

# Implementasi Deep Learning Untuk Rekomendasi Aplikasi E-learning Yang Tepat Untuk Pembelajaran Jarak Jauh

Wowon Priatna <sup>1,\*</sup>, Rakhmat Purnomo <sup>1</sup>, Tri Dharma Putra <sup>1</sup>

<sup>1</sup> Fakultas Ilmu Komputer; Universitas Bhayangkara Jakarta Raya; Jl. Perjuangan 81, Marga Mulya, Bekasi Utara; 02188955882; e-mail: [wowon.priatna@dsn.ubharajaya.ac.id](mailto:wowon.priatna@dsn.ubharajaya.ac.id), [rakhmat.purnomo@dsn.ubharajaya.ac.id](mailto:rakhmat.purnomo@dsn.ubharajaya.ac.id), [tridharma.putra@dsn.ubharajaya.ac.id](mailto:tridharma.putra@dsn.ubharajaya.ac.id)

\* Korespondensi: e-mail: [wowon.priatna@dsn.ubharajaya.ac.id](mailto:wowon.priatna@dsn.ubharajaya.ac.id)

Submitted: 26/05/2021; Revised: 07/06/2021; Accepted: 11/06/2021; Published: 30/09/2021

---

## Abstract

*The purpose of this study is to recommend e-learning applications that are appropriate for use in online learning in college environments. The large number of e-learning platforms used by lecturers for online lecture activities results in students being forced to use several e-learning applications depending on the lecturer who teaches the courses taken, for the university also finally gives lecturers policies for distance learning reports each finished giving the material. In this study the data collection method began by taking data from the faculty to find out which e-learning applications were widely used by lecturers, then distributing questionnaires to students and lecturers who used the e-learning application to measure the E-learning application with the e-learning criteria. Appropriate. The data is then processed into a dataset. The algorithm used in implementing deep learning is Artificial Neural Network (ANN). For the implementation of ANN, 27 variables were determined from the e-learning criteria and 1 target. In this ANN stage, prediction was used with classifications based on preparation, training, learning, evaluation and prediction using the python programming. The results obtained in this study that the Moodle application gets the highest score with an accuracy of 97% to be used as a recommendation for e-learning applications that are appropriate for universities to conduct online lectures.*

**Keywords:** Artificial Neural Network, Deep Learning, E-learning, Online Teaching, Python

## Abstrak

Tujuan penelitian ini adalah untuk rekomendasi aplikasi *e-learning* yang tepat untuk digunakan dalam pembelajaran online dilingkungan perguruan tinggi. Banyaknya platform *e-learning* yang digunakan oleh dosen-dosen untuk kegiatan kuliah *online* berakibat mahasiswa dalam belajar terpaksa menggunakan beberapa aplikasi *e-learning* tergantung dari dosen yang mengajar mata kuliah yang diambil, untuk pihak universitas juga akhirnya memberikan kebijakan dosen-dosen untuk laporan pembelajaran jarak setiap selesai memberikan materi. Dalam penelitian ini metode pengumpulan data dimulai dengan mengambil data dari fakultas untuk mengetahui aplikasi *e-learning* yang banyak digunakan oleh para dosen, selanjutnya membagikan kuisioner kepada mahasiswa dan dosen yang menggunakan aplikasi *e-learning* untuk mengukur aplikasi *e-learning* tersebut dengan kriteria *e-learning* yang sesuai. Data kemudian diolah dijadikan dataset. Algoritma yang digunakan dalam implementasi deep learning ini adalah Artificial Neural Network (ANN). Untuk implementasi ANN ditentukan 27 variabel yang didapat dari kriteria *e-learning* dan 1 target, dalam tahapan ANN ini menggunakan prediksi dengan klasifikasi berdasarkan preposisi training, learning, evaluation dan prediction dengan menggunakan pemrograman *python*. Hasilnya yang didapat penelitian ini aplikasi moodle mendapatkan nilai tertinggi dengan akurasi 97% untuk dijadikan rekomendasi aplikasi *e-learning* yang tepat digunakan untuk perguruan tinggi dalam melakukan perkuliahan *online*.

**Kata kunci:** *Artificial Neural Network, Deep Learning, E-learning, Online Teaching, Python*

## 1. Pendahuluan

Dengan perkembangan teknologi di seluruh dunia dan dengan booming akses informasi, pembelajaran jarak jauh telah menjadi lebih populer karena memungkinkan individu untuk mempelajari keterampilan baru tanpa kehadiran mentor secara fisik dalam mengajar. Pembelajaran jarak jauh merupakan bagian dari proses e-learning karena memungkinkan orang untuk berbagi pengetahuan meskipun ada batasan dan batasan geografis. Seperti disebutkan sebelumnya, e-learning dapat didefinisikan sebagai akses ke kurikulum pendidikan di luar kelas tradisional dengan memanfaatkan teknologi elektronik (Moubayed, Injadat, Nassif, Lutfiyya, & Shami, 2018).

Sejak diberlakukannya larangan kegiatan belajar mengajar di sekolah, di universitas dan sejenisnya oleh mendikbud untuk mencegah penularan virus covid 19 maka dengan itu diberlakukan kuliah daring. Sekolah dan universitas dalam melakukan daring atau e-learning memutuskan untuk menggunakan aplikasi e-learning baik yang gratis, berbayar dipasaran atau membuat aplikasi e-learning sendiri.

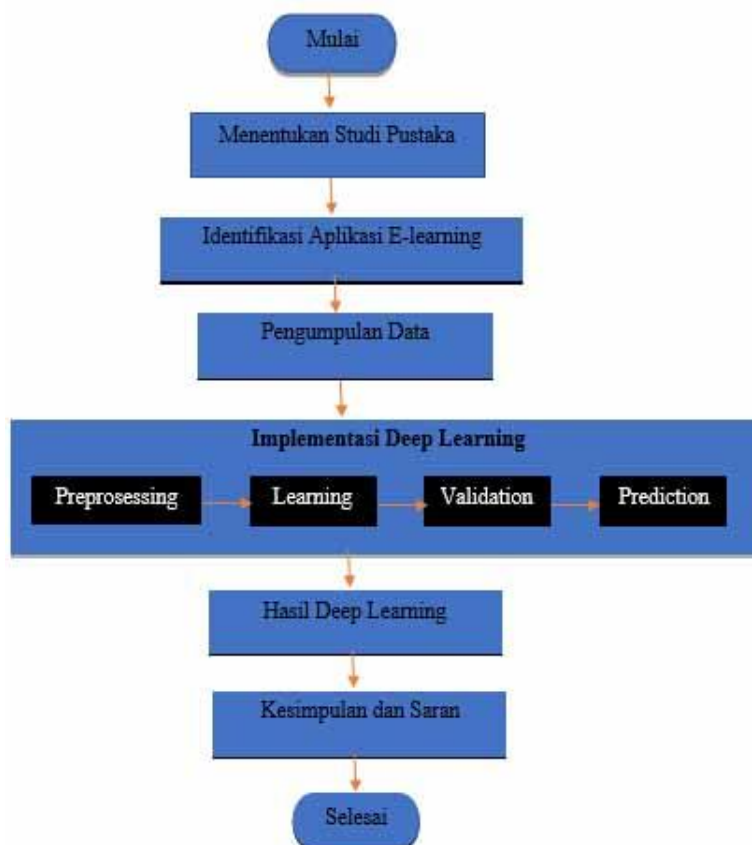
Untuk memilih aplikasi e-learning yang tepat untuk mendukung pembelajaran harus memperhatikan dari fitur-fitur, kecepatan mengakses aplikasi, kemudahan, user friendly, support multi platform. Maka itu perlu dilakukan klasifikasi aplikasi e-learning yang tepat untuk mendukung kegiatan pembelajaran. Salah satu metode untuk melakukan klasifikasi adalah menggunakan teknik deep learning. Deep Learning dapat memberikan rekomendasi klasifikasi berdasarkan input output pembelajaran (Liu & WU, 2017). Deep learning dapat memberikan klasifikasi dari model integrasi pembelajaran (Fakhfakh, Ben, & Ben, 2017)

Deep learning untuk klasifikasi gambar hewan menggunakan (Chauhan & Ram, 2018), prediksi kesehatan jembatan menggunakan algoritma Artificial Neural Network jenis deep learning (Reza, 2017). Deep learning dapat digunakan untuk rekomendasi system (Shiddieqy, Hariadi, & Adiono, 2017), rekomendasi tantangan saat ini (Fakhfakh et al., 2017). Deep learning dapat digunakan untuk prediksi perkembangan e-learning dimasa depan (Muniasamy & Alasiry, 2020). Deep learning dapat digunakan juga untuk prediksi hasil belajar di lingkungan pendidikan tinggi.

Dari latar belakang masalah dan dari beberapa penelitian diatas maka tujuan dalam penelitian ini adalah untuk memprediksi aplikasi e-learning yang tepat sebagai rekomendasi untuk perkuliahan jarak jauh di lingkungan Perguruan Tinggi.

## 2. Metode Penelitian

Terdapat beberapa tahapan dalam melakukan implementasi deep learning terhadap rekomendasi aplikasi e-learning yang tepat untuk pendidikan jarak jauh ditunjukkan pada gambar 1.



Sumber: Hasil Pengolahan Data (2021)

Gambar 1. Tahapan Penelitian

### 2.1 Menentukan studi pustaka

Menentukan Studi pustaka untuk mendapatkan referensi untuk penerapan deep learning untuk mendapatkan algoritma yang tepat untuk memberikan rekomendasi e-learning untuk pengguna

### 2.2 Identifikasi Aplikasi E-learning

Identifikasi e-learning adalah indentifikasi aplikasi e-learning yang digunakan dilingkungan universitas oleh dosen-dosen dalam melakukan pembelajaran online di masa pademi covid 19. Identifikasi aplikasi e-learning digunakan 4 jenis aplikasi yang paling banyak digunakan dalam pembelajaran online diataranya moodle, google classroom, slack dan edmodo.

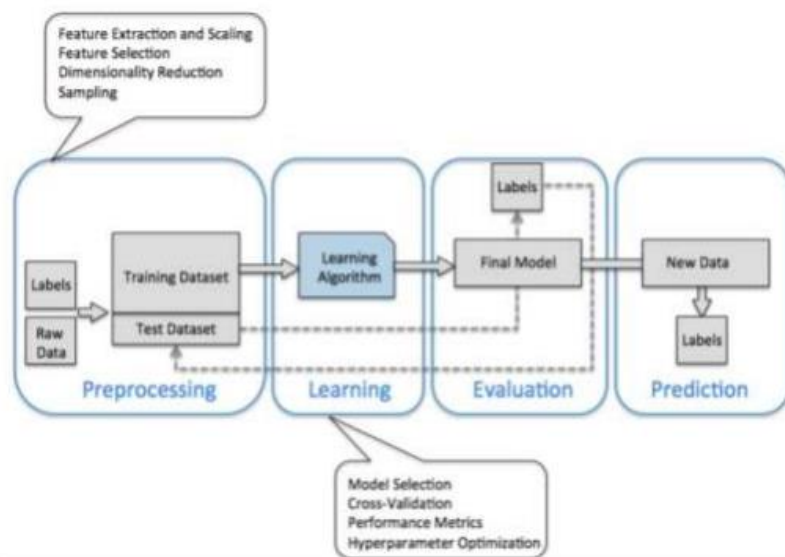
### 2.3 Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penelitian ini adalah membagikan kuisioner dengan responden dosen dan Mahasiswa Fakultas Ilmu Komputer Universitas Bhayangkara Jakarta Raya.

### 2.4 Implementasi Deep Learning

Implementasi deep learning adalah proses untuk mengolah data untuk prediksi aplikasi e-learning yang tepat menggunakan algoritma artificial neural network. Dimana prediksi dilakukan berdasarkan data yang telah didapat dan bagaimana system dapat

mengklasifikasikannya. Tahapan prediksi menggunakan klasifikasi diantaranya Preprocessing, Learning, Evaluation, Prediction (Raschka, 2015), tahapan klasifikasi ditunjukkan oleh gambar 2.



Sumber: (Raschka, 2015)

Gambar 2. Tahapan Klasifikasi

Dari gambar 2 dapat dijelaskan prosesnya, diantaranya *pre-processing*, *learning* atau *training*, *validation* dan *prediction*. *Pre-processing* pada tahapan ini adalah mengelompokan data dengan membagi data menjadi training, validasi dan data untuk tes. Training ada digunakan untuk menentukan bobot ( $w$ ), validasi data dalam penelitian data adalah untuk parameter tuning sedangkan data tes dalah untuk memvalidasi apakah model yang telah dibangun sudah optimal. *Learning* atau *training* dilakukan dengan menciptakan model untuk membuat pola dengan mempelajari data. Untuk melakukan prediktif harus melakukan training data menggunakan algoritma machine learning, dimana algoritma pembelajaran yang dipakai adalah *Artificial Neural Network*. *Validation* yaitu dengan mengevaluasi model yang telah diciptakan pada tahap learning dan training. Memastikan bahwa model prediktif sesuai dengan data dan model siap untuk digunakan. *Prediction* pada tahap ini adalah model yang telah didapatkan untuk memprediksi tes data

## 2.5 Instrumen penelitian

Penelitian ini menggunakan rubric untuk evaluasi *tool e-learning* dalam artikel (Anstey & Watson, 2018) untuk menentukan kriteria aplikasi e-learning yang sesuai untuk online e-learning. Instrument penelitian ditunjukkan pada tabel 1.

Tabel 1 Instrumen Penelitian

Dimensi	Indikator	Pernyataan
Functionality	<i>Scale.</i>	Menguji skala (Chou, 2014).
	<i>Ease of Use</i>	Aplikasi mudah digunakan (Dodun et al., 2015).
	<i>Tech Support / Help Availability.</i>	Sebagai pendukung SDM (Nawaz & Zubair Khan, 2012).
	<i>Hypermediality</i>	Mendukung komunikasi (Anstey & Watson, 2018).

Dimensi	Indikator	Pernyataan
Accessibility	Accessibility Standards	Aplikasi ada informasi terperinci (Bühler & Fisseler, 2007).
	Accessibility Initiative.	Tersedia panduan akses (Bühler & Fisseler, 2007).
	User-Focused Participation	user dapat berpartisipasi diplatform yang sama (Phipps & Kelly, 2006)
Technical	Required Equipment	Fasilitas lengkap (Kisworo, 2016)
	Learning Management System	Software terintegrasikan (Prof. Indira & MS. Sakhi, 2017).
	Desktop/Laptop Operating Systems and Browser	Aplikasi bisa berjalan di pc atau mobile (Kumar Basak, Wotto, & Bélanger, 2018).
Mobile Design	Additional Downloads	Tersedia fitur download (Baidawi & Sandi, 2013).
	Access	Dengan mudah dapat melakukan akses e-learning dengan perangkat ponsel (Moubayed et al., 2018).
	Functionality	Semua fitur berfungsi dengan baik (Chen & He, 2013).
Privacy, Data Protection, and Rights	Offline Access	Aplikasi dapat diakses secara offline (Akcaoglu & Lee, 2016)
	Sign Up / Sign In	Terdapat login dan registrasi yang kuat (Maeda & Ono, 2019).
	Data Privacy and Ownership	Aplikasi aman dan menjaga integritas (Richardson et al., 2015)
Social Presence	Archiving, Saving, and Exporting Data	Pengguna dapat mengarsipkan, menyimpan, atau mengimpor dan mengekspor data (Rapanta, Botturi, Goodyear, Guàrdia, & Koole, 2020).
	Collaboration	Pengguna dapat kolaborasi. (Gregory & Bannister-Tyrrell, 2017).
	User Accountability.	Tersedia fitur group kelas (Akcaoglu & Lee, 2016).
Teaching Presence	Diffusion	Ada jalur untuk komunikasi (Akcaoglu & Lee, 2016).
	Facilitation	Tersedia fasilitas diskusi. (Richardson et al., 2015).
	Customization	Tersedia fitur perubahan (Rapanta et al., 2020).
Cognitive Presence	Learning Analytics	Tersedia fitur untuk evaluasi (Rapanta et al., 2020).
	Enhancement of Cognitive Task(s)	Tersedia untuk menilai kognisi (Gregory & Bannister-Tyrrell, 2017).
	Higher-Order Thinking	Aplikasi dapat menjadi inspirasi (Lee, 2014).

Sumber: Hasil Pengolahan Data (2021)

## 2.6 Teknik Pengumpulan Data

Penelitian ini menggunakan angket yang dikirim menggunakan google form kepada mahasiswa dan dosen dilingkungan univristas bhayangkara Jakarta Raya. Angket berisikan pertanyaan mengenai function *Accessibility*, *technical*, *mobile design*, *Privacy*, *Data Protection*, *Social Presence*, *Teaching Presence*, *Cognitive Presence* pada beberapa aplikasi e-learning

yang digunakan. Didalam kuisiener digunakan skala linkert untuk masing-masing responden. Kriteria penilaian responden dapat dilihat pada table 2.

Tabel 2. Bobot Jawaban Responden

Jawaban	Nilai
Tidak ada	1
Ragu-Ragu	2
Ada	3

Sumber: Hasil Pengolahan Data (2021)

### 3. Hasil dan Pembahasan

#### 3.1. Profil Data

Adapun data frekuensi hasil kuisiener yang didapat dari 197 sampel penelitian untuk mengukur aplikasi *slack,moodle,google class room* dan *Edmodo*. Sehingga dari hasil kuisiener digunakan untuk dataset, dimana setiap indicator dijadikan sebagai variable untuk input. Untuk variabel yang digunakan dalam penelitian ini dapat dilihat pada tabel 3.

Table 3. Percobaan penggunaan Hidden Layer

Variabel	Keterangan
X1	Scale
X2	Ease of Use
X3	Tech Support
X4	Hypermediality
X5	Accessibility Standards
X6	Accessibility Initiative
X7	User Focused
X8	Participation
X9	Required Equipment
X10	Learning Management System
X11	Dekstop/Laptop OS Browser
X12	Additional Downloads
X13	Access
X14	Fuctionality
X15	Offline Acess
X16	Sign up/ Sign In
X17	Data Privacy
X18	Archiving, saving and Exporting Data
X19	Collaboration
X20	User Accountability
X21	Disffusion
X22	Facilitation
X23	Customization
X24	Learning Analytics
X25	Enhancement Cognitive Task
X26	Higher-Order Thinking
X27	Metacognitive
Y	Target

Sumber: Hasil Pengolahan Data (2021)

#### 3.2. Mendefinisikan Input dan Target

Untuk input data yang akan diolah oleh algoritma ANN menggunakan 27 variabel x yang sudah didefinisikan tahap sebelumnya. Sedangkan untuk mendefinisikan target Hasil yang diharapkan dari penelitian ini adalah rekomendasi aplikasi e-learning yang tepat dengan memprediksi dari data yang telah didapatkan. Hasil yang dimaksudkan adalah Jika target

bernilai 0 berarti aplikasi tidak direkomendasikan sedangkan Jika target bernilai 1 berarti aplikasi direkomendasikan

### 3.2. Preprocessing

Pada tahap persiapan ini pengolahan data menggunakan Microsoft excel, menghitung data hasil kuisioner, selanjutnya untuk digunakan untuk kebutuhan pengolahan algoritam didalam pemograman python file disimpan dengan format csv. Dimana dalam tahapan ini data di import untuk dijadikan data array dan dataframe menggunakan library machine learning numpy dan pandas. Untuk tahapan preprocessing yang diolah menggunakan python dapat dilihat listing program pada gambar 3.

```
1 # Importing Data
2
3 df = pd.read_csv("dataset/GCR.csv")
4 data = df.copy()
5 df # Mention no of rows to be displayed from the top in the argument
```

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	...	X19	X20	X21	X22	X23	X24	X25	X26	X27	Target
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	3.0	3.0	3.0	3.0	3.0	3.0	1.0	3.0	3.0	3.0	...	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	1.0	1.0
2	3.0	3.0	3.0	3.0	3.0	3.0	1.0	3.0	3.0	3.0	...	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	1.0	1.0
3	3.0	3.0	3.0	3.0	3.0	3.0	1.0	3.0	3.0	3.0	...	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	1.0	1.0
4	3.0	3.0	3.0	1.0	3.0	1.0	3.0	3.0	3.0	1.0	...	1.0	1.0	1.0	1.0	1.0	3.0	3.0	3.0	1.0	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
193	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	...	3.0	3.0	1.0	1.0	3.0	3.0	3.0	3.0	1.0	1.0
194	3.0	3.0	3.0	3.0	3.0	1.0	3.0	3.0	3.0	1.0	...	3.0	3.0	1.0	1.0	3.0	3.0	3.0	3.0	1.0	1.0
195	3.0	3.0	3.0	3.0	1.0	3.0	1.0	3.0	3.0	1.0	...	1.0	1.0	1.0	3.0	3.0	1.0	1.0	3.0	1.0	1.0
196	3.0	3.0	3.0	3.0	1.0	1.0	3.0	3.0	3.0	1.0	...	1.0	3.0	1.0	3.0	3.0	3.0	3.0	3.0	1.0	1.0
197	3.0	3.0	3.0	3.0	3.0	1.0	1.0	3.0	3.0	3.0	...	3.0	3.0	1.0	3.0	3.0	3.0	3.0	3.0	1.0	1.0

198 rows x 28 columns

Sumber: Hasil Penelitian (2021)

Gambar 3. Tahap Preprocessing

#### 3.2.1 Pemilihan data

Pada proses pemilihan data adalah memisahkan variabel dengan target, disini attribute x1 sampai x27 adalah variabel X sedangkan attribute Target adalah variabel y sebagai output. Listing program untuk pemilihan data ditunjukkan pada gambar 4.

```
In [7]: 1 X = df.iloc[:, 0:27]
        2 y = df.iloc[:, 27]
```

Sumber: Hasil Penelitian (2021)

Gambar 4. Listing Program Untuk Pemilihan Data

#### 3.2.2 Feature Extraction

Dalam proses krasifikasi algoritma pembelajaran membutuhkan pemisahan data yang bertujuan untuk mencocokkan data sample kepada class/label tertentu (Raschka, 2015). Listing program python untuk Proses pemisahan ditunjukkan pada gambar 5.

```
1 # Splitting the dataset into the Training set and Test set
2 from sklearn.model_selection import train_test_split
3 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

Sumber: Hasil Penelitian (2021)

Gambar 5. Listing Program untuk Feature Extraction

Dari gambar 5 pemisahan data training dimulai dengan import library machine learning `sklearn_model_selection`. Data training dengan variable `X_train`, target adalah `y_train` dan untuk uji model menggunakan `x_test` dan `y_test`.

### 3.2.3 Feature Scalling

Data yang diinput hasil dari kuisisioner menunjukkan nilai angka 1 sampai 3. Pada tahapan ini adalah proses normalisasi untuk mendapatkan nilai standar deviasi yang tidak bervariasi. Rentan nilai yang bervariasi akan mengakibatkan error dalam yang tinggi, algoritma learning akan melakukan pembelajaran dengan memprioritaskan error rendah (Raschka, 2015). Gambar 6 adalah proses melakukan feature scalling.

```
: 1 # Feature Scalling
2 from sklearn.preprocessing import StandardScaler
3 sc = StandardScaler()
4 X_train = sc.fit_transform(X_train)
5 X_test = sc.transform(X_test)
```

Sumber: Hasil Penelitian (2021)

Gambar 6. Listing Program Untuk Feature Scalling

### 3.2.4 Training, Validation, Test Split

Pada tahapan ini adalah menentukan nilai training dari jumlah data 197 sample digunakan untuk proses training adalah sebanyak 105 sample, untuk test sebanyak 42 sample dan validasi adalah 52 sample dimana prosesnya dilakukan secara random. Proses training, validation dan test split dalam python dapat dilihat pada gambar 4.

### 3.3. Learning

Pada tahapan ini data hasil pembejaraan dilakukan dengan algoritma pembelajaran ditampilkan. Tahapan ini mempelajari dengan melakukan training data untuk menghasilkan akurasi dalam memprediksi aplikasi yang tepat untuk digunakan dalam e-learning. Berikut tahapan metode dalam learning.

#### 3.3.1 Training, Validation, Test Split

Kriteria untuk membangun pembelajaran dengan nerural network adalah menggunakan Input layer sebanyak 27 neuron, output layer sebanyak 1 neuron, fungsi aktivasi hidden layer menggunakan relu, fungsi aktivasi untuk output adalah sigmoid. Parameter untuk melakukan training melakukan beberapa percobaan dalam membangun arsitektur artificial Neural Network



(ANN) dengan menginput beberapa neuron. Berikut tabel 4 menampung hasil percobaan menggunakan neuron.

Table 4. Percobaan menggunakan beberapa Neuron

Jumlah Neuron	Accuracy
10	96%
15	95%
20	100%
50	85%

Sumber: Hasil Pengolahan Data (2021)

Dari tabel 4 akurasi tertinggi yang dihasilkan dalam penelitian ini adalah jumlah neuron. Selanjutnya melakukan percobaan dengan menggunakan beberapa hidden layer. Percobaan penggunaan hidden layer dapat dilihat pada tabel 5.

Table 5. Percobaan penggunaan Hidden Layer

Jumlah Hidden Layer	Training Accuracy
2	94,29%
3	97,6%
4	91,8%

Sumber: Hasil Pengolahan Data (2021)

Dari hasil percobaan yang tampak pada tabel 5 menunjukkan arsitektur ANN yang memiliki nilai training tertinggi adalah menggunakan 3 hidden layer dengan akurasi 97.6%. Untuk membangun deep learning menggunakan *neural network* dalam penelitian ini adalah menggunakan pemrograman python dimana ANN jenis deep learning membutuhkan library keras. Berikut list program dapat dilihat pada gambar 7.

```
1 import keras
2 from keras.models import Sequential
3 from keras.layers import Dense, Dropout, Activation
4 from keras.layers import LeakyReLU, PReLU, ELU
5 from keras.activations import relu, sigmoid
6 from keras.wrappers.scikit_learn import KerasClassifier
7
8 # Initialising the ANN
9 classifier = Sequential()
10
11 # Adding the input Layer and the first hidden Layer
12 classifier.add(Dense(units = 6, kernel_initializer = 'uniform', activation = 'relu', input_dim = 27))
13 # classifier.add(Dropout(p = 0.1))
14
15 # Adding the second hidden Layer
16 classifier.add(Dense(units = 6, kernel_initializer = 'uniform', activation = 'relu'))
17 # classifier.add(Dropout(p = 0.1))
18
19 # Adding the output Layer
20 classifier.add(Dense(units = 1, kernel_initializer = 'uniform', activation = 'sigmoid'))
21
22 # Compiling the ANN
23 classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
24
25 # Fitting the ANN to the Training set
26 model=classifier.fit(X_train, y_train, validation_split=0.33, batch_size = 10, epochs = 100)
```

Sumber: Hasil Pengolahan Data(2021)

Gambar 7. Listing Program Untuk Membangun Model ANN

Hasil yang didapatkan setelah menjalankan listing diatas adalah didapatkan accuracy tertinggi 0.9429 atau 94.29% untuk fase learning training.

### 3.4. Evaluation

Pada evaluation adalah melakukan evaluasi dari beberapa parameter yang digunakan untuk mencapai hasil maximal untuk memprediksi aplikasi e-learning yang direkomendasikan.

Parameter yang diuji adalah percobaan penggunaan Dropout, hasil dari percobaan penggunaan beberapa Dropout dapat dilihat pada table 6.

Table 6. Percobaan penggunaan beberapa Dropout

Dropout	Validasi Accuracy
0.1	0.9231
0.01	0.943
0.001	0.924

Sumber: Hasil Pengolahan Data (2021)

Hasil didapatkan sesuai yang tampak pada tabel 6 adalah drop out dengan nilai 0.01 memiliki validasi tertinggi. Parameter lengkap untuk membangun ANN terbaik dengan epoch, 100, optimize: adam, Dropout: 0.01, Loss function: binary\_crossentropy dan Batch size: 128.

### 3.5. Prediction

Tahap akhir dari ANN adalah prediksi, hasil yang didapatkan berupa klasifikasi hasil akhir dari tes data yang telah dilakukan learning. Berikut listing program untuk prediksi dapat dilihat pada gambar 8.

```
1 y_pred = classifier.predict(X_test)
2 y_pred = (y_pred > 0.5)
3
4 print(y_pred)
```

Sumber: Hasil Pengolahan Data (2021)

Gambar 8. Listing Program Untuk Prediksi

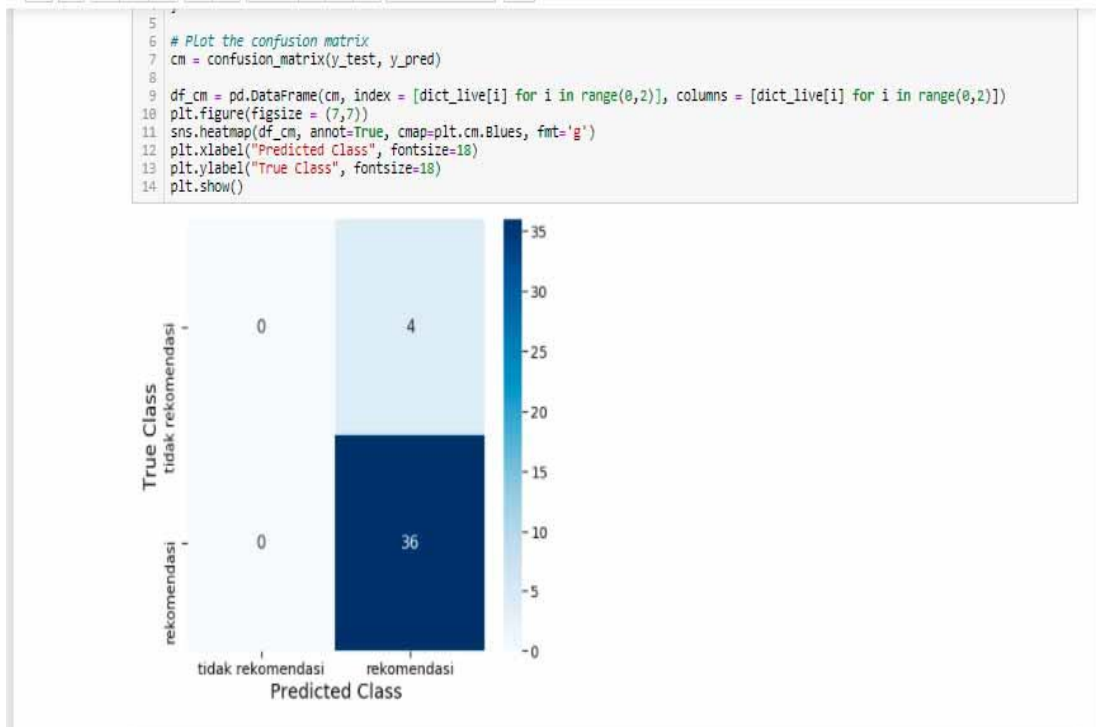
Dari hasil learning dilanjutkan dengan menggunakan confusion matrix untuk menguji akurasi dari hasil learning. Hasil ditunjukkan pada gambar 9, dimana hasil selama learning banyak terjadi error karena itu saat uji confusion matrix mendapatkan 4 data yang ditampung dalam matrix.

```
In [68]: 1 # Making the Confusion Matrix
2 from sklearn.metrics import confusion_matrix
3 print(confusion_matrix(y_test, y_pred))
4
5
[[ 0  4]
 [ 0 36]]
```

Sumber: Hasil Pengolahan Data(2021)

Gambar 9. Listing Program Untuk Menguji Confusion Matrix

Untuk mendapatkan gambaran hasil prediksi berdasarkan uji confusion matrix visualisasikan menggunakan grafik pada gambar 10.



Sumber: Hasil Pengolahan Data(2021)

Gambar 10. Listing Program Untuk Menguji Confusion Matrix

Dari gambar 10 menunjukkan tidak rekomendasi 0 dan rekomendasi 36 jadi data valid dan model ANN bias digunakan untuk melakukan prediksi.

### 3.6. Hasil implementasi Deep Learning

Setelah dilakukan proses learning dalam implementasi Deep Learning menggunakan algoritma ANN untuk implementasi e-learning dari data kuisisioner yang telah di proses. Hasil prediksi untuk rekomendasi aplikasi e-learning yang tepat digunakan untuk pembelajaran online berdasarkan nilai akurasi yang didapatkan dari setiap aplikasi e-learning. Berikut listing program untuk melihat hasil akurasi aplikasi ditunjukkan pada gambar 11.



Sumber: Hasil Pengolahan Data(2021)

Gambar 11. Listing Program accuracy

Hasil accuracy untuk setiap aplikasi e-learning dapat dilihat pada tabel 5.

Tabel 7. Hasil Accuracy Aplikasi E-learning

Aplikasi	Accuracy
Moodle	97,5%
Google Classroom	87%
Edmodo	90,5%
Slack	75.5%

Sumber: Hasil Pengolahan Data (2021)

Dari tabel 7 menunjukkan bahwa *accuracy* yang dihasilkan oleh aplikasi *moodle* lebih tinggi *disbanding* aplikasi *e-learning* lainnya, maka itu rekomendasi aplikasi *e-learning* yang tepat digunakan adalah aplikasi *moodle*.

#### 4. Kesimpulan

Adapun kesimpulan dalam penelitian ini adalah untuk mengukur efektivitas aplikasi *e-learning* yang digunakan mengambil data dari pihak universitas aplikasi *e-learning* yang sering digunakan dimana pengambilan data untuk mengambil *dataset* dilakukan dengan menyebar kuisioner kepada dosen dan mahasiswa sehingga menghasilkan prediksi untuk rekomendasi aplikasi *e-learning* yang tepat setelah dilakukan pembelajaran menggunakan algoritma *Artificial Neural Network* untuk deep learning maka dihasilkan aplikasi *moodle* dengan akurasi 97% terpilih sebagai rekomendasi aplikasi yang akan digunakan di perguruan tinggi.

#### Ucapan Terima Kasih

Ucapan Terima kasih kepada LPPMP Universitas Bhayangkara yang telah mendanai riset penelitian ini sehingga dapat selesai tepat waktu

#### Daftar Pustaka

- Akcaoglu, M., & Lee, E. (2016). Increasing social presence in online learning through small group discussions. *International Review of Research in Open and Distance Learning*, 17(3), 1–17. <https://doi.org/10.19173/irrodl.v17i3.2293>
- Anstey, L. M., & Watson, G. P. L. (2018). A Rubric for Evaluating E-Learning Tools in Higher Education.
- Baidawi, T., & Sandi, M. (2013). Desktop Computing Dengan Metode Eyeos. *Seminar Nasional Inovasi Dan Teknologi (SNIT) 2013 IMPLEMENTASI*, (May), A-221.
- Bühler, C., & Fisseler, B. (2007). Accessible E-Learning and Educational Technology - Extending Learning Opportunities for People with Disabilities. *Conference ICL2007, September 26 -28, 2007*, (August 2014), 11 pages. Retrieved from <https://hal.archives-ouvertes.fr/hal-00257138/>
- Chauhan, K., & Ram, S. (2018). Image Classification with Deep Learning and Comparison between Different Convolutional Neural Network Structures using Tensorflow and Keras. *International Journal of Advance Engineering and Research Development*, 5(02), 533–538.
- Chen, Y., & He, W. (2013). Security risks and protection in online learning: A survey. *International Review of Research in Open and Distance Learning*, 14(5), 108–127. <https://doi.org/10.19173/irrodl.v14i5.1632>
- Chou, T.-C. R. (2014). A Scale of University Students' Attitudes toward e-Learning on the Moodle System. *International Journal of Online Pedagogy and Course Design*, 4(3), 49–65. <https://doi.org/10.4018/ijopcd.2014070104>

- Dodun, O., Panaite, E., Seghedin, N., Nagîţ, G., Duşa, P., Neşţian, G., & Sletineanu, L. (2015). Analysis of an E-learning Platform use by Means of the Axiomatic Design. *Procedia CIRP*, 34, 244–249. <https://doi.org/10.1016/j.procir.2015.07.059>
- Fakhfakh, R., Ben, A., & Ben, C. (2017). Deep Learning-Based Recommendation: Current Issues and Challenges. *International Journal of Advanced Computer Science and Applications*, 8(12). <https://doi.org/10.14569/ijacsa.2017.081209>
- Gregory, S., & Bannister-Tyrrell, M. (2017). Digital learner presence and online teaching tools: higher cognitive requirements of online learners for effective learning. *Research and Practice in Technology Enhanced Learning*, 12(1). <https://doi.org/10.1186/s41039-017-0059-3>
- Kisworo, M. W. (2016). Implementing open source platform for education quality enhancement in primary education: Indonesia experience. *Turkish Online Journal of Educational Technology*, 2016(july), 295–301.
- Kumar Basak, S., Wotto, M., & Bélanger, P. (2018). E-learning, M-learning and D-learning: Conceptual definition and comparative analysis. *E-Learning and Digital Media*, 15(4), 191–216. <https://doi.org/10.1177/2042753018785180>
- Lee, S. M. (2014). The relationships between higher order thinking skills, cognitive density, and social presence in online learning. *Internet and Higher Education*, 21, 41–52. <https://doi.org/10.1016/j.iheduc.2013.12.002>
- Liu, J., & WU, C. (2017). Deep Learning Based Recommendation: A Survey. *Lecture Notes in Electrical Engineering*, 2(March 2017), 467–475. <https://doi.org/10.1007/978-981-10-4154-9>
- Maeda, M., & Ono, Y. (2019). Diffusion of lesson study as an educational innovation. *International Journal of Comparative Education and Development*, 21(1), 46–60. <https://doi.org/10.1108/IJCED-10-2018-0044>
- Moubayed, A., Injadat, M., Nassif, A. B., Lutfiyya, H., & Shami, A. (2018). E-Learning: Challenges and Research Opportunities Using Machine Learning Data Analytics. *IEEE Access*, 6, 39117–39138. <https://doi.org/10.1109/ACCESS.2018.2851790>
- Muniasamy, A., & Alasiry, A. (2020). Deep learning: The impact on future eLearning. *International Journal of Emerging Technologies in Learning*, 15(1), 188–199. <https://doi.org/10.3991/IJET.V15I01.11435>
- Nawaz, A., & Zubair Khan, M. (2012). Issues of Technical Support for e-Learning Systems in Higher Education Institutions. *International Journal of Modern Education and Computer Science*, 4(2), 38–44. <https://doi.org/10.5815/ijmecs.2012.02.06>
- Phipps, L., & Kelly, B. (2006). Holistic approaches to e-learning accessibility. *Alt-J*, 14(1), 69–78. <https://doi.org/10.1080/09687760500479860>
- Prof. Indira, D., & MS. Sakhi. (2017). Online learning. *International Education & Research Journal [IERJ]*, 3(8), 32–34. <https://doi.org/10.4324/9780429355097-7>
- Rapanta, C., Botturi, L., Goodyear, P., Guàrdia, L., & Koole, M. (2020). Online University

- Teaching During and After the Covid-19 Crisis: Refocusing Teacher Presence and Learning Activity. *Postdigital Science and Education*, 2(3), 923–945. <https://doi.org/10.1007/s42438-020-00155-y>
- Raschka, S. (2015). *Python Machine Learning*. Birmingham: Packt Publishing Limited.
- Reza, R. (2017). *Penerapan Jaringan Saraf Tiruan Jenis Deep Learning Untuk Sistem Informasi Kesehatan Jembatan*. Universitas Bina Nusantara.
- Richardson, J. C., Koehler, A. A., Besser, E. D., Caskurlu, S., Lim, J. E., & Mueller, C. M. (2015). Conceptualizing and investigating instructor presence in online learning environments. *International Review of Research in Open and Distance Learning*, 16(3), 256–297. <https://doi.org/10.19173/irrodl.v16i3.2123>
- Shiddieqy, H. A., Hariadi, F. I., & Adiono, T. (2017). Implementation of deep-learning based image classification on single board computer. *2017 International Symposium on Electronics and Smart Devices, ISESD 2017, 2018-Janua*, 133–137. <https://doi.org/10.1109/ISESD.2017.8253319>