

Eksperimen 1: Dengan semua hyperparameter bernilai default, latih dan uji model dengan dua input berbeda: judul dan deskripsi berita.

- **Input Judul**

```
# Load dataset training dan testing
train_df = pd.read_csv('train.csv')
test_df = pd.read_csv('test.csv')

# Input Title
texts = train_df['title'].values
labels = train_df['news_type'].values
```

Hasil Epoch:

```
Epoch 1/5
3000/3000 [=====] - 21s 7ms/step - loss: 1.3844 - accuracy:
0.2749 - val_loss: 1.3806 - val_accuracy: 0.2644
Epoch 2/5
3000/3000 [=====] - 18s 6ms/step - loss: 1.3708 - accuracy:
0.4180 - val_loss: 1.3549 - val_accuracy: 0.4533
Epoch 3/5
3000/3000 [=====] - 16s 5ms/step - loss: 1.3237 - accuracy:
0.6091 - val_loss: 1.2853 - val_accuracy: 0.7288
Epoch 4/5
3000/3000 [=====] - 16s 5ms/step - loss: 1.2318 - accuracy:
0.7184 - val_loss: 1.1759 - val_accuracy: 0.7621
Epoch 5/5
3000/3000 [=====] - 16s 5ms/step - loss: 1.1082 - accuracy:
0.7645 - val_loss: 1.0529 - val_accuracy: 0.7540
```

Hasil Akurasi Test:

```
238/238 [=====] - 0s 1ms/step - loss: 0.7530 - accuracy: 0.8189
Test Accuracy: 0.8189473748207092
```

Training Accuracy: 76.45%

Validation Accuracy: 75.40%

Test Accuracy: 81.89%

Training Time per Epoch: sekitar 16-21 detik

Analisis:

- Model mengalami peningkatan akurasi dari epoch pertama (27.49%) hingga (76.45%). Val_loss mengalami penurunan dan val_accuracy mengalami peningkatan. Hal ini menunjukkan bahwa model bisa mempelajari model data dari judul dengan baik.
- Akurasi pada data tes juga tinggi yakni sebesar 81.89%. Hal ini menunjukkan bahwa model mampu memprediksi cukup baik dengan hanya menggunakan data dari judul berita.

- Meskipun Hanya menggunakan judul berita yang relatif singkat, Model masih dapat mempelajari pola yang cukup baik dari judul berita untuk melakukan klasifikasi jenis berita.

- **Input deskripsi berita**

Load dataset training dan testing

```
train_df = pd.read_csv('train.csv')
```

```
test_df = pd.read_csv('test.csv')
```

Input Description

```
texts = train_df['description'].values
```

```
labels = train_df['news_type'].values
```

Hasil Epoch:

Epoch 1/5

3000/3000 [=====] - 19s 6ms/step - loss: 1.3318 - accuracy: 0.5278 - val_loss: 1.2164 - val_accuracy: 0.7897

Epoch 2/5

3000/3000 [=====] - 21s 7ms/step - loss: 1.0349 - accuracy: 0.8121 - val_loss: 0.8657 - val_accuracy: 0.8242

Epoch 3/5

3000/3000 [=====] - 24s 8ms/step - loss: 0.7351 - accuracy: 0.8473 - val_loss: 0.6318 - val_accuracy: 0.8584

Epoch 4/5

3000/3000 [=====] - 20s 7ms/step - loss: 0.5570 - accuracy: 0.8626 - val_loss: 0.5061 - val_accuracy: 0.8676

Epoch 5/5

3000/3000 [=====] - 26s 9ms/step - loss: 0.4608 - accuracy: 0.8741 - val_loss: 0.4423 - val_accuracy: 0.8747

<keras.src.callbacks.History at 0x16a155ded30>

Hasil Akurasi test:

238/238 [=====] - 1s 2ms/step - loss: 0.4516 - accuracy: 0.8676

Test Accuracy: 0.8676315546035767

Training Accuracy: 87.41%

Validation Accuracy: 87.47%

Test Accuracy: 86.76%

Training Time per Epoch: sekitar 19-26 detik

Analisis:

- Pada eksperimen ini, akurasi model lebih cepat meningkat, mulai dari epoch pertama (52,78%) dan mencapai akurasi pelatihan sebesar 87.41% di epoch kelima. Val_accuracy

- juga mengalami peningkatan yang signifikan, Hal ini menunjukkan bahwa model mempelajari data dari deskripsi lebih baik daripada hanya menggunakan judul.
- Akurasi pada data tes juga lebih tinggi yakni sebesar 86.67 %.
 - Dengan menggunakan deskripsi berita, model dapat belajar lebih banyak informasi yang berguna untuk klasifikasi. Hal ini bisa terjadi karena deskripsi berita cenderung lebih kaya akan konteks dan informasi dibandingkan judul yang lebih ringkas dan terbatas.

Kesimpulan:

Walaupun model dapat bekerja dengan cukup baik hanya dengan menggunakan judul berita, deskripsi berita dapat memberikan hasil yang lebih baik secara keseluruhan. Hal ini disebabkan deskripsi berita memberikan lebih banyak informasi yang relevan untuk membantu model memahami konteks jenis berita dengan lebih mendalam. Hal ini dapat terlihat dari training accuracy, validation accuracy, serta test accuracy yang lebih tinggi pada data dengan input deskripsi berita bila dibandingkan dengan input judul. yakni $87.41\% > 76.45\%$, $87.47\% > 75.40\%$, $86.76\% > 81.89\%$.

Eksperimen 2: Dengan input deskripsi berita dan hyperparameter lain bernilai default, latih dan uji model dengan tiga nilai maxlen berbeda: 10, 100, dan 1000.

Input deskripsi berita

```
# Load dataset training dan testing
```

```
train_df = pd.read_csv('train.csv')
```

```
test_df = pd.read_csv('test.csv')
```

```
# Input Description
```

```
texts = train_df['description'].values
```

```
labels = train_df['news_type'].values
```

- **Maxlen 10**

```
# Melakukan padding teks sepanjang maksimal maxlen
```

```
maxlen = 10
```

```
X_train = pad_sequences(sequences, maxlen=maxlen)
```

Hasil Epoch:

```
Epoch 1/5
```

```
3000/3000 [=====] - 19s 6ms/step - loss: 0.8923 - accuracy:  
0.7189 - val_loss: 0.6284 - val_accuracy: 0.7854
```

```
Epoch 2/5
```

```
3000/3000 [=====] - 20s 7ms/step - loss: 0.5421 - accuracy:  
0.8154 - val_loss: 0.5516 - val_accuracy: 0.8069
```

```
Epoch 3/5
```

```
3000/3000 [=====] - 20s 7ms/step - loss: 0.4593 - accuracy:  
0.8438 - val_loss: 0.5321 - val_accuracy: 0.8118
```

```
Epoch 4/5
```

3000/3000 [=====] - 20s 7ms/step - loss: 0.4109 - accuracy:
0.8599 - val_loss: 0.5290 - val_accuracy: 0.8120
Epoch 5/5
3000/3000 [=====] - 20s 7ms/step - loss: 0.3764 - accuracy:
0.8717 - val_loss: 0.5334 - val_accuracy: 0.8110

Hasil Akurasi Test:

238/238 [=====] - 0s 1ms/step - loss: 0.5506 - accuracy: 0.8033
Test Accuracy: 0.8032894730567932

Training Accuracy: 87.17%

Validation Accuracy: 81.10%

Test Accuracy: 80.33%

Training Time per Epoch: sekitar 19-20 detik

Analisis:

Maxlen 10 hanya menggunakan 10 kata pertama dari setiap deskripsi berita. Akibatnya model tidak bisa mendapatkan konteks secara utuh dari deskripsi yang panjang. Hal ini dapat dilihat dari performa pada data validation (81.10%) dan data testing yang turun yakni hanya 80.33%. Dari awalnya, Training Accuracy: 87.17%. Hal ini mengindikasikan bahwa model sedikit overfitting pada data training.

- **Maxlen 100**

Melakukan padding teks sepanjang maksimal maxlen
maxlen = 100
X_train = pad_sequences(sequences, maxlen=maxlen)

Hasil Epoch:

Epoch 1/5
3000/3000 [=====] - 19s 6ms/step - loss: 0.9121 - accuracy:
0.7969 - val_loss: 0.5184 - val_accuracy: 0.8690
Epoch 2/5
3000/3000 [=====] - 20s 7ms/step - loss: 0.4147 - accuracy:
0.8844 - val_loss: 0.3701 - val_accuracy: 0.8894
Epoch 3/5
3000/3000 [=====] - 19s 6ms/step - loss: 0.3217 - accuracy:
0.9030 - val_loss: 0.3275 - val_accuracy: 0.8999
Epoch 4/5
3000/3000 [=====] - 18s 6ms/step - loss: 0.2786 - accuracy:
0.9146 - val_loss: 0.3043 - val_accuracy: 0.9051
Epoch 5/5

3000/3000 [=====] - 20s 7ms/step - loss: 0.2496 - accuracy:
0.9224 - val_loss: 0.2916 - val_accuracy: 0.9066

Hasil Akurasi Test:

238/238 [=====] - 0s 1ms/step - loss: 0.3020 - accuracy: 0.9047
Test Accuracy: 0.9047368168830872

Training Accuracy: 92.24%

Validation Accuracy: 90.66%

Test Accuracy: 90.47%

Training Time per Epoch: sekitar 18-20 detik

Analisis:

Maxlen 100 memberikan Hasil yang lebih baik daripada maxlen 10. Pada nilai ini, model bisa menangkap lebih banyak konteks dan deskripsi berita, sehingga meningkatkan akurasi pada validation set dan test set. Hasil akurasi model pada data testing sebesar 90.47 % menunjukkan bahwa panjang 100 cukup untuk menyimpan informasi yang relevan dari deskripsi berita tanpa kehilangan konteks atau menambah noise yang berlebihan. Selain itu juga hasil akurasi model pada data training (92.24%) dan pada validation set (90.66%). Hal ini mengindikasikan bahwa tidak terjadi overfitting pada data training karena perbedaan akurasi model pada data validation dan data testing tidak jauh (hanya sekitar 1-2 %)

- **Maxlen 1000**

Melakukan padding teks sepanjang maksimal maxlen
maxlen = 1000
X_train = pad_sequences(sequences, maxlen=maxlen)

Hasil Epoch:

Epoch 1/5

3000/3000 [=====] - 22s 7ms/step - loss: 1.3377 - accuracy:
0.5047 - val_loss: 1.2207 - val_accuracy: 0.7655

Epoch 2/5

3000/3000 [=====] - 22s 7ms/step - loss: 1.0369 - accuracy:
0.8061 - val_loss: 0.8629 - val_accuracy: 0.8345

Epoch 3/5

3000/3000 [=====] - 22s 7ms/step - loss: 0.7310 - accuracy:
0.8445 - val_loss: 0.6267 - val_accuracy: 0.8512

Epoch 4/5

3000/3000 [=====] - 24s 8ms/step - loss: 0.5539 - accuracy:
0.8604 - val_loss: 0.5052 - val_accuracy: 0.8642

Epoch 5/5

3000/3000 [=====] - 25s 8ms/step - loss: 0.4602 - accuracy:
0.8726 - val_loss: 0.4400 - val_accuracy: 0.8723

Hasil Akurasi Test:

238/238 [=====] - 1s 2ms/step - loss: 0.4495 - accuracy: 0.8663
Test Accuracy: 0.8663157820701599

Training Accuracy: 87.26%

Validation Accuracy: 87.23%

Test Accuracy: 86.63%

Training Time per Epoch: sekitar 22-25 detik

Analisis:

Maxlen 1000 mencakup seluruh kata dalam setiap deskripsi berita (hingga 1000 kata). Meski pada dasarnya hal ini memberikan informasi yang lengkap, test accuracy sebesar 86.63% lebih rendah dibandingkan maxlen 100. Ini menunjukkan bahwa meskipun model mendapatkan lebih banyak informasi, Informasi yang didapatkan mungkin kurang relevan. Hal ini menyebabkan penurunan akurasi karena noise tambahan dari kata-kata yang kurang penting. Akurasi model training dan validasi yang lebih rendah juga mengindikasikan bahwa model kesulitan untuk belajar dari sequence yang terlalu panjang, mungkin karena model menjadi terlalu kompleks dan kesulitan untuk menangkap pola yang ada.

Kesimpulan:

Maxlen 100 memberikan hasil yang optimal dalam eksperimen ini, dengan akurasi test tertinggi sebesar 90.47%. Pada maxlen ini, model mampu menangkap konteks yang cukup dari deskripsi berita tanpa kelebihan informasi atau kehilangan konteks. Maxlen 10 mungkin terlalu pendek untuk menangkap konteks secara utuh. Sedangkan maxlen 1000 tidak memberikan keuntungan yang signifikan (bahkan memberikan kerugian) dan mungkin menyebabkan model sulit untuk mempelajari informasi penting dari deskripsi yang panjang.

Eksperimen 3: Dengan input deskripsi berita dan hyperparameter lain bernilai default, latih dan uji model dengan dua nilai epochs berbeda: 1 dan 5.

Input deskripsi berita

```
# Load dataset training dan testing
```

```
train_df = pd.read_csv('train.csv')
```

```
test_df = pd.read_csv('test.csv')
```

```
# Input Description
```

```
texts = train_df['description'].values
```

```
labels = train_df['news_type'].values
```

Maxlen

```
# Melakukan padding teks sepanjang maksimal maxlen
```

```
maxlen = 1000
```

```
X_train = pad_sequences(sequences, maxlen=maxlen)
```

- **Epochs = 1**

Hasil Epochs:

3000/3000 [=====] - 22s 7ms/step - loss: 1.3372 - accuracy: 0.5167 - val_loss: 1.2257 - val_accuracy: 0.7310

Hasil Akurasi Test:

238/238 [=====] - 0s 2ms/step - loss: 1.2281 - accuracy: 0.7233
Test Accuracy: 0.7232894897460938

Training Accuracy: 51.67%

Validation Accuracy: 73.10%

Test Accuracy: 72.33%

Training Time: 22 detik.

Analisis:

Pada epochs = 1, akurasi training sebesar 51.67% menunjukkan bahwa model baru mulai belajar dan belum mampus menangkap pola secara menyeluruh. Validation accuracy cukup besar yakni sekitar 73.10% dan test accuracy sebesar 72.33% (cukup tinggi untuk model yang hanya dilatih dengan menggunakan 1 epoch). Namun, ini menunjukkan bahwa model masih belum cukup matang dalam mempelajari fitur dari data. Model cenderung mengalami underfitting karena belum sepenuhnya mempelajari pola yang ada dalam data.

- **Epochs = 5**

Hasil Epochs:

Epoch 1/5

3000/3000 [=====] - 24s 8ms/step - loss: 1.3532 - accuracy: 0.4796 - val_loss: 1.2707 - val_accuracy: 0.7824

Epoch 2/5

3000/3000 [=====] - 22s 7ms/step - loss: 1.1054 - accuracy: 0.7966 - val_loss: 0.9404 - val_accuracy: 0.7945

Epoch 3/5

3000/3000 [=====] - 24s 8ms/step - loss: 0.8026 - accuracy: 0.8287 - val_loss: 0.6927 - val_accuracy: 0.8339

Epoch 4/5

3000/3000 [=====] - 22s 7ms/step - loss: 0.6142 - accuracy: 0.8450 - val_loss: 0.5580 - val_accuracy: 0.8468

Epoch 5/5

3000/3000 [=====] - 22s 7ms/step - loss: 0.5083 - accuracy: 0.8591 - val_loss: 0.4810 - val_accuracy: 0.8606

Hasil Akurasi Test:

238/238 [=====] - 1s 2ms/step - loss: 0.4897 - accuracy: 0.8554
Test Accuracy: 0.855394721031189

Training Accuracy: 85.91%

Validation Accuracy: 86.06%

Test Accuracy: 85.54%

Training Time per Epoch: sekitar 22-24 detik.

Analisis:

Ketika epochs = 5, model sudah mulai memahami pola dengan lebih dalam, dengan training accuracy yang meningkat hingga 85.91%. Hal ini juga tercermin dari validation accuracy yang mencapai 86.06% dan test accuracy sebesar 85.54%. Penurunan training loss dan validation loss menunjukkan bahwa model secara bertahap mengoptimalkan dirinya selama 5 epoch. Testing yang relatif tinggi menunjukkan bahwa model mampu melakukan klasifikasi dengan baik ke data yang belum pernah dilihat sebelumnya.

Kesimpulan:

Melatih model hanya selama 1 epoch menghasilkan akurasi yang tidak optimal (model mengalami underfitting). Jika waktu pelatihan menjadi faktor yang membatasi, melatih model untuk 1 epoch bisa memberikan hasil yang cukup baik, walaupun model belum bisa menangkap pola secara menyeluruh. Melatih model selama 5 epochs menghasilkan peningkatan performa yang signifikan. Dengan test accuracy sebesar 85.54%, hal ini dikarenakan model telah mempelajari pola dengan lebih dalam sehingga memberikan hasil yang lebih baik. 5 epochs tampak menjadi jumlah yang optimal dalam eksperimen ini, karena peningkatan akurasi cukup besar dibandingkan 1 epoch, tanpa ada indikasi overfitting.