

TESIS

**KLASIFIKASI KEPERIBADIAN DENGAN METODE DOMINANCE,
INFLUENCE, STEADINESS, COMPLIANCE (DISC) PADA TWITTER
MENGUNAKAN ALGORITMA ARTIFICIAL NEURAL NETWORK
CLASIFICATION (BACKPROPAGATION)**



Disusun oleh:

Nama : Idris
NIM : 18.52.1153
Konsentrasi : Informatics Technopreneurship

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2020

TESIS

**KLASIFIKASI KEPERIBADIAN DENGAN METODE DOMINANCE,
INFLUENCE, STEADINESS, COMPLIANCE (DISC) PADA TWITTER
MENGUNAKAN ALGORITMA ARTIFICIAL NEURAL NETWORK
CLASIFICATION (BACKPROPAGATION)**

**PERSONALITY CLASSIFICATION WITH DOMINANCE, INFLUENCE,
STEADINESS, COMPLIANCE (DISC) METHOD ON TWITTER USING
ARTIFICIAL NEURAL NETWORK CLASSIFICATION ALGORITM
(BACKPROPAGATION)**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

Nama : Idris
NIM : 18.52.1153
Konsentrasi : Informatics Technopreneurshp

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2020

HALAMAN PENGESAHAN

KLASIFIKASI KEPERIBADIAN DENGAN METODE DOMINANCE, INFLUENCE, STEADINESS, COMPLIANCE (DISC) PADA TWITTER MENGGUNAKAN ALGORITMA ARTIFICIAL NEURAL NETWORK CLASSIFICATION (BACKPROPAGATION)

PERSONALITY CLASSIFICATION WITH DOMINANCE, INFLUENCE, STEADINESS, COMPLIANCE (DISC) METHOD ON TWITTER USING ARTIFICIAL NEURAL NETWORK CLASSIFICATION ALGORITHM (BACKPROPAGATION)

Dipreparasi dan Disusun oleh

Idris

18.52.1153

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Kamis, 6 Agustus 2020

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 6 Agustus 2020

Rektor

Prof. Dr. M. Suvanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

KLASIFIKASI KEPERIBADIAN DENGAN METODE DOMINANCE, INFLUENCE, STEADINESS, COMPLIANCE (DISC) PADA TWITTER MENGGUNAKAN ALGORITMA ARTIFICIAL NEURAL NETWORK CLASIFICATION (BACKPROPAGATION)

PERSONALITY CLASSIFICATION WITH DOMINANCE, INFLUENCE, STEADINESS, COMPLIANCE (DISC) METHOD ON TWITTER USING ARTIFICIAL NEURAL NETWORK CLASSIFICATION ALGORITHM (BACKPROPAGATION)

Dipersiapkan dan Disusun oleh

Idris

18.52.1153

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Kamis, 6 Agustus 2020

Pembimbing Utama

Prof. Dr. Ema Utami, S.Si., M.Kom
NIK. 190302037

Pembimbing Pendamping

Anggit Dwi Hartanto, M.Kom
NIK. 190302163

Anggota Tim Penguji

Dr. Kusriani, M.Kom.
NIK. 190302106

Dr. Arief Setyanto, S.Si., M.T.
NIK. 190302036

Prof. Dr. Ema Utami, S.Si., M.Kom
NIK. 190302037

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 6 Agustus 2020
Direktur Program Pascasarjana

Dr. Kusriani, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Idris
NIM : 18.52.1153
Konsentrasi : Informatics Technopreneurshlp

Menyatakan bahwa Tesis dengan judul berikut:

Klasifikasi Kepribadian Dengan Metode Dominance, Influence, Steadiness, Compliance (DISC) Pada Twitter Menggunakan Algoritma Artificial Neural Network Clasification (Backpropagation)

Dosen Pembimbing Utama : Prof. Dr. Ema Utami, S.Si., M.Kom
Dosen Pembimbing Pendamping : Anggit Dwj Hartanto, M.Kom

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 6 Agustus 2020

Yang Menyatakan,

Meterai Asli
Rp 6.000

Idris

PERSEMBAHAN

Dengan rasa penuh rasa syukur karya ini penulis persembahkan untuk :

- Allah SWT, yang memberikan hidup dan memegang matinya setiap makhluk tanpa-Nya tulisan ini tiada bermakna.
- Istri saya tercinta.
- Kedua anak saya yang saya cintai.
- Bapak dan Ibu saya tercinta.
- Ibu Ema Utami , Prof. Dr., S.Si., M.Kom. dan Bapak Anggit Dwi Hartanto, M.Kom yang selalu memberikan arahan, dukungan, dan ilmu didalam menjalankan penelitian ini.

MOTTO

“Maka sesungguhnya bersama kesulitan ada kemudahan. Sesungguhnya bersama kesulitan ada kemudahan. Maka apabila engkau telah selesai (dari sesuatu urusan), tetaplah bekerja keras (untuk urusan yang lain). Dan hanya kepada Tuhanmulah engkau berharap. (QS. Al-Insyirah,6-8)”

“Jawaban sebuah keberhasilan adalah terus belajar dan tak kenal putus asa”

“Rahasia keberhasilan adalah kerja keras dan belajar dari kegagalan”



KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas segala limpahan rahmat dan hidayah-Nya serta shalawat serta salam penulis curahkan kepada junjungan kita Nabi Muhammad SAW sehingga penulis dapat menyelesaikan laporan tesis dengan judul “*KLASIFIKASI KEPERIBADIAN DENGAN METODE DOMINANCE, INFLUENCE, STEADINESS, COMPLIANCE (DISC) PADA TWITTER MENGGUNAKAN ALGORITMA ARTIFICIAL NEURAL NETWORK CLASIFIKATION (BACKPROPAGATION)*” dengan baik.

Tesis ini disusun guna memenuhi salah satu persyaratan untuk mendapatkan gelar Magister pada jurusan Magister Teknik Informatika di Universitas Amikom Yogyakarta.

Dalam penyusunan tesis ini, penulis sangat menyadari masih banyak kekurangan. Selain itu, terselesainya penyusunan tesis ini adalah berkat bantuan dan bimbingan dari berbagai pihak baik secara langsung maupun tidak langsung. Oleh karena itu penulis ingin menyampaikan rasa terimakasih kepada :

1. Allah SWT atas segala berkah, hidayah bimbingan dan keridhoan ilmu-Nya sehingga penulis dapat menyelesaikan tugas ini dengan baik.
2. Nabi Muhammad SAW yang selalu menjadi panutan dan suri tauladan.
3. Istri penulis sebagai sosok luar biasa yang memberikan segala dukungan dan semangat baik moril maupun material.

4. Bapak Prof. Dr. M. Suyanto, M.M., Rektor Universitas Amikom Yogyakarta, yang telah memberikan kebijaksanaan akademik kepada penulis dalam menyelesaikan tesis.
5. Ibu Ema Utami , Prof. Dr., S.Si., M.Kom. dan Bapak Anggit Dwi Hartanto, M.Kom, selaku dosen pembimbing penulis, yang telah memberikan bimbingan, arahan dan masukan, sehingga penulis dapat menyelesaikan tesis ini dengan baik.
6. Seluruh staff pengajar dan karyawan Universitas Amikom Yogyakarta yang telah memberikan ilmu, bimbingan, arahan dan masukan sehingga penulis dapat memperoleh ilmu yang bermanfaat.
7. Semua pihak yang telah membantu dalam pelaksanaan dan penyusunan tugas ini.

Semoga segala bantuan dan dukungan yang telah diberikan bermanfaat untuk penulis maupun pembaca.

DAFTAR ISI

TESIS	II
HALAMAN PENGESAHAN.....	III
HALAMAN PERSETUJUAN.....	IV
HALAMAN PERNYATAAN KEASLIAN TESIS.....	V
PERSEMBAHAN.....	VI
MOTTO.....	VII
KATA PENGANTAR.....	VIII
DAFTAR ISI.....	X
DAFTAR TABEL.....	XII
DAFTAR GAMBAR.....	XIII
INTISARI.....	XV
<i>ABSTRACT</i>	XVI
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	6
1.3 Batasan Masalah.....	7
1.4 Tujuan Penelitian.....	7
1.5 Manfaat Penelitian.....	8
1.6 Hipotesis.....	8
BAB II TINJAUAN PUSTAKA.....	9
BAB III METODE PENELITIAN.....	266
3.1 Jenis, Sifat, dan Pendekatan Penelitian.....	266
3.2 Metode Pengumpulan Data.....	266

3.2.1	Pemakaian Kuis.....	277
3.2.2	Penyaringan Data.....	29
3.2.3	Mining Data.....	300
3.3	Metode Analisis Data.....	322
3.3.1	Preprocessing.....	33
3.3.1.1	Tweet Cleaning.....	34
3.3.1.2	Tokenization.....	37
3.3.1.3	Normalisasi Kata.....	39
3.3.1.4	Stopword.....	40
3.3.1.5	Stemming.....	42
3.3.1.8	POS Tagging.....	43
3.3.1.6	POS Filtering.....	46
3.3.1.7	Tweet Merging.....	48
3.3.2	Pelatihan Data.....	49
3.3.2.1	Term Frequency.....	49
3.3.2.2	Konversi ke Biner.....	55
3.3.2.3	Pembuatan Model Neural Network.....	58
3.3.2.4	Feed Forward.....	59
3.3.2.5	Back Propagation.....	62
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....		63
BAB V PENUTUP.....		91
5.1	Kesimpulan.....	91
5.2	Saran.....	91
DAFTAR PUSTAKA.....		93
LAMPIRAN.....		96
1.	Alat Kuis DISC.....	96
2.	Skenario Pencarian Model Terbaik.....	99

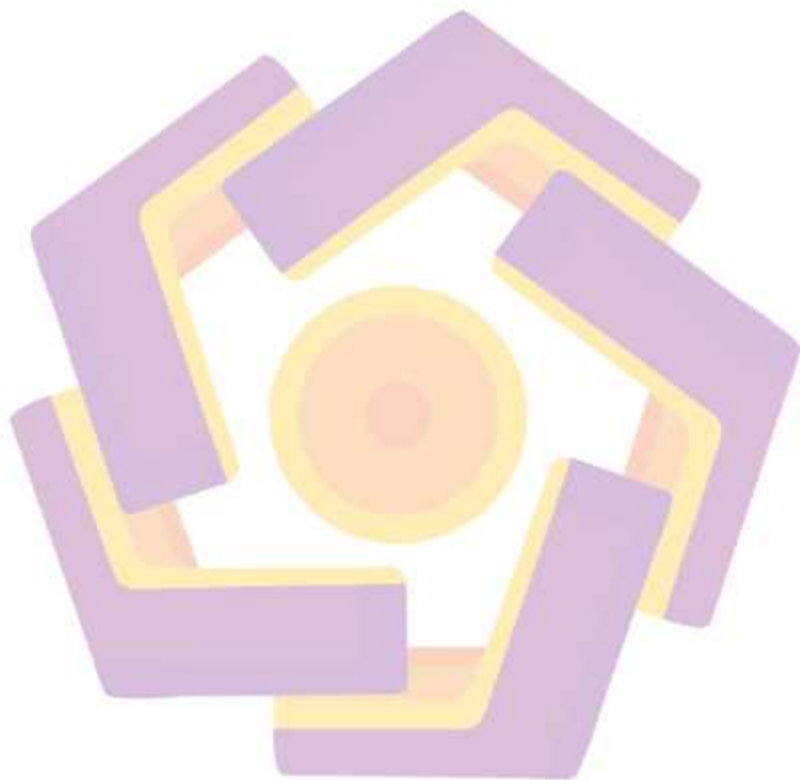
DAFTAR TABEL

Tabel 2.1 Matriks Literatur Review dan Posisi Penelitian.....	11
Tabel 3.1 Tweet yang telah di <i>scrape</i>	31
Tabel 3.2 Sebelum proses <i>Tweet Cleaning</i>	35
Tabel 3.3 Setelah proses <i>Tweet Cleaning</i>	36
Tabel 3.4 Setelah proses Tokenization.....	38
Tabel 3.5 Setelah proses normalisasi kata.....	39
Tabel 3.6 Setelah proses penghapusan stopword.....	41
Tabel 3.7 Setelah proses stemming.....	42
Tabel 3.8 Tagset.....	42
Tabel 3.9 Setelah proses POS tagging	44
Tabel 3.10 Setelah proses POS filtering	46
Tabel 3.11 Setelah proses tweet merging.....	48
Tabel 3.12 Proses term frequency untuk dominance	49
Tabel 3.13 Proses term frequency untuk influence	50
Tabel 3.14 Proses term frequency untuk steadiness.....	51
Tabel 3.15 Proses term frequency untuk compliance	52
Tabel 3.16 Tweet yang di scrape.....	53
Tabel 3.17 Hasil term frequency.....	54
Tabel 3.18 Tiga kata teratas	53
Tabel 3.19 Perhitungan layer input ke layer hidden	57
Tabel 3.20 Perhitungan layer hidden ke layer output	59
Tabel 4.1 Tabel hasil_kuis	62
Tabel 4.2 Tabel tweets	63
Tabel 4.3 Tabel Skenario Pencarian Akurasi Terbaik dengan K-Fold Cross Validation.....	81

DAFTAR GAMBAR

Gambar 3.1 Alur pengumpulan data	28
Gambar 3.2 Rancangan halaman Kuis utama	29
Gambar 3.3 Rancangan halaman Kuis hasil	30
Gambar 3.4 Alur metode analisis data	34
Gambar 3.5 Alur preprocessing	34
Gambar 3.6 Model neural network	56
Gambar 4.1 Alur sistem	61
Gambar 4.2 Data untuk input layer	64
Gambar 4.2 Data untuk output layer	64
Gambar 4.4 <i>Flow chart system</i>	65
Gambar 4.6 Antarmuka halaman Kuis	66
Gambar 4.7 Antarmuka halaman hasil Kuis	67
Gambar 4.8 Tabel hasil_kuis yang telah disaring	68
Gambar 4.9 Scrape data dengan twitter scraper	69
Gambar 4.10 Tabel tweets	70
Gambar 4.11 Tweet cleaning	71
Gambar 4.12 Tokenisasi	71
Gambar 4.13 Bahasa tidak baku	72
Gambar 4.14 Replace kata tidak baku	73
Gambar 4.15 Menghilangkan <i>stopword</i>	73
Gambar 4.16 <i>Stemming</i>	73
Gambar 4.17 Pelatihan <i>POS Tagging</i>	74
Gambar 4.18 Pemberian <i>tag</i>	74
Gambar 4.19 <i>POS Filtering</i>	74
Gambar 4.20 Penggabungan <i>tweet</i>	75
Gambar 4.21 Fungsi term frequency	76
Gambar 4.22 Konversi ke biner	78

Gambar 4.23 <i>Training data</i>	79
Gambar 4.24 <i>K-Fold Cross Validation</i>	80
Gambar 4.25 Tes akurasi menggunakan <i>K-Fold Cross Validation</i>	84
Gambar 4.26 Antarmuka halaman utama.....	85
Gambar 4.27 Antarmuka halaman prediksi	85



INTISARI

Maju mundurnya suatu perusahaan biasanya didukung oleh adanya sumber daya yang handal, terutama sumber daya manusia. Perekrutan dan penempatan pegawai pada posisi yang tepat akan membawa dampak yang signifikan bagi suatu perusahaan. Di dunia ini sifat dan karakter manusia sangat beraneka ragam bentuknya. Teori DISC mengklasifikasikan kepribadian menjadi empat tipe yaitu *dominance, influence, steadiness dan compliance*.

Perbedaan karakter setiap tipe tentu saja akan berpengaruh pada gaya perilaku, cara menghadapi tekanan hidup dan juga cara berkomunikasi baik secara langsung maupun dengan media sosial. Melalui sosial media, seseorang dapat meluapkan perasaannya melalui postingan yang diunggahnya. Dari postingan tersebut dapat dilakukan analisis mengenai karakter kepribadian yang ia dimiliki.

Penelitian ini bertujuan untuk mengetahui seberapa besar akurasi analisis profiling pada Twitter sehingga bisa menjadi acuan untuk proses perekrutan pegawai. Penelitian ini menggunakan algoritma *Artificial Neural Network* untuk mengklasifikasikan 275 akun Twitter kedalam teori DISC dan mendapatkan akurasi sebesar 42,91% dari 72 skenario yang dijalankan.

Kata Kunci: Kepribadian DISC, Media Sosial, Analisis Profiling, Sumber Daya Manusia.

ABSTRACT

The success or failure of a company is usually supported by the presence of reliable resources, especially human resources. Recruitment and placement of employees in the right position will have a significant impact on a company. Human nature and character are very diverse in their forms.

DISC theory classifies personality into four types namely dominance, influence, steadiness, and compliance. The difference in the character of each type will of course affect the behavior style, how to deal with life pressures, and also how to communicate both directly and with social media. Through social media, a person can vent his feelings through the posts he uploaded. From these posts an analysis of the personality character he is possessed can be carried out.

This study uses Artificial Neural Network algorithm to classify 275 Twitter accounts into DISC theory and got an accuracy of 42.91% from 72 scenarios.

Keywords: DISC, Social Media, Profiling Analysis, Human resources.

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Di dunia ini sifat dan karakter manusia sangat beranekaragam bentuknya. Pengklasifikasiannya pun sangat beranekaragam menurut sudut pandang dari tokoh/ahli yang mengklasifikasikannya. Karakter sifat seseorang tentu saja akan berbeda dengan karakter sifat orang lain karena mengingat setiap kepribadian mempunyai ciri karakteristik yang berbeda satu sama lain. Perbedaan karakter tersebut tentu saja akan berpengaruh pada perilakunya.

Saat ini terdapat beberapa cara dan pendekatan untuk dapat mengevaluasi dan memprediksi kecenderungan perilaku seseorang. Pada salah satu sisi, ada yang sekedar menggunakan suatu tes sederhana untuk menguji keterampilan dan kemampuan sedangkan pada sisi lainnya terdapat penggunaan tes kepribadian yang dibuat untuk memberikan gambaran umum tentang gaya dan perilaku seseorang selengkap mungkin. Ada beberapa metode tes yang digunakan untuk mengetahui karakter dan kepribadian seseorang. Alat tes model MBTI (*Myers, Briggs Type Indicator*) mengukur kepribadian seseorang menjadi empat dikotomi yaitu *introversion-extraversion*, *intuition-sensing*, *feeling-thinking* dan *perception judging* yang kemudian hasil dari tes tersebut akan dimasukkan ke dalam salah satu dari enam belas *personalities*. Di samping model tes MBTI ada juga model OCEAN atau *Big Five Personality Model* yang menganalisis tiga puluh *trait* dalam kepribadian di bawah *openness*, *conscientious*, *extraversion*, *agreeableness* dan *neuroticism*. Serta ada juga model tes kepribadian yang

menggunakan pendekatan DISC (*dominance, influence, steadiness, atau compliance*). Konsep DISC ini pertama kali dikembangkan oleh psikolog bernama *William Moulton Marston*. DISC adalah sebuah alat ukur psikometri yang mengukur gaya kepribadian seseorang mengenai perilaku kerjanya (Nofiar, 2009).

Pada saat ini pendekatan DISC memang bukanlah sebuah alat tes kepribadian yang lengkap atau tes psikometrik dalam pengertian teknis, tetapi alat ini memberikan gambaran mengenai gaya seseorang yang dapat memprediksi kecenderungan perilakunya di masa yang akan datang. Hal ini diperoleh dengan mengevaluasi faktor-faktor kepribadian utama yang ada dalam diri seseorang.

DISC ini memberikan banyak keuntungan dalam penggunaannya, jika alat tes yang lengkap sering berisi ratusan pertanyaan, dan membutuhkan waktu lama dalam melengkapinya, kuis DISC hanya berisi dua puluh empat pertanyaan, dan dapat diselesaikan dalam waktu hanya lima belas menit atau bahkan kurang (Shin, 2013). Keuntungan lainnya ada pada interpretasinya, pada tes lengkap merupakan hasil pekerjaan para ahli atau *expert*-nya, hasil DISC dapat dikerjakan dengan menggunakan suatu *software* dan dapat dikerjakan dengan otomatisasi, dengan demikian waktu pelaporan yang dibutuhkan juga akan jauh lebih cepat.

Seorang individu dapat dikatakan memiliki gaya kepribadian tertentu apabila memiliki skor tertinggi pada salah satu kategori dari keempat kategori DISC, yaitu *dominance, influence, steadiness, atau compliance*. Tes DISC ini biasanya mengukur bagaimana gaya perilaku seseorang, cara berkomunikasi, cara menghadapi tekanan dan sebagainya.

Twitter adalah situs jejaring sosial mikroblogging yang digemari oleh masyarakat Indonesia. Survei yang dilakukan oleh Asosiasi Penyelenggara Jasa Internet Indonesia menyebutkan bahwa Twitter merupakan media sosial yang menempati peringkat 4 dalam hal kunjungan oleh masyarakat Indonesia (*Asosiasi Penyelenggara Jasa Internet Indonesia, n.d.*). Kelebihan media sosial Twitter adalah dibatasinya jumlah karakter menjadi maksimal 280 karakter, sehingga di twitter tidak ada fitur “*read more*” karena karakter sudah ditampilkan semuanya di layar. *Tweet* yang diposting oleh pengguna tersebut dapat di analisa dan diolah menjadi informasi yang bermanfaat dengan teknik analisa sentimen.

Sentiment analysis yaitu proses memahami, mengekstrak dan mengolah data secara otomatis untuk mendapat informasi sentimen yang terdapat dari suatu opini (Buntoro et al., 2014). *Sentiment analysis* bisa juga disebut dengan *Opinion Mining* yang merupakan salah satu cabang penelitian *Text Mining*.

Opinion Mining merupakan riset berbasis komputasional yang dilakukan pada emosi dan opini sentimen yang diekspresikan secara tekstual (Zulfa & Winarko, 2017). Sentimen Analisis menjadi topik yang banyak dijadikan bahan penelitian oleh para peneliti.

Pada penelitian ini, penulis mengklasifikasikan opini sentimen tersebut kedalam empat tipe kepribadian berdasarkan teori DISC yaitu *dominance*, *influence*, *steadiness*, atau *compliance*. Teori kepribadian ini cukup banyak digunakan untuk penelitian seperti pada penelitian sebelumnya yang dilakukan oleh (Dewi Puri Astiti, S.Fil & Dr. Ni Made Swasti Wulanyani, S.Psi, 2015). Hasil penelitian yang dilakukan menunjukkan bahwa dari sebagian besar tenaga

medis ditemukan memiliki kepribadian *compliance* dan *steadiness*. Tipe kepribadian tersebut memiliki pengaruh yang signifikan terhadap perilaku kontraproduktif yang muncul.

Peneliti dalam mendapatkan hasil uji kepribadian pada *tweet* menggunakan metode *Jaringan Saraf Tiruan (JST)* atau *Artificial Neural Network (ANN)*. *Jaringan Saraf Tiruan (JST)* atau yang biasa disebut *Artificial Neural Network (ANN)* atau *Neural Network (NN)* merupakan paradigma pengolahan informasi yang terinspirasi dari sistem biologis saraf manusia seperti proses pengolahan informasi pada otak manusia (Sutojo et al., 2011). Paradigma ini didasari oleh bagaimana otak manusia mengatur sel-sel saraf (neuron) sehingga mampu melaksanakan tugas tertentu, misalnya pengenalan pola dengan efektivitas yang tinggi. JST memiliki struktur tersebar paralel yang besar dan mempunyai kemampuan belajar, sehingga dapat melakukan generalisasi atau menghasilkan keluaran yang benar untuk masukan yang belum pernah dilatihkan (Suyanto, 2014). Dengan kedua kemampuan ini, JST dapat menyelesaikan masalah-masalah yang kompleks.

Salah satu pendekatan dalam pelatihan JST adalah *supervised learning* dimana jaringan diberi pola dari masukan dan target/keluaran yang berfungsi sebagai pembimbing untuk melatih jaringan sehingga diperoleh bobot yang terbaik. Pendekatan *supervised learning* yang banyak digunakan dalam pelatihan JST yaitu *backpropagation* karena *backpropagation* mampu memperkecil tingkat *error* dengan cara menyesuaikan bobotnya berdasarkan *output* dan target yang diinginkan. Hal ini dikarenakan *backpropagation* memiliki tiga layer dalam proses

pelatihannya yaitu *input layer*, *hidden layer* dan *output layer*, dimana *backpropagation* ini merupakan perkembangan dari *single layer network* (jaringan layer tunggal) yang memiliki dua layer yaitu *input layer* dan *output layer*. Dengan adanya *hidden layer* pada *backpropagation* dapat menyebabkan besarnya tingkat *error* pada *backpropagation* lebih kecil dibanding dengan tingkat *error* pada *single layer network*. Hal tersebut dikarenakan *hidden layer* pada *backpropagation* sebagai tempat untuk mengupdate dan menyesuaikan bobot, sehingga didapatkan nilai bobot yang baru yang bisa diarahkan mendekati dengan target output yang diinginkan.

Beberapa peneliti telah menggunakan *backpropagation* di dalam penelitiannya dan hasil yang didapatkanpun menunjukkan hasil yang optimal. (Lhaksana, KM; Nhita, Fhira & Anggraini, 2017) menggunakan metode *backpropagation* untuk mengklasifikasi kepribadian seseorang berdasarkan status facebooknya. Hasil dari penelitian ini menunjukkan bahwa metode *backpropagation* menghasilkan performansi sistem yang baik dan mendapatkan nilai error dengan min kesalahan 0,00001% pada pengujian 2 dan 3. Dan menghasilkan performansi sistem dengan akurasi tertinggi 84,00% didapatkan pada proporsi data 70:30 % data yang diujikan. (Andrianto & Adinugroho, 2018) juga menggunakan metode *backpropagation* untuk menganalisis sentimen konten radikal melalui dokumen twitter. Dari hasil penelitian yang dilakukan menunjukkan bahwa metode ini mempunyai tingkat akurasi data sebesar 90 % yang diperoleh dari nilai *learning rate* 0,5 jumlah data latih 60 dan iterasi maksimum 40. Jurnal penelitian yang ditulis oleh (Solikhun et al., 2017) pun

menunjukkan bahwa penggunaan algoritma *backpropagation* dalam memprediksi tingkat pemahaman siswa terhadap mata pelajaran mempunyai tingkat akurasi yang tinggi. Hasil penelitian yang mereka lakukan menunjukkan bahwa JTS dapat melakukan prediksi terhadap pemahaman siswa pada mata pelajaran di atas 90 % keakuratan kebenarannya.

Berdasarkan beberapa latar belakang dan permasalahan yang telah disampaikan, penulis tertarik ingin melakukan penelitian tentang *tweet* seseorang pada *twitter* yang akan diklasifikasikan kepribadiannya menggunakan metode DISC dan mengolah data tersebut dengan algoritma *backpropagation*. Untuk itu penulis mengambil judul penelitian "Klasifikasi Kepribadian dengan Metode Dominance, Influence, Steadiness, Compliance (DISC) Pada Twitter Menggunakan Algoritma Artificial Neural Network Classification (Backpropagation)".

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, maka dapat diambil perumusan masalah dalam penelitian ini, yaitu:

1. Dapatkah postingan twitter digunakan sebagai fitur klasifikasi kepribadian dengan metode DISC?
2. Berapa besar tingkat akurasi pemprediksian kepribadian terhadap akun twitter dengan metode DISC menggunakan *algoritma Backpropagation*?

1.3 Batasan Masalah

Agar penelitian dalam penelitian ini lebih terarah dan memudahkan dalam pembahasan, maka perlu adanya batasan masalah, yaitu:

1. Tweet yang dianalisis adalah tweet berbahasa Indonesia
2. Tweet yang digunakan hanya tweet yang berupa teks, tidak mengandung gambar, video, gif, maupun emoticon.
3. Data kepribadian diklasifikasikan kedalam 4 kategori, yaitu: *Dominance, Influence, Steadiness, dan Compliance.*
4. Alat kuis diambil dari buku psikologi DISC berjudul "The DISC Codes: Cara Cepat Menguasai Kode Sukses Manusia" (Shin, 2013)
5. Kuis disebarakan selama 1-31 Maret 2020
6. Jumlah data *training* adalah 275 akun twitter dengan jumlah *tweet* sebanyak 356.415 tweet.
7. Proses pemrediksian menggunakan data yang sudah dilatih.
8. Pemvalidasian menggunakan *k-fold cross validation.*

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah

- a) Untuk dapat mengetahui apakah postingan twitter berbahasa Indonesia dapat digunakan dalam pengklasifikasian empat tipe kepribadian, yaitu *Dominance, Influence, Steadiness, dan Compliance.*
- b) Untuk mengetahui tingkat akurasi pemrediksian kepribadian terhadap akun twitter dengan metode DISC menggunakan algoritma *Backpropagation (ANN).*

1.5 Manfaat Penelitian

Manfaat yang diberikan pada penelitian ini adalah

1. Untuk membantu mengklasifikasikan akun twitter ke dalam empat tipe kepribadian, yaitu *Dominance*, *Influence*, *Steadiness*, dan *Compliance*.
2. Untuk mengetahui tingkat akurasi pemprediksian kepribadian terhadap akun twitter seseorang dengan metode DISC menggunakan algoritma *Backpropagation* (ANN).

Selain itu penelitian ini juga dapat digunakan untuk referensi bagi penelitian selanjutnya yang berhubungan dengan pengklasifikasian kepribadian akun twitter, maupun penelitian yang berhubungan dengan Analisis Sentimen dan juga *Document Classification*.

1.6 Hipotesis

Ho : Klasifikasi kepribadian dengan metode *dominance*, *influence*, *steadiness*, dan *compliance* (DISC) tidak dapat digali dari status twitter dengan menggunakan algoritma *backpropagation* (ANN).

Ha : Klasifikasi kepribadian dengan metode *dominance*, *influence*, *steadiness*, dan *compliance* (DISC) dapat digali dari status twitter dengan menggunakan algoritma *backpropagation* (ANN).

BAB II TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Penelitian ini mengacu kepada beberapa referensi yang memuat teori-teori yang berkaitan dengan penelitian yang dilakukan. Referensi-referensi tersebut nantinya akan digunakan sebagai landasan dan sekaligus sebagai pembatas dalam melakukan penelitian. Berikut adalah beberapa referensi yang peneliti gunakan didalam penelitian ini.

Penelitian yang dilakukan oleh (Dewi Puri Astiti, S.Fil & Dr. Ni Made Swasti Wulanyani, S.Psi, 2015) dengan menggunakan metode DISC menunjukkan bahwa dari sebagian besar tenaga medis ditemukan memiliki kepribadian *compliance* dan *steadiness*. Tipe kepribadian tersebut memiliki pengaruh yang signifikan terhadap perilaku kontraproduktif yang muncul.

Selanjutnya adalah penelitian yang dilakukan oleh (Andrianto & Adinugroho, 2018) dengan menggunakan algoritma *backpropagation* untuk proses klasifikasi analisis sentimen konten radikal melalui dokumen twitter. Penelitian ini menghasilkan akurasi sebesar 90% dengan data latih sebanyak 60 dataset, *learning rate* sebesar 0,5, dan iterasi maksimum sebanyak 40 kali.

Penelitian (Lhaksana, KM; Nhita, Fhira & Anggraini, 2017) melakukan penelitian dengan mengumpulkan data pelamar yaitu berupa data status facebook menggunakan 100 orang pelamar yang akan di ambil maksimal 5 status dari setiap akun. Jadi total data ada 468 data status dari seluruh pelamar. Dengan menggunakan metode *backpropagation* untuk mengklasifikasi kepribadian menghasilkan performansi sistem yang baik dan mendapatkan nilai

error dengan min kesalahan 0,00001 % pada pengujian 2 dan 3. Adapun hasil pengujian yang telah dilakukan hasil performansi sistem dengan akurasi tertinggi 84,00 % didapatkan pada proporsi data 70:30 % data yang diujikan.

Penelitian (Solikhun et al., 2017) juga melakukan penelitian tentang penggunaan algoritma *backpropagation* untuk memprediksi tingkat pemahaman siswa dalam suatu mata pelajaran. Hasil yang diperoleh menunjukkan bahwa JTS dapat melakukan prediksi terhadap pemahaman siswa dalam mata pelajaran tertentu di atas 90 % keakuratan kebenarannya.



2.2 Keaslian Penelitian

Tabel 2. Matriks literatur review dan posisi penelitian
 “Klasifikasi Kepribadian Dengan Metode Dominance, Influence, Steadiness, Compliance (DISC) Pada Twitter Menggunakan Algoritma Artificial Neural Network Clasification (Backpropagation)”

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1.	Klasifikasi Kepribadian Berdasarkan Status Facebook Menggunakan Metode <i>Backpropagation</i> []	Kemas Muslim Lhaksana, Fhira Nhita, Duwi Anggraini, e-Proceeding of Engineering ISSN : 2355-9365, 2017 Nama peneliti, Sumber, Tahun penelitian	Mengetahui kepribadian seseorang pelamar pekerjaan melalui status facebook yang terdiri dari <i>social word</i> , <i>positive emotions</i> , dan <i>negative emotions</i>	<ul style="list-style-type: none"> Berdasarkan hasil pengujian yang dilakukan menggunakan metode <i>backpropagation</i> menghasilkan performansi sistem yang baik dan mendapatkan nilai <i>error</i> dengan min kesalahan 0,00001 % pada pengujian 2 dan 3. Hasil pengujian yang telah dilakukan hasil peromansi sistem dengan akurasi tertinggi 84,00 % didapatkan pada proporsi data 70:30 % data yang diujikan 	Pada penelitian penggunaan <i>neuron hidden layer</i> tidak bisa menentukan neuron mana yang paling bagus menentukan nilai, karena neuron itu trial dan error	<ul style="list-style-type: none"> Dalam rencana penelitian kami, yang menjadi fokus yaitu pengklasifikasian kepribadian seseorang dengan metode DISC (dominance, influence, steadiness, compliance) media sosial yang kami gunakan dalam penelitian adalah twitter yang jumlah karakternya sudah ditentukan yaitu maksimal 280 karakter

Tabel 2. Matriks literatur review dan posisi penelitian (lanjutan)
 “Klasifikasi Kepribadian Dengan Metode Dominance, Influence, Steadiness, Compliance (DISC) Pada Twitter Menggunakan
 Algoritma Artificial Neural Network Clasification (Backpropagation)”

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
2	Analisis Sentimen Konten Radikal Melalui Dokumen Twitter Menggunakan Metode Backpropagation[]	Brian Andrianto, Indriati, Sigit Adimugroho, Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, e-ISSN: 2548-964X http://j-ptiik.ub.ac.id , 2018	Menganalisis tweet berbahasa Indonesia yang mengandung kata radikal, termasuk dalam konten radikal positif atau radikal negatif	Berdasarkan proses pengujian, sistem menghasilkan parameter terbaik di nilai learning rate: 0,5, jumlah data latih: 60, Iterasi maksimum 40 dengan hasil akurasi sebesar 90%	Pada penelitian ini jumlah data latih yang dipakai hanya 100. Diharapkan pada nanti kedepannya pada penelitian selanjutnya, data latih bisa ditambah	<ul style="list-style-type: none"> • Dalam rencana penelitian kami, yang menjadi fokus yaitu pengklasifikasian kepribadian seseorang dengan metode DISC (dominance, influence, steadiness, compliance) • media sosial yang kami gunakan dalam penelitian adalah twitter yang jumlah karakternya sudah ditentukan yaitu maksimal 280 karakter

Tabel 2. Matriks literatur review dan posisi penelitian (lanjutan)
 “Klasifikasi Kepribadian Dengan Metode Dominance, Influence, Steadiness, Compliance (DISC) Pada Twitter Menggunakan
 Algoritma Artificial Neural Network Clasification (Backpropagation)”

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
3	Jaringan Saraf Tiruan Untuk Memprediksi Tingkat Pemahaman Sisiwa Terhadap Matapelajaran Dengan Menggunakan Algoritma Backpropagation[]	Solikhun, M. Safi, Agus Trisno, Jurnal Sains Komputer & Informatika (J-SAKTI) ISSN:2548-9771/EISSN: 2549-7200 http://ejournal.tun.asbangsa.ac.id/index.php/lsakti , 2017	Untuk memprediksi tingkat pemahaman siswa terhadap mata pelajaran	Algoritma <i>backpropagation</i> dapat melakukan prediksi tingkat pemahaman siswa terhadap mata pelajaran dengan menunjukkan performa diatas 90%.	-	<ul style="list-style-type: none"> • Dalam rencana penelitian kami, yang menjadi fokus yaitu pengklasifikasian kepribadian seseorang dengan metode DISC (dominance, influence, steadiness, compliance) • media sosial yang kami gunakan dalam penelitian adalah twitter yang jumlah karakternya sudah ditentukan yaitu maksimal 280 karakter

Tabel 2. Matriks literatur review dan posisi penelitian (lanjutan)
 “Klasifikasi Kepribadian Dengan Metode Dominance, Influence, Steadiness, Compliance (DISC) Pada Twitter Menggunakan Algoritma Artificial Neural Network Clasification (Backpropagation)”

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
4	Perilaku Kontraproduktif Dan Model Kepribadian Disc Pada Tenaga Kesehatan Di Rumah Sakit Umum Denpasar[]	Dewi Puri Astiti, Ni Made Swasti Wulanyani, Sosial-humaniora/Psikologi Universitas Udayana Bali, 2015	<ul style="list-style-type: none"> • Untuk mendeteksi perilaku kontraproduktif tenaga kesehatan RSU di Denpasar • Untuk melihat profile kepribadian tenaga kesehatan yang berperilaku kontraproduktif dengan model DISC • Untuk melihat hubungan antara variabel kepribadian Model DISC dengan variabel perilaku kontraproduktif. 	Sebagian besar tenaga medis memiliki kepribadian Compliance dan Steadiness. Tipe kepribadian tersebut memiliki pengaruh yang signifikan terhadap perilaku kontra produktif yang muncul		<ul style="list-style-type: none"> • Dalam rencana penelitian kami, yang menjadi fokus yaitu pengklasifikasian kepribadian seseorang dengan metode DISC (dominance, influence, steadiness,compliance) • media sosial yang kami gunakan dalam penelitian adalah twitter yang jumlah karakternya sudah ditentukan yaitu maksimal 280 karakter

2.3 Landasan Teori

Penulis dalam melakukan penelitian sebagai bahan untuk mendukung penelitian merujuk pada beberapa teori dan referensi, beberapa sumber teori dan referensi yang berkaitan dengan penelitian klasifikasi kepribadian dengan menggunakan metode DISC dan sistem pengklasifikasian data menggunakan metode *Artificial Neural Network*.

2.3.1 Klasifikasi Kepribadian dengan Metode Dominance, Influence, Steadiness, Compliance (DISC)

Karakter dan kepribadian seseorang sangat bermacam-macam tipenya. Ada beberapa metode untuk mengetahui karakter dan kepribadian seseorang, salah satunya yaitu dengan metode DISC. DISC merupakan alat psikometri yang di buat untuk mengukur perilaku seseorang dan bagaimana seseorang bereaksi dalam situasi tertentu (Arga January & Adrian, 2014). DISC mencoba mengkategorikan individu dalam empat elemen kepribadian yaitu *dominan*, *influence*, *steadiness*, dan *compliance*. Alat ukur DISC dapat menentukan elemen apa yang paling tinggi dari seseorang sehingga individu akan memiliki gaya bekerja seperti apa dalam organisasi atau situasi kerja. DISC dapat menunjukkan kelemahan dan kelebihan seseorang sesuai dengan elemen kepribadian.

Individu dengan kepribadian dominan adalah bagaimana seseorang merespon masalah dan tantangan dengan menggunakan kekuasaan, karakteristik cerdas, tegas, dan langsung. *Influence* adalah bagaimana seseorang berinteraksi dan mencoba mempengaruhi orang lain, karakteristik mudah bergaul, supel dan komunikatif. Dan *Steadiness* adalah bagaimana seseorang merespon perubahan,

variasi dan kecepatan di lingkungan secara konsisten, karakteristik keras hati, gigih dan sabar. Kemudian, *Compliance* adalah bagaimana seseorang merespon peraturan dan prosedur yang di tetapkan pihak lain, karakteristik teratur, akurasi dan fokus pada fakta (Dewi Puri Astiti, S.Fil & Dr. Ni Made Swasti Wulanyani, S.Psi, 2015).

Dalam penerapannya di dunia bisnis dan usaha, alat ini telah membuka wawasan dan pemikiran, baik secara profesional maupun secara personal. Penerapan metode tes kepribadian DISC pertama kali digunakan dalam bidang militer dan kemudian untuk selanjutnya digunakan pada perekrutan tentara Amerika Serikat pada tahun – tahun menjelang Perang Dunia II setelah keandalannya terbukti kemudian digunakan untuk kepentingan umum.

2.3.2 Twitter

Perkembangan teknologi komunikasi memberikan pengaruh yang besar terhadap kegiatan komunikasi kehidupan manusia saat ini. Tamburaka menjelaskan bahwa kehadiran media sosial sebagai salah satu wujud perkembangan teknologi yang dimanfaatkan manusia modern untuk berkomunikasi (Tamburaka, 2013). Twitter merupakan salah satu dari sekian banyak media sosial yang menarik perhatian pengguna internet. Hal tersebut disebabkan penggunaannya yang mudah guna saling bertukar informasi sehingga setiap individu di planet ini dapat saling terhubung (MADCOMS, 2010).

Twitter, menurut situs resmi twitter, berdiri di San Francisco pada tanggal 19 April 2007. Misinya adalah untuk memberi kesempatan bagi setiap orang untuk

dapat saling menciptakan dan berbagi ide ide serta informasi secara langsung tanpa hambatan(*Twitter - Perusahaan, n.d.*).

Twitter juga menjadi salah satu situs jejaring sosial mikroblogging yang digemari oleh masyarakat Indonesia. Survei yang dilakukan oleh Asosiasi Penyelenggara Jasa Internet Indonesia menyebutkan bahwa Twitter merupakan media sosial yang menempati peringkat 4 dalam hal kunjungan oleh masyarakat Indonesia (*Asosiasi Penyelenggara Jasa Internet Indonesia, n.d.*). Kelebihan media sosial Twitter adalah dibatasinya jumlah karakter menjadi maksimal 280 karakter, sehingga di twitter tidak ada fitur "read more" karena karakter sudah ditampilkan semuanya di layar. *Tweet* yang diposting oleh pengguna tersebut dapat di analisa dan diolah menjadi informasi yang bermanfaat dengan teknik analisa sentimen.

2.3.3 Artificial Neural Network

JST atau yang biasa disebut *Artificial Neural Network* (ANN) atau *Neural Network* (NN) merupakan paradigma pengolah informasi yang terinspirasi dari sistem biologis saraf manusia seperti proses pengolahan informasi pada otak manusia(Sutojo et al., 2011). Paradigma ini didasari oleh bagaimana otak manusia mengatur sel-sel saraf (neuron) sehingga mampu melaksanakan tugas tertentu, misalnya pengenalan pola dengan efektivitas yang tinggi. JST memiliki struktur tersebar paralel yang besar dan mempunyai kemampuan belajar, sehingga dapat melakukan generalisasi atau menghasilkan keluaran yang benar untuk masukan yang belum pernah dilatihkan(Suyanto, 2014). Dengan kedua kemampuan ini, JST dapat menyelesaikan masalah-masalah yang kompleks.

Metode Artificial Neural Network pada saat ini sudah banyak diimplementasikan untuk pengklasifikasian di berbagai bidang karena memiliki kelebihan-kelebihan antara lain *adaptive*, *self-organization*, *real time operation*, kemampuan mengakuisisi pengetahuan yang baik, kemampuan untuk memberikan toleransi atas suatu distorsi (*error/fault*), dan kemampuan memproses pengetahuan secara efisien karena memakai sistem paralel, sehingga waktu yang diperlukan untuk mengoperasikannya jauh lebih singkat.

Pelatihan pada JST memiliki beberapa pendekatan, yaitu *unsupervised learning*, *reinforcement learning*, dan *supervised learning*. *Unsupervised learning* atau *selforganized learning* tidak membutuhkan pembimbing untuk memantau proses pembelajaran. Pada *unsupervised learning* jaringan hanya diberi masukan tetapi tidak mendapatkan target yang diinginkan, sehingga modifikasi bobot pada jaringan dilakukan berdasarkan parameter tertentu. Sedangkan pada *supervised learning*, diberikan pola dari masukan dan target (keluaran) yang berfungsi sebagai pembimbing untuk melatih jaringan sehingga diperoleh bobot yang terbaik. Beberapa metode pembelajaran dengan supervisi di antaranya *Hebb Rule*, *Perceptron*, *Back Propagation*, *Bidirectional Associative Memory (BAM)*, dan *Learning Vector Quantization (LVQ)*.

2.3.3.1 Algoritma Backpropagation

Salah satu pendekatan dalam pelatihan JST adalah *supervised learning* dimana jaringan diberi pola dari masukan dan target/keluaran yang berfungsi sebagai pembimbing untuk melatih jaringan sehingga diperoleh bobot yang terbaik. Pendekatan *supervised learning* yang banyak digunakan dalam pelatihan

JST yaitu *backpropagation* karena *backpropagation* mampu memperkecil titik *error* dengan cara menyesuaikan bobotnya berdasarkan *output* dan target yang diinginkan. Hal ini dikarenakan *backpropagation* memiliki tiga layer dalam proses pelatihannya yaitu *input layer*, *hidden layer* dan *output layer*, dimana *backpropagation* ini merupakan perkembangan dari *single layer network* (jaringan layer tunggal) yang memiliki dua layer yaitu *input layer* dan *output layer* (Kusumadewi, 2003). Dengan adanya *hidden layer* pada *backpropagation* dapat menyebabkan besarnya tingkat *error* pada *backpropagation* lebih kecil dibanding dengan tingkat *error* pada *single layer network*. Hal tersebut dikarenakan *hidden layer* pada *backpropagation* sebagai tempat untuk mengupdate dan menyesuaikan bobot, sehingga didapatkan nilai bobot yang baru yang bisa diarahkan mendekati dengan target output yang diinginkan.

Secara umum langkah utama dari algoritma *backpropagation* adalah pengambilan *input*, penelusuran *error* dan penyesuaian bobot. *Error* ditelusuri mulai dari *output layer*, kemudian ke *hidden layer* dan akhirnya ke *input layer*. Setiap iterasi dalam *backpropagation* dilakukan dengan dua arah yaitu *Forward* (tahap maju) dan *Backward* (tahap mundur). Berikut langkah-langkah pelatihan *backpropagation* (Nurmila et al., 2010):

2.3.3.1.1 Langkah 0

- a. Inisialisasi bobot (ambil bobot awal dengan nilai *random* yang cukup kecil)
- b. Tetapkan maksimum target *error* dan *learning rate*.

2.3.3.1.2 Feed Forward

Langkah 1 :

a. Menjumlahkan sinyal-sinyal *input* yang memiliki bobot:

$$Z_{in_j} = b1_j + \sum_{i=1}^n w_{ij} \cdot x_i \quad (1)$$

Keterangan :

Z_{in_j} : Hasil penjumlahan sinyal input (*hidden layer*)

$b1_j$: bias

w_{ij} : bobot (*weight*) yang menghubungkan input layer dengan *hidden layer*

x_i : Input data yang akan diolah

b. Fungsi aktivasi untuk menghitung sinyal output

$$z_j = f(z_{in_j}) \quad (2)$$

Keterangan:

z_j : Nilai *hidden output*

$f(z_{in_j})$: Fungsi aktivasi untuk menghitung nilai *hidden output*

Langkah 2 :

a. Menjumlahkan sinyal-sinyal *input* terbobot:

$$Y_{in_k} = b2_k \sum_{i=1}^p Z_i \cdot W_{jk} \quad (3)$$

Keterangan :

Y_{in_k} : Hasil penjumlahan sinyal *hidden output*

$b2_k$: Tambahan *input* yang biasa di sebut bias

Z_i : Nilai *input* dari *hidden output*

W_{jk} : Bobot dari koneksi antara dua *node*

b. Fungsi aktivasi untuk menghitung sinyal *output* :

$$Y_k = f(Y_{in_k}) \quad (4)$$

Y_k : Nilai output prediksi

$f(Y_{in_k})$: Fungsi aktivasi untuk menghitung nilai output prediksi

2.3.3.1.3 Back Propagation

Langkah 3 :

- a. Hitung informasi *error*-nya:

$$\delta = (t_k - T_k)f'(Y_{in_k}) \quad (5)$$

Keterangan

δ : Menghitung informasi error

T_k : *Output prediksi*

$f'(Y_{in_k})$: Turunan fungsi aktivasi

- b. Hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai):

$$\Delta W_{jk} = \alpha \delta_{jk} x_{jk} \quad (6)$$

Keterangan:

ΔW_{jk} : Menghitung koreksi bobot

α : Learning rate

δ_{jk} : Informasi error

x_{jk} : Output hasil prediksi

- c. Hitung koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai):

$$\Delta b_{2k} = \alpha \delta_k \quad (7)$$

Keterangan :

Δb_{2k} : Menghitung koreksi bobot

α : Learning rate

δ_k : Informasi error

Langkah 4 :

- a. Menjumlahkan delta *input* (dari unit-unit yang berada pada lapisan di atasnya :

$$\delta_{in_j} = \sum_{k=1}^m \delta_{2k} W_{jk} \quad (8)$$

- b. Kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi *error*:

$$\delta_{1j} = \delta_{in_j} f'(Z_{in_j}) \quad (9)$$

- c. Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai):

$$\Delta V_{ij} = \alpha \varphi_{ij} \quad (10)$$

- d. Hitungkan juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai):

$$\Delta b_{1j} = \alpha \beta_{1j} \quad (11)$$

2.3.3.1.4 Memperbarui Bobot

Langkah 5 :

- a. Tiap-tiap unit *output* memperbaiki bias dan bobotnya :

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \quad (12)$$

$$b2_k(\text{baru}) = b2_k(\text{lama}) + \Delta b2_k \quad (13)$$

b. Tiap-tiap unit tersembunyi memperbaiki bias dan bobotnya :

$$V_{ij}(\text{baru}) = V_{ij}(\text{lama}) + \Delta V_{ij} \quad (14)$$

$$b1_j(\text{baru}) = b1_j(\text{lama}) + \Delta b1_j \quad (15)$$

Langkah 6 : Hitung total *error*

Menghitung total *error* menggunakan *mean squared error* dengan rumus perhitungan:

$$MSE = \frac{1}{2} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (16)$$

Keterangan :

MSE : Merupakan total *error*

Y_i : Merupakan target *output* ke- i

\hat{Y}_i : Merupakan *output* dari pelatihan ke- i

Fase tersebut diulang hingga kondisi *error* terpenuhi

berbagai macam pola yang dinyatakan oleh setiap pasangan masukan keluaran akan meningkat.

2.3.4 K-Fold Cross Validation

K-Fold Cross Validation adalah salah satu metode yang digunakan untuk validasi sebuah dataset. Cara kerja dari metode ini adalah dengan cara data akan di partisi / dibagi menjadi k bagian/ *fold* segment dengan ukuran yang sama. Proses *training* dan *testing* data dilakukan sebanyak k kali dimana pada setiap

iterasi ke- i , segment K_i akan menjadi data testing sedangkan yang lain berperan sebagai data training.

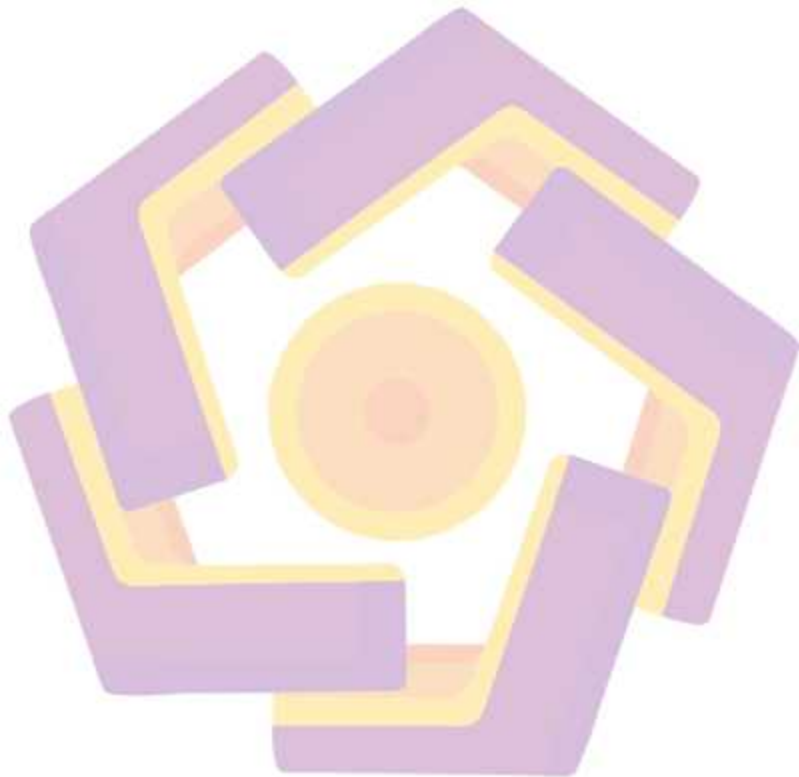
2.3.5 Preprocessing

Preprocessing merupakan bagian dari penyiapan dataset, Sebelum proses pengklasifikasian, dataset yang didapat harus di olah terlebih dahulu agar siap untuk digunakan untuk tahap selanjutnya. Proses ini disebut sebagai *preprocessing*. *Preprocessing* dibutuhkan agar data yang diolah nantinya berkualitas dan dapat meningkatkan proses penggalian informasi, yaitu pada penanganan *noisy text* / teks yang tidak memiliki makna, *incomplete data* / data yang tidak lengkap, dan *inconsistent data* / data yang tidak konsisten (Kadir, 2005).

Penelitian ini menggunakan dataset yang didapat dari Twitter, yaitu berupa tweet. *Tweet-tweet* tersebut tidak rapi/terstruktur, karena tidak ada aturan bagi pengguna untuk membuat *tweet*. Sehingga, seringkali didalam tweet tersebut terdapat kata-kata yang tidak baku, simbol-simbol, tagar, dan lain-lain. Untuk itu, tweet akan diolah terlebih dahulu dengan urutan sebagai berikut:

1. *Case Folding*, merupakan tahapan untuk mengubah keseluruhan kalimat menjadi huruf kecil.
2. Menghilangkan URL/Tautan
3. Menghilangkan nama user (*username*) dan simbol at (@)
4. Menghilangkan tagar/hashtag
5. Tokenisasi, yaitu membagi/memisahkan kalimat menjadi perkata
6. Mengubah kata yang tidak baku menjadi kata baku

7. *Stemming*, yaitu menghilangkan imbuhan pada kata
8. *Stopwords removal* yaitu tahapan untuk menghilangkan kata-kata yang kurang memiliki makna, seperti kata: yang, tetapi, atau, ke, di, dengan, dan sebagainya



BAB III

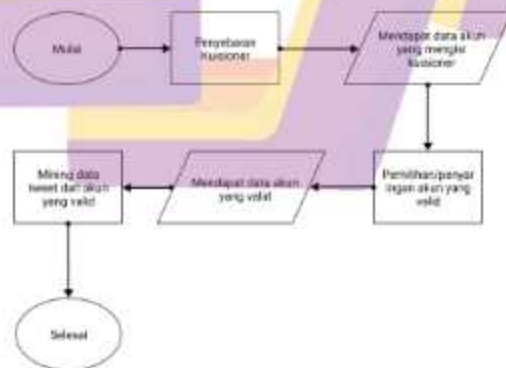
METODE PENELITIAN

3.1 Jenis, Sifat, dan Pendekatan Penelitian

Dalam penelitian ini peneliti menggunakan jenis penelitian deskriptif untuk membuat gambaran secara sistematis, faktual dan akurat mengenai fakta-fakta dan sifat-sifat pada suatu obyek penelitian. Penelitian ini bersifat kuantitatif dimana dalam mengumpulkan data menggunakan kuisioner berupa kuis setelah data terkumpul, data diolah dan diinterpretasikan sebuah hasil pengklasifikasian jenis kepribadian *dominance*, *influence*, *steadiness*, dan *compliance*.

3.2 Metode Pengumpulan Data

Sebelum mengumpulkan data, penelitian ini harus mencari nama pengguna dan tipe kepribadian dari pengguna terlebih dahulu. Label tipe kepribadian pengguna dibutuhkan agar bisa digunakan untuk *data training* maupun *data testing*.



Gambar 3.1 Alur Pengumpulan Data

3.2.1 Pemakalan Kuis

Pada penelitian ini, peneliti memakai sebuah kuis DISC dari dari sumber berupa buku psikolog berjudul “The DISC Codes: Cara Cepat Menguasai Kode Sukses Manusia” (Shin, 2013) yang berisi dua puluh empat pertanyaan, kemudian dikonversi menjadi halaman web agar semua partisipan bisa mengakses secara daring. Kuis ini dapat diselesaikan dalam waktu hanya lima belas menit atau kurang. Sumber dari kuis ini, yaitu buku “The DISC Codes: Cara Cepat Menguasai Kode Sukses Manusia” terbukti membuat beberapa perusahaan lebih mengenali para karyawannya, serta lebih bisa menempatkan, mengembangkan, memotivasi, menghargai, dan mempertahankan para kadernya (Shin, 2013). Uji statistik pada kuis ini tidak diperlukan, karena alat kuis berasal dari buku.

Setelah kuis tersebut dikonversikan ke sebuah website, tautan website di publikasikan di Twitter agar pengguna twitter tahu kemudian mengisi kuis tersebut. kemudian pengguna twitter bisa memilih sifat dan karakter yang mereka miliki. Setelah pengguna memilih, pengguna harus memasukkan username mereka sehingga mereka dapat mensubmit kuis tersebut dan sistem kuis akan menampilkan hasil berupa tipe kepribadian berdasarkan sifat dan karakter yang telah mereka isikan sebelumnya. Selain itu, pengguna juga dapat mengetahui deskripsi daripada tipe kepribadian tersebut, serta kelebihan dan juga kelemahannya. Data pengguna akan disimpan kedalam *database* yang berisi sejumlah kolom.

Dalam rancangan halaman utama Kuis, pengguna akan diperintah untuk memilih sifat atau karakter yang ada pada diri mereka. Isian pada kuis terdiri dari

M (*Most*) dan L (*Least*). *Most* adalah pilihan paling mendekati sifat pengguna, sedangkan *Least* adalah pilihan yang paling tidak mendekati sifat pengguna. Setelah pengguna mengisi semua isian sifat, mereka harus mengisikan *username* twitter mereka agar nantinya datanya bisa di *scrape*.

Tes Kepribadian DISC

M	L
<input checked="" type="radio"/>	<input type="radio"/>
<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	<input checked="" type="radio"/>
<input type="radio"/>	<input type="radio"/>

M	L
<input checked="" type="radio"/>	<input type="radio"/>
<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	<input checked="" type="radio"/>

Masukkan username twitter: _____

Submit

Gambar 3.2 Rancangan halaman kuis utama

Setelah partisipan mengisi semua kolom input di halaman utama dan mensubmitnya, maka partisipan akan langsung diarahkan ke halaman hasil kuis. Rancangan halaman hasil kuis berisi hasil tipe kepribadian berdasarkan sifat yang pengguna pilih di halaman utama (*Dominance, Influence, Steadiness, dan Compliance*). Halaman ini juga menampilkan informasi tentang tipe kepribadian tersebut dan juga kelebihan serta kelemahannya.

@idris adalah seorang

Compliance

Karakter Umum Compliance

Kelebihan Compliance

Kekurangan

Pilihan Kari Compliance

Submit

Gambar 3.3 Rancangan halaman hasil kuis

3.2.2 Penyaringan Data

Pengguna yang telah mensubmit data tentunya bisa saja tidak serius dalam mengisi data yang ia submit. Untuk itu dilakukanlah penyaringan data agar data yang didapatkan valid. Peneliti menerapkan beberapa aturan berikut untuk menyaring data yang tidak valid berdasarkan penelitian (X. Liu & Zhu, 2016) :

1. Jika total *tweet* dari pengguna sebanyak 500 *tweet* atau lebih dari itu, pengguna adalah sampel yang valid. Aturan ini dapat memastikan bahwa pengguna adalah pengguna yang aktif di jejaring sosial twitter.
2. Saat pengguna mengisi Kuis, sistem akan mencatat berapa waktu yang pengguna lewati saat mengisi Kuis tersebut. Jika waktu pengisian terlalu sebentar, maka pengguna adalah sampel yang tidak valid.

Didalam penelitian ini, peneliti menentukan bahwa lama waktu pengisian harus lebih dari 10 detik.

3. Saat pengguna mengisikan Kuis, pengguna yang valid adalah pengguna yang mengisi lebih dari 10 kotak sifat dan kepribadian.
4. Pengguna yang valid adalah pengguna yang tidak memiliki skor akhir kepribadian yang sama. Dikarenakan penelitian ini hanyalah mengklasifikasikan kedalam empat tipe kepribadian saja.
5. Peneliti harus mengecek apakah nama pengguna yang dimasukkan di form username benar-benar ada di twitter. Jika tidak ada, berarti pengguna yang mengisikan test tersebut adalah *invalid*.
6. Setelah mengecek username tersebut ada di twitter, peneliti harus mengecek apakah pengguna tersebut sama dengan pengguna yang membalas hasil kepribadiannya di tweet peneliti yang menyebar tautan sebelumnya. Hal ini ditujukan agar username tersebut benar-benar valid milik pengguna tersebut.

3.2.3 Mining Data

Penelitian ini menggunakan *library* Twitter Scraper untuk mengunduh semua *tweet* pengguna dari pertama kali pengguna tersebut bergabung ke jejaring sosial Twitter sampai pengambilan data dilakukan agar data yang diteliti benar-benar valid. *Library* Twitter Scraper dipilih karena kemampuannya yang dapat mengunduh seluruh *tweet* pengguna dari awal waktu. Cara-cara lainnya seperti menggunakan Twitter API tidak dipilih dikarenakan limitasi yang diberikan oleh

Twitter API, seperti hanya dapat mengunduh 3000 tweet terakhir termasuk didalamnya retweet, yang mana retweet tidak dibutuhkan didalam penelitian ini.

Tweet yang telah di scrape selanjutnya akan dimasukkan kedalam basis data MYSQL.

Tabel 3.1 Tweet yang telah di scrape

No.	Tweet	Username	Label
1	laper... ada yang mau beliin w makanan?	@psamathea	Dominance
2	Woi film yg anak sma ribut ribut kkg kena? Lah kartun kena? Mau dibawa kemana sih anak kecil skrng? Ke arah cinta cintaan?	@iLvneG	Dominance
3	Kalau renkarnasi itu ada, gw ingin tetep terlahir lagi dari rahim ibu gw. Thanks so much my spirit :* #mom	@dewinuraini13	Influence
4	Otak lg gabisa diajak kompromi -_-	@fadillahfatah7	Influence
5	mungkin kan katanya hapenya doi ilang nah hape ente direset sama yang ngambil hape vivo itu bisa jadi sih	@syihabws	Steadiness

Tabel 3.1 Tweet yang telah di *scrape* (lanjutan)

No.	Tweet	Username	Label
6	Sepi itu indah, percayalah @ Sabana Gunung Merbabu https://www.instagram.com/p/BdPFGEEng-R/	@lhamutomoaji	Steadiness
7	Ga bisa berhenti dengerin satu album dari kemaren. Krn emng se bagus itu semua lagunya, tak terkecuali. Good Job @NCTsmtown_127 #NCT127KickItpic.twitter.com/Vyj1jOJi8	@bananachacaa	Compliance
8	Semoga anakmu kelak harapannya tidak dipermainkan orang terdekatnya.	@ermiauw	Compliance

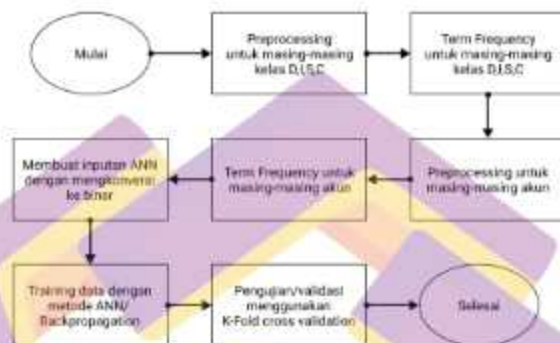
3.3 Metode Analisis Data

Data akan dianalisa melalui beberapa langkah, diantaranya preprocessing, Term Frequency, konversi ke biner untuk inputan algoritma ANN, sampai akhirnya data di validasi menggunakan K-Fold Cross Validation.

Penelitian (Hartanto et al., 2019) menggunakan proses validasi ke psikolog terlebih dahulu sebelum proses Preprocessing. Tetapi, di penelitian ini proses validasi ke psikolog sudah dilakukan di awal yaitu pada saat pembuatan soal untuk Kuis, kemudian tidak lagi di validasi ke psikolog setelah proses data mining untuk meminimalisir human error karena penelitian sebelumnya (Hartanto et al., 2019) hanya mendapatkan akurasi 36.67% saat divalidasi lagi ke psikolog. Karena soal Kuis tersebut berasal dari psikolog, kata-kata yang didapat pada saat

preprocessing sampai term frequency akan sesuai dengan akun yang mempunyai sifat berdasarkan soal Kuis.

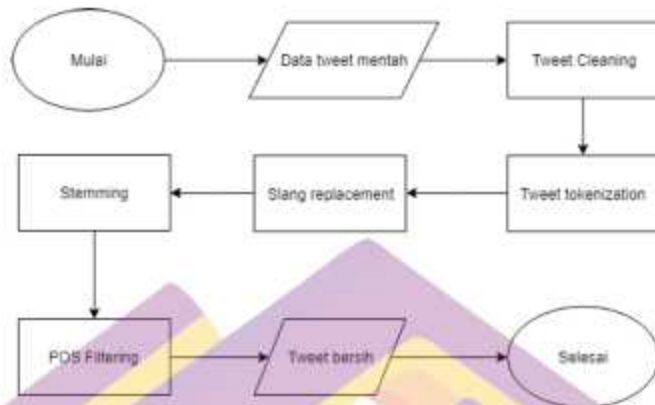
Alur analisis data akan dijelaskan pada gambar 3.4 dibawah ini.



Gambar 3.4 Alur Metode Analisis Data

3.3.1 Preprocessing

Sebelum proses klasifikasi oleh Artificial Neural Network, *tweet* harus diolah dan disiapkan terlebih dahulu agar menghasilkan data yang bersih serta layak dipakai pada saat pengolahan nanti. Pengolahan *tweet* sebelum di analisis ini disebut dengan *preprocessing*. Ada banyak proses didalam preprocessing seperti yang dijelaskan pada alur di gambar 3.4



Gambar 3.5 Alur *preprocessing*

Tweet mentah akan dibersihkan dari terlebih dahulu dari URL, *hashtag*, *username*, dll.

3.3.1.1 Tweet Cleaning

Proses pertama yang dilakukan pada *preprocessing* adalah *tweet cleaning* atau pembersihan *tweet*. Pembersihan yang dimaksud disini adalah menghilangkan data yang dianggap tidak diperlukan pada proses analisis klasifikasi kepribadian nantinya.

Adapun urutan-urutan pada proses *cleaning tweet* adalah sebagai berikut:

1. Mengubah semua huruf menjadi huruf kecil / *case folding*
2. Menghilangkan tautan atau URL.
3. Menghilangkan nama pengguna atau *username*
4. Menghilangkan *hashtag* atau tanda pagar (tagar)
5. Menghilangkan karakter unicode
6. Menghilangkan *emoticon*
7. Menghilangkan kata yang kurang dari 3 karakter

Tabel berikut berisi data *tweet* sebelum dilakukan *tweet cleaning* dan yang sudah dibersihkan atau *tweet cleaning*.

Tabel 3.2 Sebelum proses *Tweet Cleaning*

No.	Tweet	Username	Label
1	laper... ada yang mau beliin w makanan?	psamathea	Dominance
2	Woi film yg anak sma ribut ribut kkg kena? Lah kartun kena? Mau dibawa kemana sih anak kecil skrng? Ke arah cinta cintaan?	ILvnneG	Dominance
3	Kalau renkarnasi itu ada, gw ingin tetep terlahir lagi dari rahim ibu gw. Thanks so much my spirit :* #mom	dewinuraini13	Influence
4	Otak lg gabisa diajak kompromi -_-	fadillahfatah7	Influence
5	mungkin kan katanya hapenya doi ilang nah hape ente direset sama yang ngambil hape vivo itu bisa jadi sih	syihabws	Steadiness
6	Sepi itu indah, percayalah @ Sabana Gunung Merbabu https://www.instagram.com/p/BdPFGEEEng-R/	lhamutomoaji	Steadiness

Tabel 3.2 Sebelum proses *Tweet Cleaning* (lanjutan)

7	Ga bisa berhenti dengerin satu album dari kemaren. Krn emng sebgus itu semua lagunya, tak terkecuali. Good Job @NCTsmtown_127 #NCT127Kick!pic.twitter.com/Vyj1jOlji8	bananachacaa	Compliance
8	Semoga anakmu kelak harapannya tidak dipermainkan orang terdekatnya.	ermiauw	Compliance

Tabel 3.3 Setelah proses *Tweet Cleaning*

No.	Tweet	Username	Label
1	laper ada yang mau beliin w makanan	psamathea	Dominance
2	woi film yg anak sma ribut ribut kkg kena lah kartun kena mau dibawa kemana sih anak kecil skrng ke arah cinta cinta	ILvneG	Dominance
3	kalau renkarnasi itu ada, gw ingin tetep terlahir lagi dari rahim ibu gw thanks so much my spirit	dewinuraini13	Influence
4	otak lg gabisa diajak kompromi	fadillahfatah7	Influence

Tabel 3.3 Setelah proses *Tweet Cleaning* (lanjutan)

5	mungkin kan katanya hapenya doi ilang nah hape ente direset sama yang ngambil hape vivo itu bisa jadi sih	syihabws	Steadiness
6	sepi itu indah percayalah	lhamutomoaji	Steadiness
7	ga bisa berhenti dengerin satu album dari kemaren km emng sebgus itu semua lagunya tak terkecuali good job	bananachacaa	Compliance
8	semoga anakmu kelak harapannya tidak dipermainkan orang terdekatnya	ermiauw	Compliance

3.3.1.2 Tokenization

Tokenization atau tokenisasi adalah proses dimana *tweet* dipisahkan perkata secara berurutan. Variabel yang sebelumnya bertipe string diubah menjadi variable yang bertipe array dengan kata yang telah ditokenisasi didalamnya.

Tabel 3.4 Setelah proses *Tokenization*

No.	Tweet	Username	Label
1	['laper', 'ada', 'yang', 'mau', 'beliin', 'w', 'makanan']	psamathea	Dominance

Tabel 3.4 Setelah proses Tokenization (lanjutan)

2	['woi', 'film', 'yg', 'anak', 'sma', 'ribut', 'ribut', 'kkg', 'kena', 'lah', 'kartun', 'kena', 'mau', 'dibawa', 'kemana', 'sih', 'anak', 'kecil', 'skrng', 'ke', 'arah', 'cinta', 'cintaan']	ILvneG	Dominance
3	['kalau', 'renkarnasi', 'itu', 'ada', ',', 'gw', 'ingin', 'tetep', 'terlahir', 'lagi', 'dari', 'rahim', 'ibu', 'gw', 'thanks', 'so', 'much', 'my', 'spirit']	dewinuraini13	Influence
4	['otak', 'lg', 'gabisa', 'diajak', 'kompromi']	fadillahfatah7	Influence
5	['mungkin', 'kan', 'katanya', 'hapenya', 'doi', 'ilang', 'nah', 'hape', 'ente', 'direset', 'sama', 'yang', 'ngambil', 'hape', 'vivo', 'itu', 'bisa', 'jadi', 'sih']	syihabws	Steadiness
6	['sepi', 'itu', 'indah', 'percayalah']	lhamutomoaji	Steadiness
7	['ga', 'bisa', 'berhenti', 'dengerin', 'satu', 'album', 'dari', 'kemaren', 'km', 'emng', 'sebagus', 'itu', 'semua', 'lagunya', 'tak', 'terkecuali', 'good', 'job']	bananachacia	Compliance
8	['semoga', 'anakmu', 'kelak', 'harapannya', 'tidak', 'dipermainkan', 'orang', 'terdekatnya']	ermiauw	Compliance

3.3.1.3 Normalisasi Kata

Normalisasi kata adalah proses penggantian kata tidak baku menjadi kata yang baku karena tidak jarang pengguna twitter mengirimkan tweet yang berisi kalimat yang tidak baku. Karena di indonesia belum ada daftar kata tidak baku / kata gaul, maka peneliti membuat kamus kata tidak baku untuk tujuan penelitian ini.

Tabel 3.5 Setelah proses normalisasi kata

No.	Tweet	Username	Label
1	['lapar', 'ada', 'yang', 'mau', 'beliin', 'saya', 'makanan']	psamathea	Dominance
2	['woi', 'film', 'yang', 'anak', 'sma', 'ribut', 'ribut', 'tidak', 'kena', 'lah', 'kartun', 'kena', 'mau', 'dibawa', 'kemana', 'sih', 'anak', 'kecil', 'sekarang', 'ke', 'arah', 'cinta', 'cintaan']	ILvnnG	Dominance
3	['kalau', 'renkarnasi', 'itu', 'ada', 'saya', 'ingin', 'tetap', 'terlahir', 'lagi', 'dari', 'rahim', 'ibu', 'saya', 'thanks', 'so', 'much', 'my', 'spirit']	dewinuraini13	Influence
4	['otak', 'lagi', 'tidak bisa', 'diajak', 'kompromi']	fadillahfatah7	Influence

Tabel 3.5 Setelah proses normalisasi kata (lanjutan)

5	['mungkin', 'kan', 'katanya', 'ponselnya', 'dia', 'hilang', 'nah', 'ponsel', 'anda', 'direset', 'sama', 'yang', 'mengambil', 'ponsel', 'vivo', 'itu', 'bisa', 'jadi', 'sih']	syihabws	Steadiness
6	['sepi', 'itu', 'indah', 'percayalah']	lhamutomoaji	Steadiness
7	['tidak', 'bisa', 'berhenti', 'mendengarkan', 'satu', 'album', 'dari', 'kemarin', 'karena', " 'sebagus', 'itu', 'semua', 'lagunya', 'tak', 'terkecuali', 'good', 'job']	bananachacaa	Compliance
8	['semoga', 'anakmu', 'kelak', 'harapannya', 'tidak', 'dipermainkan', 'orang', 'terdekatnya']	ermiauw	Compliance

3.3.1.4 Stopword

Stopwords adalah kata umum (*common words*) yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Contoh stop words untuk bahasa Inggris diantaranya "of", "the". Sedangkan untuk bahasa Indonesia diantaranya "yang", "di", "ke".

Penelitian ini menggunakan *stopwords* bahasa Indonesia yang disediakan oleh *library* python Sastrawi, karena didalamnya terdapat *stopword* bahasa Indonesia yang cukup lengkap dan *update*.

Tabel 3.6 Setelah proses penghapusan stopword

No.	Tweet	Username	Label
1	['lapar', 'yang', 'mau', 'beliin', 'makanan']	psamathea	Dominance
2	['woi', 'film', 'anak', 'sma', 'ribut', 'ribut', 'kena', 'lah', 'kartun', 'kena', 'mau', 'dibawa', 'sih', 'anak', 'kecil', 'sekarang', 'arah', 'cinta', 'cintaan']	ILvnnG	Dominance
3	['kalau', 'renkarnasi', 'ada', 'ingin', 'tetap', 'terlahir', 'dari', 'rahim', 'ibu', 'thanks', 'so', 'much', 'my', 'spirit']	dewinuraini13	Influence
4	['otak', 'tidak', 'diajak', 'kompromi']	fadillahfatah7	Influence
5	['mungkin', 'kan', 'katanya', 'ponselnya', 'hilang', 'nah', 'ponsel', 'direset', 'sama', 'mengambil', 'ponsel', 'vivo', 'bisa', 'jadi', 'sih']	syihabws	Steadiness
6	['sepi', 'indah', 'percayalah']	lhamutomoaji	Steadiness
7	['bisa', 'berhenti', 'mendengarkan', 'satu', 'album', 'kemarin', 'sebagus', 'semua', 'lagunya', 'tak', 'terkecuali', 'good', 'job']	bananachacaa	Compliance
8	['semoga', 'anakmu', 'kelak', 'harapannya', 'dipermainkan', 'orang', 'terdekatnya']	ermiauw	Compliance

3.3.1.5 Stemming

Proses *stemming* merupakan proses guna menghilangkan kata imbuhan sehingga hanya tersisa kata dasarnya saja dari suatu kata. Pada bahasa Indonesia, imbuhan yang umum digunakan yaitu imbuhan mem-, meny-, ber- dan lain-lain serta akhiran -an, -kan, -i, dan lain-lain.

Di penelitian ini penulis menggunakan library Sastrawi python untuk melakukan proses *stemming*. Hasil dari proses *stemming* dapat dilihat pada tabel 2.1.

Tabel 3.7 Setelah proses stemming

No.	Tweet	Username	Label
1	['lapar', 'yang', 'mau', 'beli', 'makan']	psamathea	Dominance
2	['woi', 'film', 'anak', 'sma', 'ribut', 'ribut', 'kena', 'lah', 'kartun', 'kena', 'mau', 'bawa', 'sih', 'anak', 'kecil', 'sekarang', 'arah', 'cinta', 'cinta']	ILynneG	Dominance
3	['kalau', 'renkarnasi', 'ada', 'ingin', 'tetap', 'lahir', 'dari', 'rahim', 'ibu', 'thanks', 'so', 'much', 'my', 'spirit']	dewinuraini13	Influence
4	['otak', 'tidak', 'ajak', 'kompromi']	fadillahfatah7	Influence
5	['mungkin', 'kan', 'kata', 'ponsel', 'hilang', 'nah', 'ponsel', 'reset', 'sama', 'ambil', 'ponsel']	syihabws	Steadiness

Tabel 3.7 Setelah proses stemming (lanjutan)

6	['sepi', 'indah', 'percaya']	lhamutomoaji	Steadiness
7	['bisa', 'henti', 'dengar', 'satu', 'album', 'kemarin', 'bagus', 'semua', 'lagu', 'tak', 'kecuali', 'good', 'job']	bananachacaa	Compliance
8	['moga', 'anak', 'kelak', 'harap', 'main', 'orang', 'dekat']	ermiauw	Compliance

3.3.1.8 POS Tagging

Proses selanjutnya adalah memberikan tag pada setiap kata yang telah dihilangkan imbuhan, proses ini dinamakan *POS Tagging*. Dalam tahap ini, peneliti menggunakan *library* dari Stanford Natural Language Processing yaitu Stanford POS Tagger.

Tabel 3.8 Tagset

Tag	Deskripsi	Contoh
CC	Coordinating conjunction	Dan, tetapi, atau
CD	Cardinal number	Dua, juta, enam, 213, sepertiga
OD	Ordinal number	Ketiga, ke-1, pertama
DT	Determiner / article	Para, Sang, Si
FW	Foreign word	Climage change, monkey
IN	Preposition	Dalam, dengan, di, ke

Tabel 3.8 Tagset (lanjutan)

JJ	Adjective	Bersih, panjang, hitam, lama
MD	Modal and auxiliary verb	Boleh, harus, sudah, mesti, perlu
NEG	Negation	Tidak, belum, jangan
NN	Noun	Bahwa, sekarang
NNP	Proper Noun	Jokowi, Piala Dunia, Idul Fitri
NND	Classifier, partitive, and measurement noun	Orang, ton, helai, lembar
PR	Demonstrative pronoun	Ini, itu, sini, situ
PRP	Personal pronoun	Saya, kami, kita, kamu, kalian, dia
RB	Adverb	Sangat, hanya, justru, niscaya
RP	Particle	Pun, -lah, -kah
SC	Subordinating conjunction	Sejak, jika, seandainya
SYM	Symbol	IDR, +, %
UH	Interjection	Brengsek, oh, ooh, aduh, ayo
VB	Verb	Merancang, mengatur, pergi
WH	Question	Siapa, dimana, bagaimana, kapan
X	Unknown	Statemen
Z	Punctuation	"....", ?, ..

Akan tetapi, karena belum adanya model untuk bahasa Indonesia di Stanford POS Tagger, peneliti menggunakan korpus dan konfigurasi pelatihan pada penelitian (Dinakaramani et al., 2014) untuk membangun model *tagger* Bahasa Indonesia yang akan digunakan di dalam tahap ini.

Tabel 3.9 Setelah proses POS tagging

No.	Tweet	Username	Label
1	[(‘lapar’, ‘VB’), (‘yang’, ‘SC’), (‘mau’, ‘RB’), (‘beli’, ‘VB’), (‘makan’, ‘VB’)]	psamathea	Dominance
2	[(‘woi’, ‘NN’), (‘film’, ‘NN’), (‘anak’, ‘NN’), (‘sma’, ‘NN’), (‘ribut’, ‘PR’), (‘ribut’, ‘VB’), (‘kena’, ‘SC’), (‘lah’, ‘RP’), (‘kartun’, ‘VB’), (‘kena’, ‘SC’), (‘mau’, ‘RB’), (‘bawa’, ‘VB’), (‘sih’, ‘NN’), (‘anak’, ‘NN’), (‘kecil’, ‘JJ’), (‘sekarang’, ‘NN’), (‘arah’, ‘NN’), (‘cinta’, ‘VB’), (‘cinta’, ‘VB’)]	ILvnnG	Dominance
3	[(‘kalau’, ‘SC’), (‘renkarnasi’, ‘NN’), (‘ada’, ‘VB’), (‘ingin’, ‘RB’), (‘tetap’, ‘RB’), (‘lahir’, ‘VB’), (‘dari’, ‘IN’), (‘rahim’, ‘NN’), (‘ibu’, ‘NN’), (‘thanks’, ‘FW’), (‘so’, ‘FW’), (‘much’, ‘FW’), (‘my’, ‘FW’), (‘spirit’, ‘FW’)]	dewinuraini13	Influence

Tabel 3.9 Setelah proses POS tagging (lanjutan)

4	[(‘otak’, ‘NN’), (‘tidak’, ‘NEG’), (‘ajak’, ‘VB’), (‘kompromi’, ‘NN’)]	fadillahfatah7	Influence
5	[(‘mungkin’, ‘MD’), (‘kan’, ‘MD’), (‘kata’, ‘VB’), (‘ponsel’, ‘NN’), (‘hilang’, ‘VB’), (‘nah’, ‘NN’), (‘ponsel’, ‘NN’), (‘reset’, ‘NN’), (‘sama’, ‘JJ’), (‘ambil’, ‘VB’), (‘ponsel’, ‘NN’), (‘vivo’, ‘NN’), (‘bisa’, ‘MD’), (‘jadi’, ‘VB’), (‘sih’, ‘VB’)]	syihabws	Steadiness
6	[(‘sepi’, ‘NN’), (‘indah’, ‘MD’), (‘percaya’, ‘VB’)]	lhamutomoaji	Steadiness
7	[(‘bisa’, ‘MD’), (‘henti’, ‘VB’), (‘dengar’, ‘VB’), (‘satu’, ‘CD’), (‘album’, ‘NN’), (‘kemarin’, ‘NN’), (‘bagus’, ‘JJ’), (‘semua’, ‘CD’), (‘lagu’, ‘NN’), (‘tak’, ‘NEG’), (‘kecuali’, ‘VB’), (‘good’, ‘FW’), (‘job’, ‘FW’)]	bananachacaa	Compliance
8	[(‘moga’, ‘SC’), (‘anak’, ‘NN’), (‘kelak’, ‘NN’), (‘harap’, ‘VB’), (‘main’, ‘IN’), (‘orang’, ‘NN’), (‘dekat’, ‘JJ’)]	ermiauw	Compliance

3.3.1.6 POS Filtering

Setelah proses *POS tagging*, tahan selanjutnya adalah memilih kata mana saja yang dapat menjadi data latih. Dalam penelitian ini, peneliti merujuk ke

penelitian (B. Liu, 2010) bahwa ada 4 tag yang memiliki aspek sentimen yang tinggi, yaitu kata kerja atau VB, kata sifat atau JJ, kata benda atau NN, dan kata keterangan atau RB. Selain ke empat tag tersebut akan tidak dimasukkan ke dalam array pada tweet.

Tabel 3.10 Setelah proses POS filtering

No.	Tweet	Username	Label
1	[('lapar', 'VB'), ('mau', 'RB'), ('beli', 'VB'), ('makan', 'VB')]	psamathea	Dominance
2	[('woi', 'NN'), ('film', 'NN'), ('anak', 'NN'), ('sma', 'NN'), ('ribut', 'VB'), ('kartun', 'VB'), ('mau', 'RB'), ('bawa', 'VB'), ('sih', 'NN'), ('anak', 'NN'), ('kecil', 'JJ'), ('sekarang', 'NN'), ('arah', 'NN'), ('cinta', 'VB'), ('cinta', 'VB')]	ILvnnG	Dominance
3	[('renkarnasi', 'NN'), ('ada', 'VB'), ('ingin', 'RB'), ('tetap', 'RB'), ('lahir', 'VB'), ('rahim', 'NN'), ('ibu', 'NN')]	dewinuraini13	Influence
4	[('otak', 'NN'), ('ajak', 'VB'), ('kompromi',	fadillahfatah7	Influence

Tabel 3.10 Setelah proses POS filtering (lanjutan)

5	[(kata', 'VB'), ('ponsel', 'NN'), ('hilang', 'VB'), ('nah', 'NN'), ('ponsel', 'NN'), ('reset', 'NN'), ('sama', 'JJ'), ('ambil', 'VB'), ('ponsel', 'NN'), ('vivo', 'NN'), ('jadi', 'VB'), ('sih', 'VB')]	syihabws	Steadiness
6	[(sepi', 'NN'), ('percaya', 'VB')]	lhamutomoaji	Steadiness
7	[(henti', 'VB'), ('dengar', 'VB'), ('album', 'NN'), ('kemarin', 'NN'), ('bagus', 'JJ'), ('lagu', 'NN'), ('kecuali', 'VB')]	bananachacaa	Compliance
8	[(anak', 'NN'), ('kelak', 'NN'), ('harap', 'VB'), ('orang', 'NN'), ('dekat', 'JJ')]	ermianuw	Compliance

3.3.1.7 Tweet Mergling

Setelah tahap *POS filtering*, tahap selanjutnya adalah proses *tweet mergling* atau penggabungan *tweet*. Karena proses klasifikasi pada penelitian ini adalah klasifikasi per akun, maka seluruh tweet dalam akun yang sama akan dijadikan satu buah array.

Tabel 3.11 Setelah proses tweet merging

No.	Tweet	Label
1	['lapar', 'mau', 'beli', 'makan', 'woi', 'film', 'anak', 'sma', 'ribut', 'kartun', 'mau', 'bawa', 'sih', 'anak', 'kecil', 'sekarang', 'arah', 'cinta', 'cinta']	Dominance
2	['renkarnasi', 'ada', 'ingin', 'tetap', 'lahir', 'rahim', 'ibu', 'otak', 'ajak', 'kompromi']	Influence
3	['kata', 'ponsel', 'hilang', 'nah', 'ponsel', 'reset', 'sama', 'ambil', 'ponsel', 'vivo', 'jadi', 'sih', 'sepi', 'percaya']	Steadiness
4	['henti', 'dengar', 'album', 'kemarin', 'bagus', 'lagu', 'kecuali', 'anak', 'kelak', 'harap', 'orang', 'dekat']	Compliance

3.3.2 Pelatihan Data

3.3.2.1 Term Frequency

Term Frequency adalah proses penghitungan kata berdasarkan seberapa sering kata tersebut muncul dalam suatu dokumen. Rumus perhitungan dari *term frequency* adalah:

$$\text{Term Frequency (z)} = \frac{\text{Angka kemunculan kata z didalam dokumen}}{\text{Total seluruh kata didalam dokumen}}$$

Dalam penelitian ini, seluruh akun akan digabungkan menjadi satu dokumen guna mencari kata yang sering muncul pada suatu kepribadian. Setelah

kata yang paling sering muncul, dipilih beberapa kata yang paling sering muncul. Dalam contoh ini, peneliti akan memilih tiga kata teratas yang frekuensi kemunculannya paling sering.

Tabel 3.12 Proses term frequency untuk dominance

Term	Rumus	Hasil
lapar	$\frac{1}{19}$	0,052631578947368
mau	$\frac{2}{19}$	0,105263157894737
beli	$\frac{1}{19}$	0,052631578947368
makan	$\frac{1}{19}$	0,052631578947368
woi	$\frac{1}{19}$	0,052631578947368
Film	$\frac{1}{19}$	0,1428571428571429
Anak	$\frac{2}{19}$	0,105263157894737
sma	$\frac{1}{19}$	0,052631578947368
Rebut	$\frac{1}{19}$	0,052631578947368

Tabel 3.12 Proses term frequency untuk dominance (lanjutan)

kartun	$\frac{1}{19}$	0,052631578947368
Bawa	$\frac{1}{19}$	0,052631578947368
Sih	$\frac{1}{19}$	0,052631578947368
kecil	$\frac{1}{19}$	0,052631578947368
Sekarang	$\frac{1}{19}$	0,052631578947368
arah	$\frac{1}{19}$	0,052631578947368
cinta	$\frac{2}{19}$	0,105263157894737

Tabel 3.13 Proses term frequency untuk influence

Term	Rumus	Hasil
reinkarnasi	$\frac{1}{10}$	0,1
ada	$\frac{1}{10}$	0,1
ingin	$\frac{1}{10}$	0,1
tetap	$\frac{1}{10}$	0,1

Tabel 3.13 Proses term frequency untuk influence (lanjutan)

lahir	$\frac{1}{10}$	0,1
rahim	$\frac{1}{10}$	0,1
Ibu	$\frac{1}{10}$	0,1
otak	$\frac{1}{10}$	0,1
ajak	$\frac{1}{10}$	0,1
kompromi	$\frac{1}{10}$	0,1

Tabel 3.14 Proses term frequency untuk steadness

Term	Rumus	Hasil
kata	$\frac{1}{14}$	0,071428571428571
ponsel	$\frac{3}{14}$	0,214285714285714
hilang	$\frac{1}{14}$	0,071428571428571
nah	$\frac{1}{14}$	0,071428571428571
reset	$\frac{1}{14}$	0,071428571428571

Tabel 3.14 Proses term frequency untuk steadiness (lanjutan)

sama	$\frac{1}{14}$	0,071428571428571
ambil	$\frac{1}{14}$	0,071428571428571
vivo	$\frac{1}{14}$	0,071428571428571
jadi	$\frac{1}{14}$	0,071428571428571
sih	$\frac{1}{14}$	0,071428571428571
sepi	$\frac{1}{14}$	0,071428571428571
percaya	$\frac{1}{14}$	0,071428571428571

Tabel 3.15 Proses term frequency untuk compliance

['henti', 'dengar', 'album', 'kemarin', 'bagus', 'lagu', 'kecuali', 'anak', 'kelak',
'harap', 'orang', 'dekat']

Term	Rumus	Hasil
henti	$\frac{1}{12}$	0,0833333333333333
dengar	$\frac{1}{12}$	0,0833333333333333
album	$\frac{1}{12}$	0,0833333333333333

Tabel 3.15 Proses term frequency untuk compliance (lanjutan)

kemarin	$\frac{1}{12}$	0,0833333333333333
bagus	$\frac{1}{12}$	0,0833333333333333
lagu	$\frac{1}{12}$	0,0833333333333333
kecuali	$\frac{1}{12}$	0,0833333333333333
anak	$\frac{1}{12}$	0,0833333333333333
kelak	$\frac{1}{12}$	0,0833333333333333
harap	$\frac{1}{12}$	0,0833333333333333
orang	$\frac{1}{12}$	0,0833333333333333
dekat	$\frac{1}{12}$	0,0833333333333333

Sehingga didapat tiga kata yang paling sering muncul adalah sebagai berikut:

Tabel 3.16 Tweet yang di scrape

Kata	Sifat
Mau	Dominance
Anak	Dominance

Tabel 3.16 Tweet yang di scrape (lanjutan)

Reinkarnasi	Influence
Ada	Influence
Ingin	Influence
Ponsel	Steadiness
Kata	Steadiness
Hilang	Steadiness
Henti	Compliance
Dengar	Compliance
Album	Compliance

3.3.2.2 Konversi ke Biner

Setelah melakukan *term frequency*, langkah selanjutnya adalah mempersiapkan data yang telah diolah menjadi data yang siap diinput kedalam layer input pada algoritma *Artificial Neural Network*. Inputan algoritma *Artificial Neural Network* ini berbentuk array dari layer-layer yang berisi angka, sehingga kita akan mengubah kata-kata yang ada pada *term frequency* ke angka biner (0 dan 1) dengan cara mencocokkan kata per kata dari suatu akun kedalam term frequency. Jika suatu kata cocok/sama dengan salah satu kata yang ada didalam term frequency, maka nilai kata tersebut adalah 1, akan tetapi jika suatu kata tidak ada yang cocok didalam term frequency, maka nilai kata tersebut adalah 0. Perubahan kedalam bentuk biner ini dilakukan agar bisa diproses oleh algoritma

Artificial Neural Network, karna algoritma ini hanya bisa memproses inputan dalam bentuk angka saja. Langkah-langkahnya adalah sebagai berikut:

1. Kumpulkan semua *term frequency* kepribadian yang telah dibuat sebelumnya dan jadikan satu array. Tabel berikut ini adalah hasil dari seluruh *term frequency* sebelumnya yang telah digabungkan menjadi satu array.

Tabel 3.17 Hasil term frequency

Mau
Anak
Cinta
Reinkarnasi
Ada
Ingin
Ponsel
Kata
Hilang
Henti
Dengar
Album

2. Pilih satu akun yang ingin dijadikan *data training* dan *data testing* dan pilih seluruh tweet akun tersebut pada *database* yang telah *disrape* sebelumnya.

3. Lakukan proses *preprocessing* dan *tweet merging* pada seluruh tweet dari akun tersebut agar seluruh tweet dalam satu akun tersebut digabungkan.
4. Setelah itu, lakukan *term frequency* untuk menentukan beberapa kata yang muncul paling atas pada akun tersebut, dalam contoh ini peneliti akan mengambil tiga kata paling atas sesuai dengan langkah pada *term frequency* sebelumnya pada kepribadian.

Tabel 3.18 Tiga kata teratas

mau
anak
cinta

5. Buat array berisi angka nol (0) berjumlah 12. Angka 12 didapatkan dari $3 * 4$, yang mana tiga adalah jumlah kata teratas yang muncul dan empat adalah jumlah seluruh kepribadian. Hasil dari array tersebut adalah seperti ini:

[0,0,0,0,0,0,0,0,0,0,0,0]

6. Cocokkan setiap kata yang ada pada array akun dan kata pada array kepribadian. Jika ada kata yang cocok, ubahlah array berisi angka nol yang telah dibuat tadi menjadi angka 1 sesuai urutan array pada *term frequency* kepribadian. Sehingga, hasilnya akan menjadi seperti ini:

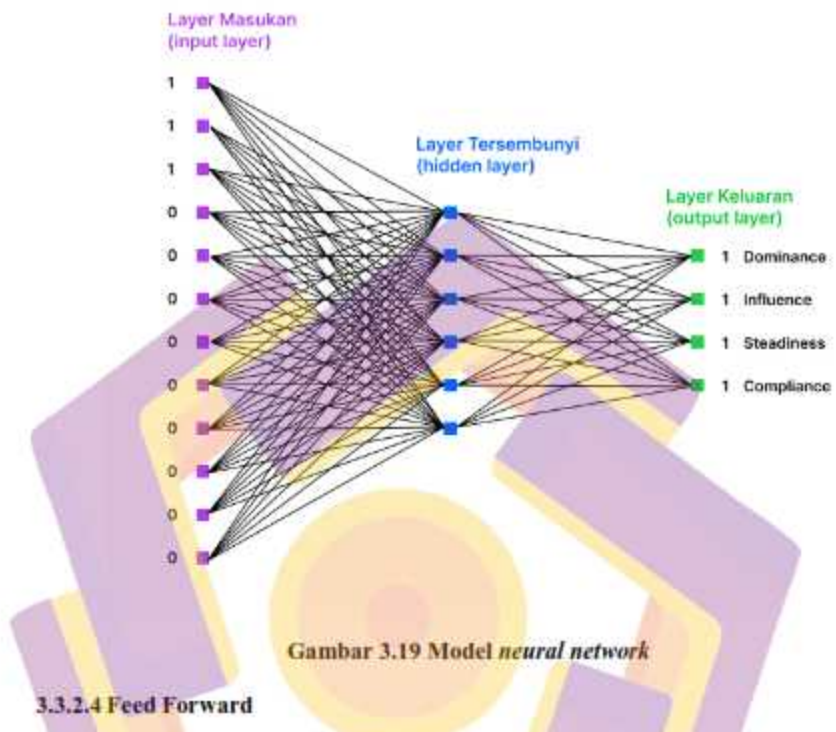
[1,1,1,0,0,0,0,0,0,0,0,0]

Array berisi angka 0 dan 1 itulah nanti yang akan digunakan sebagai inputan pada layer input *Artificial Neural Network*.

3.3.2.3 Pembuatan Model Neural Network

Setelah tahap pengkonversian akun twitter ke dalam input *Artificial Neural Network*, tahap selanjutnya yang dilakukan adalah menentukan jumlah layer input, layer tersembunyi, dan layer keluaran pada arsitektur jaringan syaraf buatan nanti. Pada kasus ini, layer input berjumlah 12 *node*. 12 *node* didapat dari jumlah *term frequency* yang diambil dikalikan jumlah total tipe kepribadian, atau tiga dikalikan 4. Pada *hidden layer* peneliti menentukan bahwa pada contoh ini jumlah *hidden layer* adalah 6 *node*, jumlah ini bisa berubah-ubah menyesuaikan hasil pengujian paling baik yang didapat. Sedangkan pada *output layer* berjumlah 4 *node* karena jumlah kepribadian ada 4 tipe. Sehingga modelnya akan menjadi seperti pada gambar 3.3





3.3.2.4 Feed Forward

Feed Forward atau biasa juga disebut *forward propagation* adalah proses dimana kita membawa data pada input melewati tiap *neuron* pada *hidden layer* sampai kepada *output layer* yang nanti akan dihitung *error*nya. Caranya adalah dengan cara menjumlahkan keseluruhan layer input dikalikan bobot yang dimilikinya. Rumus lengkapnya adalah sebagai berikut:

$$dot_j = \sum_i^{12} w_{ij}x_i + b_j$$

$$h_j = \sigma(dot_j) = \max(0, dot_j)$$

Dimana j adalah *node* pada *hidden layer*, i adalah *node* pada *input layer*, 12 adalah jumlah *node input layer*, w adalah *weight* atau bobot pada *node input layer*, x adalah nilai inputan pada *node input layer*, b adalah bias pada *hidden layer*, dan h adalah output dari *node* pada *hidden layer*. Persamaan diatas menggunakan ReLU sebagai fungsi aktivasi:

Dalam contoh yang diberikan pada bagian pengkonversian ke biner, untuk menghitung nilai keluaran pada *hidden layer* menggunakan fungsi aktivasi ReLU adalah sebagai berikut:

Tabel 3.20 Perhitungan layer input ke layer hidden

Hidden	Rumus	Hasil Rumus	ReLU
h1	$(0*1)+(0*1)+(0*1)+(1*1)+$ $(1*1)+(1*1)+(0*1)+(0*1)+$ $(0*1)+(0*1)+(0*1)+(0*1)+1$	4	4
h2	$(0*1)+(0*1)+(0*1)+(1*1)+$ $(1*1)+(1*1)+(0*1)+(0*1)+$ $(0*1)+(0*1)+(0*1)+(0*1)+1$	4	4
h3	$(0*1)+(0*1)+(0*1)+(1*1)+$ $(1*1)+(1*1)+(0*1)+(0*1)+$ $(0*1)+(0*1)+(0*1)+(0*1)+1$	4	4

Tabel 3.20 Perhitungan layer input ke layer hidden (lanjutan)

h4	$(0*1)+(0*1)+(0*1)+(1*1)+$ $(1*1)+(1*1)+(0*1)+(0*1)+$ $(0*1)+(0*1)+(0*1)+(0*1)+1$	4	4
h5	$(0*1)+(0*1)+(0*1)+(1*1)+$ $(1*1)+(1*1)+(0*1)+(0*1)+$ $(0*1)+(0*1)+(0*1)+(0*1)+1$	4	4
h6	$(0*1)+(0*1)+(0*1)+(1*1)+$ $(1*1)+(1*1)+(0*1)+(0*1)+$ $(0*1)+(0*1)+(0*1)+(0*1)+1$	4	4

Sedangkan, untuk menghitung nilai keluaran pada *output layer*, menggunakan inputan dari nilai *node hidden layer* dikalikan bobot *node hidden layer* adalah seperti yang ditunjukkan tabel 3.19 dimana bobot hidden node kita beri nilai 1. Pada awal pelatihan, bobot suatu node ditentukan secara acak karena yang ingin kita cari/hitung agar pengklasifikasian berhasil adalah bobot tersebut dengan metode *Backpropagation*.

Tabel 3.21 Perhitungan layer hidden ke layer output

Output	Rumus	Hasil Rumus	Sigmoid
o1	$(4*1)+(4*1)+(4*1)+$ $(4*1)+1$	25	0.9999999999986112

Tabel 3.21 Perhitungan layer hidden ke layer output (lanjutan)

o2	$(4*1)+(4*1)+(4*1)+$ $(4*1)+1$	25	0.999999999986112
o3	$(4*1)+(4*1)+(4*1)+$ $(4*1)+1$	25	0.999999999986112
o4	$(4*1)+(4*1)+(4*1)+$ $(4*1)+1$	25	0.999999999986112

3.3.2.5 Back Propagation

Tentunya, nilai keluaran dari output layer tidak sesuai dengan apa yang kita harapkan yaitu $[0,1,0,0]$. Ini dikarenakan bobot yang kita beri diawal tadi adalah ditentukan secara acak. Untuk mendapatkan bobot yang sesuai, kita harus menyesuaikan nilai keluaran yang dihitung dengan nilai keluaran yang diinginkan. Pertama, kita harus menghitung nilai error dari output layer. Nilai error adalah nilai yang telah dihitung dikurangi nilai yang diinginkan:

$$\delta_k = (t_k - Y_k) g'(y_k)$$

Setelah menghitung error, kita harus menyesuaikan bobot pada hidden layer dan input layer berdasarkan error yang tersebut:

$$w_{jk} \leftarrow w_{jk} - \alpha \delta_k Z_j$$

Setelah bobot pada hidden layer dan juga input layer disesuaikan, ulangi proses *feed forward* dan *back propagation* berulang-ulang sehingga kita menemukan error yang paling minimal.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

4.1 Analisis Sistem

4.1.1 Analisis Kebutuhan Fungsional

Untuk melaksanakan penelitian ini, dibutuhkan sebuah sistem yang harus terpenuhi, yaitu:

1. Sistem mampu menampilkan antarmuka kedalam web browser
2. Sistem mampu mengunduh/*scrape* semua *tweet* user dari awal sampai akhir
3. Sistem mampu melakukan serangkaian tahap *preprocessing*, meliputi:
 - a. Pembersihan *tweet* dari *username*, url, *case folding*, *hashtag*, dan *unicode*.
 - b. Tokenisasi
 - c. Normalisasi Kata
 - d. Penghapusan *stopword*
 - e. *Stemming*
 - f. *POS Tagging*
 - g. *POS Filtering*
4. Sistem mampu melakukan perhitungan *term frequency*
5. Sistem mampu mengklasifikasikan kepribadian akun twitter kedalam empat kategori DISC, yaitu *Dominance*, *Influence*, *Steadiness*, dan *Compliance*.

Sistem mampu menampilkan hasil klasifikasi suatu akun beserta keterangan tentang hasil tersebut.

4.1.2 Analisis Kebutuhan Non Fungsional

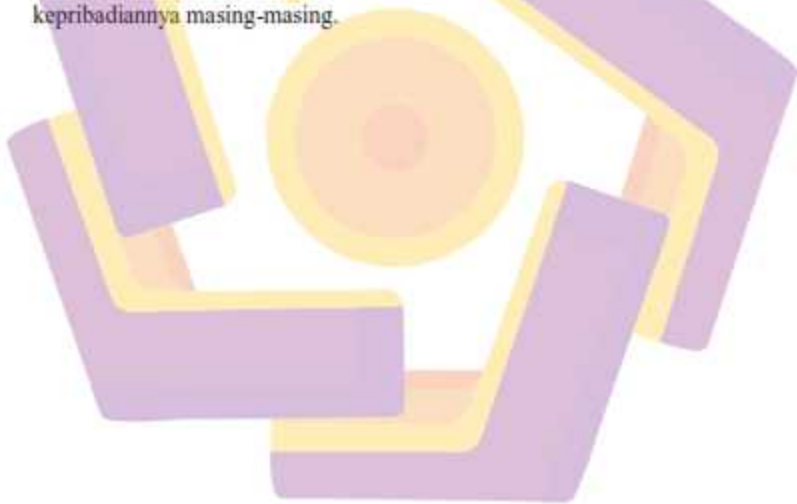
Kebutuhan fungsional adalah kebutuhan yang diperlukan agar fungsi pada sistem dapat tercapai. Kebutuhan tersebut meliputi kebutuhan perangkat keras dan perangkat lunak yang di digunakan pada saat membangun dan menjalankan sistem tersebut.

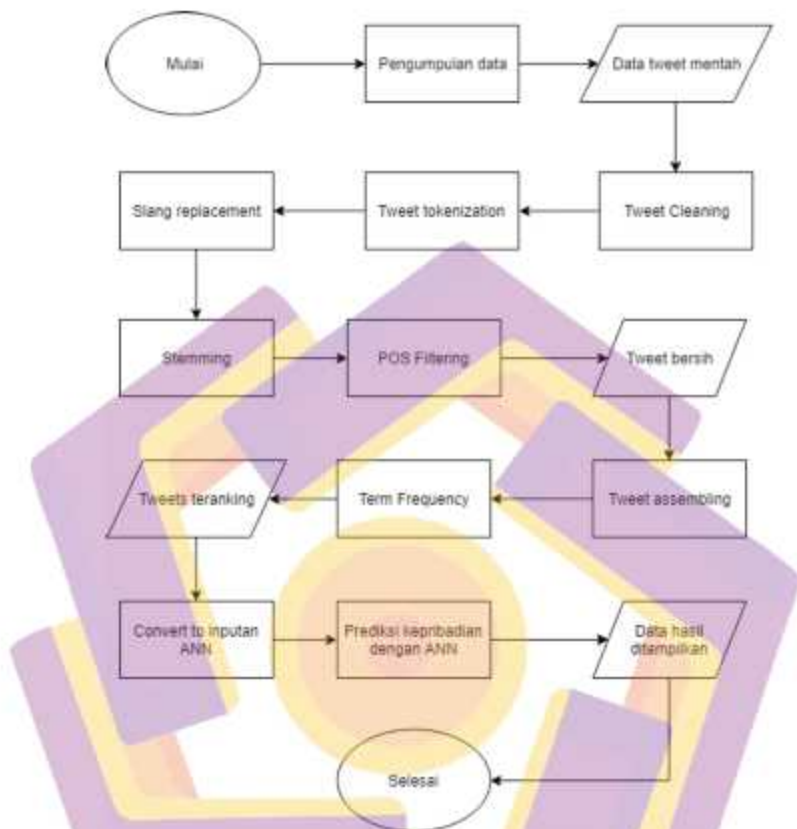
1. Perangkat keras yang peneliti gunakan:
 - a. RAM sebesar 8192 MB
 - b. Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz (4 CPUs), ~2.4GHz
 - c. Intel(R) HD Graphics 520
 - d. NVIDIA GeForce 930M
2. Perangkat lunak yang peneliti gunakan:
 - a. Python 3.6.8 (flask, tensorflow, numpy, nltk, sastrawi, twitterscraper, sklearn)
 - b. Windows 10 Home Single Language 64-bit (10.0, Build 17134) (17134.rs4_release.180410-1804)
 - c. Firefox Web Browser
 - d. Mysql
 - e. Sublime Text
 - f. JQuery
 - g. HTML
 - h. CSS

Windows Powershell

4.1.3 Gambaran Umum Sistem

Sistem yang akan dibangun didalam penelitian ini adalah sistem yang mampu untuk mengunduh semua *tweet* pengguna dari awal sampai akhir, mengubahnya ke dalam satu dokumen, kemudian sistem mampu menganalisis jenis kepribadian dari pengguna menurut kepribadian DISC. Didalam penelitian ini, pengumpulan data dilakukan guna menjadikannya sebagai data latih dan data uji. Data yang dikumpulkan merupakan data yang berasal dari Kuis yang telah di publikasikan di media sosial twitter sehingga datanya sudah berlabelkan jenis kepribadiannya masing-masing.





Gambar 4.1 Alur sistem

Adapun penjelasan terkait gambar 4.1 diatas adalah:

1. Pengumpulan dan pengambilan data dilakukan dengan menggunakan *library* twitterscraper
2. Data melalui proses *preprocessing* dimana *tweet* akan dibersihkan dan digabungkan antara *tweet* satu dengan *tweet* yang lain ke dalam satu array

3. Setelah digabungkan, pada array tersebut akan dilakukan *term frequency* untuk meranking tiga ratus kata yang paling banyak muncul didalam dokumen
4. Tiga ratus kata yang muncul tersebut kemudian dikonversi menjadi inputan ANN dengan mencocokkan 200 kata yang muncul dengan 800 kata yang ada didalam *term frequency* yang sudah dilakukan pada proses *training*. Pada akhirnya, 200 kata tersebut akan diubah menjadi *layer input* dengan 800 inputan.
5. *Artificial Neural Network* akan memprediksi kepribadian berdasarkan inputan yang telah dimasukkan.
6. Hasil prediksi akan ditampilkan ke dalam *web browser*.

4.2 Analisis Kebutuhan Data

4.2.1 Perancangan Basis Data

Database yang digunakan pada penelitian ini adalah MySQL. Database digunakan untuk menyimpan hasil pengumpulan data dan penyaringan pengumpulan data. Berikut merupakan tabel-tabel yang akan dibuat.

Tabel 3.20 Tabel hasil_kuis

No	Nama Kolom	Tipe Data	Keterangan
1	id	int(255)	Primary Key
2	nama	varchar(255)	
3	usia	varchar(25)	

Tabel 3.20 Tabel hasil_kuis (lanjutan)

4	jenis_kelamin	varchar(1)	
5	tanggal_tes	date	
6	twitter_username	varchar(255)	
7	hasil	varchar(1)	
8	hasil_kedua	varchar(1)	
9	most_d	varchar(255)	
10	most_i	varchar(255)	
11	most_s	varchar(255)	
12	most_c	varchar(255)	
13	most_x	varchar(255)	
14	less_d	varchar(255)	
15	less_i	varchar(255)	
16	less_s	varchar(255)	
17	less_c	varchar(255)	
18	less_x	varchar(255)	
19	timer	varchar(255)	
20	number_of_tweets	varchar(255)	

Tabel 3.22 Tabel tweets

No	Nama Kolom	Tipe Data	Keterangan
1	id	int(255)	Primary Key
2	text	text	
3	username	varchar(255)	
4	username_db_id	varchar(255)	
5	verified	varchar(255)	
6	hasil	varchar(1)	
7	Konversi_hasil_ke_angka	varchar(1)	

4.2.2 Perancangan Dataset

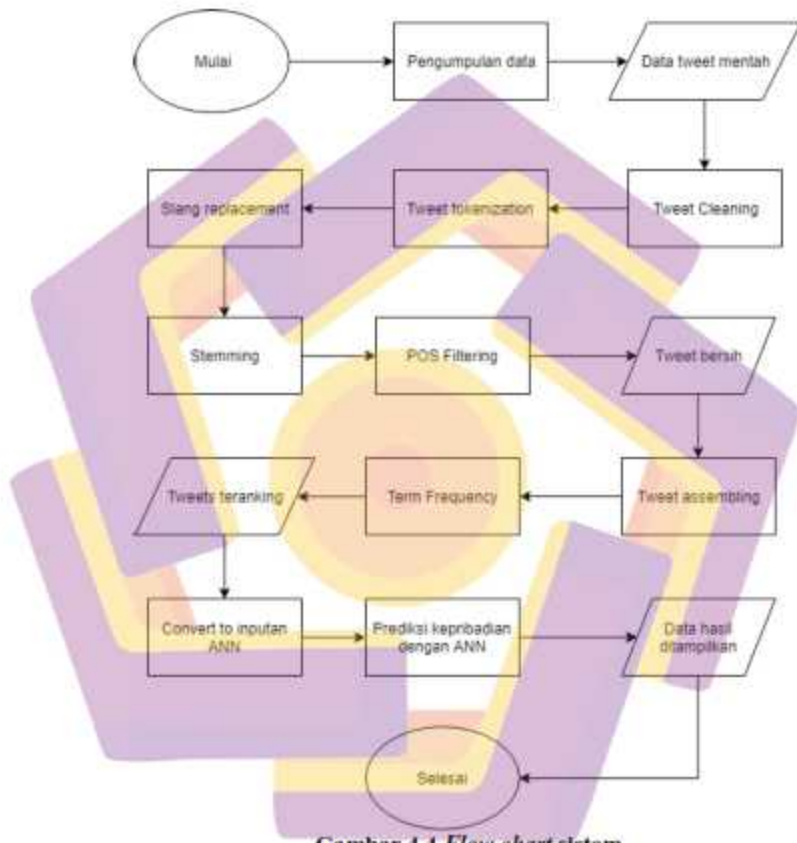
Perancangan dataset dilakukan setelah melalui proses *preprocessing*, *term frequency*, dan juga konversi ke biner. Perancangan dataset terbagi menjadi perancangan data training, label data training, data testing, dan label data testing. Berkas *data training*, label *data training*, *data testing*, dan label *data testing* akan disimpan kedalam file txt.

Berikut merupakan contoh dataset yang sudah disiapkan.

4.3 Perancangan Sistem

4.3.1 Flowchart Sistem

Flow chart pada sistem yang akan dirancang adalah sebagai berikut.

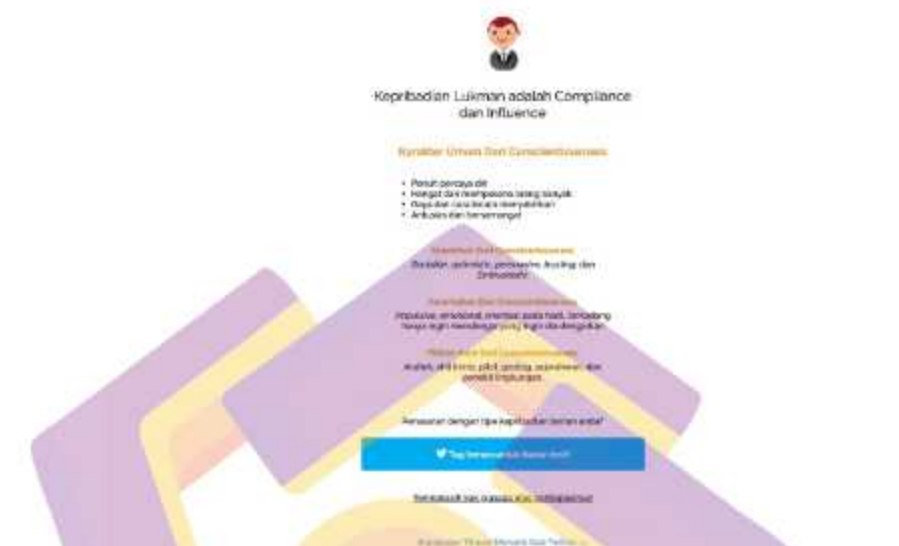


Gambar 4.4 Flow chart sistem

4.4 Implementasi

4.4.1 Deskripsi Implementasi

Penelitian ini menggunakan algoritma *Artificial Neural Network* untuk mengklasifikasikan suatu akun kedalam empat tipe kepribadian berdasarkan



Gambar 4.6 Antarmuka halaman hasil kuis

Data yang telah disubmit oleh pengguna kemudian disimpan kedalam basis data yang telah dirancang pada bab sebelumnya di tabel hasil_kuis. Berikut adalah tangkapan layar dari tabel hasil kuis.

id	nama	usia	jenis kelamin	tanggal_bes	total jawaban	benar	salah	nilai_a	nilai_b	nilai_c	nilai_d	nilai_e
1	Adi	19	P	2019-12-12	10	5	5	2	2	4	0	0
2	Budi	20	L	2019-12-12	10	7	3	3	3	3	1	0
3	Cici	21	L	2019-12-12	10	8	2	4	2	0	0	0
4	Dede	22	P	2019-12-12	10	9	1	3	3	3	0	0
5	Evi	23	L	2019-12-12	10	6	4	3	0	0	0	0
6	Fani	24	P	2019-12-12	10	7	3	1	7	0	0	0
7	Gani	25	P	2019-12-12	10	8	2	2	7	0	0	0
8	Hani	26	P	2019-12-12	10	9	1	0	3	6	0	0
9	Iani	27	P	2019-12-12	10	8	2	3	6	0	0	0
10	Jani	28	P	2019-12-12	10	7	3	1	6	0	0	0
11	Kani	29	L	2019-12-12	10	6	4	0	7	0	0	0
12	Lani	30	L	2019-12-12	10	7	3	1	3	3	0	0
13	Mani	31	L	2019-12-12	10	8	2	4	2	3	0	0
14	Nani	32	L	2019-12-12	10	9	1	1	3	1	0	0
15	Oani	33	L	2019-12-12	10	8	2	1	7	0	0	0
16	Pani	34	L	2019-12-12	10	7	3	1	4	0	0	0
17	Qani	35	L	2019-12-12	10	6	4	0	1	4	0	0

Gambar 4.7 Tabel hasil_kuis

4.4.2.2 Penyaringan Data

Data yang didapatkan kemudian disaring agar mendapatkan hasil yang valid. Sebelum penyaringan data dilakukan, peneliti memilih 500 data pertama, dan dihasilkan 283 data yang dianggap valid setelah melalui proses penyaringan. Berikut adalah tangkapan layar dari basis data tabel hasil_kuis yang telah disaring datanya kedalam sampel yang valid.

id	nama	jenis_kelamin	tanggal_lahir	tempat_lahir	pendidikan	nilai_1	nilai_2	nilai_3	nilai_4	nilai_5
1	Muhammad	P	2004-10-01	Sumedang	S	0	0	0	0	10
2	Wati	L	2003-03-08	Indragiri	S	0	0	0	0	0
3	Wati	L	2000-03-08	Kabupaten	S	0	0	0	0	0
4	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
5	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
6	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
7	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
8	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
9	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
10	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
11	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
12	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
13	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
14	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
15	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
16	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
17	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
18	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
19	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
20	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
21	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
22	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
23	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
24	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
25	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
26	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
27	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
28	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
29	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
30	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
31	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
32	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
33	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
34	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
35	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
36	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
37	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
38	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
39	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
40	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
41	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
42	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
43	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
44	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
45	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
46	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
47	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
48	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
49	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0
50	Wati	L	2001-02-01	Kabupaten	S	0	0	0	0	0

Gambar 4.8 Tabel hasil_kuis yang telah disaring

4.4.2.3 Scrape Data

Setelah penyaringan data dilakukan, langkah selanjutnya adalah pengunduhan data *tweet / tweet scrapping*. Peneliti menggunakan *library* bernama Twitter Scraper untuk mengunduh semua *tweet* pengguna dari pertama kali pengguna tersebut bergabung ke twitter hingga sampai pada tanggal dimana pengambilan data dilakukan. Berikut adalah script untuk pengunduhan data menggunakan library Twitter Scraper.

```

1 import mysql.connector
2 import datetime as dt
3 import csv
4 from twitterscraper import query_tweets
5
6
7 #
8 mydb = mysql.connector.connect(
9     host = "localhost",
10    user = "root",
11    passwd = "",
12    database = "tes_disc_final"
13 )
14 mycursor = mydb.cursor()
15
16
17
18 if __name__ == '__main__':
19
20     sql = "SELECT * FROM result_tweet WHERE sub_kas AND sub_tersj"
21     mycursor.execute(sql)
22     selected = mycursor.fetchall()
23
24     for user in selected:
25
26         utama = user[5]
27         hasil = user[6]
28
29         if hasil == 'D':
30             konversiHasilKeAngka = '0'
31         elif hasil == '3':
32             konversiHasilKeAngka = '1'
33         elif hasil == '5':
34             konversiHasilKeAngka = '2'
35         elif hasil == 'C':
36             konversiHasilKeAngka = '3'
37
38         thetweets = query_tweets("from:"+utama)
39
40     for tweet in thetweets:
41         sql = "INSERT INTO tweets VALUES (%s, %s, %s, %s, %s, %s, %s)"
42         sql = ("", tweet.text, tweet.user, user[0], tweet.verified, hasil, konversiHasilKeAngka)
43         mycursor.execute(sql, sql)
44         mydb.commit()
45         print("hasil is: "+tweet.text)

```

Gambar 4.9 Scrape data dengan Twitter Scraper

Setelah data diunduh/ di *mining*, maka tabel tweets akan terisi seperti pada gambar berikut.

id	text	username	username_db_id	verified	hasil	konver
31276	jem 10en, nhe RT @nonopia: lntas kayeny. Ngamp...	ikasyaf	124	None	C	3
31277	kangan talah @yanjuf...	ikasyaf	124	None	C	3
31278	knapa??? RT @nonopia: @ikasyaf muk beneran mata ...	ikasyaf	124	None	C	3
31279	kapan pada digantungi RT @izakcyber: polekkes i...	ikasyaf	124	None	C	3
31280	@Hannoy hoy camk, knp ya akun kmu kirim2 spam d...	ikasyaf	124	None	C	3
31281	halahha RT @nonopia: (-_-) RT @ikasyaf: knp? jw...	ikasyaf	124	None	C	3
31282	ya sa... RT @baessa: @ikasyaf @mihul @izakcyber ...	ikasyaf	124	None	C	3
31283	selamat pagi ochaas RT @tauzhaw: @ikasyaf selamat...	ikasyaf	124	None	C	3
31284	cars kerjanya gmnah leh? bu sampetnya dilerkati...	ikasyaf	124	None	C	3
31285	nico stroo @tauzhaw :)	ikasyaf	124	None	C	3
31286	I get it RT @camilistrua: 1 + 1 = Watlow. 2 + 2 = ...	ikasyaf	124	None	C	3
31287	knp? angah mangah aku mas, dlcok: D RT @nonopi...	ikasyaf	124	None	C	3
31288	ya, bglu >> RT @mihul: @ikasyaf @izakcyber @yas...	ikasyaf	124	None	C	3
31289	aku juga RT @nonopia: @ikasyaf lakin aku ga be li...	ikasyaf	124	None	C	3
31290	ngga RT @mihul: Kalo misalkan semua kmu angah...	ikasyaf	124	None	C	3
31291	RT @Holestau: lomading lita yy behat mata solat...	ikasyaf	124	None	C	3
31292	@izakcyber mangah dm2 profil tamasi emang lebi...	ikasyaf	124	None	C	3
31293	RT @wacela: hema kasih kepada semua rekan yang...	ikasyaf	124	None	C	3
31294	lebay halha RT @nonopia: Mengonang masa abab @...	ikasyaf	124	None	C	3
31295	amanah woa neag? RT @shibauchedat: @ikasyaf abi...	ikasyaf	124	None	C	3
31296	iy dari anals bandung, meh mhewa bglat 3... dula...	ikasyaf	124	None	C	3
31297	Gaku destiasi di kimbir (? , ?)	ikasyaf	124	None	C	3
31298	RT @nla_neth: pernah an... itu udah bu silang ...	ikasyaf	124	None	C	3
31299	sok RT @tauzhaw: @ikasyaf gah ada daban... pulang ...	ikasyaf	124	None	C	3
31300	kmu ga kulan? RT @tauzhaw: gah mau > RT @ikasya...	ikasyaf	124	None	C	3

Gambar 4.10 Tabel tweets

4.4.3 Preprocessing

Preprocessing merupakan tahap dimana dokumen harus diolah terlebih dahulu agar data yang diproses nantinya hanya data yang dibutuhkan saja.

4.4.3.1 Tweet Cleaning

Pada tahap *tweet cleaning*, dilakukan penghapusan beberapa parameter yang nantinya tidak dibutuhkan didalam pengklasifikasian data. Langkah-langkah *preprocessing* adalah sebagai berikut:

1. Mengubah semua huruf menjadi huruf kecil / *case folding*
2. Menghilangkan tautan atau URL
3. Menghilangkan nama pengguna atau *username*

4.4.3.3 Normalisasi Kata

Implementasi selanjutnya yaitu adalah tahap normalisasi kata, dimana kata slang atau kata tidak baku diubah menjadi kata baku. Berikut ini adalah kamus kata tidak baku yang telah dibuat oleh peneliti.

```

74 #bahasa gaul
75 ▾ slang = {
76   "org": "orang",
77   "rangorang": "orang-orang",
78   "hihi": "tawa",
79   "jgn": "jangan",
80   "eng": "menang",
81   "trs": "tefus",
82   "ujan": "hujan",
83   "pake": "pakai",
84   "nek": "kalau",
85   "kau": "kamu",
86   "u": "kamu",
87   "akuu": "aku",
88   "ajaa": "seja",
89   "tpi": "tapi",
90   "dri": "dari",
91   "ingat": "ingat",
92   "ban": "banget",
93   "teeen": "teman",
94   "gabisa": "tidak bisa",
95   "denger": "dengar",
96   "jgaa": "juga",
97   "jadi": "jadi",
98   "tdi": "tadi",
99   "tdi": "tadi",
100  "nnti": "nanti",
101  "marahh": "marah",
102  "lngt": "banget",
103  "apani": "apa ini",
104  "hahaa": "tawa",
105  "kasi": "kasih",
106  "dapat": "dapat",
107  "can": "bisa",
108  "wis": "sudah",
109  "adek": "adik",
110  "doi": "dia",
111  "makasih": "terimakasih",
112  "thanks": "terimakasih",
113  "gatau": "tidak mau",
114  "hho": "tawa",
115  "hahahah": "tawa",
116  "malen": "malan",
117  "utk": "untuk",
118  "krn": "karena"
119 }

```

Gambar 4.13 Bahasa tidak baku

Kemudian berikut ini adalah *script* untuk mengganti kata tidak baku menjadi kata yang baku.

```
#replace slang
featureList = [slang[word] if word in slang else word for word in
               featureList]
```

Gambar 4.14 Replace kata tidak baku

4.4.3.4 Stopword

Berikutnya yaitu implementasi *stopword*. Stop words adalah kata umum (*common words*) yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Berikut ini adalah *script*nya.

```
untokenize = ' '.join(featureList)
#remove stopwords
featureList = stopwords.remove(untokenize)
```

Gambar 4.15 Menghilangkan *stopword*

4.4.3.5 Stemming

Setelah penghapusan *stopwords*, lalu kita implementasikan *stemming* guna menghapus imbuhan pada kata sehingga kata tersebut menjadi kata dasar. Berikut adalah *script*nya.

```
#stemming
featureList = stemmer.stem(featureList)
```

Gambar 4.16 Stemming

4.4.3.6 POS Tagging

Langkah berikutnya adalah pemberian *tag* pada setiap kata baku. Dalam pemberian *tag* kepada kata, peneliti menggunakan *tools* dari Stanford Natural Language Processing yaitu Stanford *POS Tagger*. Tetapi karena belum ada model

POS tagger untuk bahasa Indonesia, peneliti menggunakan korpus dari penelitian [26] untuk membangun model *tagger* bahasa Indonesia yang akan digunakan pada tahap ini. Script untuk melatih model adalah sebagai berikut.

```

1 from nltk.tag import CRFTagger
2
3 jumlah = 50000
4 namaFile = "Indonesian_Manually_Tagged_Corpus.tsv"
5 with open(namaFile, 'r', encoding='utf-8') as f:
6     lines = f.read().split('\n')
7
8 pasangan = []
9 allPasangan = []
10
11 for line in lines[:min(jumlah, len(lines))]:
12     if line == '':
13         allPasangan.append(pasangan)
14         pasangan = []
15     else:
16         kata, tag = line.split('\t')
17         p = (kata, tag)
18         pasangan.append(p)
19
20 ct = CRFTagger()
21 ct.train(allPasangan, 'Indo.crf.tagger')

```

Gambar 4.17 Pelatihan *POS Tagging*

Setelah dilatih, maka kita akan memiliki model *POS tagger* untuk bahasa Indonesia. Tahap selanjutnya yang dilakukan adalah pemberian *tag* pada setiap kata. Scriptnya adalah sebagai berikut.

```

from nltk.tag import CRFTagger

featureList = ct.tag_sents([FeatureList])

```

Gambar 4.18 Pemberian *tag*

4.4.3.7 POS Filtering

Setelah kata diberi *tag*, selanjutnya adalah penyaringan kata berdasarkan kata yang dibutuhkan saja. Scriptnya adalah sebagai berikut.

```

featureList = [(w, t) for (w, t) in featureList if t == 'NN' or t.startswith('RB') or t.startswith('JJ') or t.startswith('VB')]

featureList = [n[0] for n in featureList]

```


Gambar 4.19 POS Filtering

4.4.3.8 Tweet Mergling

Setelah *tweet* di bersihkan, di *stemming*, dan disaring berdasarkan tagnya, langkah selanjutnya adalah menggabungkan seluruh *tweet* dalam satu akun tersebut menjadi satu. Berikut adalah scriptnya.

```
mergedTweets = []
for word in featureList:
    mergedTweets.append(word)
```

Gambar 4.20 Penggabungan *tweet*

4.4.4 Pelatihan

4.4.4.1 Term Frequency

Setelah tahap *preprocessing*, tahap selanjutnya adalah merangking 200 kata pertama di setiap tipe kepribadian dengan menggunakan metode *term frequency*. Berikut ini adalah script dari *term frequency*.

```

1 import time
2 import math
3 import json
4
5 def tf(bow):
6     panjangKata = len(bow)
7     hasil = {}
8     for word in bow:
9         hasil[word] = bow.count(word)/panjangKata
10        print(word)
11
12    #mengubah tipe menjadi tuple dan mengurutkan dari terkecil ke terbesar
13    hasil = sorted(hasil.items(), key = lambda kv:(kv[1], kv[0]))
14
15    #reverse hasil, menjadi terbesar ke terkecil
16    hasil = hasil[::-1]
17
18    #mengubah tipe menjadi dictionary kembali, dan memilih 200 teratas
19    hasil = dict((k,v) for k,v in hasil[:200])
20
21    return hasil
22
23 if __name__ == "__main__":
24
25    #load kalimat
26    with open("3.txt", "r") as kalimat:
27        kalimat = json.loads(kalimat.read())
28        executeTF = tf(kalimat)
29
30    file = open("3-tf.txt", "w")
31    file.write(json.dumps(executeTF))
32    file.close()

```

Gambar 4.21 Fungsi term frequency

4.4.4.2 Konversi Ke Biner

Setelah proses term frequency, dokumen kemudian dikonversi menjadi inputan algoritma *Artificial Neural Network* dengan cara mengubahnya kedalam biner. Scriptnya adalah sebagai berikut.

```

351 if __name__ == "__main__":
352
353     #load kalimat
354     with open("0-tf200.txt", "r") as kalimat:
355         tfidf1 = json.loads(kalimat.read())
356
357     with open("1-tf200.txt", "r") as kalimat:
358         tfidf2 = json.loads(kalimat.read())
359
360     with open("2-tf200.txt", "r") as kalimat:
361         tfidf3 = json.loads(kalimat.read())
362
363     with open("3-tf200.txt", "r") as kalimat:
364         tfidf4 = json.loads(kalimat.read())
365
366     tfidf = tfidf1 + tfidf2 + tfidf3 + tfidf4
367
368     # sql = "SELECT * FROM tweets WHERE id = 22" #210
369     finalOutputPathOutput = 11
370     finalOutputPathInput = 11
371
372     for i in range(1, 1033): #200
373
374         sql = "SELECT * FROM tweets WHERE username_id=" + str(i) +
375             " AND id=" + str(i)
376
377         selected = MyCursor.fetchall()
378
379         if selected:
380             mergedTweets = []
381
382             for user in selected:
383                 tweet = preprocessing(user[1])
384                 userID = user[4]
385                 featureList = word_tokenize(tweet)
386
387                 #jangan slang
388                 featureList = [word for word in featureList if word in stopwords]
389                 #remove stopwords
390                 featureList = stopwords.remove(featureList)
391
392                 #stemming
393                 featureList = stemmer.stem(featureList)
394
395                 #remove stopwords for the second time
396                 featureList = stopwords.remove(featureList)
397
398                 featureList = word_tokenize(featureList)
399
400                 featureList = ct.tag_sents(featureList)
401
402                 featureList = FeatureList(f)
403
404                 featureList = [w, 1] for (w, 1) in featureList if w != 'NN' or (w.startswith('NN')
405                     or w.startswith('JJ') or w.startswith('VB'))
406
407                 featureList = [w] for w in featureList
408
409                 # print(featureList)
410
411                 for word in featureList:
412                     mergedTweets.append(word)
413
414                 # print("done for " + user[1])
415
416     mergedTweets = tfmergedTweets
417
418     tfidf = np.array(tfidf)
419
420     inputTab = []
421
422     for row in range(100):
423         inputTab.append(0)
424
425

```

```

426     for word in mergedTweets:
427         indexIntfidf = np.where(tfidf == word)
428         indexIntfidf = indexIntfidf[0].tolist()
429
430     if indexIntfidf:
431         for i in indexIntfidf:
432             inputToNo[i] = 1
433
434     # hasilToNo = {}
435
436     if hasilTes=="1":
437         hasilToNo=0
438     elif hasilTes=="2":
439         hasilToNo=1
440     elif hasilTes=="3":
441         hasilToNo=2
442     elif hasilTes=="4":
443         hasilToNo=3
444
445
446     file = open("inputs200.txt", "w")
447     file.write(json.dumps(resultInputValueToNo))
448     file.close()
449
450     file = open("outputs200.txt", "w")
451     file.write(json.dumps(resultOutputValueToNo))
452     file.close()

```

Gambar 4.22 Konversi ke biner

4.4.4.3 Training Data

Setelah data diubah menjadi biner, layer input memiliki 800 *node*. Angka 800 didapat dari 200 kata teratas dari setiap tipe kepribadian dijumlahkan dengan jumlah seluruh tipe kepribadian. Pada *hidden layer*, peneliti menentukan *hidden layer* sebanyak 400 node karena itu adalah jumlah node yang mengeluarkan output dengan hasil terbaik. Sedangkan pada output layer terdapat 4 node yang didapat dari jumlah seluruh tipe kepribadian yang berjumlah 4 tipe.

Pada penelitian ini peneliti menggunakan library python yaitu tensorflow untuk menjalankan algoritma *Artificial Neural Network* ini. Berikut adalah script untuk pelatihan data sebanyak 275 akun.

```

3 # TensorFlow and tf.keras
4 import tensorflow as tf
5 from tensorflow import keras
6 import json
7 import os
8
9 # Helper libraries
10 import numpy as np
11
12 checkpoint_path = "ckpt/ckpt"
13 checkpoint_dir = os.path.dirname(checkpoint_path)
14
15 # Monitor checkpoint callback
16 cp_callback = tf.keras.callbacks.ModelCheckpoint(checkpoint_path, save_weights_only=True, verbose=1)
17
18 # Load dataset
19 with open("inputs200.txt", "r") as file_obj:
20     inputs200 = np.array(json.loads(file_obj.read()))
21
22 with open("outputs200.txt", "r") as file_obj:
23     outputs200 = np.array(json.loads(file_obj.read()))
24
25 model = keras.Sequential([
26     keras.layers.Dense(100, input_shape=(100,1), activation='tanh'),
27     keras.layers.Dense(400, activation='tanh'),
28     keras.layers.Dense(1, activation='sigmoid')
29 ])
30
31 model.compile(optimizer='adam',
32               loss='sparse_categorical_crossentropy',
33               metrics=['accuracy'])
34
35 model.fit(inputs200, outputs200, epochs=100, callbacks=[cp_callback])

```

Gambar 4.23 Training data

4.4.5 Pengujian

4.4.5.1 K-Fold Cross Validation

K-Fold Cross Validation adalah salah satu metode yang digunakan untuk validasi sebuah dataset. Cara kerja dari metode ini adalah dengan cara data akan di partisi / dibagi menjadi k bagian/ *fold* segment dengan ukuran yang sama. Proses *training* dan *testing* data dilakukan sebanyak k kali dimana pada setiap iterasi ke-*i*, segment *K_i* akan menjadi data testing sedangkan yang lain berperan sebagai data training. Proses tersebut akan dilakukan sebanyak k kali dengan segment *D_i* akan menjadi data testing tepat satu kali dan akan menjadi data training.

Pada penelitian ini, dilakukan pencarian model neural network sebanyak 72 skenario guna mencari input layer, hidden layer, dan epoch (perulangan) yang mendapat akurasi paling tinggi menggunakan K-Fold Cross Validation. Nantinya

model yang mendapat akurasi paling baik akan diterapkan pada aplikasi prediksi kepribadian. Seluruh skenario dan hasilnya dapat dilihat di tabel 4.1. Sedangkan, script untuk menjalankan K-fold Cross validation adalah sebagai berikut.

```

1 import tensorflow as tf
2 from tensorflow import keras
3 from sklearn.model_selection import StratifiedKFold, StratifiedShuffleSplit, StratifiedKFold
4 import numpy as np
5 import time
6 import os
7
8 checkpoint_path = "checkpoint"
9 checkpoint_dir = os.path.dirname(checkpoint_path)
10
11 # Monitor checkpoint callback
12 cp_callback = tf.keras.callbacks.Callback(checkpoint(checkpoint_path, save_weights_only=True, verbose=0))
13
14 seed = 7
15 np.random.seed(seed)
16
17 with open("inputs200.txt", "r") as f:
18     theInput = np.array([int(line) for line in f.readlines()])
19 with open("outputs200.txt", "r") as f:
20     theOutputs = np.array([int(line) for line in f.readlines()])
21 n_folds = 10
22 kf = StratifiedKFold(n_splits=n_folds, shuffle=True, random_state=seed)
23
24 scores = []
25
26 for i, (train, test) in enumerate(kf.split(theInput, theOutputs)):
27
28     if i == 0:
29         model = keras.Sequential()
30         keras.layers.Dense(100, input_shape=(100,)), activation=tf.nn.relu,
31         keras.layers.Dense(100, activation=tf.nn.relu),
32         keras.layers.Dense(10, activation=tf.nn.softmax)
33         # 300 layer Aktivasi antara 200 ke 400
34         model.compile(optimizer="adam",
35                       loss="sparse_categorical_crossentropy",
36                       metrics=["accuracy"])
37         # model.load_weights(checkpoint_path)
38         model.fit(theInput[train], theOutputs[train], epochs=10, verbose=0, callbacks = [cp_callback])
39     else:
40         model = keras.Sequential()
41         keras.layers.Dense(100, input_shape=(100,)), activation=tf.nn.relu,
42         keras.layers.Dense(100, activation=tf.nn.relu),
43         keras.layers.Dense(10, activation=tf.nn.softmax)
44         # 300 layer Aktivasi antara 200 ke 400
45         model.compile(optimizer="adam",
46                       loss="sparse_categorical_crossentropy",
47                       metrics=["accuracy"])
48         model.fit(theInput[train], theOutputs[train], epochs=100, verbose=0, callbacks = [cp_callback])
49
50     scores = model.evaluate(theInput[test], theOutputs[test], verbose=0)
51
52     print ("Running Fold", i+1, " ", n_folds)
53     print ("ur %2f%%" % (model.metrics_names[1], scores[i]*100))
54     scores.append(scores[i] + 100)
55
56 print("rata-rata akurasi adalah %2f%%" % (np.mean(scores)))

```

Gambar 4.24 K-Fold Cross Validation

Tabel 4.3 Skenario Pencarian Akurasi Terbaik dengan K-Fold Cross Validation

No	Jumlah Input Layer	Jumlah Hidden Layer	Jumlah Epoch	Akurasi
1.	400	200	10	35.87%
2.	400	200	11	41.40%
3.	400	200	12	38.24%
4.	400	200	13	37.70%
5.	400	200	14	34.58%
6.	400	200	15	32.90%
7.	400	200	50	35.43%
8.	400	200	100	35.41%
9.	400	400	10	33.71%
10.	400	400	11	33.78%
11.	400	400	12	36.79%
12.	400	400	13	36.89%
13.	400	400	14	32.84%
14.	400	400	15	38.58%
15.	400	400	50	35.93%
16.	400	400	100	34.07%
17.	400	800	10	33.73%
18.	400	800	11	32.42%
19.	400	800	12	36.00%
20.	400	800	13	38.61%
21.	400	800	14	33.69%
22.	400	800	15	32.71%
23.	400	800	50	35.05%
24.	400	800	100	34.50%
25.	800	400	10	42.91%
26.	800	400	11	39.96%
27.	800	400	12	41.09%
28.	800	400	13	39.84%
29.	800	400	14	39.80%
30.	800	400	15	40.31%
31.	800	400	50	38.10%
32.	800	400	100	40.31%
33.	800	800	10	39.43%
34.	800	800	11	38.53%
35.	800	800	12	40.71%
36.	800	800	13	40.73%

Tabel 4.3 Skenario Pencarian Akurasi Terbaik dengan K-Fold Cross Validation (lanjutan)

37.	800	800	14	40.73%
38.	800	800	15	39.86%
39.	800	800	50	39.44%
40.	800	800	100	41.66%
41.	800	1600	10	38.02%
42.	800	1600	11	40.66%
43.	800	1600	12	39.37%
44.	800	1600	13	41.66%
45.	800	1600	14	40.79%
46.	800	1600	15	37.74%
47.	800	1600	50	38.04%
48.	800	1600	100	41.96%
49.	1200	600	10	28.53%
50.	1200	600	11	30.26%
51.	1200	600	12	28.57%
52.	1200	600	13	30.65%
53.	1200	600	14	30.74%
54.	1200	600	15	31.27%
55.	1200	600	50	29.40%
56.	1200	600	100	30.31%
57.	1200	1200	10	29.75%
58.	1200	1200	11	32.92%
59.	1200	1200	12	32.01%
60.	1200	1200	13	28.78%
61.	1200	1200	14	26.93%
62.	1200	1200	15	31.16%
63.	1200	1200	50	31.60%
64.	1200	1200	100	29.04%
65.	1200	2400	10	32.02%
66.	1200	2400	11	34.69%
67.	1200	2400	12	30.20%
68.	1200	2400	13	34.33%
69.	1200	2400	14	28.42%
70.	1200	2400	15	31.62%
71.	1200	2400	50	29.78%
72.	1200	2400	100	33.31%

Pada tabel 4.1, dapat dilihat bahwa skenario terbaik adalah skenario ke 25 dimana modelnya terdiri dari 800 input layer, 400 hidden layer, dan 10 epoch.

Akurasi yang didapat adalah sebesar 52.91% dengan akurasi terbesar sebesar mencapai 62.50% dan akurasi terkecil sebesar hanya 30.43% seperti yang ditunjukkan pada gambar berikut.

```
Running Fold 1 / 10
acc: 62.50%
Running Fold 2 / 10
acc: 56.52%
Running Fold 3 / 10
acc: 30.43%
Running Fold 4 / 10
acc: 47.83%
Running Fold 5 / 10
acc: 40.91%
Running Fold 6 / 10
acc: 31.82%
Running Fold 7 / 10
acc: 40.91%
Running Fold 8 / 10
acc: 45.45%
Running Fold 9 / 10
acc: 40.91%
Running Fold 10 / 10
acc: 31.82%
rata-rata akurasi adalah 42.91%
```

Gambar 4.25 Tes akurasi menggunakan K-Fold Cross Validation

4.4.0 Antarmuka

4.4.6.1 Antarmuka Halaman Utama

Pada halaman utama terdapat inputan dimana pengguna dapat menginputkan *username* twitter yang ingin mereka klasifikasikan dan tombol submit. Seperti yang ditunjukkan pada gambar berikut.

Prediksi Kepribadian DISC

Olan Idm



Aplikasi pengklasifikasian kepribadian kedalam DISC menggunakan algoritma Back Propagation.

Username Twitter

Masukkan username twitter...

Lihat Hasilnya!

Gambar 4.26 Antarmuka halaman utama

4.4.6.2 Antarmuka Halaman Prediksi

Antarmuka hasil prediksi menampilkan tipe temperamen dari akun yang telah di *scrape* dan di analisis datanya. Keterangan tipe temperamen tersebut juga ditampilkan beserta kekurangan dan kelebihanannya.

Kepribadian Lukman adalah Compliance dan Influence

Karakteristik Utama dari Kepribadian tersebut

- Perak dan ramah
- Mengikuti dan menanggapi komentar
- Dapat berinteraksi dengan banyak orang
- Adanya rasa kepedulian

Kelebihan dan Kekurangan

Kelebihan: Memiliki kemampuan komunikasi yang baik, mampu menjalin hubungan yang baik dengan orang lain, dan memiliki rasa tanggung jawab yang tinggi.

Kekurangan: Mudah terpengaruh oleh orang lain, cenderung menghindari konflik, dan kurang berani dalam mengambil keputusan.

Rekomendasi: Perlu meningkatkan kemampuan komunikasi yang lebih baik, berani dalam mengambil keputusan, dan meningkatkan kemampuan dalam menghadapi konflik.

Perilaku: Menghindari konflik, berinteraksi dengan orang lain.

Tag: #Kepribadian #DISC

Terima kasih telah membaca dan berkunjung!

Home / About / Contact Us / Privacy Policy / Terms of Service /

Gambar 4.27 Antarmuka halaman prediksi

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, kesimpulan yang didapatkan adalah:

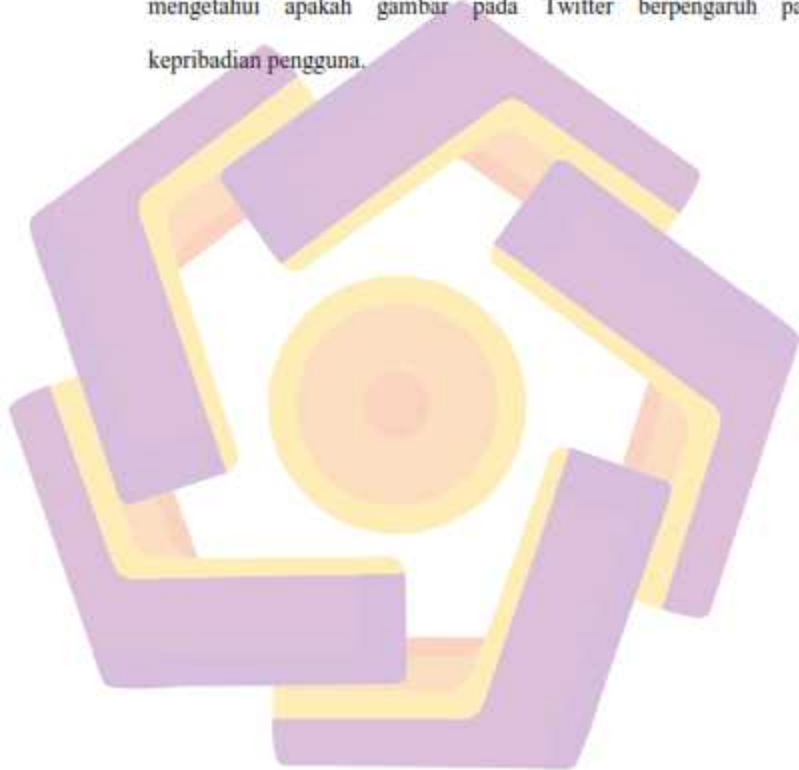
1. Pengklasifikasian dilakukan dengan melewati tahap *preprocessing*, meliputi: *tweet cleaning*, *tokenization*, *stemming*, *POS tagging*, dan juga *POS filtering*. Setelah tahap *preprocessing*, data akan melalui proses perankingan dengan metode *term frequency* dan perubahan menjadi biner.
2. Klasifikasi kepribadian berdasarkan teori kepribadian DISC dengan algoritma *Artificial Neural Network* menghasilkan akurasi sebesar 42.91% jika diuji *K-Fold Cross Validation*.

5.2 Saran

Penelitian ini masih terdapat beberapa keterbatasan dan kekurangan yang dapat menjadi acuan bagi penelitian dan pengembangan selanjutnya. Adapun saran yang didapat dari penelitian ini adalah:

1. Perlu penambahan kamus untuk kata tidak baku agar semakin banyak kata tidak baku yang tersaring dan dokumen menjadi semakin bersih.
2. *POS tagging* pada penelitian ini berasal dari korpus yang terbentuk dari kalimat-kalimat baku, sedangkan di dalam tweet banyak ditemukan kata yang tidak baku. Sehingga dibutuhkan korpus yang berasal dari pola-pola kalimat yang tidak baku.

3. Penelitian ini masih meneliti sebatas teks pada tweet saja. Pengembangan penelitian bisa dilakukan dengan menambahkan variabel-variabel lainnya seperti emoji/emotikon.
4. Penelitian pada gambar di Twitter juga layak untuk dilakukan untuk mengetahui apakah gambar pada Twitter berpengaruh pada kepribadian pengguna.



DAFTAR PUSTAKA

PUSTAKA BUKU

- Arga January, M., & Adrian., 2014, *Modul Pelatihan DISC*. One Spirit Consulting, Jakarta
- Kadir, A., 2005, *Dasar Pemrograman Python*, Andi Publisher, Yogyakarta
- Kusumadewi, S., 2003, *Artificial intelligence (teknik dan aplikasinya)*, Graha Ilmu, Yogyakarta
- MADCOMS, 2010, *Facebook, Twitter, dan Plurk Dalam Satu Genggaman*, Andi Publisher, Yogyakarta
- Nofiar, Y., 2009, *Handbook of disc: alat ukur perilaku kerja*, Quantum Quality International, <https://opac.perpusnas.go.id/DetailOpac.aspx?id=122252>
- Shin, E., 2013, *The DISC Codes: Cara Cepat Menguasai Kode Sukses Manusia*, Bor Monda Tour, Jakarta
- Sutojo, T., Mulyanto, E., & Suhartono, V., 2011, *Kecerdasan buatan*. Andi Offset, Yogyakarta
- Suyanto, 2014, *Artificial Intelligence*. Penerbit Informatika. Bandung
- Tamburaka, A., 2013, *Literasi media: cerdas bermedia khalayak media massa*. PT. Raja Grafindo Persada. <https://opac.perpusnas.go.id/DetailOpac.aspx?id=1193237>

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

- Andrianto, B., & Adinugroho, S., 2018, Analisis Sentimen Konten Radikal Melalui Dokumen Twitter Menggunakan Metode Backpropagation. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(12), 7380–7385. *Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya*, Malang
- Buntoro, G. A., Adji, T. B., & Purnamasari, A. E., 2014, Sentiment Analysis Twitter dengan Kombinasi Lexicon Based dan Double Propagation, *Citee*, January 2014, 39–43
- Dewi Puri Astiti, S.Fil, M. S., & Dr. Ni Made Swasti Wulanyani, S.Psi, M. E. P., 2015, PERILAKU KONTRAPRODUKTIF DAN MODEL KEPRIBADIAN

DISC PADA TENAGA KESEHATAN DI RUMAH SAKIT UMUM
DENPASAR, *Sosial-Humaniora/Psikologi*, 1–40

- Hartanto, A. D., Utami, E., Adi, S., & Hudnanto, H. S. , 2019, Job seeker profile classification of twitter data using the naïve bayes classifier algorithm based on the DISC method. *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2019*. <https://doi.org/10.1109/ICITISEE48480.2019.9003963>
- Lhaksana, KM; Nhita, Fhira & Anggraini, D., 2017, Klasifikasi Kepribadian Berdasarkan Status Facebook Menggunakan Metode Backpropagation, *E-Proceeding of Engineering*, 4(3), 5174–5183
- Liu, X., & Zhu, T, 2016, Deep learning for constructing microblog behavior representation to identify social media user's personality, *PeerJ Computer Science*, <https://doi.org/10.7717/peerj-cs.81>
- Nurmila, N., Sugiharto, A., & Sarwoko, E. A., 2010, Algoritma Back Propagation Neural Network Untuk Pengenalan Pola Karakter Huruf Jawa, *Jurnal Masyarakat Informatika*, 1(1), 1–10. <https://doi.org/10.14710/jmasif.1.1>
- Solikhun, S., Safii, M., & Trisno, A., 2017, Jaringan Saraf Tiruan Untuk Memprediksi Tingkat Pemahaman Sisiwa Terhadap Matapelajaran Dengan Menggunakan Algoritma Backpropagation, *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, 1(1), 24, <https://doi.org/10.30645/j-sakti.v1i1.26>
- Zulfa, L., & Winarko, E., 2017, Sentimen Analisis Tweet Berbahasa Indonesia Dengan Deep Belief Network. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*. <https://doi.org/10.22146/ijccs.24716>

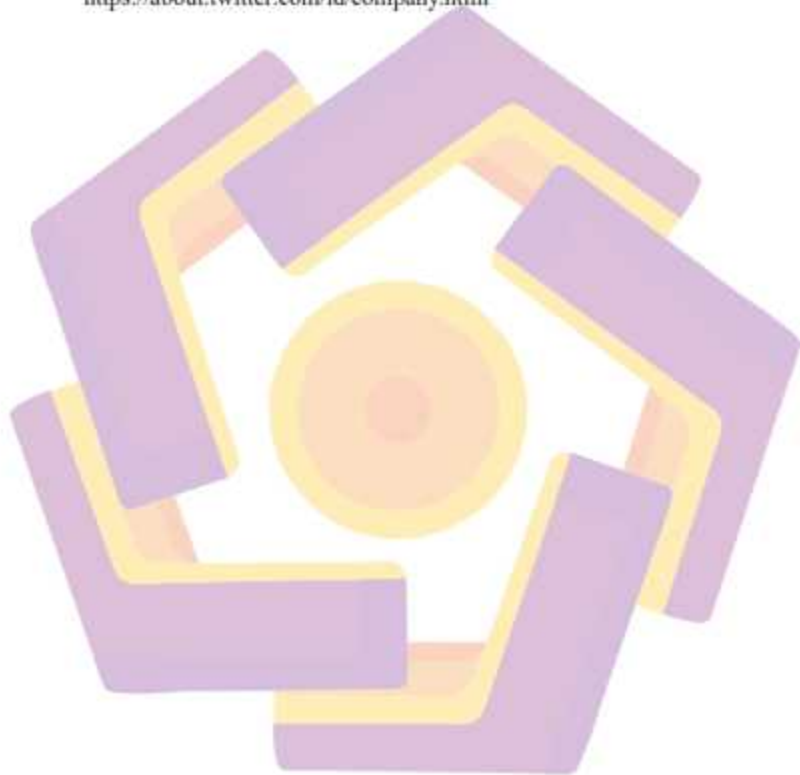
PUSTAKA LAPORAN PENELITIAN

- Dinakaramani, A., Fam, R., Luthfi, A., & Manurung, R., 2014, *Designing an Indonesian part of speech tagset and manually tagged Indonesian corpus*. <https://doi.org/10.1109/IALP.2014.6973519>
- Liu, B., 2010, *Sentiment Analysis and Subjectivity: Handbook of Natural Language Processing, Second Edition*, (editors: N. Indurkha and F. J. Damerau, Chapman and Hall/CRC, USA

PUSTAKA INTERNET

Asosiasi Penyelenggara Jasa Internet Indonesia. (n.d.). Retrieved June 23, 2020, from <https://apjii.or.id/survei2018s>

Twitter - Perusahaan. (n.d.). Retrieved June 23, 2020, from <https://about.twitter.com/id/company.html>



LAMPIRAN

1. Alat Kuis DISC

Alat kuis yang berasal dari buku "The DISC Codes: Cara Cepat Menguasai Kode Sukses Manusia" dikonversi menjadi halaman web.

Tes Kepribadian DISC

DISC adalah

Salah satu metode penilaian

yang digunakan untuk menilai perilaku yang ada.

Salah satu menggunakan kode yang dapat digunakan oleh orang (DISC) dan perilaku yang paling baik menurut Anda (L, S, dan D).

Siapa adalah orang yang...

M **S** **D**

- Cara yang Mudah untuk
- Perilaku yang sangat stabil orang
- Melayak, Melayak orang
- Tolak, Hingamati

M **L** **D**

- Cara yang Mudah
- Stabil, Stabil
- Mudah Perilaku, Tidak baik
- Reaksi, Menunggu

Tes Kepribadian DISC

DISC adalah

Salah satu metode penilaian yang digunakan untuk menilai perilaku yang ada.

Salah satu menggunakan kode yang dapat digunakan oleh orang (DISC) dan perilaku yang paling baik menurut Anda (L, S, dan D).

Siapa adalah orang yang...

M **S** **D**

- Cara yang Mudah untuk
- Perilaku yang sangat stabil orang
- Melayak, Melayak orang
- Tolak, Hingamati

M **L** **D**

- Cara yang Mudah
- Stabil, Stabil
- Mudah Perilaku, Tidak baik
- Reaksi, Menunggu

Ulasan Tes

Ulasan tes ini akan...

Pilih Tes

M	L	2/74
<input type="radio"/>	<input type="radio"/>	Membantu orang
<input type="radio"/>	<input type="radio"/>	Donor darah bersama
<input type="radio"/>	<input type="radio"/>	Bagian dari keluarga
<input type="radio"/>	<input type="radio"/>	Ingin membuat teman
M	L	4/74
<input type="radio"/>	<input type="radio"/>	Melakukan hal baik
<input type="radio"/>	<input type="radio"/>	Mengucapkan perasaan saya
<input type="radio"/>	<input type="radio"/>	Membantu orang lain
<input type="radio"/>	<input type="radio"/>	Siapa pun saja
M	L	5/74
<input type="radio"/>	<input type="radio"/>	Hidup lebih tenang
<input type="radio"/>	<input type="radio"/>	Demak capai, Tuhan
<input type="radio"/>	<input type="radio"/>	Utakir membuat kebahagiaan
<input type="radio"/>	<input type="radio"/>	Udah nggak ada urusan
M	L	6/74
<input type="radio"/>	<input type="radio"/>	Belaku selalu merasa aman
<input type="radio"/>	<input type="radio"/>	Senang ketika sudah selesai bekerja
<input type="radio"/>	<input type="radio"/>	Membantu orang lain
<input type="radio"/>	<input type="radio"/>	Suka melakukan sesuatu yang positif
M	L	7/74
<input type="radio"/>	<input type="radio"/>	Siapa pun berubah menjadi
<input type="radio"/>	<input type="radio"/>	Endung yang berkesan
<input type="radio"/>	<input type="radio"/>	Siapa saja di kalangan
<input type="radio"/>	<input type="radio"/>	Tidak selalu beruntung
M	L	8/74
<input type="radio"/>	<input type="radio"/>	Mengurangi yang baik
<input type="radio"/>	<input type="radio"/>	Mengurangi yang baik
<input type="radio"/>	<input type="radio"/>	Mengurangi yang baik
<input type="radio"/>	<input type="radio"/>	Mengurangi yang baik
M	L	9/74
<input type="radio"/>	<input type="radio"/>	Hal yang penting
<input type="radio"/>	<input type="radio"/>	Lakukan dengan benar, Allah yang penting
<input type="radio"/>	<input type="radio"/>	Siapa yang penting
<input type="radio"/>	<input type="radio"/>	Hal yang penting
M	L	10/74
<input type="radio"/>	<input type="radio"/>	Allah sebagai sumber segala kebaikan
<input type="radio"/>	<input type="radio"/>	Allah membuat semua dengan baik
<input type="radio"/>	<input type="radio"/>	Allah membuat Tuhan yang baik
<input type="radio"/>	<input type="radio"/>	Allah membuat Tuhan yang baik
M	L	11/74
<input type="radio"/>	<input type="radio"/>	Siapa saja, Allah yang penting
<input type="radio"/>	<input type="radio"/>	Allah, siapa yang penting
<input type="radio"/>	<input type="radio"/>	Allah yang penting
<input type="radio"/>	<input type="radio"/>	Ingin melakukan yang baik
M	L	12/74
<input type="radio"/>	<input type="radio"/>	Siapa pun saja, Allah yang penting
<input type="radio"/>	<input type="radio"/>	Siapa pun saja, Allah yang penting
<input type="radio"/>	<input type="radio"/>	Siapa pun saja, Allah yang penting
<input type="radio"/>	<input type="radio"/>	Siapa pun saja, Allah yang penting
M	L	13/74
<input type="radio"/>	<input type="radio"/>	Ingin melakukan
<input type="radio"/>	<input type="radio"/>	Pada setiap kesempatan
<input type="radio"/>	<input type="radio"/>	Tertarik mempelajari perasaan
<input type="radio"/>	<input type="radio"/>	Siapa pun saja, Allah yang penting
M	L	14/74
<input type="radio"/>	<input type="radio"/>	Siapa pun saja, Allah yang penting
<input type="radio"/>	<input type="radio"/>	Siapa pun saja, Allah yang penting
<input type="radio"/>	<input type="radio"/>	Siapa pun saja, Allah yang penting
<input type="radio"/>	<input type="radio"/>	Siapa pun saja, Allah yang penting

M L 16/74

- Mengaparkan video berbudak dgn benar
- Pergerakan mata kanan-kiri
- Dengan cara kedudukan baru
- Meminta gerakan oleh lawan yg dihadap

M L 17/74

- Aliran perlu dipertanyakan
- Aliran membuat sil
- Aliran membuat bunyi
- Aliran membuat ariat

M L 17/74

- Pergerakan kebudayaan
- Perseki Gerakan
- Keindahan, kesihatan
- Bina, daya, kecekapan, kecekapan

M L 17/74

- Menunggu Pergerakan lanjutan
- Sedia bergerak, bergerak
- Dengan arahan, kemudian
- Menunggu, tidak baik

M L 18/74

- Tidak membuat perubahan
- Melakukan aliran peribadi, ikut arahan
- Mula bergerak, bergerak
- Menghentikan gerakan tidak

M L 18/74

- Gerakan peribadi mereka
- Saya akan melakukan
- Saya akan melakukan mereka
- Saya dipukul video

M L 18/74

- Memainkan orang di dalam
- Mengajar 2 orang berlawanan
- Gerakan, tidak
- Pergerakan, kemudian

M L 18/74

- Menyempatkan orang, tidak selesa
- Terletak lapis, tidak
- Mula, tidak bergerak
- Tanpa, kemudian

M L 18/74

- Menunggu, tidak
- Menghentikan, tidak
- Menghentikan, tidak
- Menunggu, tidak

M L 18/74

- Dengan pergerakan, dengan pergerakan
- Menunggu, tidak
- Gerakan, kemudian
- Menunggu, kemudian yang bergerak, tidak

Nama Lengkap


Masukkan nama lengkap anda.

Uraikan Halayati

2. Skenario Pencarian Model Terbaik

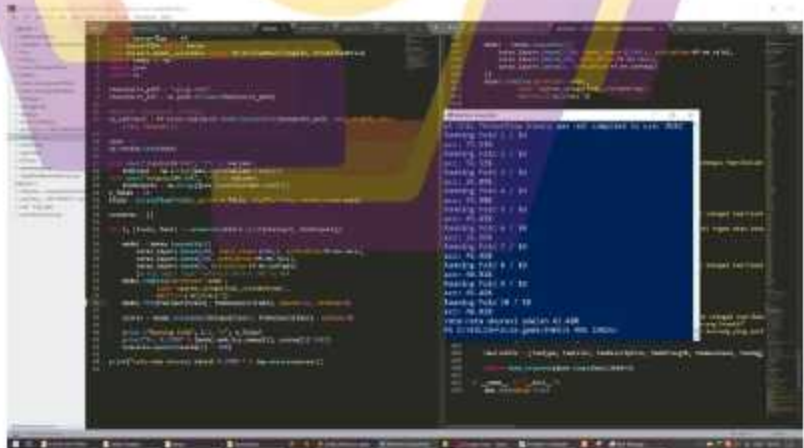
Terdapat sebanyak 72 skenario untuk mencari model terbaik dari algoritma ANN.

1. Input layer: 400, Hidden layer: 200, Epoch: 10



```
python3 main.py --input 400 --hidden 200 --epoch 10
[INFO] Training neural network with 400 input nodes, 200 hidden nodes, and 10 epochs.
[INFO] Training progress:
Epoch 1: Loss: 0.85, Accuracy: 0.15
Epoch 2: Loss: 0.75, Accuracy: 0.25
Epoch 3: Loss: 0.65, Accuracy: 0.35
Epoch 4: Loss: 0.55, Accuracy: 0.45
Epoch 5: Loss: 0.45, Accuracy: 0.55
Epoch 6: Loss: 0.35, Accuracy: 0.65
Epoch 7: Loss: 0.25, Accuracy: 0.75
Epoch 8: Loss: 0.15, Accuracy: 0.85
Epoch 9: Loss: 0.1, Accuracy: 0.9
Epoch 10: Loss: 0.05, Accuracy: 0.95
[INFO] Training completed. Best model saved.
```

2. Input layer: 400, Hidden layer: 200, Epoch: 11



```
python3 main.py --input 400 --hidden 200 --epoch 11
[INFO] Training neural network with 400 input nodes, 200 hidden nodes, and 11 epochs.
[INFO] Training progress:
Epoch 1: Loss: 0.85, Accuracy: 0.15
Epoch 2: Loss: 0.75, Accuracy: 0.25
Epoch 3: Loss: 0.65, Accuracy: 0.35
Epoch 4: Loss: 0.55, Accuracy: 0.45
Epoch 5: Loss: 0.45, Accuracy: 0.55
Epoch 6: Loss: 0.35, Accuracy: 0.65
Epoch 7: Loss: 0.25, Accuracy: 0.75
Epoch 8: Loss: 0.15, Accuracy: 0.85
Epoch 9: Loss: 0.1, Accuracy: 0.9
Epoch 10: Loss: 0.05, Accuracy: 0.95
Epoch 11: Loss: 0.02, Accuracy: 0.98
[INFO] Training completed. Best model saved.
```

3. Input layer: 400, Hidden layer: 200, Epoch: 12


```
def model(hidden):  
    model = Sequential()  
    model.add(Dense(400, activation='tanh'))  
    model.add(Dense(hidden, activation='tanh'))  
    model.add(Dense(1, activation='sigmoid'))  
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])  
    return model  
  
def train(hidden):  
    model = model(hidden)  
    model.fit(X_train, y_train, epochs=100, validation_data=(X_test, y_test))  
    return model  
  
def test(hidden):  
    model = model(hidden)  
    model.evaluate(X_test, y_test)  
    return model
```

100 Epochs Training Results for 400 Input, 200 Hidden Nodes

Epoch	Train Accuracy	Test Accuracy
1	0.00	0.00
2	0.00	0.00
3	0.00	0.00
4	0.00	0.00
5	0.00	0.00
6	0.00	0.00
7	0.00	0.00
8	0.00	0.00
9	0.00	0.00
10	0.00	0.00
11	0.00	0.00
12	0.00	0.00
13	0.00	0.00
14	0.00	0.00
15	0.00	0.00
16	0.00	0.00
17	0.00	0.00
18	0.00	0.00
19	0.00	0.00
20	0.00	0.00
21	0.00	0.00
22	0.00	0.00
23	0.00	0.00
24	0.00	0.00
25	0.00	0.00
26	0.00	0.00
27	0.00	0.00
28	0.00	0.00
29	0.00	0.00
30	0.00	0.00
31	0.00	0.00
32	0.00	0.00
33	0.00	0.00
34	0.00	0.00
35	0.00	0.00
36	0.00	0.00
37	0.00	0.00
38	0.00	0.00
39	0.00	0.00
40	0.00	0.00
41	0.00	0.00
42	0.00	0.00
43	0.00	0.00
44	0.00	0.00
45	0.00	0.00
46	0.00	0.00
47	0.00	0.00
48	0.00	0.00
49	0.00	0.00
50	0.00	0.00
51	0.00	0.00
52	0.00	0.00
53	0.00	0.00
54	0.00	0.00
55	0.00	0.00
56	0.00	0.00
57	0.00	0.00
58	0.00	0.00
59	0.00	0.00
60	0.00	0.00
61	0.00	0.00
62	0.00	0.00
63	0.00	0.00
64	0.00	0.00
65	0.00	0.00
66	0.00	0.00
67	0.00	0.00
68	0.00	0.00
69	0.00	0.00
70	0.00	0.00
71	0.00	0.00
72	0.00	0.00
73	0.00	0.00
74	0.00	0.00
75	0.00	0.00
76	0.00	0.00
77	0.00	0.00
78	0.00	0.00
79	0.00	0.00
80	0.00	0.00
81	0.00	0.00
82	0.00	0.00
83	0.00	0.00
84	0.00	0.00
85	0.00	0.00
86	0.00	0.00
87	0.00	0.00
88	0.00	0.00
89	0.00	0.00
90	0.00	0.00
91	0.00	0.00
92	0.00	0.00
93	0.00	0.00
94	0.00	0.00
95	0.00	0.00
96	0.00	0.00
97	0.00	0.00
98	0.00	0.00
99	0.00	0.00
100	0.95	0.95

8. Input layer: 400, Hidden layer: 200, Epoch: 100

```
def model(hidden):  
    model = Sequential()  
    model.add(Dense(400, activation='tanh'))  
    model.add(Dense(hidden, activation='tanh'))  
    model.add(Dense(1, activation='sigmoid'))  
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])  
    return model  
  
def train(hidden):  
    model = model(hidden)  
    model.fit(X_train, y_train, epochs=10, validation_data=(X_test, y_test))  
    return model  
  
def test(hidden):  
    model = model(hidden)  
    model.evaluate(X_test, y_test)  
    return model
```

10 Epochs Training Results for 400 Input, 400 Hidden Nodes

Epoch	Train Accuracy	Test Accuracy
1	0.00	0.00
2	0.00	0.00
3	0.00	0.00
4	0.00	0.00
5	0.00	0.00
6	0.00	0.00
7	0.00	0.00
8	0.00	0.00
9	0.00	0.00
10	0.95	0.95

9. Input layer: 400, Hidden layer: 400, Epoch: 10


```
Epoch 15/15: 100%|#####| 1/1 [0:00<]
Epoch 15: training loss: 0.188, testing loss: 0.248
Epoch 15: training accuracy: 0.92, testing accuracy: 0.88
```

14. Input layer: 400, Hidden layer: 400, Epoch: 15

```
Epoch 50/50: 100%|#####| 1/1 [0:00<]
Epoch 50: training loss: 0.188, testing loss: 0.248
Epoch 50: training accuracy: 0.92, testing accuracy: 0.88
```

15. Input layer: 400, Hidden layer: 400, Epoch: 50

```

11 # Parameters
12 # Input layer: 400, Hidden layer: 400, Epoch: 100
13 # Training parameters
14 # Learning rate: 0.001, Batch size: 100, Number of epochs: 100
15 # Loss function: cross_entropy, Optimizer: Adam
16 # Model architecture: fully_connected
17 # Data loader: DataLoader
18 # Model: MLP
19 # Training process
20 # Training loop
21 # Evaluation process
22 # Save model
23 # Load model
24 # Test model
25 # Main function
26 # If __name__ == '__main__':
27 #     main()
28 # End of file

```

```

Epoch: 0/100, Loss: 0.0000, Accuracy: 0.0000
Epoch: 10/100, Loss: 0.0000, Accuracy: 0.0000
Epoch: 20/100, Loss: 0.0000, Accuracy: 0.0000
Epoch: 30/100, Loss: 0.0000, Accuracy: 0.0000
Epoch: 40/100, Loss: 0.0000, Accuracy: 0.0000
Epoch: 50/100, Loss: 0.0000, Accuracy: 0.0000
Epoch: 60/100, Loss: 0.0000, Accuracy: 0.0000
Epoch: 70/100, Loss: 0.0000, Accuracy: 0.0000
Epoch: 80/100, Loss: 0.0000, Accuracy: 0.0000
Epoch: 90/100, Loss: 0.0000, Accuracy: 0.0000
Epoch: 100/100, Loss: 0.0000, Accuracy: 0.0000

```

16. Input layer: 400, Hidden layer: 400, Epoch: 100

```

11 # Parameters
12 # Input layer: 400, Hidden layer: 800, Epoch: 10
13 # Training parameters
14 # Learning rate: 0.001, Batch size: 100, Number of epochs: 10
15 # Loss function: cross_entropy, Optimizer: Adam
16 # Model architecture: fully_connected
17 # Data loader: DataLoader
18 # Model: MLP
19 # Training process
20 # Training loop
21 # Evaluation process
22 # Save model
23 # Load model
24 # Test model
25 # Main function
26 # If __name__ == '__main__':
27 #     main()
28 # End of file

```

```

Epoch: 0/10, Loss: 0.0000, Accuracy: 0.0000
Epoch: 1/10, Loss: 0.0000, Accuracy: 0.0000
Epoch: 2/10, Loss: 0.0000, Accuracy: 0.0000
Epoch: 3/10, Loss: 0.0000, Accuracy: 0.0000
Epoch: 4/10, Loss: 0.0000, Accuracy: 0.0000
Epoch: 5/10, Loss: 0.0000, Accuracy: 0.0000
Epoch: 6/10, Loss: 0.0000, Accuracy: 0.0000
Epoch: 7/10, Loss: 0.0000, Accuracy: 0.0000
Epoch: 8/10, Loss: 0.0000, Accuracy: 0.0000
Epoch: 9/10, Loss: 0.0000, Accuracy: 0.0000
Epoch: 10/10, Loss: 0.0000, Accuracy: 0.0000

```

17. Input layer: 400, Hidden layer: 800, Epoch: 10

```

Epoch 11/100:
  train_loss: 0.0000
  train_acc: 0.0000
  val_loss: 0.0000
  val_acc: 0.0000
  
```

18. Input layer: 400, Hidden layer: 800, Epoch: 11

```

Epoch 12/100:
  train_loss: 0.0000
  train_acc: 0.0000
  val_loss: 0.0000
  val_acc: 0.0000
  
```

19. Input layer: 400, Hidden layer: 800, Epoch: 12


```
Epoch: 11 / 100
Loss: 0.0000
Accuracy: 100.00%
Time: 0:00:00
```

34. Input layer: 800, Hidden layer: 800, Epoch: 11

```
Epoch: 12 / 100
Loss: 0.0000
Accuracy: 100.00%
Time: 0:00:00
```

35. Input layer: 800, Hidden layer: 800, Epoch: 12


```

Epoch 10:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 1:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 2:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 3:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 4:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 5:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 6:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 7:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 8:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 9:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 10:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%

```

40. Input layer: 800, Hidden layer: 800, Epoch: 100

```

Epoch 10:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 1:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 2:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 3:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 4:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 5:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 6:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 7:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 8:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 9:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%
...
Epoch 10:
Training Loss: 0.0000
Validation Loss: 0.0000
Accuracy: 100.00%

```

41. Input layer: 800, Hidden layer: 1600, Epoch: 10

The screenshot shows a terminal window with a dark background. On the left, there is a column of code, likely Python, defining a neural network architecture and training process. On the right, the terminal output displays training progress for epoch 11. A small window titled 'Epoch 11' is overlaid on the output, showing a table of metrics:

Epoch	Train Loss	Val Loss	Train Acc	Val Acc
11	0.0000	0.0000	1.0000	1.0000

42. Input layer: 800, Hidden layer: 1600, Epoch: 11

This screenshot is similar to the one above, showing the terminal output for epoch 12. The 'Epoch 12' window displays the following metrics:

Epoch	Train Loss	Val Loss	Train Acc	Val Acc
12	0.0000	0.0000	1.0000	1.0000

43. Input layer: 800, Hidden layer: 1600, Epoch: 12

```
import tensorflow as tf
import tensorflow.keras as keras
import numpy as np

# Load data
train_data = tf.keras.datasets.mnist.load_data()
(x_train, y_train), (x_test, y_test) = train_data

# Preprocess data
x_train = x_train / 255.0
x_test = x_test / 255.0

# Build model
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(800, activation='relu'),
    keras.layers.Dense(1600, activation='relu'),
    keras.layers.Dense(10, activation='softmax')
])

# Compile model
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Train model
model.fit(x_train, y_train,
        validation_data=(x_test, y_test),
        epochs=13)
```

```
Epoch 1/13: 100% 10000/10000 [1.000s] - loss: 2.3042 - accuracy: 0.0000
Epoch 2/13: 100% 10000/10000 [1.000s] - loss: 1.7512 - accuracy: 0.0400
Epoch 3/13: 100% 10000/10000 [1.000s] - loss: 1.5000 - accuracy: 0.0800
Epoch 4/13: 100% 10000/10000 [1.000s] - loss: 1.3000 - accuracy: 0.1200
Epoch 5/13: 100% 10000/10000 [1.000s] - loss: 1.1500 - accuracy: 0.1600
Epoch 6/13: 100% 10000/10000 [1.000s] - loss: 1.0500 - accuracy: 0.2000
Epoch 7/13: 100% 10000/10000 [1.000s] - loss: 0.9800 - accuracy: 0.2400
Epoch 8/13: 100% 10000/10000 [1.000s] - loss: 0.9200 - accuracy: 0.2800
Epoch 9/13: 100% 10000/10000 [1.000s] - loss: 0.8700 - accuracy: 0.3200
Epoch 10/13: 100% 10000/10000 [1.000s] - loss: 0.8300 - accuracy: 0.3600
Epoch 11/13: 100% 10000/10000 [1.000s] - loss: 0.7900 - accuracy: 0.4000
Epoch 12/13: 100% 10000/10000 [1.000s] - loss: 0.7600 - accuracy: 0.4400
Epoch 13/13: 100% 10000/10000 [1.000s] - loss: 0.7300 - accuracy: 0.4800
```

44. Input layer: 800, Hidden layer: 1600, Epoch: 13

```
import tensorflow as tf
import tensorflow.keras as keras
import numpy as np

# Load data
train_data = tf.keras.datasets.mnist.load_data()
(x_train, y_train), (x_test, y_test) = train_data

# Preprocess data
x_train = x_train / 255.0
x_test = x_test / 255.0

# Build model
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(800, activation='relu'),
    keras.layers.Dense(1600, activation='relu'),
    keras.layers.Dense(10, activation='softmax')
])

# Compile model
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Train model
model.fit(x_train, y_train,
        validation_data=(x_test, y_test),
        epochs=14)
```

```
Epoch 1/14: 100% 10000/10000 [1.000s] - loss: 2.3042 - accuracy: 0.0000
Epoch 2/14: 100% 10000/10000 [1.000s] - loss: 1.7512 - accuracy: 0.0400
Epoch 3/14: 100% 10000/10000 [1.000s] - loss: 1.5000 - accuracy: 0.0800
Epoch 4/14: 100% 10000/10000 [1.000s] - loss: 1.3000 - accuracy: 0.1200
Epoch 5/14: 100% 10000/10000 [1.000s] - loss: 1.1500 - accuracy: 0.1600
Epoch 6/14: 100% 10000/10000 [1.000s] - loss: 1.0500 - accuracy: 0.2000
Epoch 7/14: 100% 10000/10000 [1.000s] - loss: 0.9800 - accuracy: 0.2400
Epoch 8/14: 100% 10000/10000 [1.000s] - loss: 0.9200 - accuracy: 0.2800
Epoch 9/14: 100% 10000/10000 [1.000s] - loss: 0.8700 - accuracy: 0.3200
Epoch 10/14: 100% 10000/10000 [1.000s] - loss: 0.8300 - accuracy: 0.3600
Epoch 11/14: 100% 10000/10000 [1.000s] - loss: 0.7900 - accuracy: 0.4000
Epoch 12/14: 100% 10000/10000 [1.000s] - loss: 0.7600 - accuracy: 0.4400
Epoch 13/14: 100% 10000/10000 [1.000s] - loss: 0.7300 - accuracy: 0.4800
Epoch 14/14: 100% 10000/10000 [1.000s] - loss: 0.7000 - accuracy: 0.5200
```

45. Input layer: 800, Hidden layer: 1600, Epoch: 14


```
def model():  
    model = Sequential([  
        Dense(800, activation='tanh'),  
        Dense(1600, activation='tanh'),  
        Dense(10, activation='softmax')  
    ])  
    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])  
    model.fit(x_train, y_train, epochs=15, validation_data=(x_val, y_val))  
    return model
```

```
Epoch 1/15: 100% 10/10 [1.000s] - loss: 0.6172 - accuracy: 0.1000  
Epoch 2/15: 100% 10/10 [1.000s] - loss: 0.5852 - accuracy: 0.1000  
Epoch 3/15: 100% 10/10 [1.000s] - loss: 0.5532 - accuracy: 0.1000  
Epoch 4/15: 100% 10/10 [1.000s] - loss: 0.5212 - accuracy: 0.1000  
Epoch 5/15: 100% 10/10 [1.000s] - loss: 0.4892 - accuracy: 0.1000  
Epoch 6/15: 100% 10/10 [1.000s] - loss: 0.4572 - accuracy: 0.1000  
Epoch 7/15: 100% 10/10 [1.000s] - loss: 0.4252 - accuracy: 0.1000  
Epoch 8/15: 100% 10/10 [1.000s] - loss: 0.3932 - accuracy: 0.1000  
Epoch 9/15: 100% 10/10 [1.000s] - loss: 0.3612 - accuracy: 0.1000  
Epoch 10/15: 100% 10/10 [1.000s] - loss: 0.3292 - accuracy: 0.1000  
Epoch 11/15: 100% 10/10 [1.000s] - loss: 0.2972 - accuracy: 0.1000  
Epoch 12/15: 100% 10/10 [1.000s] - loss: 0.2652 - accuracy: 0.1000  
Epoch 13/15: 100% 10/10 [1.000s] - loss: 0.2332 - accuracy: 0.1000  
Epoch 14/15: 100% 10/10 [1.000s] - loss: 0.2012 - accuracy: 0.1000  
Epoch 15/15: 100% 10/10 [1.000s] - loss: 0.1692 - accuracy: 0.1000  
Total time taken: 15.000s
```

46. Input layer: 800, Hidden layer: 1600, Epoch: 15

```
def model():  
    model = Sequential([  
        Dense(800, activation='tanh'),  
        Dense(1600, activation='tanh'),  
        Dense(10, activation='softmax')  
    ])  
    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])  
    model.fit(x_train, y_train, epochs=50, validation_data=(x_val, y_val))  
    return model
```

```
Epoch 1/50: 100% 10/10 [1.000s] - loss: 0.6172 - accuracy: 0.1000  
Epoch 2/50: 100% 10/10 [1.000s] - loss: 0.5852 - accuracy: 0.1000  
Epoch 3/50: 100% 10/10 [1.000s] - loss: 0.5532 - accuracy: 0.1000  
Epoch 4/50: 100% 10/10 [1.000s] - loss: 0.5212 - accuracy: 0.1000  
Epoch 5/50: 100% 10/10 [1.000s] - loss: 0.4892 - accuracy: 0.1000  
Epoch 6/50: 100% 10/10 [1.000s] - loss: 0.4572 - accuracy: 0.1000  
Epoch 7/50: 100% 10/10 [1.000s] - loss: 0.4252 - accuracy: 0.1000  
Epoch 8/50: 100% 10/10 [1.000s] - loss: 0.3932 - accuracy: 0.1000  
Epoch 9/50: 100% 10/10 [1.000s] - loss: 0.3612 - accuracy: 0.1000  
Epoch 10/50: 100% 10/10 [1.000s] - loss: 0.3292 - accuracy: 0.1000  
Epoch 11/50: 100% 10/10 [1.000s] - loss: 0.2972 - accuracy: 0.1000  
Epoch 12/50: 100% 10/10 [1.000s] - loss: 0.2652 - accuracy: 0.1000  
Epoch 13/50: 100% 10/10 [1.000s] - loss: 0.2332 - accuracy: 0.1000  
Epoch 14/50: 100% 10/10 [1.000s] - loss: 0.2012 - accuracy: 0.1000  
Epoch 15/50: 100% 10/10 [1.000s] - loss: 0.1692 - accuracy: 0.1000  
Epoch 16/50: 100% 10/10 [1.000s] - loss: 0.1372 - accuracy: 0.1000  
Epoch 17/50: 100% 10/10 [1.000s] - loss: 0.1052 - accuracy: 0.1000  
Epoch 18/50: 100% 10/10 [1.000s] - loss: 0.0732 - accuracy: 0.1000  
Epoch 19/50: 100% 10/10 [1.000s] - loss: 0.0412 - accuracy: 0.1000  
Epoch 20/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 21/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 22/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 23/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 24/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 25/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 26/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 27/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 28/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 29/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 30/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 31/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 32/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 33/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 34/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 35/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 36/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 37/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 38/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 39/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 40/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 41/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 42/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 43/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 44/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 45/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 46/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 47/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 48/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 49/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Epoch 50/50: 100% 10/10 [1.000s] - loss: 0.0092 - accuracy: 0.1000  
Total time taken: 50.000s
```

47. Input layer: 800, Hidden layer: 1600, Epoch: 50


```
Epoch 50: Train Accuracy: 0.85, Train Loss: 0.15, Val Accuracy: 0.82, Val Loss: 0.18
```

50. Input layer: 1200, Hidden layer: 600, Epoch: 11

```
Epoch 51: Train Accuracy: 0.86, Train Loss: 0.14, Val Accuracy: 0.83, Val Loss: 0.17
```

51. Input layer: 1200, Hidden layer: 600, Epoch: 12

The screenshot shows a terminal window with a dark background. On the left, there is a column of code with various colors (red, green, blue, yellow). On the right, there is a column of output text. A small white window is overlaid on the right side, displaying a table of data. The table has 10 columns and 10 rows, with values ranging from 0.000000 to 0.999999. The text in the terminal includes "Epoch 11" and "Input layer: 1200, Hidden layer: 1200, Epoch: 11".

58. Input layer: 1200, Hidden layer: 1200, Epoch: 11

The screenshot shows a terminal window with a dark background. On the left, there is a column of code with various colors (red, green, blue, yellow). On the right, there is a column of output text. A small white window is overlaid on the right side, displaying a table of data. The table has 10 columns and 10 rows, with values ranging from 0.000000 to 0.999999. The text in the terminal includes "Epoch 12" and "Input layer: 1200, Hidden layer: 1200, Epoch: 12".

59. Input layer: 1200, Hidden layer: 1200, Epoch: 12

The screenshot shows a terminal window with a dark background. On the left, there is a column of code, likely Python, defining a neural network architecture. On the right, there is a window titled "Terminal Output" showing the execution results. The output includes the following text:

```
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
```

64. Input layer: 1200, Hidden layer: 1200, Epoch: 100

The screenshot shows a terminal window with a dark background, similar to the one above. It displays code on the left and terminal output on the right. The output text is as follows:

```
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
Epoch 100: Loss: 0.0000, Accuracy: 100.00%
```

65. Input layer: 1200, Hidden layer: 2400, Epoch: 10

