

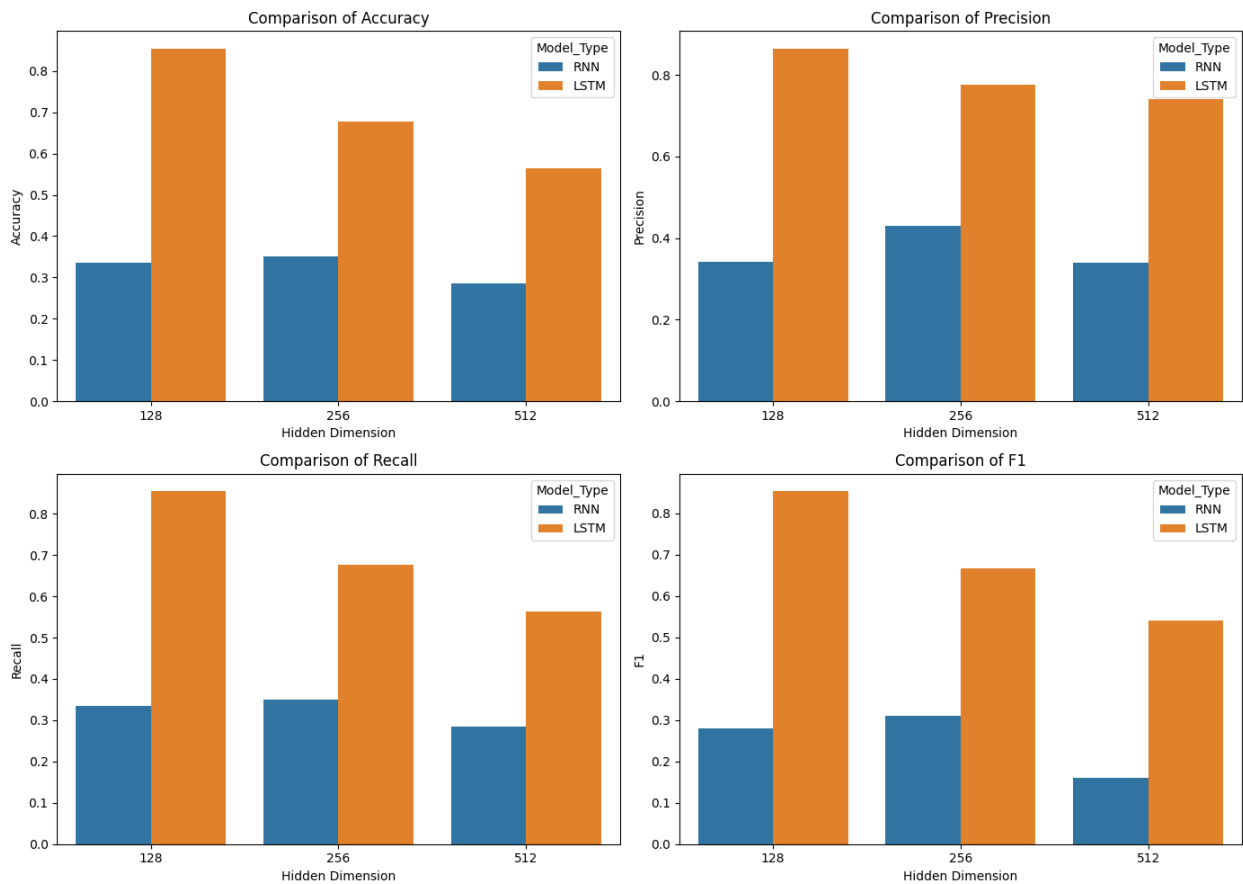
**Analisis Performa Model RNN dan LSTM untuk Klasifikasi Artikel Berita BBC**  
Oleh : Risma Widiyanti ( 24/545884/PPA/06830)

Dataset : <https://www.kaggle.com/datasets/jacopoferretti/bbc-articles-dataset/data>

Github : <https://github.com/rismawidiyanti/news-rnn-lstm>

**Tabel Hasil Perbandingan Model**

Model	Dimensi Hidden	Akurasi	Presisi	Recall	F1 Score
RNN	128	0.3348	0.3419	0.3348	0.2791
	256	0.3506	0.4296	0.3506	0.3098
	512	0.2854	0.3388	0.2854	0.1613
LSTM	128	0.8539	0.8645	0.8539	0.8530
	256	0.6764	0.7754	0.6764	0.6667
	512	0.5640	0.7414	0.5640	0.5400



## Visualisasi dan Temuan Kinerja Model

Dalam tugas ini, saya berusaha memahami bagaimana arsitektur RNN dan LSTM. Prosesnya tidak selalu mulus—terdapat banyak tantangan:

### 1. Menjalankan RNN Awal

Pertama kali menjalankan model RNN, saya terkejut dengan hasil yang buruk. Model dengan 512 unit (yang seharusnya lebih kuat) malah memberikan hasil terburuk (28.5%). Setelah beberapa kali percobaan, saya sadar bahwa model ini mengalami masalah vanishing gradient—sebuah fenomena di mana model kehilangan kemampuan untuk "mengingat" informasi dari awal kalimat saat memproses artikel panjang.

```
14/14 35 195ms/step
RNN_512 Performance:
Accuracy: 0.2854
Precision: 0.3388
Recall: 0.2854
F1 Score: 0.1613

Classification Report:

```

	precision	recall	f1-score	support
business	0.00	0.00	0.00	102
entertainment	0.00	0.00	0.00	77
politics	0.23	0.73	0.35	84
sport	1.00	0.01	0.02	102
tech	0.37	0.81	0.51	80
accuracy			0.29	445
macro avg	0.32	0.31	0.17	445
weighted avg	0.34	0.29	0.16	445

Contoh artikel yang selalu salah diklasifikasi oleh RNN:

```
Example 379:
True category: sport
RNN prediction: politics
LSTM prediction: sport
Excerpt: ferguson fear milan cut edg manchest unit manag sir alex ferguson said side task ac milan would made easier absenc andriy shevchenko milan talisman european footbal year miss wednesday char
Sequence length: 243
Number of unique tokens: 164
```

RNN memprediksi sebagai "politik" padahal sebenarnya "olahraga".

### 2. Menjalankan LSTM

Situasi berubah saat beralih ke LSTM. Model 128-unit langsung menunjukkan peningkatan signifikan, mencapai akurasi 85.4%.

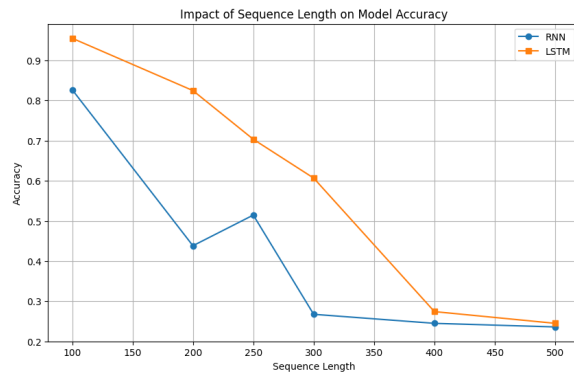
```
LSTM_128 Performance:
Accuracy: 0.8539
Precision: 0.8645
Recall: 0.8539
F1 Score: 0.8530

Classification Report:

```

	precision	recall	f1-score	support
business	0.98	0.84	0.91	102
entertainment	0.84	0.74	0.79	77
politics	0.89	0.74	0.81	84
sport	0.85	0.97	0.91	102
tech	0.74	0.95	0.83	80
accuracy			0.85	445
macro avg	0.86	0.85	0.85	445
weighted avg	0.86	0.85	0.85	445

Tantangan sebenarnya muncul saat mencoba mengoptimalkan panjang sekuens. Melalui eksperimen panjang sekuens (100-500 token), saya menemukan bahwa:



1. Sekuens 100 token: Memberikan hasil terbaik untuk LSTM (95.5%)
2. Sekuens 250 token: Masih memberikan hasil baik (70.3%)
3. Sekuens 400-500 token: Performa menurun drastis (24-26%)

Eksperimen dengan embedding pun menunjukkan bahwa menggunakan GloVe pre-trained memberikan peningkatan dibanding embedding yang dilatih dari awal.

### 3. Analisa Temuan

Dalam proses ini saya menemukan beberapa hal :

#### a. Paradoks Kompleksitas Model

Salah satu temuan mengejutkan adalah bahwa model LSTM terbesar (512 unit) malah memberikan hasil terburuk dibanding variannya. Setelah menganalisis kurva pembelajaran, saya menemukan bahwa model besar cenderung mengalami overfitting.

Model LSTM-128 mencapai konvergensi lebih cepat dan stabil:

```
Epoch 8/15
42/42 ----- 0s 445ms/step - accuracy: 0.5997 - loss: 1.0446WARNING:absl:You are saving your model as an HDF5 file which will be overwritten every time you call model.save(). Consider using the --save_to_text_format flag to save to a text-based format instead.
42/42 ----- 21s 511ms/step - accuracy: 0.6016 - loss: 1.0411 - val_accuracy: 0.7663 - val_loss: 0.7247
Epoch 9/15
42/42 ----- 41s 524ms/step - accuracy: 0.7680 - loss: 0.6770 - val_accuracy: 0.7618 - val_loss: 0.7646
Epoch 10/15
42/42 ----- 42s 545ms/step - accuracy: 0.5231 - loss: 1.2200 - val_accuracy: 0.5303 - val_loss: 1.2242
Epoch 11/15
42/42 ----- 22s 538ms/step - accuracy: 0.5458 - loss: 1.1209 - val_accuracy: 0.6315 - val_loss: 0.9693
Epoch 12/15
42/42 ----- 0s 442ms/step - accuracy: 0.6766 - loss: 0.8416WARNING:absl:You are saving your model as an HDF5 file which will be overwritten every time you call model.save(). Consider using the --save_to_text_format flag to save to a text-based format instead.
42/42 ----- 21s 508ms/step - accuracy: 0.6768 - loss: 0.8409 - val_accuracy: 0.6787 - val_loss: 0.6894
Epoch 13/15
42/42 ----- 0s 474ms/step - accuracy: 0.7728 - loss: 0.5809WARNING:absl:You are saving your model as an HDF5 file which will be overwritten every time you call model.save(). Consider using the --save_to_text_format flag to save to a text-based format instead.
42/42 ----- 41s 520ms/step - accuracy: 0.7736 - loss: 0.5802 - val_accuracy: 0.8337 - val_loss: 0.5395
```

Sementara model LSTM-512 menunjukkan pembelajaran yang tidak stabil:

```
Epoch 9/15
42/42 0s 3s/step - accuracy: 0.3939 - loss: 1.4386WARNING:absl:You are saving your model as an HDF5 f
42/42 143s 3s/step - accuracy: 0.3949 - loss: 1.4367 - val_accuracy: 0.5326 - val_loss: 1.1773
Epoch 10/15
42/42 200s 3s/step - accuracy: 0.4972 - loss: 1.2221 - val_accuracy: 0.5011 - val_loss: 1.2612
Epoch 11/15
42/42 144s 3s/step - accuracy: 0.5129 - loss: 1.2723 - val_accuracy: 0.4809 - val_loss: 1.3111
Epoch 12/15
42/42 141s 3s/step - accuracy: 0.4772 - loss: 1.2710 - val_accuracy: 0.5303 - val_loss: 1.1852
Epoch 13/15
42/42 0s 3s/step - accuracy: 0.5196 - loss: 1.0856WARNING:absl:You are saving your model as an HDF5 f
42/42 143s 3s/step - accuracy: 0.5202 - loss: 1.0851 - val_accuracy: 0.5618 - val_loss: 1.1122
Epoch 14/15
42/42 141s 3s/step - accuracy: 0.5904 - loss: 1.0191 - val_accuracy: 0.3438 - val_loss: 2.2914
```

## b. Kesalahan dalam Mengungkap Pola

Analisis artikel yang salah diklasifikasi mengungkap bahwa RNN sering bingung antara kategori "politik" dan "olahraga" yang memiliki beberapa kata kunci serupa (misalnya "pemimpin", "menang", "kalah"). LSTM lebih mampu memahami konteks keseluruhan.

Contoh artikel yang benar diklasifikasi oleh LSTM namun salah oleh RNN:

```
True category: business
RNN prediction: politics
LSTM prediction: business
Excerpt: record year chilean copper chile copper industri regist record earn bn government chilean copper commiss cochilco report stron
```

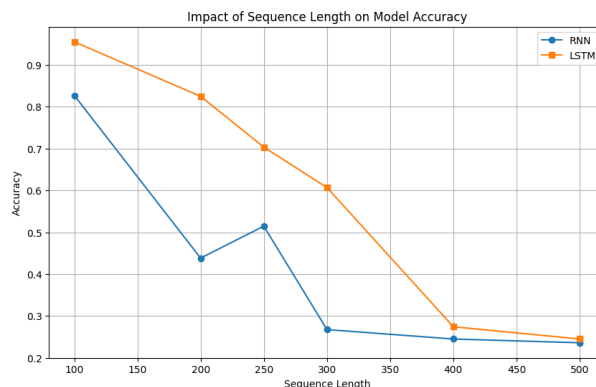
*"record year chilean copper chile copper industri regist record earn bn government chilean copper commiss..."*

- RNN: politik
- LSTM: bisnis (benar)

## c. Efek Panjang Sekuens

Melalui percobaan, saya menemukan bahwa panjang sekuens optimal tidak selalu "lebih panjang = lebih baik". Terlalu pendek, konteks hilang. Terlalu panjang, model tenggelam dalam noise. Setelah percobaan berulang, sekuens 100-250 token memberikan keseimbangan terbaik.

Performa LSTM pada berbagai panjang sekuens



## Kesimpulan

Melalui proses trial-and-error, saya memperoleh pemahaman tentang:

1. **Pilihan arsitektur yang tepat:** LSTM > RNN untuk teks panjang
2. **Fenomena overfitting:** Model besar tidak selalu lebih baik
3. **Keseimbangan panjang input:** Tidak terlalu pendek, tidak terlalu panjang
4. **Power representasi kata:** Embedding pre-trained sangat membantu