



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB V

ANALISA DATA DENGAN LSTM

5.1 Analisa Hasil Tes pada Konfigurasi LSTM

Agar dapat memperoleh konfigurasi terbaik untuk *datasets* yang dimiliki, maka dilakukan tes terhadap konfigurasi model terlebih dahulu dengan menggunakan *datasets* yang sama. Selain itu, uji coba konfigurasi model dimaksudkan untuk mengamati seberapa besar dampak satu konfigurasi terhadap kinerja model LSTM. Adapun konfigurasi yang diamati pada model LSTM yakni: (1) jumlah iterasi pada training model Word2vec: skip-grams, (2) jumlah epoch, (3) jumlah layer, (4) jumlah cell unit, (5) learning-rate, (6) jumlah batch data, (7) inisialisasi weight, (8) inisialisasi bias, (9) L2 pada weight, (10) L2 pada bias. Model diujicoba dengan mengubah konfigurasi satu per satu dan hasil akurasiya kemudian dibandingkan. Tes dilakukan dengan menggunakan *datasets* proporsional, dimana jumlah *tweets* kelas positif dan negatif seimbang, yakni 1,422 *tweets*. Hasil tes konfigurasi yang dilakukan terhadap model dapat dilihat pada halaman selanjutnya.

Tabel 5.1. Hasil Tes Konfigurasi Model LSTM

NO	CONFIG										RESULT				
	WORD2VEC	EPOCH	LAYER	CELL	LEARNING-R	BATCH-DATA	WEIGHT-INIT	BIAS-INIT	L2-WEIGHT	L2-BIAS	AVG LOSS	TRAINING ACC	TRUE POS	TRUE NEG	EV. ACC
1	10000	12	1	128	0.001	1					0.0052	90	88.23	65.85	77.04
2	50000	12	1	128	0.001	1					0.0025639	99.96	81.02	72.97	76.995
3	10000	1	10	128	0.001	1					0.44872618	78.97	72.26	72.29	72.275
4	10000	20	1	128	0.001	1					0.00342192	99.9	78.1	74.32	76.21
5	10000	50	1	128	0.001	1					0.00221528	99.96	83.94	66.89	75.415
6	10000	12	10	128	0.001	1					0.00371617	99.92	10.21	92.56	51.385
7	10000	12	1	256	0.001	1					0.0022180005	100	78.01	75	76.505
8	10000	12	1	512	0.001	1					0.00453826	99.96	69.34	81.08	75.21
9	10000	12	1	128	0.0001	1					0.1221303	97.69	85.4	58.1	71.75
10	10000	12	1	128	0.001	10					0.008715446	99.9	99.27	10.81	55.04
11	10000	12	1	128	0.001	1	-0.003, 0.003				0.0024826	99.99	66.42	85.81	76.115
12	10000	12	1	128	0.001	1		-0.003, 0.003			0.0024316	99.92	60.58	88.51	74.545
13	10000	12	1	128	0.001	1	-0.003, 0.003	-0.003, 0.003			0.002486403	99.92	0	100	50
14	10000	12	1	128	0.001	1			0.001		0.001973001	99.9	87.59	62.16	74.875
15	10000	12	1	128	0.001	1				0.001	0.002583653	99.96	85.4	66.21	75.805
16	10000	12	1	128	0.001	1			0.001	0.001	0.001792818	100	85.4	66.89	76.145

UMN

Tes dilakukan sebanyak 16 kali. Setelah tes dilakukan, terdapat pengamatan lebih mendalam terhadap setiap konfigurasi pada model dan pengaruhnya terhadap tingkat akurasi model. Adapun konfigurasi yang diamati terhadap *datasets* antara lain:

1. Jumlah iterasi *training* pada *word embedding*

Jumlah iterasi *training* pada *word embedding*, yakni *word2vec* tidak mempengaruhi secara signifikan tingkat akurasi model dalam mengklasifikasi *tweets*. Jumlah iterasi yang dites yakni: 10,000 dan 50,000.

2. Jumlah *epoch*

Epoch dalam jumlah tertentu dapat mempengaruhi tingkat akurasi model. Jumlah *epoch* yang terlampaui banyak cenderung membuat model menjadi *overfit* pada *training datasets*, sehingga hasil akurasi yang dihasilkan tidak maksimal. *Epoch* perlu ditetapkan dengan jumlah yang tertentu, dimana jumlah tersebut tak terlampaui banyak dan tidak sedikit. Jumlah *training steps* yang dites yakni: 1, 12, 20, 50.

3. Jumlah *layer*

Layer dapat mempengaruhi tingkat akurasi model secara signifikan. *Layer* dengan jumlah yang terlampaui banyak cenderung membuat model menjadi *overfit*. Sebaliknya, *layer* dengan jumlah yang sedikit dapat meningkatkan akurasi model secara signifikan, terlebih lagi jika diterapkan dalam *datasets* dengan jumlah yang sedikit. Jumlah *layer* yang dites: 1, 10.

4. Jumlah *cell units*

Cell units juga ikut mempengaruhi tingkat akurasi model secara signifikan. Penetapan *cell units* dengan jumlah tertentu bisa meningkatkan tingkat akurasi model. Namun, penetapan *cell units* yang terlampau banyak tidak berpengaruh terhadap peningkatan akurasi model. Jumlah *cell units* yang dites: 128, 256, 512.

5. *Learning-rate*

Learning-rate tidak mempengaruhi tingkat akurasi model secara signifikan.

Learning-rate yang diuji coba: 0.001, 0.0001.

6. Jumlah *batch data*

Jumlah *batch data* dapat mempengaruhi tingkat akurasi model. Semakin sedikit jumlah *batch data*, maka semakin besar pula pengaruhnya terhadap peningkatan tingkat akurasi model. *Batch data* yang diuji coba: 1, 10.

7. Nilai inisialisasi *weight* dan *bias*

Dalam tes yang dilakukan, nilai inisialisasi *weight* dan *bias* tidak mempengaruhi tingkat akurasi model. Namun bila diimplementasi bersamaan, tingkat akurasi model cenderung menurun. Nilai inisialisasi *weight* dan *bias* yang diuji coba: -0.003, 0.003.

8. Regularisasi L2 pada *weight* dan *bias*

Regularisasi L2 pada *weight* dan *bias* tidak mempengaruhi tingkat akurasi model. Nilai regularisasi L2 yang diuji coba: 0.001.

Berdasarkan hasil tes, konfigurasi terbaik untuk model LSTM yakni dengan menerapkan: 1 *layer*, 12 *epoch* model LSTM, 10,000 *training iteration word2vec*, 128 *cell units*, *learning rate* sebesar 0.001 dan *batch data* sejumlah 1. Konfigurasi

terbaik digunakan dalam eksperimen berikutnya untuk mengoptimalkan hasil akurasi model. Opsi konfigurasi lainnya juga ikut diterapkan dalam model guna melakukan eksplorasi lebih mendalam mengenai kinerja model LSTM terhadap *datasets* yang diberikan.

5.2 Analisa Hasil Klasifikasi Data dengan LSTM

Fokus utama pada eksperimen yang dilakukan yakni mengukur kinerja klasifikasi model LSTM dengan menggunakan jumlah layer yang berbeda. Konfigurasi lainnya mengikuti konfigurasi model terbaik yang diperoleh pada hasil tes sebelumnya. Secara keseluruhan, terdapat 4 macam eksperimen yang dilakukan pada model:

1. LSTM dengan *datasets* jumlah kelas tidak proporsional

Total *datasets* yang digunakan dalam eksperimen ini yakni: 13,715 *tweets*.

Dalam eksperimen ini, jumlah data antar kelas dalam *datasets* yang diberikan tidak seimbang, dimana *tweets* bermakna positif jauh lebih banyak dibandingkan *tweets* negatif dengan selisih 10,871 *tweets*.

Konfigurasi model yang digunakan yakni: 10 *layers* dengan 128 *cell unit*.

2. LSTM dengan *datasets* proporsional yang jumlahnya diperbanyak sejumlah *ratio*: 1.2 (3,412 *tweets*).

Datasets proporsional yang dimaksud yakni *datasets* dengan jumlah *tweets* seimbang untuk setiap kelas; positif dan negatif. *Datasets* proporsional diambil secara acak dari *datasets* keseluruhan yang berjumlah 13,715 *tweets*. Setiap eksperimen berikut dilakukan sebanyak 7 kali. Hal tersebut

dimaksudkan agar seluruh *tweets* bermakna positif divalidasi, sehingga dapat melihat hasil akurasi model LSTM dengan *datasets* keseluruhan namun dengan proporsi yang seimbang.

Eksperimen ini menggunakan *datasets* proporsional yang diperbanyak sejumlah *ratio*: 1.2 kali dari seluruh jumlah kelas *tweets* bermakna negatif, yakni 1,422 *tweets*. Dalam hal ini, jumlah untuk kedua kelas *tweets*, positif dan negatif ditetapkan dari jumlah maksimal *tweets* kelas negatif, sebanyak 1,422 *tweets*, kemudian dikalikan dengan *ratio*: 1.2. Perkalian jumlah *datasets* dengan *ratio* dimaksudkan untuk memperbanyak jumlah *datasets* demi memaksimalkan latihan pada model LSTM. Total *datasets* yang digunakan yakni: 3,412 *tweets*. Adapun konfigurasi model yang diterapkan yakni: 10 *layers* dengan 128 *cell unit*.

3. Eksperimen dengan *datasets* proporsional yang diperbanyak sebesar 2 kali dari jumlah *datasets* pada eksperimen no: (2) dengan konfigurasi 10 *layers* (6,824 *tweets*).

Total *datasets* yang diterapkan yakni: 6,824 *tweets*. Jumlah tersebut ditetapkan dari jumlah maksimal kelas negatif: 1,422 *tweets* dikalikan dengan *ratio*: 1.2 dan hasil akhirnya dikalikan dengan 2.

4. Eksperimen dengan *datasets* proporsional yang diperbanyak sebesar 2 kali dari jumlah *datasets* pada eksperimen no: (2) dengan konfigurasi 1 *layer* (6,824 *tweets*).

Adapun total *tweets* yang digunakan pada eksperimen berikut sama dengan eksperimen sebelumnya (3), namun dengan konfigurasi jumlah *layer* yang berbeda, yakni 1 *layer*.

Berikut pembahasan setiap eksperimen yang dilakukan pada model LSTM:

A. LSTM dengan Datasets Tidak Proporsional

Adapun *datasets* yang digunakan sejumlah: 13,715 *tweets*. Konfigurasi yang diterapkan pada model yakni: *layers*: 10, *cell units*: 128, *learning-rate*: 0.001, *batch data*: 1, jumlah iterasi *training word2vec*: 10,000. Dalam menganalisa sentimen, model LSTM meraih akurasi sebesar: 48.72%.

Tingkat akurasi yang diperoleh menandakan bahwa model tidak cukup mumpuni dalam mengklasifikasi *tweets* dengan makna positif dan negatif.

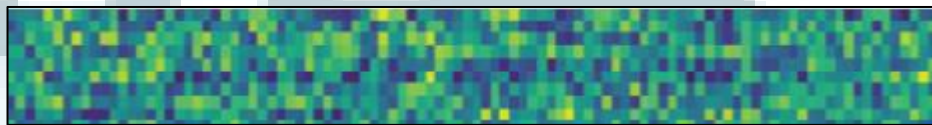
Berikut tingkat akurasi model pada eksperimen ini:

OVERALL EXPERIMENT RESULT-1					
FOLD	LOSS	TRAINING ACC	TRUE POS	TRUE NEG	EV. TESTING ACC
1	1.3254	89.11	38.68	60.42	49.55
2	2.213	89.29	77.31	25.27	51.29
3	11.4201	89.79	75.207	25.88	50.5435
4	4.033	89.14	48.65	61.41	55.03
5	1.8127	89.97	32.68	65.95	49.315
6	3.583	89.54	38.26	43.7	40.98
7	3.959	89.039	38.83	62.3	50.565
8	4.7318	87.67	71.87	25	48.435
9	2.00051	88.69	81.41	18.8	50.105
10	2.53884	89.13	41.93	40.9	41.415
		AVG. ACC	54.4827	42.963	48.72285

Tabel 5.2. Hasil Akurasi LSTM dengan Datasets Tidak Proporsional

Rata-rata akurasi model yang sudah divalidasi dengan *10-fold cross validation* yakni: 48.72%. Tingkat akurasi yang diraih oleh model menandai bahwa model tersebut mengalami *overfit* pada training datasets. Besarnya

nilai *loss* yang terdapat pada model LSTM juga menandakan bahwa sesi *training* model tidak berlangsung secara optimal, meskipun akurasi *training* sudah mencapai 85% ke atas. Setelah proses *training* pada model berakhir, *weight* pada model juga disimpan demi mengetahui distribusi nilai *weight* pada model. Berikut salah satu persebaran *weight* pada sesi *training* model:



Gambar 5.1. Persebaran *Weight* pada Salah Satu Sesi *Training* LSTM

Pada gambar di atas (Gambar 5.1.), intensitas kedalaman warna merepresentasikan nilai *weight*. Semakin gelap kedalaman warnanya, maka semakin kecil pula nilai *weight*, begitu juga sebaliknya. Nilai *weight* yang tersebar yakni dalam rentang: -0.7807065 hingga 0.7294444. Persebaran *weight* tersebut menandakan dua hal:

- Distribusi *weight* didominasi oleh warna gelap. Hal ini menandakan bahwa *weight* yang dihasilkan bernilai kecil, dan tidak berpengaruh banyak pada peningkatan akurasi sesi latihan model.
- *Weight* dengan nilai yang besar, yang ditandai dengan intensitas warna cerah tidak terdistribusi dengan baik. Hal ini menandakan bahwa model tidak terlatih secara optimal.

Dalam eksperimen ini, model mengalami kecenderungan kepada salah satu kelas *tweets*, baik positif ataupun negatif. Hal tersebut diduga akibat ketidakseimbangan jumlah *datasets* antar kelas, dimana *tweets* bermakna positif jauh lebih banyak frekuensinya (12,293 *tweets*) dibandingkan *tweets* negatif (1,422 *tweets*).

B. LSTM dengan *datasets* proporsional yang jumlahnya diperbanyak sejumlah *ratio*: 1.2 (3,412 *tweets*).

Adapun konfigurasi model yang diterapkan: 10 *layers* dan 128 *cell units*. Jumlah *tweets* untuk setiap kelas yakni 1,706, dengan total: 3,412. Eksperimen ini diuji coba sebanyak 7 ronde. Berikut hasil akurasinya:

OVERALL EXPERIMENT RESULT-2			
ROUND	TRUE POSITIVE	TRUE NEGATIVE	EV. TESTING ACC
1	60.7321	38.643	49.68
2	31.236	69.069	50.15
3	57.2977	46.688	51.99
4	53.571	46.657	50.11
5	79.937	21.108	50.82
6	41.1104	60.097	50.6
7	56.869	42.292	49.58
AVG.ACC	54.39331429	46.36485714	50.41857143

Tabel 5.3. Hasil Akurasi LSTM Datasets Proporsional dengan 10 Layers (3,412 Tweets)

Rata-rata akurasi model yang sudah divalidasi dengan *10-fold cross validation* yakni: 50.41%. Berdasarkan tingkat akurasi yang diperoleh di atas, model LSTM masih mengalami kesulitan dalam mengklasifikasi *tweets* bermakna positif dan negatif meski jumlah *datasets* telah

diseimbangkan. Meski diuji coba dengan *datasets tweets* bermakna positif yang berbeda-beda, namun tingkat akurasi masih berada di rentang 49% - 51%. Dugaan atas rendahnya tingkat akurasi tersebut yakni ialah banyaknya *layer* yang diimplementasikan pada model, yakni 10 *layers*.

Selain itu, jika diamati untuk setiap ronde, *loss* juga mengalami fluktuasi. Dalam hal ini, proses *training* model LSTM tidak berlangsung secara optimal. Detail *loss* dan akurasi setiap ronde disertakan di dalam lampiran.

C. Eksperimen dengan *datasets* proporsional yang diperbanyak sebesar 2 kali dari jumlah *datasets* pada eksperimen no: (2) dengan konfigurasi 10 *layers* (6,824 *tweets*).

Eksperimen dengan *datasets* proporsional yang diperbanyak sebesar 2 kali dari jumlah *datasets* eksperimen sebelumnya (B) dengan konfigurasi: 10 *layers*. Pada eksperimen kali ini, konfigurasi yang diimplementasikan pada model masih serupa dengan eksperimen sebelumnya (A): 10 *layers* dan 128 *cell units*. Adapun jumlah *datasets* yang digunakan pada eksperimen ini sebanyak 2 kali lipat dibandingkan *datasets* proporsional pada eksperimen sebelumnya, yakni 6,824 *tweets*. Eksperimen ini dimaksudkan untuk mengetahui seberapa besar pengaruh jumlah *datasets* terhadap kinerja model LSTM. Berikut hasil akurasi model untuk 7 ronde:

OVERALL EXPERIMENT RESULT-3			
ROUND	TRUE POSITIVE	TRUE NEGATIVE	EV. TESTING ACC
1	43.991	57.959	50.975
2	48.288	52.179	50.2335
3	60.835	39.4905	50.16275
4	49.977	51.3445	50.66075
5	40.4694	58.719	49.5942
6	48.358	51.928	50.143
7	76.428	23.751	50.0895
AVG.ACC	52.62091429	47.91014286	50.26552857

Tabel 5.4. Hasil Akurasi LSTM *Datasets* Proporsional dengan 10 Layers (6,824 Tweets)

Rata-rata akurasi model yang sudah divalidasi dengan *10-fold cross validation* yakni: 50.26%. Berdasarkan hasil yang diperoleh, model LSTM masih mengalami kesulitan dalam mengklasifikasi *tweets* meskipun jumlah *datasets* proporsional telah diperbanyak. Sama seperti eksperimen sebelumnya, hasil akurasi berada di rentang 49% - 51%. Meski jumlah *datasets* sudah diperbanyak, namun hasil akurasi model LSTM sama sekali tidak mengalami peningkatan. Model LSTM diduga mengalami *overfitting* terhadap *datasets* yang diberikan.

D. Eksperimen dengan *datasets* proporsional yang diperbanyak sebesar 2 kali dari jumlah *datasets* pada eksperimen no: (2) dengan konfigurasi 1 layer (6,824 tweets).

Eksperimen dengan *datasets* proporsional yang diperbanyak sebesar 2 kali dari jumlah *datasets* eksperimen (C) dengan konfigurasi: 1 layer. Total *datasets* pada eksperimen ini sama seperti eksperimen sebelumnya (2), yakni 6,824 *tweets*. Perbedaan dengan eksperimen sebelumnya (C) yakni

hanya terletak pada jumlah *layer*, dimana pada eksperimen ini hanya menggunakan 1 *layer*. Eksperimen kali ini dilakukan untuk mengetahui seberapa besar pengaruh *layer* terhadap kinerja model LSTM. Berikut hasil tingkat akurasi model LSTM dengan konfigurasi 1 *layer*:

OVERALL EXPERIMENT RESULT-4			
ROUND	TRUE POSITIVE	TRUE NEGATIVE	EV. TESTING ACC
1	91.435	62.365	76.9
2	65.542	88.838	77.19
3	85.75	71.353	78.5515
4	60.159	92.689	76.424
5	91.334	64.0465	78.21025
6	79.539	79.83	79.6845
7	75.478	81.547	78.5125
AVG.ACC	78.46242857	77.23835714	77.92467857

Tabel 5.5. Hasil Akurasi LSTM *Datasets* Proporsional dengan 1 Layer (6,824 Tweets)

Rata-rata akurasi model yang sudah divalidasi dengan *10-fold cross validation* yakni: 77.92%. Berdasarkan hasil yang diperoleh, model LSTM mengalami peningkatan akurasi sekitar 20%. Tingkat akurasi model LSTM juga cenderung stabil, tidak mengalami peningkatan ataupun penurunan secara signifikan selama ronde berlangsung. Nilai *loss* pada setiap ronde juga masih mengalami fluktuasi, namun dengan frekuensi fluktuasi yang lebih minim, serta lebih kecil dibanding eksperimen sebelumnya. Hal ini menandakan bahwa sesi *training* model LSTM berlangsung secara optimal, sehingga hasil akurasinya pun mengalami peningkatan. Pada eksperimen ini, model LSTM sudah tidak mengalami *overfitting*. Detail *loss* dan akurasi setiap ronde disertakan di dalam lampiran.

5.3 Diskusi

Terdapat dua hal utama yang mempengaruhi tingkat akurasi kedua model, diantaranya: (1) jumlah *datasets* antar kelas, dan (2) fitur ataupun konfigurasi model. Sebelum diberikan *datasets* dengan jumlah data yang seimbang antar kelas, kinerja kedua model tidaklah optimal. Kedua model sering kali mengalami kecenderungan pada salah satu kelas yang memiliki banyak data, yakni *tweets* kelas positif. Untuk itu, penyeimbangan jumlah *datasets* kedua kelas dilakukan agar model mampu mengklasifikasi *tweets* secara optimal. Setiap model diberikan jumlah *datasets* proporsional yang berbeda-beda. Pada *Naïve Bayes*, jumlah *datasets* proporsional ditetapkan sebanyak jumlah *tweets* kelas negatif, yakni 1,422 *tweets*. Sedangkan pada LSTM, jumlah *datasets* proporsional ditentukan dari jumlah *tweets* kelas negatif (1,422 *tweets*) dikali dengan *ratio*: 1.2.

Pada *datasets* proporsional yang disiapkan untuk model *Naïve Bayes*, jumlah *datasets* proporsional untuk kedua kelas ditentukan dari jumlah seluruh *tweets* kelas negatif, yakni 1,422 *tweets* dan diambil secara acak dari *datasets* utama. Total *datasets* proporsional yang diuji coba pada model *Naïve Bayes* mencapai: 2,844 *tweets*. Penyeimbangan *datasets* tentunya berpengaruh besar terhadap kinerja model *Naïve Bayes*. Hal ini dapat dibuktikan dari hasil akurasi yang diraih, yakni: 61.821%. Hasil akurasi *Naïve Bayes* mengalami peningkatan sekitar lebih dari 10% dari *datasets* tidak proporsional.

Sementara itu, model LSTM memiliki 2 rangkaian *datasets* proporsional dengan jumlah yang berbeda. Jumlah *datasets* untuk rangkaian pertama ditentukan

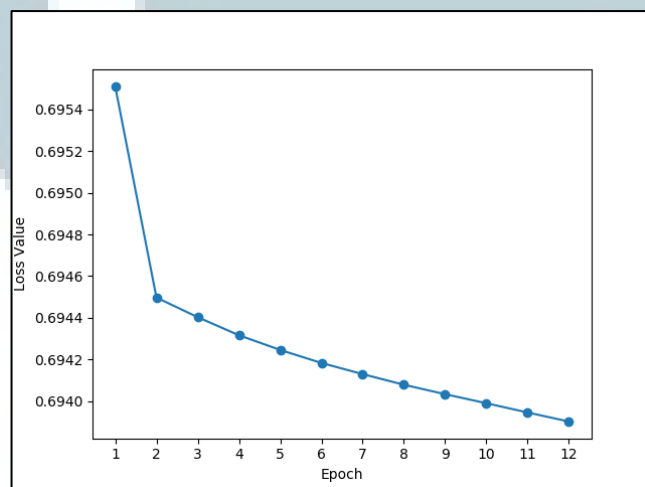
dari jumlah maksimal *tweets* kelas negatif, yakni: 1,422 *tweets* kemudian dikali dengan *ratio* yang ditentukan; 1.2. Jumlah *tweets* untuk masing-masing kelas: 1,706 dengan total mencapai: 3,412 *tweets*. Rangkaian kedua *datasets* merupakan rangkaian *datasets* pertama yang digandakan. Total dari rangkaian kedua *datasets*: 6,824 *tweets*. Agar hasil akurasi model LSTM dapat tervalidasi dengan seluruh *tweets* kelas positif, maka kedua rangkaian *datasets* tersebut dijalankan sebanyak 7 kali. Penyeimbangan *datasets* menjadi salah satu faktor yang mampu meningkatkan hasil akurasi model LSTM, disamping konfigurasi model lainnya.

Selain jumlah *datasets*, hal lain yang mempengaruhi kinerja kedua model ialah fitur ataupun konfigurasi pada model. Pada *Naïve Bayes*, terdapat dua fitur yang diterapkan, antara lain:

- Add-One Laplace Smoothing
- Kapitalisasi kata

Di antara kedua fitur tersebut, fitur yang mempengaruhi kinerja model ialah *Add-One Laplace Smoothing*. Implementasi *Add-One Laplace Smoothing* pada model *Naïve Bayes* mampu meningkatkan tingkat akurasi model sebanyak 5% pada *datasets proporsional*. Hal tersebut dikarenakan fitur *Add-One Laplace Smoothing* berperan besar dalam memperhitungkan probabilitas kata yang tidak terdapat pada *training datasets* namun hadir di dalam *testing datasets*. Fitur kapitalisasi kata pada *Naïve Bayes* tidak mempengaruhi tingkat akurasi model. Rata-rata tingkat akurasi yang diraih oleh *Naïve Bayes* dengan fitur *Add-One Laplace Smoothing* ialah: 66.31% pada *datasets proporsional*.

Sementara itu, pada model LSTM terdapat beberapa konfigurasi yang sangat mempengaruhi kinerja akurasi model. Konfigurasi utama yang paling berpengaruh berdasarkan hasil uji coba sebelumnya yakni: (1) jumlah *layer*, dan (2) jumlah *batch data*. Peningkatan jumlah *layer* pada model LSTM cenderung membuat model menjadi *overfit*, sehingga dapat menurunkan hasil akurasi model. Kondisi *overfit* pada model LSTM dapat dihindari dengan menyederhanakan struktur model, salah satunya dengan menurunkan jumlah *layer*. *Loss* yang dihasilkan pada jumlah *layer* yang banyak juga cenderung stagnan dan tidak mengalami penurunan secara signifikan. Berikut contoh nilai *loss* yang stagnan yang bersumber dari salah satu hasil akurasi model LSTM:



Gambar 5.2. Nilai *Loss* yang Stagnan pada Model LSTM

Gambar tersebut (Gambar 5.2.) merepresentasikan ketidakefektifan proses *training* yang dilalui oleh model. Nilai *loss* yang tidak kunjung menurun menandakan proses *training model* sudah tidak lagi bermanfaat. *Loss* yang cenderung stagnan mengakibatkan rendahnya tingkat akurasi yang dihasilkan.

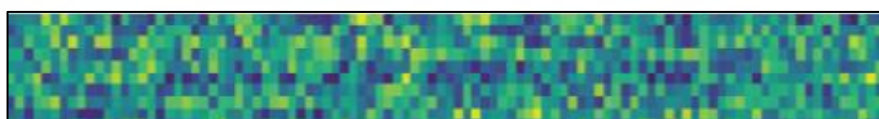
Selain *loss*, jumlah *batch data* juga mempengaruhi tingkat akurasi. Hasil akurasi tertinggi diraih dengan *batch size* paling minim yakni 1.

Setelah memperoleh konfigurasi terbaik untuk model LSTM, maka model LSTM diujicoba dengan menjalankan 4 eksperimen. Berikut hasil dari keempat eksperimen yang telah dilakukan:

1. Eksperimen dengan *datasets* tidak proporsional (13,715 *tweets*).

Pada eksperimen pertama, hasil akurasi yang diperoleh oleh model LSTM hanya mencapai 48.72%. Rendahnya hasil akurasi menunjukkan bahwa LSTM mengalami *overfitting*. Hal tersebut dikarenakan adanya kecenderungan model terhadap salah satu kelas yang memiliki jumlah data terbesar, yakni *tweets* dengan kelas positif. Dalam hal ini, model cenderung mengklasifikasi seluruh *tweets* yang diberikan ke dalam kelas positif. Adapun jumlah *layer* yang digunakan yakni: 10 *layers*.

Berikut persebaran *weight*nya:



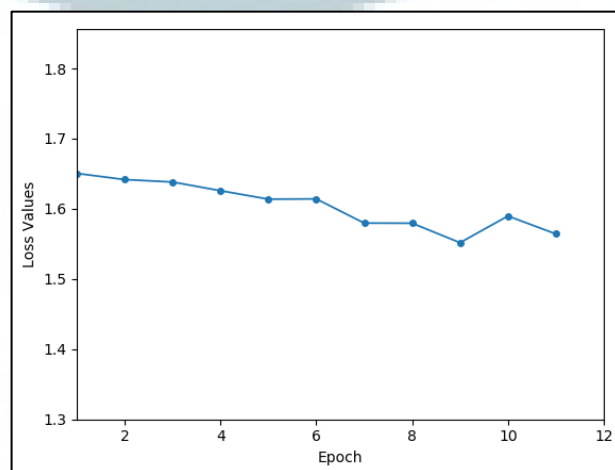
Gambar 5.3. Distribusi *weight* pada eksperimen pertama dengan 10 *layers* (13,715 *tweets*)

Pada gambar di atas (Gambar 5.3.), intensitas kedalaman warna merepresentasikan nilai *weight*. Semakin gelap kedalaman warnanya, maka semakin kecil pula nilai *weight*, begitu juga sebaliknya. Sumbu x pada gambar tersebut merepresentasikan jumlah *weight*, dan sumbu y

menampilkan jumlah *fold*, yakni 1-10. Persebaran *weight* tersebut menandakan dua hal:

- Distribusi *weight* didominasi oleh warna gelap. Hal ini menandakan bahwa *weight* yang dihasilkan bernilai kecil, dan tidak berpengaruh banyak pada peningkatan akurasi sesi latihan model.
- *Weight* dengan nilai yang besar, yang ditandai dengan intensitas warna cerah tidak terdistribusi dengan baik. Hal ini menandakan bahwa model tidak terlatih secara optimal.

Dalam eksperimen ini, model mengalami kecenderungan kepada salah satu kelas *tweets*, yakni kelas positif. Hal tersebut diduga akibat ketidakseimbangan jumlah *datasets* antar kelas, dimana *tweets* bermakna positif jauh lebih banyak frekuensinya (12,293 *tweets*) dibandingkan *tweets* negatif (1,422 *tweets*). Berikut *loss*nya:

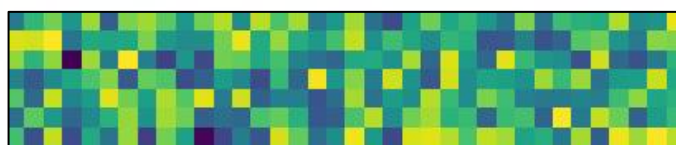


Gambar 5.4. *Loss* pada eksperimen pertama dengan 10 *layers* (13,715 *tweets*)

Pada gambar di atas (Gambar 5.4.), nilai *loss* masih berkisar antara: 1.5 – 1.6. Nilai *loss* yang dihasilkan masih cenderung tinggi, dan hanya menurun sedikit pada saat proses *training* berlangsung. Hal ini menandakan bahwa proses *training* tidak berlangsung secara optimal.

2. Eksperimen dengan *datasets* proporsional yang jumlahnya diperbanyak sejumlah *ratio*: 1.2 (3,412 *tweets*) menggunakan 10 *layers*.

Pada eksperimen kedua, hasil akurasi model yakni: 50.41%. Hasil akurasi mengalami sedikit peningkatan, meski hasilnya masih tak terpaut jauh dengan eksperimen sebelumnya (1). Meski model LSTM telah diberikan *datasets* proporsional, dimana jumlah *datasets* kelas positif sama dengan kelas negatif, hasil akurasinya masih rendah. Sama dengan eksperimen sebelumnya (1), model LSTM masih mengalami *overfitting*. Penyebab terjadinya *overfitting* diduga implementasi jumlah *layers* yang terlalu kompleks, sehingga model LSTM gagal mempelajari data secara optimal. Berikut persebaran *weight*nya:

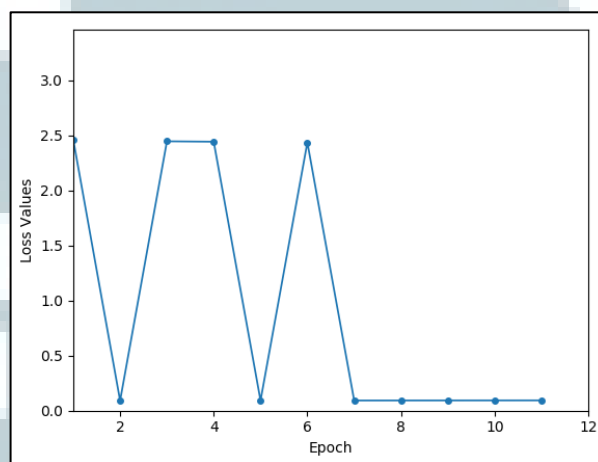


Gambar 5.5 Distribusi *weight* pada eksperimen kedua dengan 10 *layers* (3,412 *tweets*)

Gambar tersebut (Gambar 5.5.) merepresentasikan distribusi *weight* yang sudah tersebar sedikit lebih baik dari eksperimen sebelumnya (1). Pada gambar di atas (Gambar 5.5.), warna kuning sudah cukup

mendominasi, meski dengan distribusi yang cukup acak. Hal ini menandakan bahwa model sudah terlatih sedikit jauh lebih baik dibandingkan eksperimen sebelumnya (1), meski sesi latihan model masih jauh dari optimal.

Berikut *loss* pada eksperimen kedua:



**Gambar 5.6. *Loss* pada eksperimen kedua dengan 10 layers
(3,412 tweets)**

Berdasarkan gambar (Gambar 5.6.), *loss* yang dihasilkan cukup tinggi pada awal proses *training* model berlangsung, kemudian mengalami penurunan seiring dilatihnya model, atau tepatnya ketika model memasuki *epoch* ke-8. Hal ini menandakan bahwa model sudah cukup terlatih dengan lebih baik dibandingkan eksperimen sebelumnya: (1), meskipun masih belum terlatih secara optimal. Adapun nilai *loss* berkisar antara: 0.09 – 2.4.

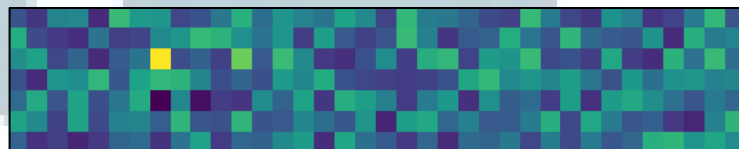
3. Eksperimen dengan *datasets* proporsional yang diperbanyak sebesar 2 kali dari jumlah *datasets* pada eksperimen no: (2) dengan konfigurasi 10 *layers* (6,824 *tweets*).

Pada eksperimen ketiga, model LSTM memperoleh hasil akurasi yang hampir serupa dengan eksperimen sebelumnya (2), yakni: 50.26%.

Meski jumlah *datasets* proporsional telah digandakan, namun hasil akurasi model masih belum mengalami peningkatan secara signifikan.

Model LSTM masih *overfitting* seperti eksperimen sebelumnya (1), dan (2). Penyebab *overfitting* diduga atas kompleksnya jumlah *layer* yang dimiliki oleh model. Untuk membuktikan hal tersebut, model LSTM pada eksperimen selanjutnya (4) diberikan jumlah *layer* sebanyak: 1.

Berikut distribusi *weight* pada eksperimen ketiga:

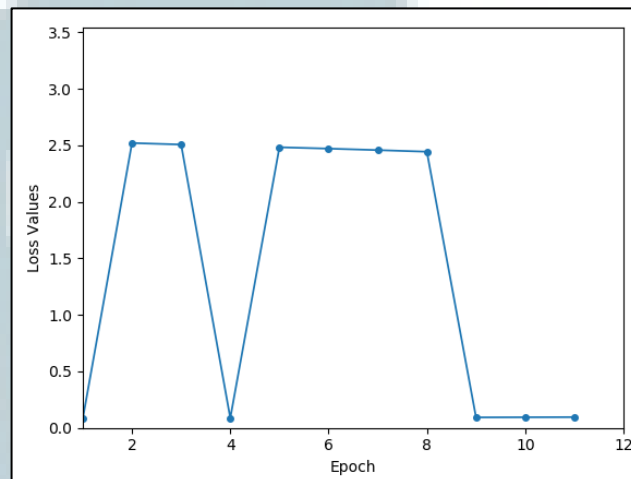


Gambar 5.7. Distribusi *weight* pada eksperimen ketiga dengan 10 *layers* (6,824 *tweets*)

Gambar di atas (Gambar 5.7.) menunjukkan bahwa *weight* tidak terdistribusi dengan baik. Hal tersebut ditandai dengan dominasi warna biru, dan hanya satu warna kuning yang terdapat pada gambar tersebut (Gambar 5.7.). Dalam hal ini, *weight* yang dihasilkan bernilai kecil, sehingga tidak terlalu berpengaruh dalam meningkatkan akurasi model LSTM. *Weight* yang bernilai besar yang ditandai dengan warna kuning

justru tidak tersebar dan hanya ada satu. *Weight* yang tidak terdistribusi dengan baik menyebabkan model LSTM tidak terlatih secara optimal.

Berikut *loss* pada eksperimen ketiga:



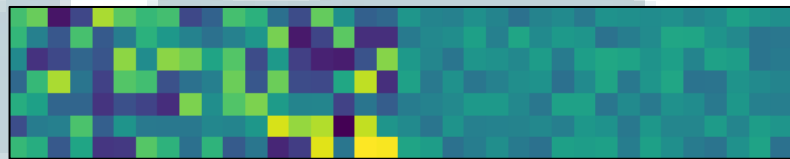
Gambar 5.8. *Loss* pada eksperimen ketiga dengan 10 layers (6,824 tweets)

Sesuai gambar di atas (Gambar 5.8.), nilai *loss* masih berkisar antara: 0.08 – 2.51. *Loss* sempat mengalami fluktuasi pada awal *training* model, namun pada akhirnya mengalami penurunan yang cukup drastis. Hal ini menandakan bahwa model sudah cukup terlatih dengan baik, meski masih belum optimal.

4. Eksperimen dengan *datasets* proporsional yang diperbanyak sebesar 2 kali dari jumlah *datasets* pada eksperimen no: (2) dengan konfigurasi 1 layer (6,824 tweets).

Pada eksperimen keempat, model LSTM berhasil meraih hasil akurasi sebanyak: 77.92%. Implementasi jumlah *layer* yang minim membuat model tidak lagi mengalami *overfitting*, dan mulai terlatih secara

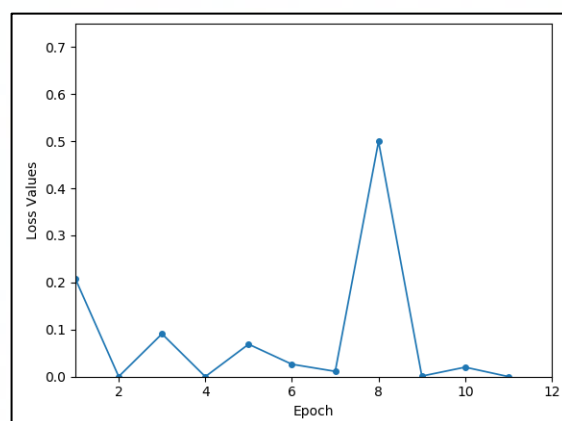
optimal. Berdasarkan seluruh hasil eksperimen, model LSTM membutuhkan jumlah *datasets* yang seimbang antar kelas, dan konfigurasi yang tepat. Dalam hal ini, konfigurasi sederhana mencegah model LSTM mengalami *overfitting*. Berikut distribusi *weight* pada eksperimen keempat:



Gambar 5.9. Distribusi *weight* pada eksperimen keempat dengan 1 layer (6,824 tweets)

Pada gambar di atas (Gambar 5.9.), mulanya *weight* sudah cukup terdistribusi dengan baik. Namun, *weight* hampir mengalami *zero-gradient* pada bagian akhir.

Berikut *loss* pada eksperimen keempat:



Gambar 5.10. *Loss* pada eksperimen keempat dengan 10 layers (6,824 tweets)

Berdasarkan gambar (Gambar 5.10.), nilai *loss* yang dihasilkan sudah jauh lebih rendah dibandingkan eksperimen sebelumnya; (1), (2), dan (3) dengan rentang: 0.0001 – 0.02. Meski nilai *loss* sempat mengalami fluktuasi pada pertengahan sesi latihan, namun nilai *loss* kemudian menurun kembali. Hal ini menandakan bahwa model sudah terlatih dengan optimal.

Setelah melakukan keempat eksperimen pada model LSTM, kinerja tertinggi diraih pada eksperimen keempat, yakni: LSTM dengan jumlah *datasets* proporsional sebanyak 6,824 *tweets* menggunakan 1 *layer*. Hal tersebut dapat dilihat dari nilai *loss* yang diperoleh, dimana nilai *loss* dengan rentang terkecil, yakni: 0.0001 – 0.02 terdapat pada eksperimen keempat, dan *weight* sudah cukup terdistribusi dengan baik pada eksperimen keempat, meski hampir mengalami *zero-gradient*. Distribusi *weight* terburuk diperoleh pada eksperimen ketiga, dimana nilai *weight* didominasi oleh nilai *weight* yang relatif kecil, dan *weight* dengan nilai terbesar tidak terdistribusi.

U
M
N