Apabila dilihat dari Tabel 2.9. maka keputusan yang harus diambil adalah dengan bekerja untuk Perusahaan Internasional, dengan alternatif terbaik kedua adalah mengambil posisi mengajar di sebuah universitas negeri.

Tabel 2. 9 Hasil Akhir Analisis Alternatif Keputusan dengan AHP

Name	Normalised Priorities	Idealised Priorities
Domestic Co.	0.193	0.579
Int'l Co.	0.333	1.000
College	0.214	0.643
State Univ.	0.262	0.785

Sumber: Saaty (2008)

Dihasilkannya beberapa prioritas keputusan ini, sekaligus memberikan cadangan keputusan yang dapat diambil apabila kemudian keputusan yang memiliki prioritas utama untuk diambil, menjadi tidak tersedia karena satu atau lain hal. Dalam hal pencarian pekerjaan, hal ini dapat dimunculkan misalnya karena tidak adanya lowongan perusahaan internasional, dan sebagainya.

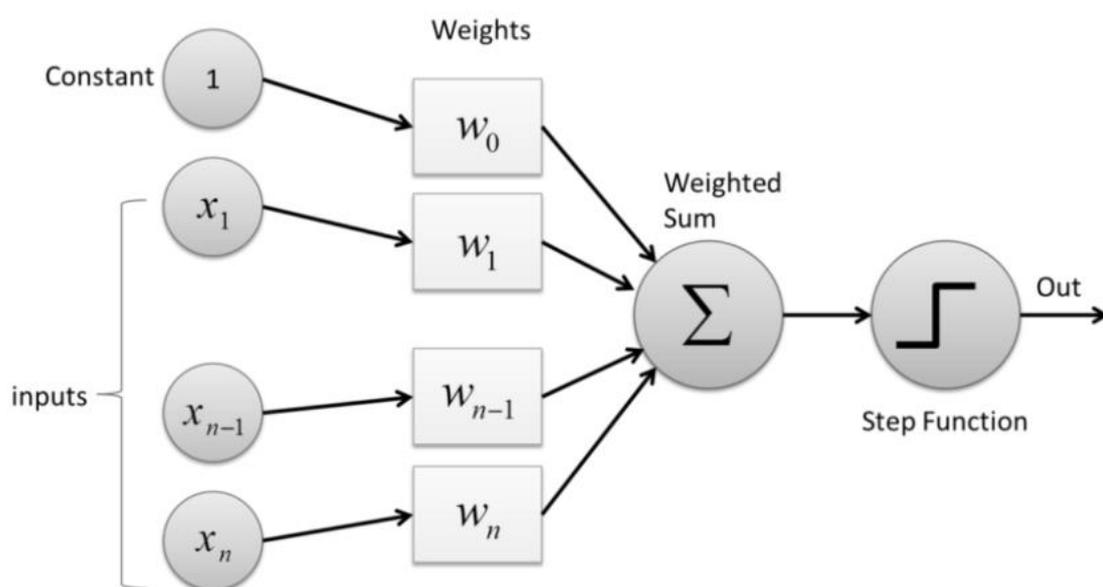
2.2.3 Deep Learning

Artificial Neural Network merupakan suatu sistem komputasi yang dianalogikan kepada model syarat dari jaringan biologis hewan. Sebuah ANN terdiri dari kumpulan beberapa unit *neuron* yang terhubung, dimana setiap hubungan, dapat mengalirkan sinyal.

Masing-masing *neuron* akan melakukan pengolahan terhadap sinyal yang diterima, dan menghasilkan suatu luaran dari hasil pengolahan tersebut. Sinyal pada hubungan antar *neuron* ini adalah nilai-nilai *real*, dan luaran dari penjumlahan dari seluruh hasil pengolahan *neuron* dimasukkan kedalam sebuah fungsi non-linier. Pada masing-masing hubungan, biasanya terdapat bobot untuk menyesuaikan tingkat skala masukan pada masing-masing

neuron. Salah satu model paling awal yang diusulkan dari *neuron* ini adalah model *perceptron* yang pertama kali dikembangkan oleh (Rosenblatt, 1958).

Model *perceptron*, mengambil masukan, baik berupa konstanta, maupun masukan lainnya, kemudian menerapkan bobot terhadap masukan-masukan tersebut, menjumlahkan hasilnya, sebelum kemudian memasukkannya kedalam sebuah fungsi aktivasi, yang umumnya adalah fungsi non-linier. Kelebihan utama dari model *perceptron* ini adalah kemampuannya dalam menerapkan fungsi aktivasi yang bersifat *non-linear* terhadap hasil masukan, yang menjadikannya mampu memproyeksikan fungsi-fungsi *non-linear* yang sering ditemukan pada fenomena fisis. Model ini ditunjukkan pada Gambar 2.2.



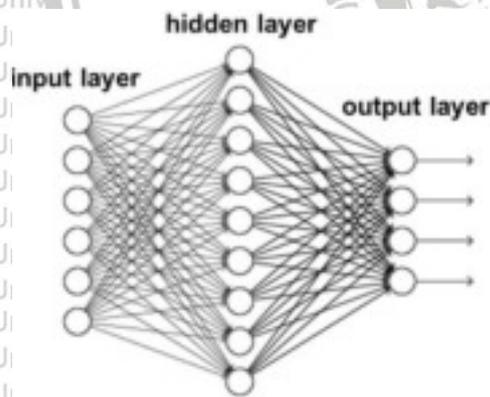
Gambar 2.2 Model *perceptron*

Sumber: Sagar Sharma (2017)

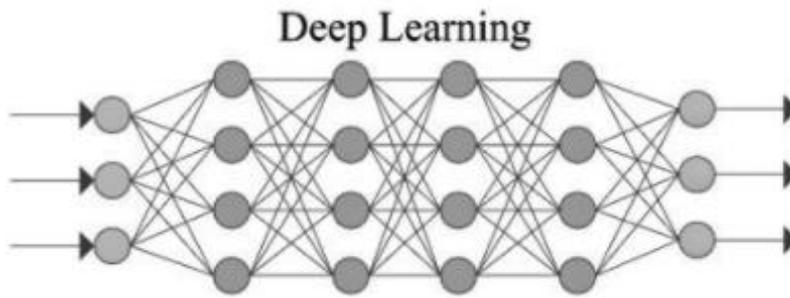
Penggunaan *perceptron* pada awalnya dianggap menjanjikan, karena menawarkan model komputasi yang memungkinkan proses pembelajaran untuk mendapatkan pola inherent dari sebuah fenomena fisis. Namun suatu permasalahan ditemukan, ketika didapati bahwa *perceptron* tidak mampu menyelesaikan permasalahan yang sangat sederhana, yaitu *XOR problem*. Temuan ini kemudian mengarahkan penggunaan *perceptron* ke arah *deep learning*, dimana *perceptron* disusun kedalam beberapa lapisan, yang mana tiap lapisan menyelesaikan suatu permasalahan atau membuat sebuah fitur yang merupakan sintesis dari beberapa fitur pada lapisan sebelumnya. Penerapan *deep learning* meningkatkan kinerja dari model ini secara keseluruhan.

Deep learning sendiri menjadi sebuah bagian dari lingkup metode pembelajaran mesin yang berbasis pada *neural network*. Penggunaan *deep learning* dapat dilakukan secara terawasi, *semi-supervised*, atau tanpa pengawasan. Penggunaan *deep learning* sendiri dapat dinyatakan berdasarkan arsitektur yang digunakan, antara lain *deep neural network*, *deep belief network*, *recurrent neural network*, dan *convolutional neural network*, yang mana arsitektur-arsitektur ini telah diterapkan pada ranah pengenalan citra, pengenalan suara, pengolahan bahasa alami (NLP), penerjemahan, bioinformatika, perancangan obat-obatan, hingga analisa citra medis. Pada penelitian sebelumnya oleh (Ciresan, Meier, & Schmidhuber, 2012), juga disimpulkan bahwa pada penerapan-penerapan salah satunya pada klasifikasi citra dan pengenalan pola, kinerja *deep learning* mampu melebihi kinerja pengenalan oleh manusia.

Penggunaan istilah *deep* pada *deep learning* merujuk pada penggunaan beberapa lapisan didalam jaringan arsitektur *neural network* yang digunakan. Pada jaringan-jaringan *shallow network*, hanya terdapat sebuah lapisan tersembunyi (*hidden layer*) antara masukan dan luaran, sebagaimana ditunjukkan pada Gambar 2.3. Sementara itu, pada arsitektur *deep learning*, digunakan lebih dari sebuah lapisan tersembunyi, sebagaimana ditunjukkan pada Gambar 2.4.



Gambar 2.3 Arsitektur *shallow learning*
Sumber: Wardah (2019)



Gambar 2. 4 Arsitektur *deep learning*
Sumber: Wardah (2019)

Pada model *neural network* dengan pelatihan tersupervisi, digunakan suatu proses pembelajaran (*training*) untuk melakukan *fitting* model, terhadap data yang ada. Pada proses pembelajaran, parameter-parameter didalam model diubah untuk melakukan optimasi terhadap parameter yang diinginkan. Pada umumnya, parameter yang dioptimasi adalah parameter fungsi *cost*, yang merupakan fungsi yang menghasilkan nilai rugi-rugi berdasarkan nilai parameter yang digunakan, dengan mengambil referensi data pembelajaran sebagai data masukan. Algoritma pelatihan akan mencoba menekan nilai rugi-rugi yang dihasilkan oleh fungsi *cost*, dengan melakukan perubahan terhadap parameter-parameter didalam model. Kecepatan perubahan umumnya dinotasikan kedalam sebuah *learning rate*. Terdapat beberapa metode perhitungan untuk melakukan pelatihan ini, salah satunya adalah metode *Gradient Descent*.

Metode *Gradient Descent* (*GD*) sebagaimana diusulkan oleh (Lemarecheal, 2012), mencoba melakukan optimasi terhadap suatu fungsi, dengan secara iteratif menggerakkan masukan menuju arah dimana garis singgung kurva fungsi yang akan dioptimasi, mengarah ke arah yang paling negatif. Pada proses pelatihan *deep learning*, pendekatan ini digunakan untuk meminimalisir fungsi *cost*. Sebagai contoh, apabila fungsi *cost* dari sebuah model *deep learning*, dapat dinyatakan dengan Persamaan 2-1.

$$f(m, b) = \frac{1}{N} \sum_{i=1}^n (y_i - (mx_i + b))^2 \dots \dots \dots (2-1)$$

Dimana:

m : parameter *weight*

b : parameter *bias*

n : jumlah data

i : indeks data

Dengan menggunakan persamaan tersebut sebagai fungsi *cost*, maka dapat dicari gradien dari fungsi pada suatu titik, dengan menggunakan Persamaan 2-2.

$$f'(m, b) = \begin{bmatrix} \frac{df}{dm} \\ \frac{df}{db} \end{bmatrix} = \begin{bmatrix} \frac{1}{N} \sum -2x_i(y_i - (mx_i + b)) \\ \frac{1}{N} \sum -2(y_i - (mx_i + b)) \end{bmatrix} \dots\dots\dots (2-2)$$

Dimana:

m : parameter *weight*

b : parameter *bias*

n : jumlah data

i : indeks data

Maka, untuk menemukan gradien, dilakukan iterasi terhadap seluruh titik data yang digunakan, untuk masing-masing nilai *m* dan *b*, dan menghitung turunan persamaan sebgiaan. Gradien baru ini menyatakan nilai lereng dari fungsi *cost* pada posisi parameter saat ini, dan menyatakan arah yang harus ditempuh untuk melakukan perbaruan terhadap parameter didalam model. Besarnya perbaruan parameter dikendalikan oleh nilai *learning rate*.

Pada perkembangannya, metode pembelajaran ini dikembangkan dengan pendekatan berbasis stokastik, yaitu *Stochastic Gradient Descent (SGD)*. Pendekatan ini mengasumsikan distribusi normal terhadap data, dan melakukan *update* terhadap parameter dalam model *deep learning*, setelah melakukan perhitungan fungsi *cost* terhadap sebagian data, dan bukan terhadap seluruh data sebagaimana pendekatan *Gradient Descent*. Sebagai contoh, apabila pada pendekatan GD, terhadap 6500 data, maka pada pendekatan SGD membatasi data pada tiap siklus *update* sebanyak misalkan 32 atau 64 data.

2.2.4 *Technique For Others Reference by Similarity to Ideal Solution*

Technique For Others Reference by Similarity to Ideal Solution (TOPSIS) merupakan metode yang diusulkan oleh (Hwang & Yoon, 1981). Ide utama dari konsep ini adalah sebuah solusi yang mengkompromikan alternatif terbaik yang terdekat dengan solusi ideal positif, dan terjauh dari solusi ideal negatif. Pengurutan dari hasil penjumlahan jarak tersebut, adalah solusi yang direkomendasikan. Jarak yang dimaksud disini adalah jarak

Euclidean yang dihitung dari posisi solusi ideal positif, dan posisi solusi ideal negatif. Metode ini dijabarkan dengan perkembangan oleh (Tzeng & Huang, 2011).

Apabila diberikan himpunan alternatif, $A = \{A_k \mid k = 1, \dots, n\}$, dan himpunan kriteria $C = \{C_j \mid j = 1, \dots, m\}$, dimana $X = \{x_{kj} \mid k = 1, \dots, n; j = 1, \dots, m\}$, menggambarkan rating kinerja dan $w = \{w_j \mid j = 1, \dots, m\}$ adalah pasangan bobot, maka langkah pertama dalam melakukan normalisasi adalah dengan menggunakan persamaan 2-1.

$$r_{kj}(x) = \frac{x_{kj}}{\sqrt{\sum_{k=1}^n x_{kj}^2}}, \quad k = 1, \dots, n; j = 1, \dots, m \quad (2-1)$$

Untuk kriteria manfaat (dimana nilai lebih tinggi lebih baik), maka rating kinerja r_{kj} adalah mengikuti Persamaan 2-2.

$$r_{kj}(x) = \frac{x_{kj} - x_j^-}{x_j^* - x_j^-} \quad (2-2)$$

Dimana:

$$x_j^* = \max_k x_{kj}$$

$$x_j^- = \min_k x_{kj}$$

Atau dapat diubah, dimana x_j^* adalah nilai yang diinginkan dan x_j^- adalah nilai terendah yang mungkin dicapai. Untuk kriteria biaya (dimana lebih rendah adalah lebih baik), maka rating kinerja r_{kj} adalah mengikuti Persamaan 2-3.

$$r_{kj}(x) = \frac{x_j^- - x_{kj}}{x_j^- - x_j^*} \quad (2-3)$$

Dimana:

$$x_j^* = \max_k x_{kj}$$

$$x_j^- = \min_k x_{kj}$$

Dari rating yang diperoleh pada setiap kriteria untuk setiap alternatif, maka dapat dihitung PIS (*Positive Ideal Solution*) dan NIS (*Negative Ideal Solution*) menggunakan persamaan 2-4.

$$PIS = A^+ = \max\{v_1^+(x), v_2^+(x), \dots, v_j^+(x), \dots, v_m^+(x)\} \dots\dots\dots (2-4)$$

Dimana:

A^+ = Nilai solusi ideal positif

v_i^+ = Nilai alternatif solusi pada kriteria

x = Kriteria ke-x

m = jumlah alternatif

Langkah selanjutnya adalah menghitung jarak *Euclidean* masing-masing alternatif, terhadap PIS dan NIS. Perhitungan ini ditunjukkan berturut turut pada Persamaan 2-5 dan Persamaan 2-6.

$$D_k^* = \sqrt{\sum_{j=1}^m [v_{kj}(x) - v_j^+(x)]^2}, k = 1, \dots, n \dots\dots\dots (2-5)$$

Dimana:

D_k = Jarak *Euclidean* nilai alternatif solusi pada kriteria k, terhadap nilai terburuk

v = Nilai alternatif solusi pada kriteria

x = Kriteria ke-x

n = jumlah kriteria

$$D_k^* = \sqrt{\sum_{j=1}^m [v_{kj}(x) - v_j^+(x)]^2}, k = 1, \dots, n \dots\dots\dots (2-6)$$

Dimana:

D_k = Jarak *Euclidean* nilai alternatif solusi pada kriteria k, terhadap nilai terburuk

v = Nilai alternatif solusi pada kriteria

x = Kriteria ke-x

n = jumlah kriteria

2.2.5 Confusion Matrix

Confusion Matrix atau lebih umum dikenal sebagai *contingency table* adalah sebuah matrix yang mungkin saja sangat besar. Pada matriks ini, sebuah tindakan pengklasifikasian yang benar tertuang pada sumbu diagonal matriks. Pada kolom lainnya, seluruh matriks adalah salah. Sebuah algoritma genetika menggunakan sebuah set *rules* untuk menguji kecocokan dari *rules* tersebut terhadap permasalahan yang dihadapi, turunannya (Santra &

Christy, 2012). Dalam hal ini *confusion matrix* digunakan untuk menilai tingkat kecocokan dari proses klasifikasi, dalam mengklasifikasikan suatu keputusan dengan kondisi sebenarnya. Apabila terjadi 10 prediksi positif benar, 5 positif palsu, 4 negatif palsu, dan 11 negatif, benar, maka contoh penggunaan *confusion matrix* dapat ditunjukkan pada tabel 2.10.

Tabel 2.10 Contoh Penggunaan Confusion Matrix

	Prediksi Positif	Prediksi Negatif
Aktual Positif	10	5
Aktual Negatif	4	11

Penggunaan *confusion matrix* dalam melakukan validasi dan verifikasi terhadap suatu model analisis prediktif (dalam hal ini, SPK yang akan dibuat dengan pendekatan AHP-TOPSIS, dan *deep learning*) dapat diukur dengan berbagai kriteria. Kriteria-kriteria tersebut merupakan standar *metrics* dalam penilaian kinerja dari model SPK (Powers, 2008). Beberapa cara analisis dari penggunaan *confusion matrix* adalah untuk mengetahui nilai-nilai seperti sensitifitas, spesifisitas, presisi, npv, akurasi, f1-score, dan lain sebagainya. Nilai ini yang selanjutnya dapat dijadikan pertimbangan untuk melakukan validasi dan verifikasi terhadap model yang telah dibuat dan dilatih.

Dalam penggunaannya, terdapat beberapa cara dalam penggunaan matriks ini, yaitu memanfaatkannya sebagai *fitness*, atau memanfaatkannya untuk menghitung *cost*. Perhitungan *fitness* mengasumsikan permasalahan dimana diinginkan asosiasi yang tinggi, atau nilai kecocokan yang tinggi. Pada perhitungan *cost*, maka sasaran utama permasalahannya adalah optimasi untuk meminimalisir biaya. Keduanya secara inheren adalah sama, dalam arti keduanya digunakan untuk melakukan optimasi dalam penyelesaian suatu masalah, dimana solusi ideal mungkin tidaklah ada. Contoh pengalokasian *cost* terhadap hasil dari *confusion matrix* ditunjukkan pada tabel 2.11.

Tabel 2. 11 Contoh Alokasi Cost Terhadap Hasil Confusion Matrix

	Prediksi Positif	Prediksi Negatif
Aktual Positif	0	1
Aktual Negatif	1	0

Kemudian, pada penggunaannya, setiap keputusan klasifikasi yang diambil, dibandingkan dengan kelas aktual, untuk mendapatkan nilai dari *cost* keputusan tersebut.

Perhitungannya dilakukan dengan menggunakan Persamaan 2-7.

$$C = \sum_{i=1}^m \sum_{j=1}^n C_{ij} \dots \dots \dots (2-7)$$

Dimana:

C = *Cost* total

i = indeks keputusan klasifikasi

j = indeks aktual

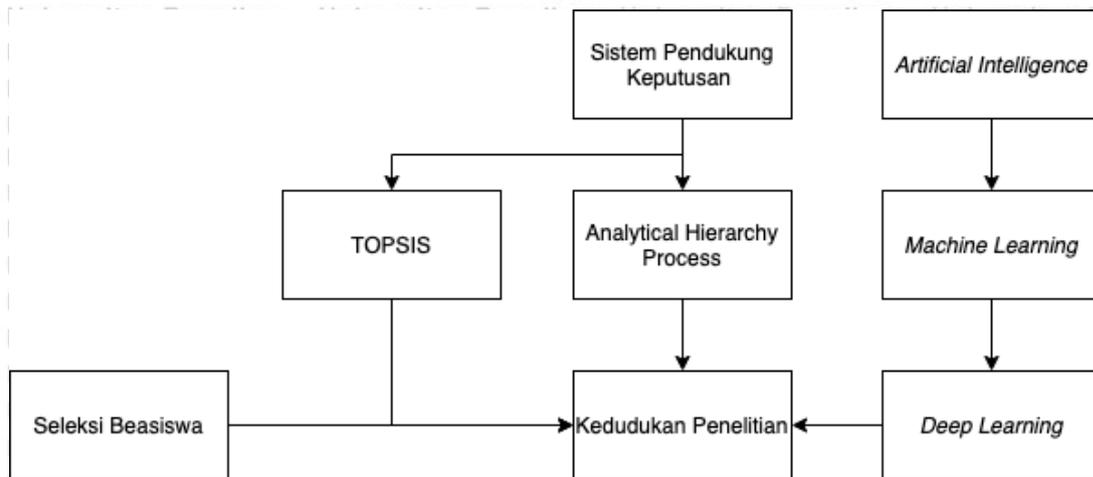
Sebagai contoh, apabila dilakukan 8 kali klasifikasi benar (A sebagai A, dan B sebagai B), dan terdapat dua kesalahan klasifikasi, yaitu satu kali A sebagai B, dan satu kali B sebagai A, maka nilainya dihitung sebagai berikut:

$$C = (8.0) + (1.1) + (1.1) = 2$$

Penggunaan *confusion matrix* untuk perhitungan memungkinkan modifikasi nilai *cost* maupun *fitness* yang digunakan. Suatu permasalahan sangat mungkin memiliki perbedaan dalam pemberian nilai *cost* dari suatu kesalahan klasifikasi. Hal ini sebagai contoh adalah ketika terjadi kesalahan klasifikasi sebuah kanker ganas sebagai kanker jinak, hal ini berimbas jauh lebih buruk terhadap pasien, dibandingkan apabila terjadi klasifikasi kanker jinak sebagai kanker ganas.

2.3 Kedudukan Penelitian

Penelitian ini merupakan persimpangan dari bidang kecerdasan buatan, khususnya topik *deep learning*, dengan sistem pendukung keputusan yang umumnya dikembangkan diatas metode-metode *model-driven*, dalam sebuah permasalahan yang membahas mengenai alokasi beasiswa. Kedudukan penelitian terhadap penelitian-penelitian sebelumnya, ditunjukkan pada Gambar 2.5.



Gambar 2.5 Kedudukan penelitian terhadap penelitian sebelumnya

Pada penelitian ini, pendekatan penggunaan *deep learning* diharapkan mampu menghasilkan SPK dengan kinerja lebih baik dibandingkan pendekatan tradisional lainnya.

Pendekatan tradisional dalam SPK umumnya menggunakan pendekatan seperti AHP dan TOPSIS. Untuk mendapat perbandingan kinerja secara lebih terukur, maka dilakukan perbandingan antara metode berbasis *deep learning* dan pendekatan tradisional gabungan AHP dan TOPSIS.

2.4 Daftar Istilah

Penelitian ini menggunakan beberapa istilah yang digunakan terkait dengan pengembangan Sistem Pendukung Keputusan (SPK) dengan berbagai metode yang digunakan. Penjelasan atas istilah tersebut ditunjukkan pada Tabel 2.12.

Tabel 2.12 Daftar Istilah Yang Digunakan

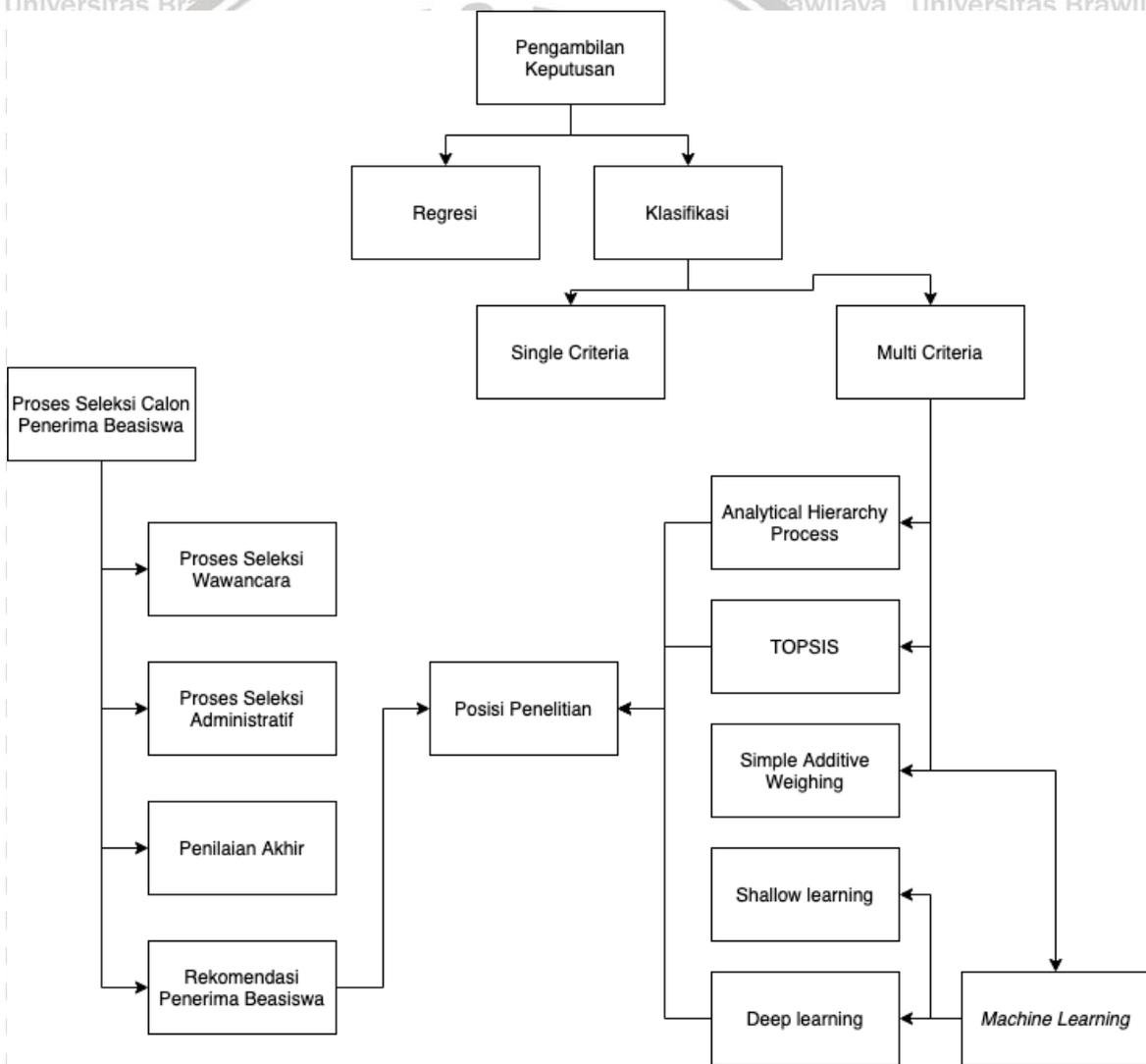
Istilah	Penjelasan
<i>AHP</i>	Metode pembobotan berdasarkan hierarki
<i>Deep learning</i>	Penggunaan model <i>neural network</i> yang disusun dalam beberapa lapisan.
<i>Pairwise Comparison</i>	Perbandingan prioritas antara kriteria yang dilakukan secara berpasangan
TOPSIS	Metode pengambilan keputusan dengan menghitung jarak <i>Euclidean</i>
<i>Accuracy</i>	Ketepatan prediksi <i>true positive</i> dan <i>true negative</i> dibandingkan total prediksi yang dihasilkan.
<i>Specificity</i>	Ketepatan dalam kejadian <i>true negative</i> dibandingkan total prediksi <i>true negative</i> dan <i>false positive</i> .
<i>Confusion Matrix</i>	Matriks pemberian <i>cost</i> kejadian prediksi dibandingkan kejadian aktual.

BAB III

KERANGKA KONSEP PENELITIAN

3.1 Kerangka Berpikir

Kerangka pemikiran dalam penyusunan tulisan ini dituangkan didalam Gambar 3.1. Gambar 3.1. ini menunjukkan hubungan antar metode yang dibahas dalam penelitian ini, yaitu *analytical hierarchy process*, TOPSIS, dan *Deep Learning*, dan beberapa alternatif metode yang tidak digunakan pada penelitian ini. Untuk topik dalam proses seleksi calon penerima beasiswa, dibahas mengenai rekomendasi penerima beasiswa saja.



Gambar 3.1 Kerangka konsep kedudukan metode dalam ranah pengambilan keputusan

Sumber: Data pribadi

Permasalahan utama yang dibahas pada penelitian disini, yaitu menyusun sebuah Sistem Pendukung Keputusan (SPK) dengan pendekatan *deep learning*, untuk melakukan rekomendasi calon penerima beasiswa, berdasarkan beberapa kriteria masukan yang dinilai.

Karena penggunaan beberapa kriteria seleksi ini, maka permasalahan yang dibahas merupakan permasalahan pengambilan keputusan *multi criteria*.

Pendekatan *deep learning* membuat sebuah model pengambilan keputusan *multicriteria*, dengan mempelajari pola yang ada dari data yang dipelajari. Proses pembelajaran ini disebut sebagai pembelajaran tersupervisi (*supervised learning*). Model *deep learning* yang cukup besar diprediksi akan dapat mencapai tingkat akurasi yang tinggi.

Di sisi lain, pendekatan pengambilan keputusan *multicriteria* tradisional, dilakukan menggunakan pendekatan seperti AHP dan TOPSIS. Pendekatan ini melakukan pengambilan keputusan dengan membuat model terlebih dahulu sebelum diujikan pada data. Pada penelitian ini, akan dibuat pula SPK berbasis pendekatan tersebut, agar dapat dibandingkan kinerjanya dengan pendekatan *deep learning*. Kemudian, dilihat dari kebutuhan daya komputasi, *deep learning* membutuhkan daya komputasi yang lebih besar dibandingkan pendekatan AHP dan TOPSIS.

3.2 Hipotesis

Berdasarkan penelitian yang telah dilakukan sebelumnya, maka disusun hipotesis yang akan dibuktikan dalam penelitian ini:

H1: Pendekatan *deep learning* memiliki kinerja lebih baik dibandingkan AHP dan TOPSIS dalam seleksi beasiswa.

3.3 Definisi Variabel Operasional

Variabel operasional penelitian akan didefinisikan sebagai berikut:

1. Kriteria seleksi, antara lain indeks prestasi, prestasi non akademik, dan kriteria lainnya.
2. *Fitness*: yaitu nilai kecocokan calon penerima beasiswa dalam suatu skema beasiswa, akan dijabarkan sebagai F_P , F_A , dan F_M yang berupa kecocokan calon penerima mahasiswa dalam skema beasiswa prestasi, beasiswa bidikmisi, dan beasiswa mandiri.
3. *Akurasi*: yaitu nilai ketepatan prediksi dari sistem pendukung keputusan dengan suatu metode, terhadap alokasi beasiswa yang sebenarnya.

BAB IV

METODE PENELITIAN

Pada bab IV disini akan dijabarkan mengenai metode dan pendekatan yang akan dipakai dalam penelitian ini. Akan dibahas teknik yang akan digunakan, jenis dan cara perolehan data, analisa, langkah dalam penelitian, dan pengujian lanjutan.

4.1 Jenis dan Cara Perolehan Data

Jenis data yang akan dipakai didalam penelitian disini ada dua data kuantitatif, yaitu data calon penerima beasiswa yang didapat dari perguruan tinggi, yang berisi data lengkap untuk pengajuan beasiswa. Kemudian, data alokasi beasiswa sesungguhnya yang diputuskan dan direalisasikan. Data alokasi beasiswa aktual ini akan digunakan untuk mengukur tingkat akurasi dari Sistem Pendukung Keputusan (SPK) dengan membandingkannya terhadap keputusan aktual yang diambil. Berdasarkan sumbernya, kedua data merupakan data primer yang diperoleh langsung dari perguruan tinggi dalam rangka seleksi beasiswa.

4.2 Variabel dan Cara Analisis Data

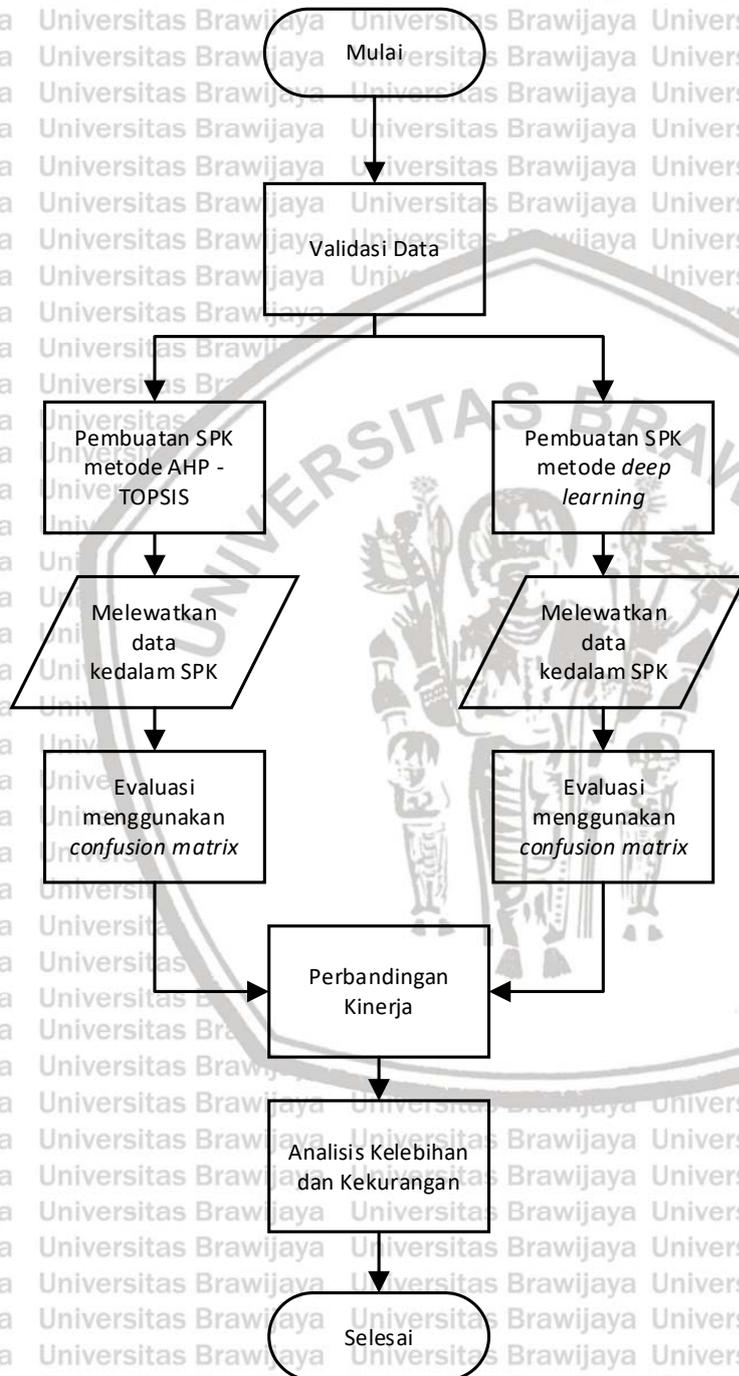
Dalam penelitian ini akan diukur dua buah variabel, yaitu:

1. Akurasi/*Sensitivity*, yaitu ketepatan dari SPK dalam merekomendasikan calon penerima beasiswa dalam suatu skema beasiswa.
2. Spesifisitas, yaitu ketepatan dari SPK dalam tidak merekomendasikan calon penerima beasiswa dalam suatu skema beasiswa.

Analisis dua variabel ini dilakukan dengan membandingkan rekomendasi dari SPK, terhadap data aktual penerima beasiswa. Apabila seorang calon penerima beasiswa direkomendasikan pada suatu skema, dan pada kenyataannya mendapatkan beasiswa tersebut, maka hal ini adalah sebuah *true positive*, dan apabila seorang calon penerima beasiswa tidak direkomendasikan pada suatu skema, dan pada kenyataannya tidak mendapatkan beasiswa tersebut, maka hal ini merupakan sebuah *true negative*. Hasil dari rekomendasi akan dievaluasi menggunakan *confusion matrix* yang akan menerapkan suatu *cost* pada kejadian kesalahan prediksi, baik *false positive* maupun *false negative*

4.3 Tahapan Penelitian

Prosedur untuk menjawab permasalahan yang ada, ditunjukkan pada Gambar 4.1. Penelitian ini akan dimulai dengan menyusun prosedur validasi, untuk memperbaiki data-data yang tidak valid yang tidak dapat digunakan pada pengolahan pendahuluan (*preprocessing*).



Gambar 4.1 Bagan alir tahapan penelitian

Didalam kegiatan penelitian disini dijabarkan tahapan-tahapan sebagai berikut:

1. Validasi Data

Pada tahapan awal disini dilakukan validasi data, yaitu pengecekan terhadap data calon penerima beasiswa yang akan digunakan. Data akan diseleksi dari data-data yang tidak valid. Data-data yang tidak dapat dilakukan *preprocessing* akan dibuang, sementara data yang dapat dilakukan *preprocessing* akan diolah dahulu.

2. Pembuatan SPK metode AHP-TOPSIS

Pada tahapan ini, dilakukan implementasi SPK memakai metode AHP dan TOPSIS. Metode AHP dipakai dalam melakukan penilaian kriteria dalam tiap-tiap skema beasiswa, sementara metode TOPSIS digunakan untuk menentukan skema beasiswa mana yang akan dialokasikan pada calon penerima beasiswa.

3. Melewatkan data ke dalam SPK AHP-TOPSIS

Data calon penerima beasiswa yang telah disiapkan, kemudian diolah menggunakan SPK yang AHP-TOPSIS yang telah dibuat. Kemudian, akan dihasilkan sebuah data rekomendasi alokasi beasiswa terhadap calon penerima beasiswa.

4. Evaluasi SPK AHP-TOPSIS Menggunakan *Confusion Matrix*

Data rekomendasi ini kemudian dievaluasi terhadap data penerima beasiswa aktual menggunakan *confusion matrix* untuk mendapatkan nilai kinerjanya dalam bentuk *cost*.

5. Pembuatan SPK Metode *Deep Learning*

Pada tahapan ini, dilakukan pembuatan sebuah model *deep learning*. Setelah dibuat, model ini kemudian akan dilatih dan diuji menggunakan data pelatihan dengan melakukan *split* data sebanyak 80-20, yaitu 80 data pelatihan, dan 20 data pengujian. Seperti pada SPK AHP-TOPSIS, model ini juga akan diuji menggunakan *confusion matrix* yang sama dengan SPK AHP – TOPSIS.

6. Perbandingan kinerja antara kedua SPK yang telah disusun.

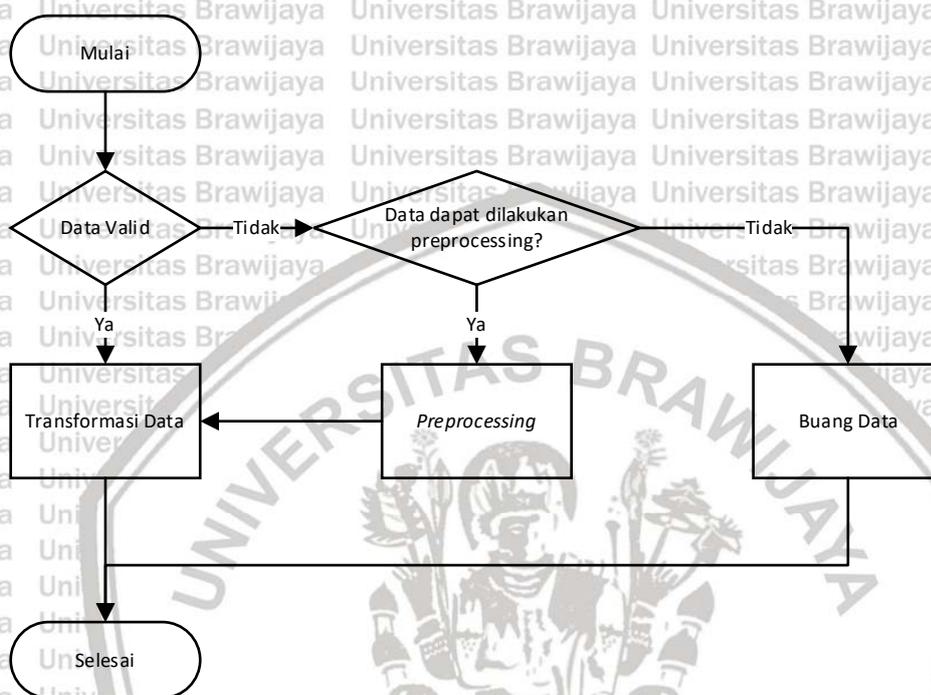
Setelah didapatkan nilai *cost* dari masing-masing SPK, pada tahap ini dilakukan perbandingan kinerja antar kedua SPK, yaitu ketepatan prediksi terhadap data sesungguhnya.

7. Analisis Kelebihan Dan Kekurangan Kedua SPK

dan analisis terhadap hasil dari kedua model SPK yang telah dibuat. Analisis akan dilihat dari segi ketepatan prediksi SPK dibanding data sebenarnya, dan sejauh apa perbedaan antar kedua SPK tersebut.

4.3.1 Validasi Data

Validasi data dilakukan dengan melewati seluruh data calon penerima beasiswa melalui suatu proses validasi data. Proses ini bertujuan untuk memastikan bahwa seluruh data, terutama yang akan digunakan dalam proses pelatihan, adalah data yang valid. Proses validasi data tersebut dijabarkan pada Gambar 4.2



Gambar 4.2 Bagan alur proses validasi data

Tiap unsur data dilewatkan ke dalam proses validasi data. Sebuah unsur pertama-tama akan dicek, apakah memiliki nilai yang valid. Satu titik data dianggap valid apabila seluruh atribut terpenuhi. Apabila data ini valid, maka data akan langsung dilewatkan ke dalam sebuah proses transformasi. Data yang memiliki kekurangan atribut, dianggap sebagai data tidak valid, untuk menjalani seleksi berikutnya. Pada seleksi berikutnya, data yang tidak valid akan dipisahkan ke dalam kategori yang dapat dilakukan *preprocessing*, atau yang tidak dapat dilakukan *preprocessing*. Apabila dapat dilakukan *preprocessing*, maka akan dilakukan sintesis untuk pengisian data terhadap data yang kosong. Apabila tidak dapat dilakukan *preprocessing*, maka data tersebut akan dibuang. Data yang telah melalui *preprocessing* ini kemudian akan diperlakukan sama seperti data yang valid.

Pertama-tama, akan dilakukan proses eksplorasi awal terhadap data. Proses eksplorasi awal data bertujuan untuk melihat struktur data dan bentuk data yang akan digunakan. Proses pengumpulan data sendiri telah dilakukan sebelumnya dengan menggunakan data

yang digunakan pada proses seleksi beasiswa yang telah lampau (periode 2019 – 2020). Untuk itu, langkah ini hanya melihat bentuk data yang telah didapatkan, dan merencanakan prosedur perubahan dan validasi data. Data pertama yang dieksplorasi adalah data untuk beasiswa prestasi, disampaikan pada Gambar 4.3.

Nama	Program Studi	Semester	Kelengkapan Berkas												IPK	IPS Terakad	beasiswa yang pernah	
Muhammad Fatur Roziqin	Teknik Mesin	5	v	v	v	v	v	v	v	v	v	v	v	v	v	3,22	3,25	
Satrio Utomo	Teknik Elektro	7	v	v	nskip nili	v	v	v	v	v	v	v	v	v	v	3,4	3,2	PPA 2019/GENA
Fajar Rahman	Teknik Elektro	7	v	v	v	v	v	v	v	v	v	v	v	v	v	3,14	3,4	tidak pernah
Nurul Wahyudi	Teknik Mekatronika	7	v	v	v	v	v	v	v	v	v	v	v	v	v	3,16	2,90	PPA 2019
Novita Anggraini	Sistem Informasi	3	v	v	v	v	v	v	v	v	v	v	v	v	v	3,7	3,68	tidak pernah
Nur Hidayat	Teknik Elektro	5	v	v	v	v	v	v	v	v	v	v	v	v	v	3,42	3,52	pernah dapet PPA 2
Nurul Lailatul Qamariyah	Teknik Industri	7	v	v	v	v	v	v	v	v	v	v	v	v	v	3,41	3,42	PPA 2017/gasal
Rizqi Afif Khairullah	Sistem Informasi	3	v	v	v	v	v	v	v	v	v	v	v	v	v	3,34	3,3	tidak pernah
Gita Savira P	Teknik Informatika	3	v	v	v	v	v	v	v	v	v	v	v	v	v	3,42	3,5	pernah ikut PPA belur
M. Rizal Palevi	Teknik Mekatronika	5	v	v	v	v	v	v	v	v	v	v	v	v	v	3,16	3,75	
Liscifara Eldyar P	Teknik Informatika	3	v	v	v	v	v	v	v	v	v	v	v	v	v	3,65	3,57	PPA 2019/GENA
Ade Novaliana Sari	Teknik Industri	7	v	v	v	v	v	v	v	v	v	v	v	v	v	3,06	3,31	Tidak Ada
Immatius Mu'arrifah	Teknik Mesin	5	v	v	v	v	v	v	v	v	v	v	v	v	v	3,35	3,27	tidak pernah
Uswatun Hassmah	Teknik Mekatronika	7	v	v	v	v	v	v	v	v	v	v	v	v	v	3,58	3,32	PPA 2017/gasal
Muhamad Hendrik Wicaksono	Teknik Informatika	5	v	v	v	v	v	v	v	v	v	v	v	v	v	3,55	3,52	bidikmisi pengganti tidak
Mukhamad Rifqi Fusti	Teknik Industri	7	v	v	v	v	v	v	v	v	v	v	v	v	v	3,25	3,42	tidak pernah

Gambar 4.3 Pratinjau data penerima beasiswa prestasi

Berdasarkan proses eksplorasi terhadap data calon penerima beasiswa prestasi, dapat dijabarkan beberapa hal yang perlu dilakukan terhadap data yang telah didapatkan. Pertama, karena beasiswa prestasi merupakan skema yang menitikberatkan pada pencapaian masing-masing calon penerima beasiswa, maka beberapa kolom yang tidak terkait tidak perlu dimasukkan, antara lain program studi, semester yang sedang dijalani, data penghasilan, dan semisalnya. Kedua, terdapat beberapa nilai kosong yang tidak diisi. Bergantung pada kolom-kolom tersebut, akan dilakukan proses *imputasi* atau pengisian terhadap data kosong tersebut. Ketiga, terdapat beberapa data yang berbentuk kategori, sehingga perlu dilakukan perubahan ke dalam data kategori. Kemudian, eksplorasi dilakukan terhadap data penerima beasiswa bidikmisi, seperti pada Gambar 4.4.

Nama	NIM	Jurusan	Alamat	No HP	IPK	Pekerjaan Bapak	Pekerja:
Sukris Susanti	150411100055	Teknik informatika	Kecamatan lempuing jaya Kab Ogan komering ilir, Sumsel	82281215179	3,25	Petani	Tidak b
dony rafika putra	180411100014	Teknik informatika	JL. K. Lemah duwur VIII/33	82228696508	3,2	Petani	tidak b
Musthofa Zuhad siroj	150421100088	Teknik industri	JL. Swoko no. 56 rt 2/ rw 2 kec. Lamongan, kab. Lamongan	85785368735	2,78	wiraswasta	Tidak b
ismi qorry aini	150411100093	Teknik informatika	kel. Kraton, bangkalan	87853044958	3,47	wiraswasta	Tidak b
nia handayani	150411100008	Teknik informatika	Kec. Galis, Kranggan timur	85231205180	3,21	Tidak bekerja	Ibu Rui
miftahul jannah	160411100016	Teknik informatika	des. Bukker kec. Jrengik kab. Sampang	8150021000	3,5	TKI	alm
yusuf satyo budi	160491100002	Teknik Mekatro	des. Mugli kec. Kalitengah kab. Lamongan	85731713656	3,05	petani	Ibu Rur
Moch. Rofi	170441100018	Sistem informasi	kamp.sawah banuajuh, kamal	82333015454	3,3	guru	guru
Fajar nur aziz	170421100061	teknik industri	Dsn. Klampisan, kec. Kandangan. Kab. Kediri	89682674657	3,02	tukang bangunan	Tidak b
farikhatus caavidah	170441100000	Sistem informasi	rt. Sekaran kec. Sekaran kab. Lamongan	85731971807	3,61	petani	Ibu Rui

Gambar 4.4 Pratinjau data penerima beasiswa bidikmisi

Proses eksplorasi terhadap data calon penerima beasiswa mandiri menemukan beberapa hal yang perlu ditindaklanjuti. Pertama, karena skema ini bertujuan menyoal calon penerima beasiswa dengan kondisi ekonomi tidak mampu, maka kriteria seleksinya lebih menitikberatkan pada kondisi perekonomian, baru kemudian diikuti pencapaian

masing-masing calon penerima beasiswa. Beberapa kolom yang tidak diperlukan dapat dikeluarkan dari data yang akan digunakan. Kedua, juga didapati beberapa nilai kosong yang tidak diisi. Tergantung dari jenisnya apabila pada kolom tersebut terdapat nilai kosong, dapat dilakukan *imputasi*, namun juga dapat dilakukan pembuangan data tersebut, atau bahkan pembuangan seluruh kolom tersebut. Ketiga, terdapat data-data yang berbentuk data kategorial. Untuk itu perlu dilakukan perubahan menjadi data-data kategorial. Hasil temuan pada eksplorasi data ini menunjukkan bahwa proses seleksi memasukkan beberapa temuan ke dalam lapangan untuk bahan perbandingan. Kemudian, dilakukan eksplorasi terhadap data-data beasiswa mandiri. Seperti pada Gambar 4.5.

No	NIM	Nama	Tempat/Tanggal Lahir	Jurusan	Fakultas	Alamat	No. HP	No. Telp	ipk	orga
1	20202011001	Fahrurrozi	MALANG,	Prodi S1 Teknik Elektro	Fakultas Sains dan Teknologi	Dusun sumbermaron	81331417372	81331417372	2,03	/rijaya
2	20202011002	Muhammad Roff Balhaqi	MALANG,	Prodi S1 Teknik Elektro	Fakultas Sains dan Teknologi	Ironggodo	82131191194	82131191194	2,08	/rijaya
3	20202011003	Muhammad Fajar Balhaqi	MALANG,	Prodi S1 Teknik Elektro	Fakultas Sains dan Teknologi				3	/rijaya
4	20202011004	Mohammad Nashir Subchan	MALANG,	Prodi S1 Teknik Elektro	Fakultas Sains dan Teknologi				3	/rijaya
5	20202011005	Stefano Fiore Al Fitzroh	MALANG,	Prodi S1 Teknik Elektro	Fakultas Sains dan Teknologi				3	/rijaya
6	20202011006	Toni Bayu Prasetya	MALANG,	Prodi S1 Teknik Elektro	Fakultas Sains dan Teknologi				3	/rijaya
7	20202011007	Mohammad Hasan	MALANG,	Prodi S1 Teknik Elektro	Fakultas Sains dan Teknologi				3	/rijaya
8	20202011008	Dedi Romadon	LUMAJANG,	Prodi S1 Teknik Elektro	Fakultas Sains dan Teknologi				3	/rijaya

Gambar 4.5 Pratinjau data penerima beasiswa mandiri

Beasiswa mandiri merupakan beasiswa yang ditetapkan oleh universitas dengan tujuan khusus. Beasiswa ini umumnya dibiayai mandiri oleh universitas dengan tujuan yang disesuaikan dengan visi dan misi universitas. Dalam penelitian ini, data calon penerima beasiswa ini merupakan data yang diusulkan dalam sebuah perguruan tinggi berbasis keagamaan. Karena kekhususan ini, data-data yang tersedia mengenai calon penerima beasiswa mandiri, juga berbeda dari data-data yang disediakan untuk skema beasiswa lainnya. Namun, dari eksplorasi awal yang dilakukan, perlu dilakukan validasi antara lain *imputasi* data kosong, dan kategorisasi terhadap data-data yang perlu dilakukan kategorisasi.

Dalam proses validasi ini, perlu dibuat daftar sebuah tabel, yang berisi mengenai parameter data, yang dapat menentukan validitas sebuah data. Parameter ini didapatkan dari masing-masing data, dan merupakan parameter yang dapat diterapkan kepada ketiga *dataset* yang dipakai pada penelitian. Tabel 4.1. menunjukkan parameter yang digunakan untuk melakukan validasi data.

Tabel 4.1 *Parameter Validasi Data*

No	Parameter	Kriteria Data Valid
1	Keberadaan data	Data harus memiliki nilai, dan tidak boleh kosong
2	Tipe data	Data harus memiliki tipe angka, dan tidak boleh berbentuk <i>string</i> , atau lainnya.
3	Jenis data	Data berbentuk kategori, harus diubah kedalam kode kategori

Secara umum, ada kesamaan permasalahan pada ketiga jenis data beasiswa tersebut. Permasalahan data kosong, pemuatan data pribadi calon penerima beasiswa, penggunaan data non numerik, dan data-data yang berbentuk pilihan merupakan tipe data yang memerlukan proses validasi. Permasalahan dan tindakan yang perlu dilakukan, disampaikan pada Tabel 4.2.

Tabel 4.2 *Prosedur Penyelesaian Permasalahan Data.*

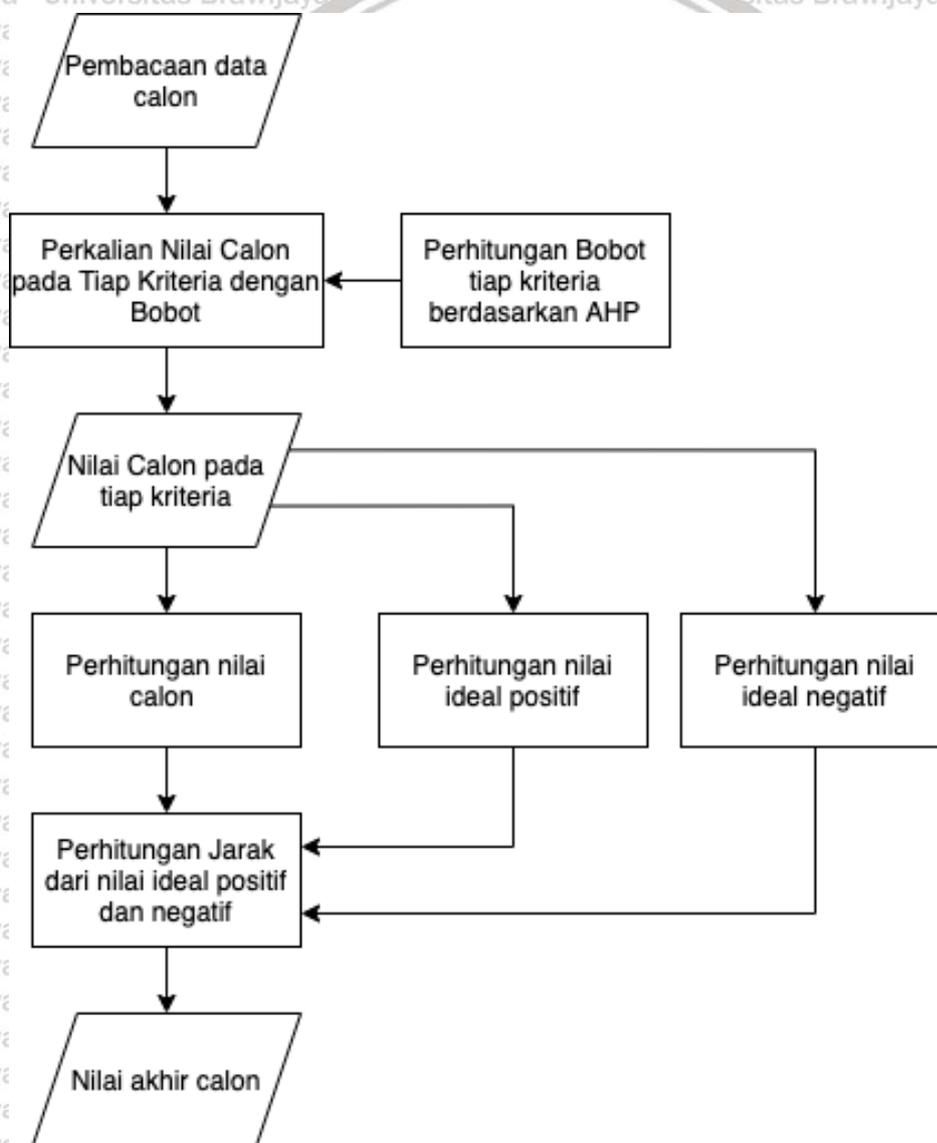
No.	Permasalahan	Tindakan
1.	Data kosong	Pemberian nilai 0 pada data-data kosong
2.	Data pribadi	Data yang terkait pribadi akan dihilangkan (dianonimkan).
3.	Data non numerik	Data diubah menjadi nilai numerik untuk pengolahan selanjutnya.
4.	Data merupakan pilihan dari beberapa alternatif	Pengubahan data menjadi bentuk data kategorial.

Pada proses validasi ini, dilakukan skala terhadap data-data dengan rentang lebar. Semisal, data IPK dengan rentang 1.00 hingga 4.00 diubah skalanya menjadi 0-1. Perubahan ini harus dilakukan untuk pendekatan AHP-TOPSIS, sementara pendekatan *deep learning* akan lebih baik dalam melakukan pengolahan data dan mencapai konvergensi (solusi optimal) ketika datanya telah diskala ke dalam rentang tersebut. Apabila hal ini tidak dilakukan, maka solusi AHP-TOPSIS akan menghasilkan data yang kacau, sementara *deep learning* akan lebih lambat dalam proses pembelajarannya.

4.3.2 Pembuatan SPK Metode AHP-TOPSIS

Pembuatan model kalkulasi untuk SPK pada metode berbasis AHP-TOPSIS dapat dibagi menjadi dua tahapan pendekatan. Tahapan pertama akan mengolah data masukan menggunakan pendekatan AHP, untuk menjadi suatu nilai komposit. Nilai komposit ini kemudian akan diolah menggunakan pendekatan TOPSIS, untuk menghasilkan suatu nilai akhir yang merupakan nilai rekomendasi terhadap seorang calon penerima beasiswa.

Pertama-tama, perlu dibuat model kalkulasi pada SPK dengan pendekatan AHP-TOPSIS. Untuk memperjelas tahapan perhitungan pada SPK AHP-TOPSIS ini, akan dibuat sebuah Gambar diagram alir perhitungannya. Diagram alir perhitungan pada SPK AHP-TOPSIS ini diperlihatkan pada Gambar 4.3.



Gambar 4.6 Diagram alir perhitungan pada algoritma AHP-TOPSIS

Pada tahapan pertama, dibuat model komputasi untuk pengolahan data dengan pendekatan AHP. Pendekatan AHP akan membagi tingkat kepentingan dari masing-masing kriteria menjadi suatu nilai komposit. Karena akan digabungkannya pendekatan AHP dengan pendekatan TOPSIS, maka nilai luaran AHP ini akan dibatasi ke dalam nilai komposit suatu kriteria gabungan. Secara matematis, perhitungan untuk nilai komposit yang dihasilkan oleh AHP adalah sebagaimana Persamaan 4.1 berikut:

$$F_s = \sum_{i=1}^n p_i \cdot w_i \dots\dots\dots (4-1)$$

Dimana:

F_s : Nilai kesesuaian (*fitness*) calon penerima dalam suatu kriteria

i : Kriteria ke - i

n : Jumlah maksimum kriteria

p_i : Nilai kinerja atau capaian calon penerima pada kriteria i

w_i : Bobot kriteria i dalam perhitungan

Pada persamaan diatas, didapatkan sebuah nilai p_i yang merupakan nilai kinerja atau capaian calon penerima pada kriteria i . Nilai ini merupakan nilai seorang calon penerima beasiswa pada kriteria tersebut berdasarkan data yang digunakan. Sebagai contoh, apabila seorang calon penerima beasiswa memiliki nilai IPK 3,5 dan IPK merupakan kriteria ke 6 dalam seleksi, maka nilai calon tersebut untuk $P_6 = 3,5$.

Pada tahapan ini, dilakukan pengembangan aplikasi pada seleksi penerima beasiswa, dengan memakai metode AHP pada penilaian kriteria pada masing-masing skema beasiswa, dan metode TOPSIS untuk penentuan alokasi beasiswa. Dasar perhitungan AHP, adalah dengan melakukan sebuah perbandingan berpasangan (*pairwise comparison matrix*), dimana tingkat kepentingan sebuah kriteria dibandingkan dengan tingkat kepentingan kriteria lainnya. Perbandingan ini diisi oleh otoritas yang berwenang dalam pengambilan keputusan ini (*expert/authority opinion*) berdasarkan dokumen panduan seleksi beasiswa.

Tabel 4.3 Contoh Matriks Perbandingan Berpasangan

	Kriteria-1	...	Kriteria-n
Kriteria-1	1	...	1/p
...
Kriteria-n	p	...	1

Dari perbandingan berpasangan yang telah dilakukan, kemudian dilakukan normalisasi prioritas. Dari hasil normalisasi tersebut, akan didapatkan bobot untuk masing-masing kriteria. Apabila terdapat lebih dari satu tingkat, maka akan dilakukan perbandingan berpasangan terhadap turunannya. Bobot dari kriteria utama, akan didistribusikan kepada kriteria turunan tersebut sesuai proporsi yang dihasilkan dari perbandingan berpasangan.

Kemudian, dilakukan perhitungan nilai kecocokan calon penerima beasiswa dalam sebuah skema beasiswa, dengan menggunakan Persamaan 4-2.

$$F_s = \sum_{i=1}^n p_i \cdot w_i \quad (4-2)$$

Dimana:

F_s : Nilai kesesuaian (*fitness*) calon penerima dalam skema beasiswa s

i : Kriteria ke - i

n : Jumlah maksimum kriteria

p_i : Nilai kinerja atau capaian calon penerima pada kriteria i

w_i : Bobot kriteria i dalam perhitungan

Setiap calon penerima beasiswa dievaluasi kesesuaiannya terhadap ketiga skema beasiswa. Berdasarkan nilai kesesuaiannya tersebut, kemudian akan digunakan metode TOPSIS untuk melakukan evaluasi terhadap satu beasiswa yang mana calon penerima tersebut akan direkomendasikan oleh SPK. Pertama-tama, pada metode ini, dilakukan normalisasi terhadap *fitness* mahasiswa pada masing-masing skema beasiswa (F_s). Normalisasi ini didasarkan pada sebuah nilai yang dihitung menggunakan Persamaan 4-3.

$$\bar{x}_{ij} = \frac{x_{ij}}{\sqrt{\sum_{j=1}^n x_{ij}^2}} \quad (4-3)$$

Dimana:

X_{ij} = Nilai kinerja alternatif j pada kriteria i

i = Kriteria

j = Alternatif

Pada normalisasi, nilai kinerja tiap alternatif pada tiap kriteria kemudian dibagi dengan nilai penormal tersebut, untuk mendapatkan sebuah nilai ternormalisasi. Dari nilai normalisasi ini, kemudian pada masing-masing kriteria, ditentukan sebuah nilai tertinggi (ideal) dan terendah (ideal negatif) dari himpunan alternatif yang ada. Kemudian, masing-masing alternatif akan dihitung jarak *Euclidean* nya terhadap solusi ideal, dan ditambahkan dengan jarak *Euclidean* nya terhadap solusi negatif ideal. Persamaan untuk menghitung jarak *Euclidean* dari solusi ideal dan solusi negatif ideal ditunjukkan berturut-turut yang ada di persamaan 4-4 dan persamaan 4-5.

$$S_i^+ = \left[\sum_{j=1}^m (V_{ij} - V_j^+)^2 \right]^{\frac{1}{2}} \dots \dots \dots (4-4)$$

Dimana:

S = Jarak *Euclidean*

V = Vektor ternormalisasi kinerja

i = Kriteria

j = Alternatif

$$S_i^- = \left[\sum_{j=1}^m (V_{ij} - V_j^-)^2 \right]^{\frac{1}{2}} \dots \dots \dots (4-5)$$

Dimana:

S = Jarak *Euclidean*

V = Vektor ternormalisasi kinerja

i = Kriteria

j = Alternatif

Pada persamaan diatas dibutuhkan sebuah nilai V_j^+ dan nilai V_j^- yang merupakan solusi ideal dari suatu kriteria. Contoh dari nilai ini adalah, bahwa pada data calon penerima beasiswa Bidikmisi, didapatkan dari 46 calon penerima beasiswa, pada kriteria IPK yang merupakan kriteria ke-1, nilai terendah dari seluruh calon adalah 2,78; dan nilai tertinggi adalah 3,85. Maka nilai $V_1^+ = 3,85$ dan nilai $V_1^- = 2,78$.

Dari masing-masing solusi ideal positif dan solusi ideal negatif kemudian didapatkan jarak *Euclidean*, setelah itu dilakukan penilaian kinerja akhir, yaitu dalam kasus ini *fitness* akhir. Perhitungan ini dilakukan menggunakan Persamaan 4-6.

$$P_i = \frac{S_i^-}{S_i^+ + S_i^-} \quad (4-6)$$

Dimana:

P_i = Kinerja akhir

S_i^+ = Jarak *Euclidean* dari solusi positif

S_i^- = Jarak *Euclidean* dari solusi negatif

4.3.3 Melewatkan Data Kedalam SPK AHP-TOPSIS

Data yang telah divalidasi, kemudian dimasukkan kedalam SPK yang telah dibuat berdasarkan metode AHP dan TOPSIS yang telah dibahas pada subbab sebelumnya. Dari data yang dimasukkan kedalam rekomendasi ini akan dihasilkan sebuah nilai rekomendasi sebagaimana ditunjukkan pada contoh Tabel 4.2.

Tabel 4.4 Contoh Rekomendasi Beasiswa

Nama	Rekomendasi B. prestasi	Rekomendasi B. bidikmisi	Rekomendasi B. mandiri
A	1	0	0
B	0	0	1
...
dst

4.3.4 Evaluasi SPK AHP-TOPSIS Menggunakan *Confusion Matrix*

Berdasarkan data rekomendasi yang dihasilkan terhadap para calon mahasiswa, akan dilakukan pengujian akurasi hasil rekomendasi dari SPK metode AHP-TOPSIS terhadap hasil seleksi beasiswa dari tahun sebelumnya. Seleksi dilakukan dengan menerapkan sebuah *cost* sesuai dengan *confusiion matrix* di Tabel 4.3.

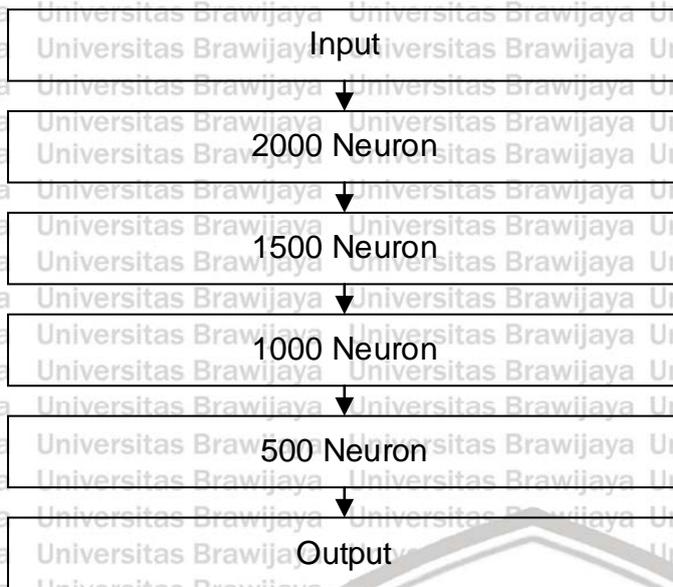
Tabel 4.5 *Confusion Matrix* Untuk Atribusi *Cost*

	Direkomendasikan Beasiswa Skema X	Tidak Direkomendasikan Beasiswa Skema X
Mendapat Beasiswa Skema X	0	1
Tidak Mendapat Beasiswa Skema X	1	0

Berdasarkan *confusion matrix* diatas dilakukan perbandingan rekomendasi dari SPK terhadap penerima beasiswa sebenarnya. Apabila terjadi kesalahan rekomendasi oleh sistem dimana seorang calon penerima direkomendasikan untuk menerima sebuah beasiswa, namun pada kenyataannya tidak mendapat, maka akan distribusikan sebuah *cost*. Nilai *cost* ini dapat diubah untuk lebih condong pada suatu skema beasiswa. Sebagai contoh, mahasiswa dari keluarga kurang mampu akan relatif lebih terdampak buruk ketika gagal mendapatkan beasiswa, dibandingkan apabila seorang mahasiswa berprestasi gagal mendapatkan beasiswa prestasi. Begitu pula dengan beasiswa bidikmisi, yang mana diperlukan untuk mendorong pemerataan pendidikan.

4.3.5 Pembuatan SPK Metode *Deep Learning*

Pembuatan sebuah SPK *deep learning* akan terdiri dari beberapa tahapan. Pertama-tama adalah melakukan validasi data, yang mana hal tersebut sudah disampaikan di sub bab sebelumnya, dan data itu juga digunakan untuk pembuatan SPK berbasis AHP-TOPSIS. Setelah dilakukan validasi data, maka tahapan selanjutnya adalah melakukan perancangan model *deep learning* yang akan digunakan. Model yang akan digunakan, merujuk pada penelitian sebelumnya (Ciresan, Meier, & Schmidhuber, 2012), yang menggunakan MLP (*Multilayer Perceptron*) yang memiliki empat buah lapisan *neuron*. Fungsi aktivasi yang digunakan adalah *rectified linear unit*. Arsitektur ini ditunjukkan pada Gambar 4.3.



Gambar 4.7 Arsitektur model *deep learning multilayer perceptron*

Model ini menerima masukan data dari lapisan *input*, untuk kemudian meneruskannya kepada lapisan-lapisan pengolahan, atau disebut *hidden layer*. Hasil pengolahan kemudian akan ditampilkan sebagai nilai pada bagian *output*, dimana nilai ini merupakan sebuah nilai probabilitas, atau rekomendasi, bahwa seorang siswa layak untuk diusulkan kedalam suatu kategori beasiswa. Perhitungan pada tiap lapisan *hidden layer*, akan menggunakan *neuron* dengan model komputasi *perceptron*, sesuai dengan Persamaan 4-6. Pada persamaan tersebut, terdapat masukan sebanyak i , untuk sebuah luaran pada *perceptron* j .

$$y_j = f_a(w_{1,j} \cdot x_1 + w_{2,j} \cdot x_2 + \dots + w_{i,j} x_i + b_j) \dots \dots \dots (4-6)$$

Dimana:

y : unsur luaran *perceptron*

x : unsur masukan *perceptron*

f_a : fungsi aktivasi

b : bias

w : bobot

i : indeks masukan

j : indeks luaran

n : indeks masukan ke- n

Setelah dilakukan pembuatan model *deep learning* ini, maka akan dilakukan proses pelatihan terhadap model. Pelatihan ini bertujuan untuk mengubah parameter yang terdapat didalam model, yaitu *weight* dan *bias* didalamnya. Perubahan terhadap parameter-parameter didalam model ini akan didasarkan pada metode *Stochastic Gradient Descent*, dimana nilai parameter yang dimiliki saat ini akan dimasukkan kedalam fungsi *cost* untuk menentukan *update* terhadap parameter yang dimiliki, untuk menekan fungsi *cost*. Fungsi *cost* yang akan digunakan untuk melakukan evaluasi terhadap kinerja model *deep learning* ini merupakan fungsi sederhana yang menghitung nilai *cost*, yaitu nilai dari prediksi yang dihasilkan, berdasarkan *confusion matrix* pada Tabel 4.3. Fungsi *cost* ini ditunjukkan pada Persamaan 4-7.

$$f_c(m, b) = \frac{1}{n} \sum_{i=1}^n (h_{\theta}(x_i) - y_i)^2 \dots\dots\dots (4-7)$$

Dimana:

f_c = fungsi *cost*

m = bobot (*weight*)

b = *bias*

n = jumlah data evaluasi, pada penelitian ini menggunakan 32 data tiap *batch*

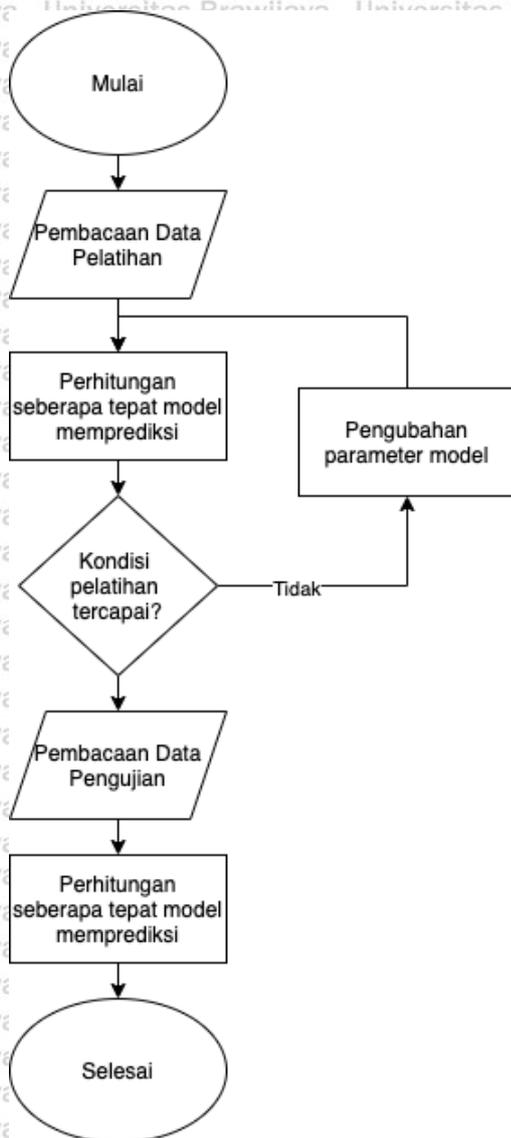
h_{θ} = fungsi prediksi dengan m dan b sebagai parameter

x_i = masukan data ke i

y_i = luaran data ke i

Berdasarkan *cost* yang dihasilkan, maka seluruh parameter yang terdapat didalam model *deep learning* ini akan diperbaharui dengan tujuan untuk menurunkan *cost*, yang diharapkan pada akhirnya akan dapat meningkatkan akurasi dari model ini. Namun, untuk memastikan bahwa tidak terjadi *overfitting* terhadap model dalam pelatihan, maka akan dilakukan sebuah validasi, dengan cara memisahkan data yang akan digunakan untuk pelatihan, kedalam 2 buah set data. Data penerima beasiswa tahun sebelumnya akan dipecah dalam rasio 80-20. Data yang lebih banyak akan digunakan untuk melatih model, dan data yang lebih sedikit akan digunakan untuk melakukan uji validasi, dimana model *deep learning* akan divalidasi dengan melihat kinerjanya terhadap kumpulan data yang belum pernah dilihat sebelumnya.

Implementasi dari model *deep learning* ini akan dibuat dengan memakai bahasa program *python*, dengan memakai bantuan beberapa pustaka (*library*) yang mendukung implementasi *deep learning* pada *python*. Beberapa situs telah menyediakan bantuan untuk pemrograman dan pelatihan terhadap model *deep learning* secara *online*, dengan menggunakan model eksekusi berbasis *notebook*. *Flowchart* yang menggambarkan proses pelatihan dan perbaruan parameter dalam model *deep learning* disini, diperlihatkan di Gambar 4.8.



Gambar 4.8 Diagram alir algoritma berbasis *deep learning*

Untuk masing-masing skema beasiswa, akan dibuat model yang berbeda. Model dibuat dengan bantuan pustaka (*library*) perangkat lunak *tensorflow*.

4.3.6 Perbandingan Kinerja Kedua SPK

Pada tahapan ini, dilakukan penilaian kinerja akhir, dengan membandingkan *cost* atau *loss* yang dihasilkan masing-masing SPK dalam merekomendasikan calon penerima beasiswa. *Cost* yang lebih rendah mengindikasikan bahwa SPK tersebut memiliki kinerja yang lebih baik dibandingkan SPK yang memiliki *cost* lebih tinggi. Selain itu, akan dilakukan perbandingan kecepatan eksekusi kedua jenis SPK. Pada model *deep learning*, kecepatan eksekusi akan diukur secara programatik, sementara pada model AHP-TOPSIS, pengukuran juga akan dilakukan secara programatik. Hasil kebutuhan eksekusi program diduga akan sangat singkat, namun akan lebih lambat pada model berbasis *deep learning* karena jumlah kalkulasi yang harus dilakukan jauh lebih banyak. Pada model *deep learning*, kecepatan eksekusi yang diukur akan didasarkan pada model yang telah selesai dilatih, dan tidak akan memasukkan waktu latihan yang dibutuhkan.

4.3.7 Analisis Kelebihan dan Kekurangan Kedua SPK

Setelah dilakukan perbandingan antar kinerja kedua SPK, pada tahapan ini dilakukan analisis, dengan mendasarkan pada kinerja masing-masing SPK. Kedua SPK dibuat dengan menggunakan tipe pendekatan berbeda, yang mana model pertama mengutamakan pembuatan model pengambilan keputusan berdasarkan konsep ideal yang telah dijabarkan, sementara konsep kedua mendasarkan pada keputusan yang telah dibuat sebelumnya. Hal ini memperkuat hipotesis awal dimana akan terjadi perbedaan antar kedua SPK.

4.3.8 Penyimpanan Hasil Pelatihan

Pembuatan SPK telah berhasil dilakukan, dan diperlukan cara untuk menyimpannya. Pada SPK pertama yang berbasis AHP-TOPSIS, data diolah untuk mengambil keputusan berdasarkan *batch*, sehingga proses perhitungan akan diulang untuk setiap *batch* data. Pada model berbasis *deep learning*, akan digunakan fungsi `model.save()` dan `model.load()` untuk menyimpan model yang telah dilatih.

4.4 Waktu Penelitian

Waktu penelitian disini akan dilakukan kurang lebih 4 bulan antara bulan Januari 2021 sampai bulan April 2021. Jadwal penelitian ini disampaikan seperti di Tabel 4.5.

Tabel 4.6 Jadwal Penelitian

No	Tahap	Bulan			
		Januari	Februari	Maret	April
1	Persiapan Penelitian	x			
2	Validasi Data Masukan	x			
3	Pembuatan SPK	x	x		
4	Pengujian SPK		x		
5	Penyusunan Artikel Jurnal		x	x	
6	Penyusunan Laporan Diakhir Penelitian			x	x



BAB V

HASIL DAN PEMBAHASAN

5.1 Persiapan Data

Sebelum dilaksanakan pembuatan purwarupa SPK, perlu dilakukan persiapan terhadap data yang akan digunakan untuk melakukan pembuatan dan pengujian SPK. Persiapan terhadap data ini terdiri dari tahap eksplorasi awal, yang dilanjutkan dengan proses validasi terhadap data yang akan digunakan.

Proses eksplorasi terhadap data dilakukan untuk melihat data-data, dan merencanakan tindakan yang perlu dilakukan terhadap data-data tersebut, agar data-data tersebut dapat digunakan. Data-data yang lengkap akan dapat langsung digunakan. Sementara itu, data-data yang kurang lengkap, perlu dilengkapi dengan melakukan beberapa tindakan, antara lain dengan *imputasi* (memasukkan nilai) terhadap data tersebut. Prosedur *imputasi* tersebut akan dijelaskan pada tahapan selanjutnya.

5.1.1 Hasil Proses Validasi Data

Pada tahapan ini, proses validasi telah dilakukan terhadap data. Dari proses ini dihasilkan beberapa kesimpulan dan tindakan yang perlu dilakukan agar data menjadi valid dan dapat digunakan didalam proses selanjutnya, yaitu dalam pembuatan SPK. Proses ini terkait dengan pengubahan data-data kategorial, dan *imputasi* terhadap data-data kosong. Prosedur validasi data akan berbeda dan dilakukan secara terpisah terhadap masing-masing data calon penerima beasiswa yang akan digunakan. Hal ini terkait dengan data-data apa saja yang tersedia untuk masing-masing beasiswa. Pada seleksi beasiswa bidikmisi, umumnya tersedia data-data finansial dari calon penerima beasiswa. Namun, pada seleksi beasiswa prestasi, umumnya hanya tersedia data-data terkait akademik. Untuk beasiswa mandiri, umumnya penekanannya mengenai keaktifan mahasiswa pada keorganisasian.

Proses pertama berhasil dilakukan terhadap data-data yang digunakan pada proses seleksi beasiswa prestasi. Tindakan yang dilakukan terhadap data-data pada proses seleksi beasiswa prestasi seperti di Tabel 5.1.

Tabel 5.1 Hasil Validasi Data Beasiswa Prestasi

No	Permasalahan	Tindakan
1	Data kosong	Pemberian nilai 0 pada data-data kosong
2	Data pribadi calon penerima beasiswa	Data akan dianonimkan, dimana data seperti nama dan kontak dihilangkan.
3	Data terisi bukan nilai numerik	Untuk data bukan nilai, dilakukan konversi menjadi nilai angka (numerik), semisal kepemilikan suatu berkas kelengkapan calon penerima.
4	Data terdiri dari beberapa kategori	Pengubahan data menjadi data kategorial, semisal program studi calon penerima beasiswa.

Untuk permasalahan data kosong yang ditemui, dilakukan pengisian nilai 0 terhadap kekosongan data, dengan asumsi bahwa nilai tidak dimasukkan karena tidak tersedia atau tidak dapat diterapkan terhadap data yang bersangkutan. Hal ini diterapkan pada kolom-kolom, utamanya tentang ketersediaan berkas-berkas penunjang seleksi calon penerima beasiswa prestasi.

Data perlu dianonimkan karena terkait dengan data pribadi mahasiswa. Untuk itu, data yang terkait dengan nama calon penerima beasiswa akan dihilangkan, sehingga hanya tersedia data terkait dengan calon penerima beasiswa itu sendiri. Data-data yang tersisa hanya data yang akan digunakan untuk seleksi menggunakan sistem yang akan dirancang.

Terhadap data non numerik, akan dilakukan konversi menjadi data numerik. Semisal data mengenai kepemilikan berkas penunjang, diisi dengan (v) dan (-) untuk menandakan kepemilikan atau tidak memiliki berkas tersebut. Data ini akan dikonversi berturut turut menjadi nilai 1 dan 0.

Kemudian, terhadap data yang berbentuk kategori, akan diubah menjadi bentuk kategori. Data berbentuk kategori ini merupakan yang terdiri dari beberapa nilai, namun tidak ada nilai antara dari beberapa nilai tersebut. Hal ini terjadi misalnya pada data program

studi calon penerima beasiswa. Selanjutnya, dilakukan validasi terhadap data calon penerima beasiswa Bidikmisi. Proses validasi ini diperlihatkan di Tabel 5.2.

Tabel 5.2 Hasil Validasi Data Beasiswa Bidikmisi

No	Permasalahan	Tindakan
1	Data kosong	Pemberian nilai 0 pada data-data kosong
2	Data pribadi calon penerima beasiswa harus anonim.	Data pribadi seperti nama, nim, alamat, dan nomor hp akan dihilangkan.
3	Data terisi bukan nilai numerik	Dilakukan konversi data menjadi nilai numerik.
4	Data terdiri dari beberapa kategori	Pengubahan data menjadi data kategorial,

Terhadap permasalahan data kosong yang ditemui pada data calon penerima beasiswa, diisi dengan nilai 0. Hal ini berdasarkan asumsi bahwa kekosongan data merupakan akibat tidak adanya data tersebut, atau tidak dapat diterapkannya nilai tersebut terhadap data. Hal ini diterapkan pada kolom-kolom semisal biaya air (karena tidak semua rumah tersambung PDAM), dan PBB (tidak semua rumah tangga melebihi batas minimum pembayaran).

Selanjutnya dilakukan penghilangan data pribadi untuk menjadikan data calon penerima beasiswa menjadi anonim. Data-data seperti nama, NIM, nomor kontak, dan alamat calon penerima beasiswa akan dihilangkan. Data-data yang tersisa hanya data yang akan digunakan untuk seleksi menggunakan sistem yang akan dirancang.

Terhadap data non numerik, akan dilakukan konversi menjadi data numerik. Semisal data mengenai penerimaan bantuan, diisi dengan (y) dan (t) untuk menandakan pernah menerima atau tidak pernah menerima bantuan jenis tersebut. Data ini akan dikonversi berturut turut menjadi nilai 1 dan 0.

Kemudian, terhadap data yang berbentuk kategori, akan diubah menjadi bentuk kategori. Data berbentuk kategori ini merupakan yang terdiri dari beberapa nilai, namun tidak ada nilai antara dari beberapa nilai tersebut. Hal ini terjadi misalnya pada data program studi calon penerima beasiswa, dan data jenis alat transportasi yang digunakan. Selanjutnya,

dilakukan validasi terhadap data calon penerima beasiswa mandiri. Proses validasi ini diperlihatkan di Tabel 5.3.

Tabel 5.3 Hasil Validasi Data Beasiswa Mandiri

No	Permasalahan	Tindakan
1	Terdapat data-data kosong	Pemberian nilai 0 pada data-data kosong
2	Data pribadi harus anonim	Data pribadi yang dapat mengidentifikasi mahasiswa perlu dihilangkan.
3	Data non numerik	Konversi data dari non numerik menjadi numerik
4	Data yang merupakan kategori ditulis sebagai non kategori	Pengubahan data menjadi data kategorial,

Terhadap permasalahan data kosong yang ditemui pada data calon penerima beasiswa, diisi dengan nilai 0. Hal ini berdasarkan asumsi bahwa kekosongan data merupakan akibat tidak adanya data tersebut, atau tidak dapat diterapkannya nilai tersebut terhadap data. Hal ini diterapkan pada kolom-kolom semisal keanggotaan organisasi, atau prestasi yang pernah dicapai sebelumnya.

Selanjutnya dilakukan penghilangan data pribadi untuk menjadikan data calon penerima beasiswa menjadi anonim. Data-data seperti nama, dan NIM, harus dihilangkan sebelum digunakan lebih lanjut. Data-data yang tersisa hanya data yang akan digunakan untuk seleksi menggunakan sistem yang akan dirancang.

Terhadap data non numerik, akan dilakukan konversi menjadi data numerik. Semisal data mengenai pernah menerima beasiswa, diisi dengan (y) dan (t) untuk menandakan pernah menerima atau tidak pernah menerima bantuan jenis tersebut. Data ini akan dikonversi berturut turut menjadi nilai 1 dan 0.

Kemudian, terhadap data yang berbentuk kategori, akan diubah menjadi bentuk kategori. Data berbentuk kategori ini merupakan yang terdiri dari beberapa nilai, namun tidak ada nilai antara dari beberapa nilai tersebut. Hal ini terjadi misalnya pada data program studi calon penerima beasiswa.

Dengan telah dilakukannya validasi terhadap tiga jenis data calon penerima beasiswa, kegiatan penelitian dilanjutkan pada tahapan selanjutnya. Tahapan selanjutnya akan membahas mengenai pembuatan SPK untuk metode AHP-TOPSIS, dan metode *deep learning*.

5.2 Pembuatan SPK Metode AHP-TOPSIS

Pembuatan SPK metode AHP-TOPSIS ini terbagi menjadi tiga langkah utama, yaitu dimulai dari pembuatan model, melewati data kedalam SPK, dan diakhiri dengan evaluasi kinerja dari model SPK. Diharapkan dari langkah ini akan didapatkan suatu model SPK yang dapat membantu proses pengambilan keputusan, yang berbasis pada metode AHP-TOPSIS.

5.2.1 Pembuatan Model Kalkulasi SPK AHP-TOPSIS.

Pada permasalahan pertama yaitu beasiswa prestasi, perlu dibuat sebuah proses dengan pendekatan AHP untuk melakukan komposit dari data-data calon penerima beasiswa menjadi sebuah data komposit nilai. Untuk pertama-tama, perlu dikategorikan seluruh data-data masukan menjadi sebuah kriteria komposit. Kriteria komposit itu disusun melalui pendapat ahli (*expert opinion*) atas beberapa data masukan kriteria dari calon penerima beasiswa diperlihatkan di Tabel 5.4

Tabel 5.4 *Sub-kriteria Penyusun Kriteria Komposit*

No	Kriteria Komposit	Sub-Kriteria
1	Administratif	12 buah kelengkapan berkas
2	Prestasi akademik	Indeks prestasi semester dan indeks prestasi kumulatif
3	Beasiswa sebelumnya	Sebelum pengajuan ini, mahasiswa pernah menerima beasiswa
4	Prestasi non akademik	Keanggotaan organisasi, dan prestasi non akademik lainnya.

Dilakukan penilaian memakai matriks perbandingan berpasangan (*pairwise comparison matrix*) berdasarkan proses penentuan bobot AHP untuk menilai bobot dari

masing-masing kriteria. Bobot yang didapatkan dari perbandingan berpasangan ini ditunjukkan di Tabel 5.5.

Tabel 5.5 *Bobot Masing-Masing Kriteria Komposit*

No	Kriteria Komposit	Bobot
1	Administratif	0.102
2	Prestasi akademik	0.448
3	Beasiswa sebelumnya	0.102
4	Prestasi non akademik	0.348

Setelah didapatkan nilai bobot untuk masing-masing kriteria komposit, langkah selanjutnya adalah menentukan bobot masing-masing sub-kriteria. Perhitungan bobot dengan menggunakan matriks perbandingan berpasangan untuk kriteria administratif, dan prestasi akademik disampaikan di Tabel 5.6 dan Tabel 5.7. Dua kriteria lain yaitu prestasi non akademik dan beasiswa sebelumnya, merupakan kriteria tunggal sehingga tidak memerlukan perbandingan berpasangan.

Tabel 5.6 *Bobot Subkriteria Administratif*

No.	Sub-kriteria	Bobot	Bobot Global
1	berkas_permohonan	0,083	0,0085
2	berkas_formulir	0,083	0,0085
3	berkas_khs	0,083	0,0085
4	berkas_ktm	0,083	0,0085
5	berkas_krs	0,083	0,0085
6	berkas_rekomendasi	0,083	0,0085
7	berkas_kartu_keluarga	0,083	0,0085
8	berkas_penghasilan	0,083	0,0085
9	piagam_prestasi	0,083	0,0085
10	aktif_kuliah	0,083	0,0085
11	surat_bebas_beasiswa	0,083	0,0085
12	surat_skck	0,083	0,0085

Tabel 5.7 Bobot Subkriteria Prestasi Akademik

No	Sub-kriteria	Bobot	Bobot Global
1	nilai_ipk	0,8	0,3584
2	nilai_ips	0,2	0,0896

Kemudian, pada permasalahan kedua, yaitu beasiswa bidikmisi, akan dilakukan langkah yang sama. Data-data dari calon penerima beasiswa akan dikategorikan kedalam empat buah kriteria komposit, yaitu prestasi, kondisi finansial, bantuan terdahulu, dan beban. Kriteria komposit ini merupakan kriteria baru yang terkomposisi dari beberapa buah sub-kriteria dari data masukan yang akan digunakan. Penentuan bobot dari masing-masing kriteria komposit ini akan menggunakan metode yang sama dengan sebelumnya, yaitu menggunakan matriks perbandingan berpasangan (*pairwise comparison matrix*). Bobot ini ditunjukkan di Tabel 5.8.

Tabel 5.8 Bobot Masing-Masing Kriteria Komposit Beasiswa Bidikmisi

No	Kriteria Komposit	Bobot
1	Prestasi	0,102
2	Kondisi Finansial	0,448
3	Bantuan Terdahulu	0,102
4	Beban	0,348

Setelah didapatkan nilai bobot untuk masing-masing kriteria komposit, langkah selanjutnya adalah menentukan bobot masing-masing sub-kriteria. Perhitungan bobot dengan menggunakan matriks perbandingan berpasangan untuk kriteria prestasi, kondisi finansial, bantuan terdahulu (yang pernah diterima), dan beban. Prestasi hanya disusun dari kriteria IPK. Pembobotan ditunjukkan di Tabel 5.9, 5.10, dan 5.11.

Tabel 5.9 *Bobot Subkriteria Kondisi Finansial*

No	Sub-kriteria	Bobot	Bobot Global
1	pekerjaan_ayah	0,167	0,075
2	pekerjaan_ibu	0,167	0,075
3	pendidikan_ayah	0,083	0,037
4	penghasilan_ayah	0,330	0,148
5	pendidikan_ibu	0,083	0,037
6	penghasilan_ibu	0,330	0,148

Tabel 5.10 *Bobot Subkriteria Bantuan Terdahulu*

No	Sub-kriteria	Bobot	Bobot Global
1	bantuan_raskin	0,2	0,0204
2	bantuan_bpjs	0,2	0,0204
3	bantuan_blt	0,2	0,0204
4	bantuan_pkh	0,2	0,0204
5	bantuan_beasiswa	0,2	0,0204

Tabel 5.11 *Bobot Subkriteria Beban*

No	Sub-kriteria	Bobot	Bobot Global
1	pengeluaran_bulanan	0.3125	0.1087
2	ukt	0.0625	0.0217
3	keluarga_tanggung	0.3125	0.1087
4	uang_kiriman	0.3125	0.1087

Kemudian, pada permasalahan ketiga, yaitu beasiswa mandiri, akan dilakukan langkah yang sama. Data-data dari calon penerima beasiswa akan dikategorikan kedalam empat buah kriteria. Karena beasiswa ini tidak membutuhkan banyak kriteria, maka hanya terdapat empat buah kriteria, dengan pembobotan dilakukan menggunakan pendekatan matriks perbandingan berpasangan (*pairwise comparison matrix*). Bobot ini ditunjukkan di Tabel 5.12.

Tabel 5.12 *Bobot Kriteria Beasiswa Mandiri*

No	Sub-kriteria	Bobot
1.	Prestasi akademik	0.2222
2.	Prestasi non akademik	0.1111
3.	Organisasi	0.2222
4.	Organisasi kekhususan	0.4444

Setelah dilakukan perhitungan bobot dengan pendekatan AHP, langkah selanjutnya adalah melakukan pendekatan TOPSIS untuk melakukan perhitungan terhadap seluruh alternatif. Pendekatan TOPSIS menilai suatu kriteria sebagai nilai relatif terhadap suatu nilai lainnya. Karena itu, pendekatan ini perlu dilakukan menggunakan data, dan terlebih dahulu dibuat sebuah normalisasi terhadap seluruh aspek masukan kriteria. Proses normalisasi ini mengambil nilai tertinggi dari tiap kriteria, dan membagi seluruh nilai dengan nilai tersebut, untuk mendapatkan suatu nilai dengan rentang dari 0 hingga 1. Setelah didapatkan nilai yang dinormalisasi, seluruh nilai kemudian dikalikan dengan bobot yang telah dibahas.

5.2.2 Melewatkan Data Kedalam SPK Metode AHP-TOPSIS

Kemudian, setelah didapatkan nilai kriteria komposit dari seluruh data yang ada, langkah selanjutnya adalah menentukan nilai positif dan negatif dari seluruh solusi yang ada. Untuk semua kriteria yang bersifat bermanfaat (misalkan nilai), maka nilai solusi positif ideal adalah nilai tertinggi, dan nilai solusi negatif adalah nilai terendah. Untuk kriteria yang bersifat tidak bermanfaat (misalkan sudah pernah menerima beasiswa), maka nilai solusi positif ideal adalah nilai terendah, dan nilai solusi negatif adalah nilai tertingginya. Pencarian nilai positif dan negatif ideal ini dilakukan dan diperlihatkan di Tabel 5.13.

Tabel 5. 13 *Solusi Positif dan Negatif Ideal*

Skema	Kriteria	Positif Ideal	Negatif Ideal
Prestasi	Administratif	0.102	0
	Prestasi akademik	0.444	0.029
	Beasiswa sebelumnya	0	0.102
	Prestasi non akademik	0.348	0
Bidikmisi	Prestasi	0.102	0
	Kondisi Finansial	0.044	0.419
	Bantuan Terdahulu	0.021	0.061
	Beban	0.348	0.048
Mandiri	Prestasi akademik	0.222	0
	Prestasi non akademik	0.111	0
	Organisasi	0.222	0
	Organisasi kekhususan	0.444	0

Setelah didapatkan nilai positif dan negatif ideal, maka proses berikutnya yaitu melakukan perhitungan S_i^+ dan S_i^- pada semua data calon penerima beasiswa pada masing-masing skema. Setelah dilakukan perhitungan S_i^+ dan S_i^- , kemudian dilakukan juga perhitungan kinerja P_i untuk masing-masing calon penerima beasiswa pada masing-masing beasiswa. Dari data ini akan dilakukan evaluasi kinerja dari SPK dalam menghasilkan sistem rekomendasi.

5.2.3 Evaluasi Kinerja SPK

Salah satu permasalahan yang ada dari SPK sistem ini adalah bahwa sistem ini menghasilkan suatu nilai rekomendasi dibandingkan nilai-nilai lainnya. Nilai ini tidak banyak memiliki arti, ketika luaran yang diinginkan adalah nilai rekomendasi dalam bentuk 0 atau 1. Untuk itu, perlu dibuat sebuah nilai ambang batas yang mana semua calon penerima beasiswa yang memiliki nilai lebih tinggi dibandingkan nilai tersebut dianggap telah memenuhi syarat untuk direkomendasikan menerima beasiswa, dan begitu pula

sebaliknya. Berdasarkan data, jumlah mahasiswa yang diterima adalah sebagaimana Tabel 5.14.

Tabel 5.14 Jumlah Calon Penerima Beasiswa Yang Mendapatkan Beasiswa

No	Skema	Jumlah Pendaftar	Jumlah Diterima
1	Prestasi	171	69
2	Bidikmisi	46	10
3	Mandiri	97	17

Berdasarkan nilai tersebut, maka penentuan keputusan rekomendasi untuk seorang calon penerima beasiswa, akan didasarkan atas nilai jumlah diterima tersebut. Hal ini berarti bahwa calon penerima beasiswa yang direkomendasikan SPK adalah calon dengan nilai tertinggi sebanyak jumlah diterima tersebut. Hasil evaluasi dilakukan dengan membandingkan hasil rekomendasi oleh SPK, dengan data penerima sebenarnya. Dengan mengambil nilai sebanyak n-jumlah diterima nilai tertinggi sebagai prediksi direkomendasikan oleh SPK, dan sisanya sebagai prediksi tidak direkomendasikan. Kemudian akurasi dihitung dengan membandingkan jumlah prediksi benar (sesuai data aktual penerima beasiswa) terhadap seluruh prediksi yang dilakukan. Dari perhitungan tersebut, didapatkan nilai akurasi sebagaimana Tabel 5.15.

Tabel 5.15 Akurasi Rekomendasi SPK Terhadap Hasil Sebenarnya

No	Skema	Akurasi Prediksi
1	Prestasi	56.72 %
2	Bidikmisi	65.21 %
3	Mandiri	95.87 %

5.3 Pembuatan SPK Metode *Deep Learning*

Pada tahapan selanjutnya, akan dilakukan pembuatan sebuah model rekomendasi keputusan dengan berdasarkan pendekatan *deep learning*. Model yang akan dibuat merupakan sebuah model *dense network* sebagaimana dijabarkan pada bab sebelumnya.

5.3.1 Pembuatan Model

Model yang digunakan untuk skema beasiswa prestasi akan menggunakan lebar data masukan 17. Begitu pula dengan model beasiswa bidikmisi dan beasiswa mandiri, akan menggunakan lebar masukan yang berbeda. Parameter untuk model beasiswa prestasi ditunjukkan pada Gambar 5.1.

Layer (type)	Output Shape	Param #
dense_25 (Dense)	(None, 2000)	36000
dense_26 (Dense)	(None, 1500)	3001500
dense_27 (Dense)	(None, 1000)	1501000
dense_28 (Dense)	(None, 500)	500500
dense_29 (Dense)	(None, 1)	501
Total params: 5,039,501		
Trainable params: 5,039,501		
Non-trainable params: 0		

Gambar 5.1 Ikhtisar model untuk beasiswa prestasi

Dari model yang telah dibuat di atas, ditemukan bahwa setidaknya ada 5.039.501 parameter pada model ini, yang memungkinkan model untuk mempelajari pola pada data beasiswa prestasi yang telah disediakan. Kemudian, pembuatan model berbasis *deep learning* ini dilanjutkan dengan model untuk beasiswa bidikmisi. Ikhtisar model ini ditunjukkan pada Gambar 5.2.

Layer (type)	Output Shape	Param #
dense_45 (Dense)	(None, 2000)	74000
dense_46 (Dense)	(None, 1500)	3001500
dense_47 (Dense)	(None, 1000)	1501000
dense_48 (Dense)	(None, 500)	500500
dense_49 (Dense)	(None, 1)	501
Total params: 5,077,501		
Trainable params: 5,077,501		
Non-trainable params: 0		

Gambar 5.2 Ikhtisar model untuk beasiswa bidikmisi

Dari gambar di atas ditunjukkan bahwa model ini memiliki sebanyak 5.077.501 parameter. Jumlah parameter tersebut sebenarnya lebih dari cukup apabila dibandingkan dengan lebar data masukan yang hanya 36 data. Kemudian, model untuk beasiswa mandiri ditunjukkan pada Gambar 5.3.

Layer (type)	Output Shape	Param #
dense_60 (Dense)	(None, 2000)	14000
dense_61 (Dense)	(None, 1500)	3001500
dense_62 (Dense)	(None, 1000)	1501000
dense_63 (Dense)	(None, 500)	500500
dense_64 (Dense)	(None, 1)	501
Total params: 5,017,501		
Trainable params: 5,017,501		
Non-trainable params: 0		

Gambar 5.3 Ikhtisar model untuk beasiswa mandiri

Model tersebut memiliki parameter sebanyak 5.017.501. Setelah pembuatan model, langkah selanjutnya adalah melakukan pelatihan terhadap model menggunakan data yang telah disiapkan.

5.3.2 Pelatihan Model

Kemudian di tahapan ini dilakukan kepada ketiga model yang telah dibuat. Algoritma untuk optimasi akan menggunakan algoritma *Adam*, sementara untuk perhitungan *loss* akan menggunakan pendekatan *binary_crossentropy*. Secara lengkap, pelatihan terhadap model dilakukan dengan menggunakan parameter pelatihan sebagaimana dicantumkan di Tabel 5.16.

Tabell 5.16 *Parameter Pelatihan Model*

No.	Parameter	Nilai
1	<i>Optimizer</i>	<i>Adam</i>
2	<i>Loss</i>	<i>Binary_crossentropy</i>
3	<i>Metrics</i>	<i>Binary_accuracy</i>
4	<i>Callbacks</i>	<i>Early Stopping</i>

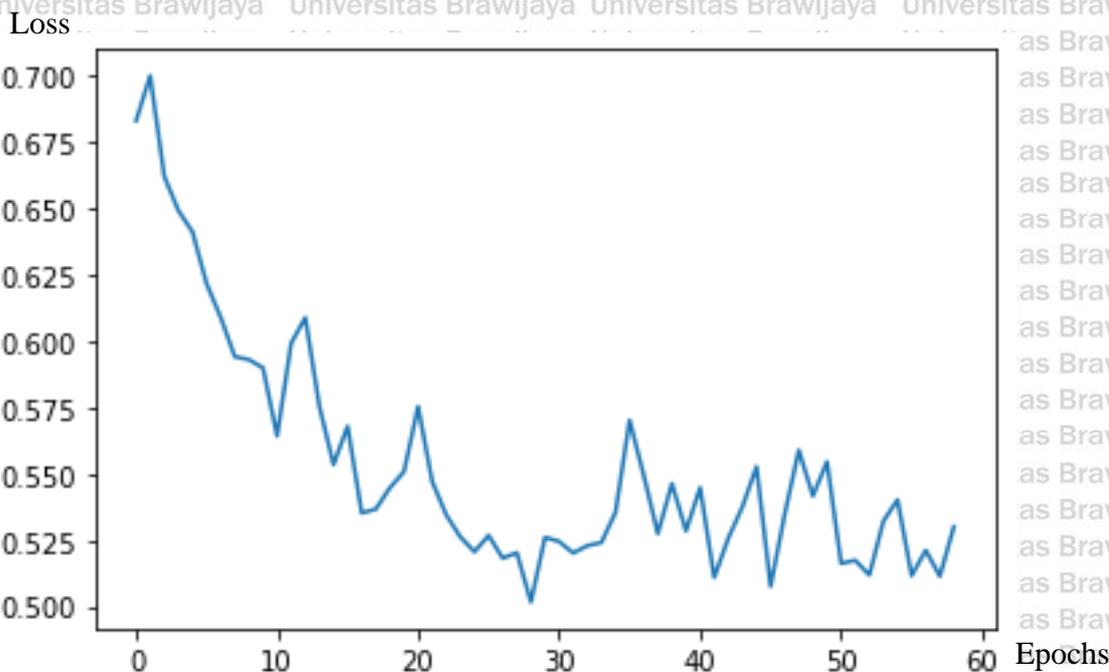
Kemudian dilakukan pelatihan terhadap model untuk prediksi beasiswa. Pelatihan terhadap model pertama berjalan selama 59 *epochs* pelatihan sebelum terjadi *callback*. Hasil pelatihan ditunjukkan di Gambar 5.4 dan Gambar 5.5. Didapatkan nilai *loss* minimal adalah 0.5017, dengan nilai akurasi 71.93%.

```

binary_accuracy: 0.6959
Epoch 27/500
6/6 [=====] - 0s 46ms/step - loss: 0.5182 -
binary_accuracy: 0.6901
Epoch 28/500
6/6 [=====] - 0s 47ms/step - loss: 0.5202 -
binary_accuracy: 0.6784
Epoch 29/500
6/6 [=====] - 0s 46ms/step - loss: 0.5017 -
binary_accuracy: 0.7193
Epoch 30/500
6/6 [=====] - 0s 46ms/step - loss: 0.5260 -
binary_accuracy: 0.6725
Epoch 31/500
6/6 [=====] - 0s 48ms/step - loss: 0.5245 -
binary_accuracy: 0.6842
Epoch 32/500
6/6 [=====] - 0s 47ms/step - loss: 0.5201 -
binary_accuracy: 0.7018

```

Gambar 5.4 Hasil pelatihan pada titik optimal *epoch* ke-29



Gambar 5.5 Grafik nilai *loss* pelatihan terhadap *epochs* pelatihan

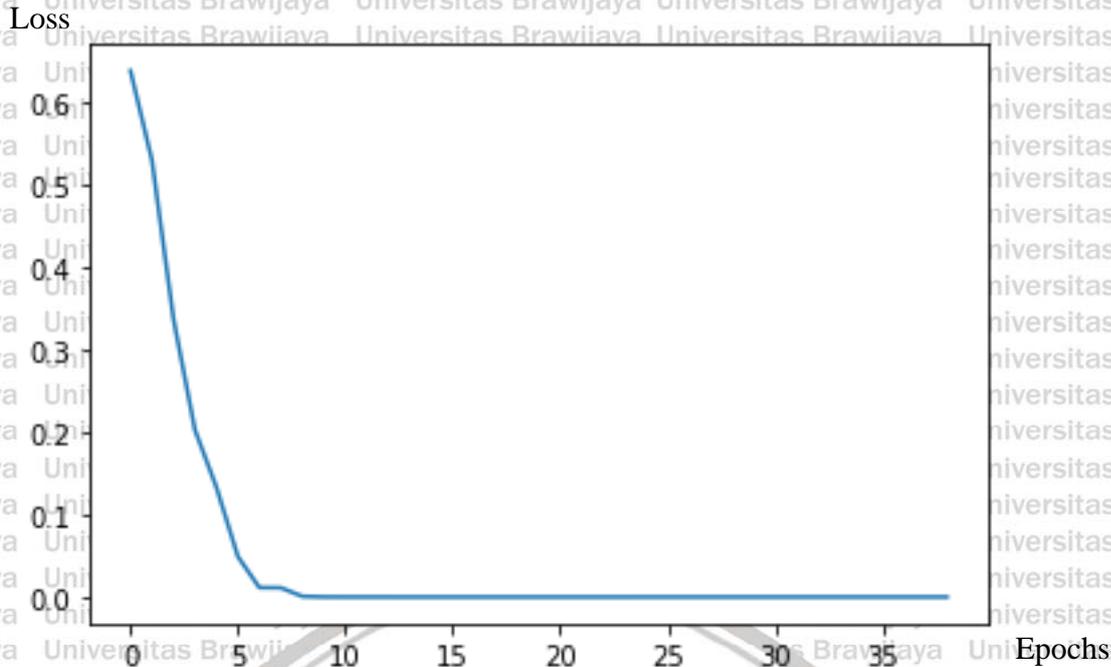
Kemudian dilakukan pelatihan terhadap model untuk prediksi beasiswa bidikmisi. Pelatihan terhadap model kedua berjalan selama 35 *epochs* pelatihan sebelum terjadi *callback*. Hasil pelatihan ditunjukkan di Gambar 5.6 dan Gambar 5.7. Didapati bahwa nilai *loss* minimal hingga mendekati 0, dengan nilai akurasi 100%.

```

Epoch 2/500
2/2 [=====] - 0s 44ms/step - loss: 0.5293 -
binary_accuracy: 0.7826
Epoch 3/500
2/2 [=====] - 0s 41ms/step - loss: 0.3374 -
binary_accuracy: 0.8043
Epoch 4/500
2/2 [=====] - 0s 39ms/step - loss: 0.2025 -
binary_accuracy: 0.8913
Epoch 5/500
2/2 [=====] - 0s 42ms/step - loss: 0.1329 -
binary_accuracy: 0.9783
Epoch 6/500
2/2 [=====] - 0s 40ms/step - loss: 0.0490 -
binary_accuracy: 1.0000
Epoch 7/500
2/2 [=====] - 0s 39ms/step - loss: 0.0113 -
binary_accuracy: 1.0000

```

Gambar 5.6 Hasil pelatihan telah konvergen pada *epochs* ke 6



Gambar 5.7 Grafik nilai *loss* pelatihan terhadap *epochs* pelatihan

Kemudian dilakukan pelatihan terhadap model untuk prediksi beasiswa mandiri.

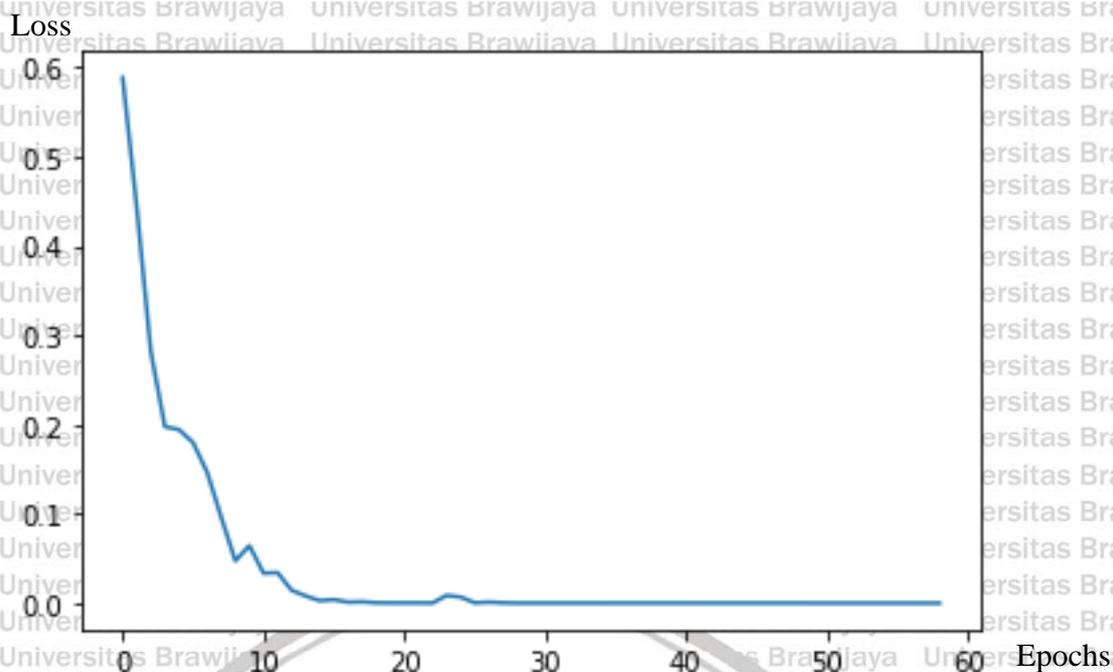
Pelatihan terhadap model ketiga ini berjalan selama 43 *epochs* pelatihan sebelum terjadi *callback*. Hasil pelatihan ditunjukkan di Gambar 5.8 dan Gambar 5.9. Didapatkan nilai *loss* minimal hingga mendekati 0, dengan nilai akurasi 100%.

```

Epoch 12/500
4/4 [=====] - 0s 43ms/step - loss: 0.0342 -
binary_accuracy: 0.9897
Epoch 13/500
4/4 [=====] - 0s 47ms/step - loss: 0.0143 -
binary_accuracy: 0.9897
Epoch 14/500
4/4 [=====] - 0s 41ms/step - loss: 0.0080 -
binary_accuracy: 1.0000
Epoch 15/500
4/4 [=====] - 0s 41ms/step - loss: 0.0028 -
binary_accuracy: 1.0000
Epoch 16/500
4/4 [=====] - 0s 42ms/step - loss: 0.0039 -
binary_accuracy: 1.0000
Epoch 17/500
4/4 [=====] - 0s 40ms/step - loss: 0.0011 -
binary_accuracy: 1.0000

```

Gambar 5.8 Hasil pelatihan telah konvergen pada *epochs* ke 14



Gambar 5.9 Grafik nilai *loss* pelatihan terhadap *epochs* pelatihan

Pada proses pembuatan, pelatihan dan pengujian model berbasis *deep learning* diatas, dilakukan proses untuk menguji memorisasi dan generalisasi dari model MLP yang telah dibuat. Proses ini dilakukan pada tahap validasi. Pada saat proses pelatihan, model *deep learning* dilatih menggunakan data yang disiapkan secara khusus untuk pelatihan, namun pada saat pengujian, model dihadapkan pada data yang tidak digunakan pelatihan. Proses pada saat validasi ini menguji apakah model telah berhasil melakukan memorisasi dan generalisasi terhadap data pelatihan. Memorisasi dan generalisasi yang baik ditunjukkan dari selisih nilai saat validasi dan pelatihan yang relatif konstan dan tidak banyak berubah.

Setelah proses pelatihan selesai dilakukan, proses untuk melakukan prediksi terhadap suatu beasiswa direkomendasikan terhadap seorang calon penerima beasiswa adalah dengan melewati data calon penerima beasiswa tersebut kedalam model. Proses ini dilakukan dengan memasukkan data mahasiswa kedalam model. Misalkan pada pengujian ini, dimasukkan data calon penerima beasiswa ke 11, yang mana data ini memiliki indeks 10. Maka dengan memasukkan kedalam model, didapatkan hasil prediksi sebagaimana Gambar 5.10. Dari contoh tersebut, didapatkan nilai 0.9999993. Pembulatan nilai tersebut adalah 1, sehingga model **merekomendasikan** calon untuk menerima beasiswa skema mandiri. Dari pengecekan data asli, didapatkan bahwa **benar** pada periode tersebut calon tersebut

menerima beasiswa mandiri. Apabila pembulatan nilai prediksi 0, maka calon tidak direkomendasikan untuk menerima beasiswa tersebut.

```
x = data_mandiri_X[10]
print(x.shape)
x.shape = (-1, 6)
print(x.shape)
print(model_mandiri.predict(x))

# Actual
print(data_mandiri_y[10])

(6,)
(1, 6)
[[0.9999993]]
1
```

Gambar 5. 10 Proses Melewatkan Data Calon Penerima Tunggal

5.3.3 Evaluasi Kinerja SPK

Dari pelatihan yang telah dilakukan, didapati bahwa 2 dari 3 model berhasil mencapai konvergensi dan mencapai nilai akurasi hingga 100%, dengan nilai *loss* hingga mendekati nilai 0. Kinerja dari model dengan pendekatan ini disampaikan di Tabel 5.17.

Tabel 5.17 Akurasi Rekomendasi SPK Terhadap Hasil Sebenarnya

No	Skema	Akurasi Prediksi
1	Prestasi	71.93 %
2	Bidikmisi	100 %
3	Mandiri	100 %

Dari hasil diatas, didapati bahwa sistem SPK berbasis pendekatan *deep learning* ini memiliki kinerja yang sangat tinggi, dimana berhasil dicapai tingkat akurasi prediksi hingga 100% terhadap dua dari tiga buah *dataset* yang diujikan. Namun, akan dilakukan analisa dan pembahasan lebih lanjut terhadap kinerja dari SPK ini, dan perbandingannya terhadap SPK berbasis AHP-TOPSIS, pada subbab selanjutnya.

5.4 Perbandingan dan Analisis Kinerja Kedua SPK

Kegiatan penelitian telah dilakukan, dan hasil kinerja akurasi dari kedua pendekatan telah ditemukan. Pendekatan pertama memanfaatkan perhitungan berbasis pendekatan AHP dan TOPSIS. Pendekatan kedua memanfaatkan pendekatan pembelajaran tersupervisi berbasis *deep learning*. Hasil dari kedua pendekatan disampaikan di Tabel 5.18.

Tabel 5.18 Perbandingan Kinerja Kedua Pendekatan

No	Skema Beasiswa	AHP-TOPSIS	Deep learning
1	Prestasi	56.72 %	71.93 %
2	Bidikmisi	65.21 %	100 %
3	Mandiri	95.87 %	100 %

Hasil penelitian telah didapatkan, dan analisis akan dilakukan terhadap masing-masing pendekatan yang digunakan, pada masing-masing beasiswa. Dengan membandingkan nilai akurasi antara AHP-TOPSIS dengan pendekatan *Deep Learning*, ditarik kesimpulan bahwa *deep learning* memiliki kemampuan prediksi yang lebih baik dibandingkan AHP-TOPSIS. Ada dua kemungkinan utama penyebab hal ini. Pertama, pembobotan dalam AHP-TOPSIS disusun secara tidak sempurna, atau pelaksanaan seleksi di lapangan berbeda (memiliki bias). Hal ini menyebabkan bobot kriteria atau bias tidak diikutsertakan secara tepat untuk menghasilkan prediksi/rekomendasi yang akurat. Kedua, *deep learning* mampu mendeteksi pola hubungan antar variabel, yang mana hal ini mungkin tidak menjadi masukan oleh pendekatan AHP-TOPSIS.

Penjelasan terhadap kedua kemungkinan ini terdapat dari analisis antara akurasi pada skema beasiswa yang berbeda. Terhadap beasiswa bidikmisi dan mandiri, *deep learning* mampu mempelajari proses penilaian dalam seleksi secara sempurna hingga mampu mencapai akurasi 100%. Pendekatan AHP-TOPSIS tidak mampu memprediksi secara sempurna, ini kemungkinan disebabkan proses penyusunan bobot dalam AHP-TOPSIS yang tidak sempurna.

Namun pada proses seleksi beasiswa prestasi, kedua pendekatan tidak mampu memprediksi dengan baik calon penerima beasiswa. Hal ini mengindikasikan bahwa dalam proses seleksi beasiswa AHP-TOPSIS, terdapat *bias-bias* atau keputusan seleksi yang tidak didasarkan pada data-data masukan yang diperoleh. Selain itu, juga dimungkinkan terdapat

data-data lain dalam seleksi beasiswa yang digunakan, namun tidak ikut disertakan dalam data yang ada (tidak dimiliki), sehingga hasil prediksi dari kedua pendekatan, tidak mampu secara efektif mendeteksi pola-pola hubungan dari masukan tersebut. Kesimpulan dari analisis ini dari sisi kemampuan kedua model, adalah bahwa kedua model cukup baik memprediksikan calon penerima, namun model dengan pendekatan *deep learning* mampu unggul dalam proses ini.

Jadi dari hasil penelitian yang sudah didapatkan, dapat disimpulkan bahwa algoritma berbasis *deep learning* lebih unggul dalam melakukan rekomendasi calon penerima beasiswa, dibandingkan pendekatan berbasis AHP-TOPSIS.



BAB VI

KESIMPULAN DAN SARAN

Penelitian ini telah selesai dan beberapa temuan dihasilkan. Dua buah model SPK telah diuji dengan menggunakan tiga buah *dataset* calon penerima beasiswa. Dapat ditarik beberapa kesimpulan dan saran yaitu sebagai berikut:

1. Validasi telah berhasil dilakukan dengan prosedur validasi yang membuang data kosong dan mengubah data menjadi data-data valid.
2. Model SPK dengan pendekatan AHP-TOPSIS telah berhasil dibuat, dengan kinerja akurasi untuk pengenalan data-data skema beasiswa prestasi, bidikmisi, dan mandiri berturut-turut adalah 56.72%, 65,21%, dan 95.87%.
3. Model SPK dengan pendekatan *deep learning* telah berhasil dibuat dengan kinerja akurasi berturut-turut adalah 71.93%, 100%, dan 100%.
4. Evaluasi terhadap kinerja kedua SPK telah dilakukan terhadap kedua SPK, yang mendapati kinerja model SPK berbasis *deep learning* lebih baik dibandingkan model berbasis AHP-TOPSIS.
5. Terhadap penelitian selanjutnya pada ranah permasalahan yang berbeda, disarankan untuk membahas penggunaan *deep learning* untuk perancangan dan pembuatan SPK dan membandingkannya dengan pendekatan yang lebih tradisional.



[Halaman Ini Sengaja Dikosongkan]

DAFTAR PUSTAKA

- Alhassan, J., & Lawal, S. (2015). Using Data Mining Technique for Scholarship Disbursement. *International Journal of Information and Communication Engineering*, 1741-1744.
- Aulck, L., Nambi, D., & West, J. (2020). Increasing Enrollment by Optimizing Scholarship Allocations Using Machine Learning and Genetic Algorithm. *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, (pp. 29-38).
- Ciresan, D., Meier, U., & Schmidhuber, J. (2012). Multi-column Deep Neural Networks for Image Classification.
- Fishburn, P. (1967). *A Problem-based Selection of Multi-Attribute Decision Making Methods*. New Jersey: Blackwell Publishing.
- Hwang, C., & Yoon, H. (1981). Multiple attribute decision making, methods and applications. *Lecture Notes in Economics and Mathematical Systems*, Vol 186.
- Josaputri, C., Sugiharti, E., & Arifudin, R. (2016). Decision Support Systems for The Determination of Cattle with Superior Seeds using AHP and SAW Method. *Scientific Journal of Informatics*, 21-30.
- Keen, P. G. (1980). *Decision Support Systems: A Research Perspective*. Cambridge: MIT.
- Lemarecheal, C. (2012). Cauchy and the Gradient Method. *Documenta Math*, 251-254.
- Puspitasari, T., Sari, E., Destarianto, P., & Riskiawan, H. (2018). Decision Support System for Determining Scholarship Selection using an Analytical Hierarchy Process. *Journal of Physics Conference Series*, 1-6.
- Putra, M., Ariyanti, W., & Cholissodin, I. (2013). Selection and Recommendation Scholarships Using AHP-SVM-TOPSIS. *Journal of Information Technology and Computer Science*, 1-13.
- Rahim, R., Supiyandi, S., Siahaan, A., Listyorini, T., & Utomo, A. (2018). TOPSIS Method Application for Decision Support System in Internal Control for Selecting Best Employees. *Journal of Physics*, 21-30.
- Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, 366-408.
- Saaty, T. (2008). *Decision making with the analytical hierarchy process*. Pittsburgh: Katz Graduate School of Business.

Sharma, S. (2017). What the Hell is Perceptron? The Fundamentals of Neural Network.

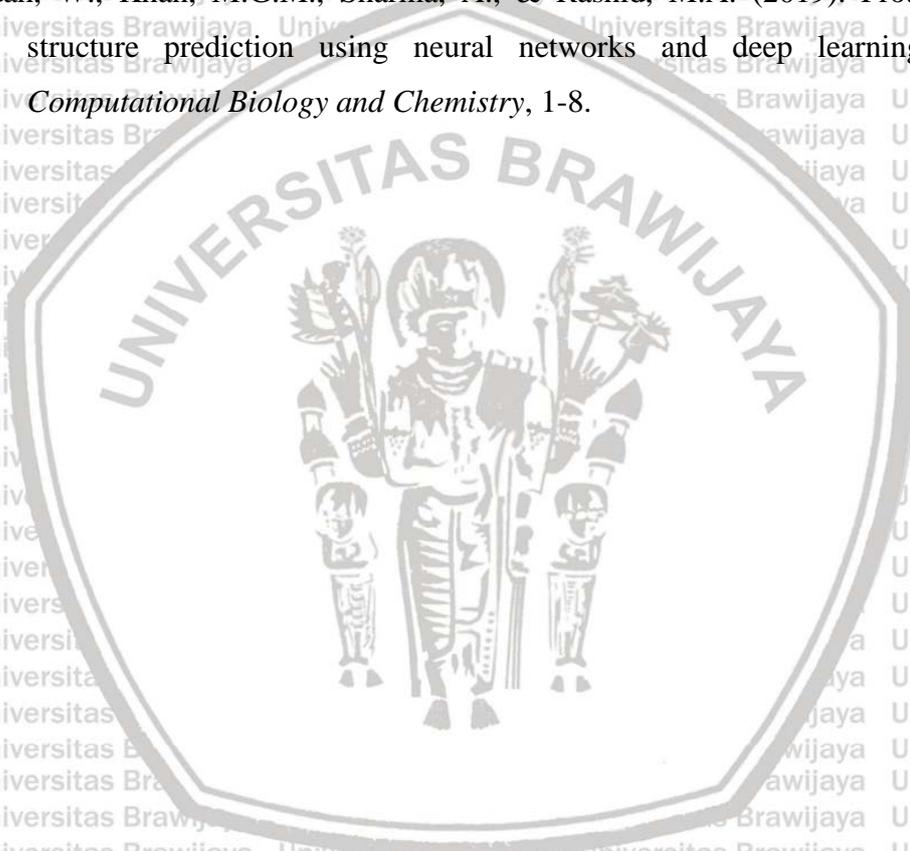
towardsdatascience.com. <https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53>. (Diakses 1 April 2021)

Santra, A., & Christy, C. (2012). Genetic Algorithm and Confusion Matrix for Document Clustering. *International Journal of Computer Science*, 322-328.

Schocken, S., & Ariav, G. (1991). *Neural Network for Decision Support: Problem and Opportunities*. Tel Aviv.

Tzeng, G., & Huang, J. (2011). *Multiple Attribute Decision Making Methods and Applications*. Boca Raton: Chapman & Hall.

Wardah, W., Khan, M.G.M., Sharma, A., & Rashid, M.A. (2019). Protein secondary structure prediction using neural networks and deep learning: A review. *Computational Biology and Chemistry*, 1-8.



Lampiran 1 Scripting Untuk Model AHP-TOPSIS

```

# Import library
import tensorflow as tf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import preprocessing
from google.colab import drive
# Mount google drive
drive.mount('/content/gdrive')
# Copy the dataset from dataset folder to current working folder
!cp gdrive/MyDrive/Colab\ Notebooks/Dataset/data_beasiswa_prestasi.csv data_prestasi.csv
!cp gdrive/MyDrive/Colab\ Notebooks/Dataset/data_beasiswa_bidikmisi.csv data_bidikmisi.csv
!cp gdrive/MyDrive/Colab\ Notebooks/Dataset/data_beasiswa_mandiri.csv data_mandiri.csv
# Load the data as pd Dataframe
data_prestasi = pd.read_csv('data_prestasi.csv', sep=',', index_col=0)
data_bidikmisi = pd.read_csv('data_bidikmisi.csv', sep=',')
data_mandiri = pd.read_csv('data_mandiri.csv', sep=',')

# Data cleaning: Remove unnamed column
data_prestasi = data_prestasi.loc[:, ~data_prestasi.columns.str.contains('^Unnamed')] # Remove unnamed column
data_bidikmisi = data_bidikmisi.loc[0:45] # Remove last data

# Separate the input data from the output data
data_prestasi_X = data_prestasi.drop('beasiswa', axis=1) # Masukan adalah seluruh kolom kecuali hasil beasiswa
data_prestasi_y = data_prestasi['beasiswa'] # Luaran adalah status penerima beasiswa
data_bidikmisi_X = data_bidikmisi.drop('beasiswa', axis=1) # Masukan adalah seluruh kolom kecuali hasil beasiswa
data_bidikmisi_y = data_bidikmisi['beasiswa'] # Luaran adalah status penerimaan beasiswa
data_mandiri_X = data_mandiri.drop('beasiswa', axis=1) # Masukan adalah seluruh kolom kecuali hasil beasiswa
data_mandiri_y = data_mandiri['beasiswa'] # Luaran adalah status penerima beasiswa

# Convert to numpy
data_prestasi_X = data_prestasi_X.to_numpy()
data_prestasi_y = data_prestasi_y.to_numpy()
data_bidikmisi_X = data_bidikmisi_X.to_numpy()
data_bidikmisi_y = data_bidikmisi_y.to_numpy()
data_mandiri_X = data_mandiri_X.to_numpy()

```

```

awijaya data_mandiri_y = data_mandiri_y.to_numpy() ijaya
awijaya # Normalize the input data ijaya
awijaya # Create the min max scaler ijaya
awijaya min_max_scaler = preprocessing.MinMaxScaler() ijaya
awijaya data_prestasi_X = min_max_scaler.fit_transform(data_prestasi_X) ijaya
awijaya data_bidikmisi_X = min_max_scaler.fit_transform(data_bidikmisi_X) ijaya
awijaya data_mandiri_X = min_max_scaler.fit_transform(data_mandiri_X) ijaya
awijaya # For the AHP Method, create the composite criterion matrices ijaya
awijaya ahp_matrix_prestasi = np.array([[0, 0.0085,0.0085,0.0085,0.0085,0.008 ijaya
awijaya 5,0.0085,0.0085,0.0085,0.0085,0.0085,0.0085,0.0085, 0, 0, 0, 0], # K ijaya
awijaya omposit Kriteria Administratif ijaya
awijaya [0, 0,0,0,0,0,0,0,0,0,0,0,0,0, 0.0896,0. ijaya
awijaya 3584, 0, 0], # Komposit Kriteria Prestasi Akademik ijaya
awijaya [0, 0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0, 0.1 ijaya
awijaya 02, 0], # Kriteria Beasiswa Sebelumnya ijaya
awijaya [0, 0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0, 0, ijaya
awijaya 0.348]] # Kriteria Prestasi non-akademik ijaya
awijaya ); # 17 kolom data masukan ijaya
awijaya ahp_matrix_prestasi = np.transpose(ahp_matrix_prestasi) ijaya
awijaya ahp_matrix_bidikmisi = np.array([[0, 0.102, 0, 0, 0, 0, 0, 0, 0, 0, 0, ijaya
awijaya , 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ijaya
awijaya , 0, 0], # Prestasi ijaya
awijaya [0, 0, 0.075, 0.075, 0.037, 0.148, 0. ijaya
awijaya 037, 0.148, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ijaya
awijaya 0, 0, 0, 0, 0, 0, 0, 0], # Kondisi Finansial ijaya
awijaya [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ijaya
awijaya 0, 0, 0, 0, 0, 0, 0.0204, 0.0204, 0.0204, 0.0204, 0.0204, 0, 0, 0, ijaya
awijaya 0, 0, 0, 0, 0, 0, 0, 0], # Bantuan Terdahulu ijaya
awijaya [0, 0, 0, 0, 0, 0, 0, 0, 0, 0.1087, 0.02 ijaya
awijaya 17, 0, 0, 0, 0.1087, 0, 0, 0.1087, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ijaya
awijaya , 0, 0, 0, 0, 0, 0, 0]] # Beban ijaya
awijaya ); ijaya
awijaya ahp_matrix_bidikmisi = np.transpose(ahp_matrix_bidikmisi) ijaya
awijaya ahp_matrix_mandiri = np.array([[0, 0, 0.2222, 0, 0, 0], # Prestasi Ak ijaya
awijaya ademik ijaya
awijaya [0, 0, 0, 0, 0.1111, 0], # Prestasi Non ijaya
awijaya Akademik ijaya
awijaya [0, 0, 0, 0.2222, 0, 0], # Organisasi ijaya
awijaya [0, 0, 0, 0, 0, 0.4444]] # Organisasi K ijaya
awijaya ekhususan ijaya
awijaya ); ijaya
awijaya ahp_matrix_mandiri = np.transpose(ahp_matrix_mandiri) ijaya
awijaya # Check the data ijaya
awijaya # Perform the matrix multiplication to get the resulting data ijaya
awijaya ahp_hasil_prestasi = np.matmul(data_prestasi_X.astype(np.float), ahp_ ijaya
awijaya matrix_prestasi) ijaya

```

```

awijaya universitas Brawijaya universitas Brawijaya universitas Brawijaya universitas Brawijaya ya
awijaya ahp_hasil_bidikmisi= np.matmul(data_bidikmisi_X.astype(np.float), ahp ya
awijaya _matrix_bidikmisi) ya
awijaya ahp_hasil_mandiri = np.matmul(data_mandiri_X.astype(np.float), ahp_ma ya
awijaya trix_mandiri) ya
awijaya # For totpsis, get positive ideal solution, and negative ideal solutio ya
awijaya n ya
awijaya # Get the max and min ya
awijaya totpsis_prestasi_max = np.amax(ahp_hasil_prestasi, axis = 0) ya
awijaya totpsis_prestasi_min = np.amin(ahp_hasil_prestasi, axis = 0) ya
awijaya totpsis_bidikmisi_max = np.amax(ahp_hasil_bidikmisi, axis = 0) ya
awijaya totpsis_bidikmisi_min = np.amin(ahp_hasil_bidikmisi, axis= 0) ya
awijaya totpsis_mandiri_max = np.amax(ahp_hasil_mandiri, axis = 0) ya
awijaya totpsis_mandiri_min = np.amin(ahp_hasil_mandiri, axis= 0) ya
awijaya # Set the positive solution and negative solution ya
awijaya # Beasiswa Prestasi ya
awijaya totpsis_prestasi_positive = np.copy(totspis_prestasi_max) ya
awijaya totpsis_prestasi_negative = np.copy(totspis_prestasi_min) ya
awijaya totpsis_prestasi_positive[2] = totpsis_prestasi_min[2] # Kriteria beas ya
awijaya iswa sebelumnya is ideally low value ya
awijaya totpsis_prestasi_negative[2] = totpsis_prestasi_max[2] # Kriteria beas ya
awijaya iswa sebelumnya, high value is not ideal ya
awijaya # Beasiswa Bidikmisi ya
awijaya totpsis_bidikmisi_positive = np.copy(totspis_bidikmisi_max) ya
awijaya totpsis_bidikmisi_negative = np.copy(totspis_bidikmisi_min) ya
awijaya totpsis_bidikmisi_positive[1] = totpsis_bidikmisi_min[1] # Kriteria ko ya
awijaya ndisi finansial, ideally low value ya
awijaya totpsis_bidikmisi_negative[1] = totpsis_bidikmisi_max[1] # ya
awijaya totpsis_bidikmisi_positive[2] = totpsis_bidikmisi_min[2] # Kriteria ba ya
awijaya ntuan terdahulu, ideally low value ya
awijaya totpsis_bidikmisi_negative[2] = totpsis_bidikmisi_max[2] # ya
awijaya # Beasiswa Mandiri ya
awijaya totpsis_mandiri_positive = np.copy(totspis_mandiri_max) ya
awijaya totpsis_mandiri_negative = np.copy(totspis_mandiri_min) ya
awijaya # Get Si+ ya
awijaya totpsis_prestasi_si_positive = np.sqrt(np.sum(np.square(ahp_hasil_pres ya
awijaya tasi - totpsis_prestasi_positive), axis=1)) ya
awijaya totpsis_bidikmisi_si_positive = np.sqrt(np.sum(np.square(ahp_hasil_bid ya
awijaya ikmisi - totpsis_bidikmisi_positive), axis=1)) ya
awijaya totpsis_mandiri_si_positive = np.sqrt(np.sum(np.square(ahp_hasil_mandi ya
awijaya ri - totpsis_mandiri_positive), axis=1)) ya
awijaya # Get Si- ya
awijaya totpsis_prestasi_si_negative = np.sqrt(np.sum(np.square(ahp_hasil_pres ya
awijaya tasi - totpsis_prestasi_negative), axis=1)) ya
awijaya totpsis_bidikmisi_si_negative = np.sqrt(np.sum(np.square(ahp_hasil_bid ya
awijaya ikmisi - totpsis_bidikmisi_negative), axis=1)) ya
awijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya ya
awijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya

```

```

awijaya topsis_mandiri_si_negative = np.sqrt(np.sum(np.square(ahp_hasil_mandi
awijaya ri - tophis_mandiri_negative), axis=1))
awijaya # Add the Si+ and Si-
awijaya topsis_prestasi_si_plus_minus = np.add(topsis_prestasi_si_positive, t
awijaya opsis_prestasi_si_negative)
awijaya topsis_bidikmisi_si_plus_minus = np.add(topsis_bidikmisi_si_positive,
awijaya tophis_bidikmisi_si_negative)
awijaya topsis_mandiri_si_plus_minus = np.add(topsis_mandiri_si_positive, to
awijaya phis_mandiri_si_negative)
awijaya
awijaya # Evaluate the Pi (final score)
awijaya topsis_prestasi_pi = np.divide(topsis_prestasi_si_negative, tophis_pr
awijaya estasi_si_plus_minus)
awijaya topsis_bidikmisi_pi = np.divide(topsis_bidikmisi_si_negative, tophis_
awijaya bidikmisi_si_plus_minus)
awijaya topsis_mandiri_pi = np.divide(topsis_mandiri_si_negative, tophis_mand
awijaya iri_si_plus_minus)
awijaya
awijaya # Create the array to compare predicted by SPK vs Actual data
awijaya spk_prestasi_pred_vs_actual = np.column_stack((tophis_prestasi_pi, da
awijaya ta_prestasi_y))
awijaya spk_bidikmisi_pred_vs_actual = np.column_stack((tophis_bidikmisi_pi,
awijaya data_bidikmisi_y))
awijaya spk_mandiri_pred_vs_actual = np.column_stack((tophis_mandiri_pi, data
awijaya _mandiri_y))
awijaya
awijaya # Convert to pandas for easy sorting
awijaya spk_prestasi_pva_df = pd.DataFrame(spk_prestasi_pred_vs_actual, colum
awijaya ns = ['Predicted', 'Actual'])
awijaya spk_bidikmisi_pva_df = pd.DataFrame(spk_bidikmisi_pred_vs_actual, col
awijaya umns = ['Predicted', 'Actual'])
awijaya spk_mandiri_pva_df = pd.DataFrame(spk_mandiri_pred_vs_actual, columns
awijaya = ['Predicted', 'Actual'])
awijaya # Evaluation
awijaya print(spk_prestasi_pva_df.sort_values(by=['Predicted'])[-
awijaya 69:]['Actual'].value_counts())
awijaya print(spk_prestasi_pva_df.sort_values(by=['Predicted'])[:102]['Actual
awijaya '].value_counts())
awijaya print(spk_bidikmisi_pva_df.sort_values(by=['Predicted'])[-
awijaya 10:]['Actual'].value_counts())
awijaya print(spk_bidikmisi_pva_df.sort_values(by=['Predicted'])[:36]['Actual
awijaya '].value_counts())
awijaya print(spk_mandiri_pva_df.sort_values(by=['Predicted'])[-
awijaya 17:]['Actual'].value_counts())
awijaya print(spk_mandiri_pva_df.sort_values(by=['Predicted'])[:80]['Actual']
awijaya .value_counts())
awijaya
awijaya # Accuracy is
awijaya acc_prestasi = (32 + 65) / 171

```


Lampiran 2 *Scripting Untuk Model Deep Learning*

```

awijaya: # Import library
awijaya: import tensorflow as tf
awijaya: import pandas as pd
awijaya: import numpy as np
awijaya: import matplotlib.pyplot as plt
awijaya: from sklearn import preprocessing
awijaya: from tensorflow import keras
awijaya: from tensorflow.keras import layers
awijaya: from google.colab import drive
awijaya: # Mount google drive
awijaya: drive.mount('/content/gdrive')
awijaya: # Copy the dataset from dataset folder to current working folder
awijaya: !cp gdrive/MyDrive/Colab\ Notebooks/Dataset/data_beasiswa_prestasi.csv
awijaya: data_prestasi.csv
awijaya: !cp gdrive/MyDrive/Colab\ Notebooks/Dataset/data_beasiswa_bidikmisi.csv
awijaya: data_bidikmisi.csv
awijaya: !cp gdrive/MyDrive/Colab\ Notebooks/Dataset/data_beasiswa_mandiri.csv
awijaya: data_mandiri.csv
awijaya: # Ensure file is copied
awijaya: !ls
awijaya: # Load the data as pd Dataframe
awijaya: data_prestasi = pd.read_csv('data_prestasi.csv', sep=',', index_col=0)
awijaya: )
awijaya: data_bidikmisi = pd.read_csv('data_bidikmisi.csv', sep=',')
awijaya: data_mandiri = pd.read_csv('data_mandiri.csv', sep=',')
awijaya: # Data cleaning: Remove unnamed column
awijaya: data_prestasi = data_prestasi.loc[:, ~data_prestasi.columns.str.contains('^Unnamed')] # Remove unnamed column
awijaya: data_bidikmisi = data_bidikmisi.loc[0:45] # Remove last data
awijaya: # Separate the input data from the output data
awijaya: data_prestasi_X = data_prestasi.drop('beasiswa', axis=1) # Masukan adalah seluruh kolom kecuali hasil beasiswa
awijaya: data_prestasi_y = data_prestasi['beasiswa'] # Luaran adalah status penerima beasiswa
awijaya: data_bidikmisi_X = data_bidikmisi.drop('beasiswa', axis=1) # Masukan adalah seluruh kolom kecuali hasil beasiswa
awijaya: data_bidikmisi_y = data_bidikmisi['beasiswa'] # Luaran adalah status penerimaan beasiswa
awijaya: data_mandiri_X = data_mandiri.drop('beasiswa', axis=1) # Masukan adalah seluruh kolom kecuali hasil beasiswa
awijaya: data_mandiri_y = data_mandiri['beasiswa'] # Luaran adalah status penerimaan beasiswa
awijaya: # Convert to numpy
awijaya: data_prestasi_X = data_prestasi_X.to_numpy()

```



```

data_prestasi_y = data_prestasi_y.to_numpy()
data_bidikmisi_X = data_bidikmisi_X.to_numpy()
data_bidikmisi_y = data_bidikmisi_y.to_numpy()
data_mandiri_X = data_mandiri_X.to_numpy()
data_mandiri_y = data_mandiri_y.to_numpy()
# Normalize the input data
# Create the min max scaler
min_max_scaler = preprocessing.MinMaxScaler()

data_prestasi_X = min_max_scaler.fit_transform(data_prestasi_X)
data_bidikmisi_X = min_max_scaler.fit_transform(data_bidikmisi_X)
data_mandiri_X = min_max_scaler.fit_transform(data_mandiri_X)
# Define early stopping callbacks
early_stopping = keras.callbacks.EarlyStopping(
    monitor='loss',
    patience=30,
    min_delta=0.001,
    restore_best_weights=True,
)

# Create the deep learning model for beasiswa prestasi
input_shape_prestasi = [17]
model_prestasi = keras.Sequential([
    layers.Dense(units=2000, activation='relu', input_shape=input_shape_prestasi),
    layers.Dense(units=1500, activation='relu'),
    layers.Dense(units=1000, activation='relu'),
    layers.Dense(units=500, activation='relu'),
    layers.Dense(units=1, activation='sigmoid')
])

model_prestasi.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['binary_accuracy'],
)

model_prestasi.summary()
# Train the model
history = model_prestasi.fit(data_prestasi_X, data_prestasi_y, batch_size=32, epochs=500, callbacks=[early_stopping])
# convert the training history to a dataframe
history_df = pd.DataFrame(history.history)
# use Pandas native plot method
history_df['loss'].plot();
# Create the deep learning model for beasiswa prestasi
input_shape_bidikmisi = [36]
model_bidikmisi = keras.Sequential([

```

```

awijaya     layers.Dense(units=2000, activation='relu', input_shape=input_sha ijaya
awijaya     pe_bidikmisi), ijaya
awijaya     layers.Dense(units=1500, activation='relu'), ijaya
awijaya     layers.Dense(units=1000, activation='relu'), ijaya
awijaya     layers.Dense(units=500, activation='relu'), ijaya
awijaya     layers.Dense(units=1, activation='sigmoid') ijaya
awijaya ] ) ijaya
awijaya model_bidikmisi.compile( ijaya
awijaya     optimizer='adam', ijaya
awijaya     loss='binary_crossentropy', ijaya
awijaya     metrics=['binary_accuracy'], ijaya
awijaya ) ijaya
awijaya model_bidikmisi.summary() ijaya
awijaya # Train the model ijaya
awijaya history = model_bidikmisi.fit(data_bidikmisi_X, data_bidikmisi_y, bat ijaya
awijaya ch_size=32, epochs=500, callbacks=[early_stopping]) ijaya
awijaya # convert the training history to a dataframe ijaya
awijaya history_df = pd.DataFrame(history.history) ijaya
awijaya # use Pandas native plot method ijaya
awijaya history_df['loss'].plot(); ijaya
awijaya # Create the deep learning model for beasiswa prestasi ijaya
awijaya input_shape_mandiri = [6] ijaya
awijaya model_mandiri = keras.Sequential([ ijaya
awijaya     layers.Dense(units=2000, activation='relu', input_shape=input_sha ijaya
awijaya     pe_mandiri), ijaya
awijaya     layers.Dense(units=1500, activation='relu'), ijaya
awijaya     layers.Dense(units=1000, activation='relu'), ijaya
awijaya     layers.Dense(units=500, activation='relu'), ijaya
awijaya     layers.Dense(units=1, activation='sigmoid') ijaya
awijaya ] ) ijaya
awijaya model_mandiri.compile( ijaya
awijaya     optimizer='adam', ijaya
awijaya     loss='binary_crossentropy', ijaya
awijaya     metrics=['binary_accuracy'], ijaya
awijaya ) ijaya
awijaya model_mandiri.summary() ijaya
awijaya # Train the model ijaya
awijaya history = model_mandiri.fit(data_mandiri_X, data_mandiri_y, batch_siz ijaya
awijaya e=32, epochs=500, callbacks=[early_stopping]) ijaya
awijaya # convert the training history to a dataframe ijaya
awijaya history_df = pd.DataFrame(history.history) ijaya
awijaya # use Pandas native plot method ijaya
awijaya history_df['loss'].plot(); ijaya
awijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya ijaya
awijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya ijaya
awijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya ijaya
awijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya ijaya

```

