

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Analisis Persoalan

Pada penelitian ini digunakan metode *Interactive Attention Network* (IAN) untuk mendapatkan sebuah model klasifikasi sentimen berbasis target terbaik pada dataset *twitter*. Adapun tahap analisisnya adalah sebagai berikut.

- a. Penelitian ini dilakukan pada *target-depedent* dataset dari Dong dkk [10] yang berisi data *tweet* pengguna *twitter* dan dikonversi dalam bentuk format csv. Sebelum dikonversi dataset *tweet* dilakukan proses *shifting* untuk mendapatkan letak awal dan letak akhir dari kata target.
- b. Membagi data *testing* menjadi dua bagian yaitu data *testing* dan data *validation*.
- c. Seluruh *word embedding* dari konteks dan target diinisialisasi menggunakan *300-dimension word vectors pre-trained* dari GloVe<sup>1</sup> [25]
- d. Dengan *word embedding* sebagai *input*, pada penelitian ini menggunakan LSTM *networks* untuk memperoleh *hidden states* dari kata untuk target dan konteks.
- e. Nilai rata-rata dari *hidden states* target dan *hidden states* konteks di-*supervise* untuk membangkitkan vektor *attention*.
- f. *Attention mechanism* digunakan untuk men-*capture* informasi penting dari konteks dan target.
- g. Selanjutnya dilakukan pembangkitan representasi dari target dan konteks secara interaktif.
- h. Representasi dari konteks dan target digabungkan sebagai representasi akhir yang nilainya dikirimkan ke fungsi *softmax* untuk klasifikasi sentimen dependen terhadap target.
- i. Melakukan proses *testing* menggunakan model dari proses *training* untuk memprediksi dataset *tweet* pengguna *twitter*.

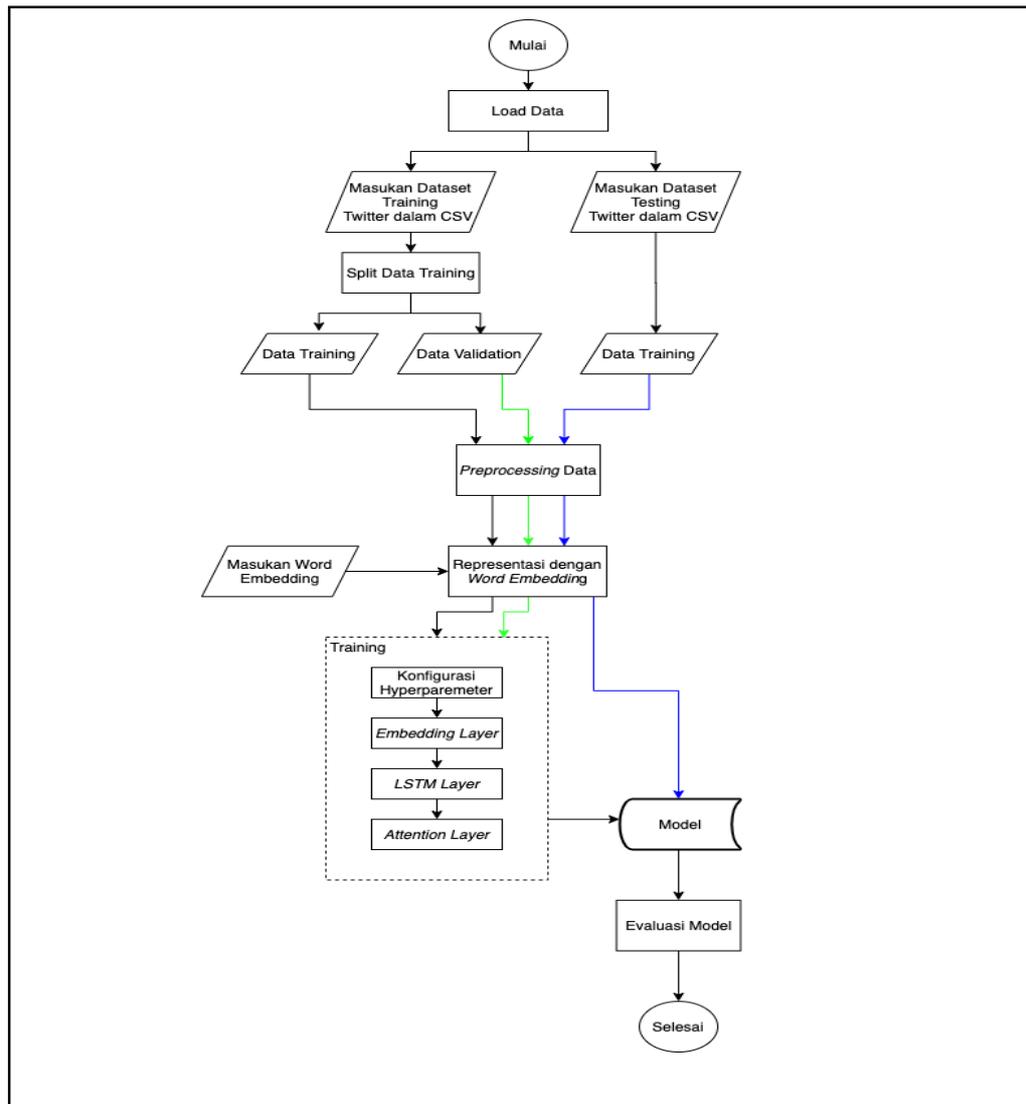
---

<sup>1</sup> Pre-trained word embedding dari Glove dapat diunduh melalui <http://nlp.stanford.edu/projects/glove>

- j. Evaluasi dengan menghitung nilai akurasi dan *macro f-1*.

### 3.2 Metodologi dan Arsitektur Umum

Pada sub bab ini, membahas langkah-langkah metodologi penelitian yang dilakukan berdasarkan tujuan pada penelitian ini. Gambar 3.1 merupakan diagram alir yang digunakan untuk mencari model IAN yang digunakan untuk klasifikasi sentimen dependen terhadap target pada dataset *twitter*.



Gambar 3. 1 Diagram alir metodologi penelitian.

### 3.3 Analisis Dataset

Pada penelitian ini, disiapkan kumpulan data *tweet* yang digunakan pada penelitian Dong dkk. [10], yang dilakukan anotasi secara manual dengan label sentimen

negatif, positif dan netral dengan terhadap target yang diberikan (contoh, "bill gates, "google", dan "xbox"). Konteks dan target sudah diidentifikasi pada dataset, Tabel 3.1 menunjukkan contoh dataset yang digunakan.

Tabel 3. 1 Contoh data pada dataset.

Konteks	Target	Sentimen
my 3-year-old was amazed yesterday to find that ' real ' 10 pin bowling is nothing like it is on the \$\$\$ ...	wii	0 (netral)
3 by \$\$\$ is an amazing song	britney spears	1 (positif)
The \$\$\$ domestic genocide hug .	obama	-1 (negatif)
i can honestly say i think lindsay lohan is crazy . but she is a riot to follow ... that 's for sure	lindsay lohan	-1 (negatif)
\$\$\$ is suing a former business manager for bad advice & mismanagement . he should file a similar suit against his hairdresser	nicolas cage	0 (netral)
google 's interview with \$\$\$ has pretty	lady gaga	1 (positif)

Konteks	Target	Sentimen
awesome intro : . via nicolehering		
really not sure i like the richard gere \$T\$ love affair thing	hilary swank	0 (e

Pada dataset terdapat 6248 *tweet* digunakan sebagai data *training* dan 692 *tweet* digunakan sebagai data *testing*, dengan jumlah positif, negatif dan netral (25%, 25%, dan 50%). Dong dkk. [10] melaporkan bahwa dataset dianotasi dengan 82.5% kesepakatan antara anotator manusia. Data tersebut dengan format *file text* (TXT) dan data terdiri dari *tweet*, target, dan sentimen. Format data tersebut dikonversi dalam bentuk *Comma Separated Value* (CSV) dengan kolom *content*, *aspect*, *sentiment*, *from*, dan *to* untuk mempermudah pemrosesan sistem pada tahap selanjutnya. Selanjutnya data *training* diambil 10% untuk digunakan sebagai data *validation*. Tabel 3.2 merupakan rincian dari dataset *twitter* yang digunakan pada penelitian ini.

Tabel 3. 2 Deskripsi dataset.

Dataset <i>training</i>	5623
Dataset <i>validation</i>	625
Dataset <i>testing</i>	692

### 3.4 Preprocessing Data

Pada tahap *preprocessing data* terdapat beberapa langkah sebagai berikut:

- a. Pada penelitian ini dilakukan *shift data* untuk membuat sampel *input* yang digunakan pada tahap *training* dan *testing*. *Shift data* yang pertama adalah membuat kolom *from*, kolom *from* adalah posisi awal dari kata target pada kalimat. Tabel 3.3 adalah hasil pembuatan kolom *from* pada dataset.

Tabel 3. 3 Dataset dengan *shift data from*.

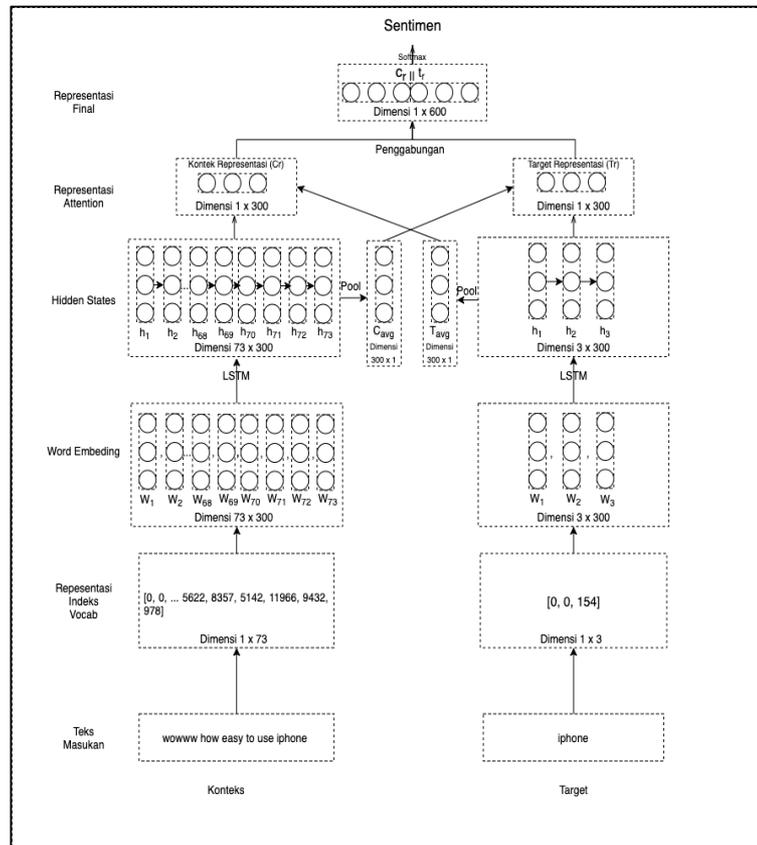
<i>Content</i>	<i>Aspect</i>	<i>Sentiment</i>	<i>From</i>
hahah yeah i know but the also good thing is that im good friends with britney spears ! lol   : p	britney spears	1	71
lawmakers challenge obama 's libya authority	obama	1	20
and they were tryna kill us all and everyone was like shit run ! then it went normal again : l and nick jonas was sitting on a chair all sad	nick jonas	0	99

- b. Selanjutnya, *shift data* yang kedua adalah membuat kolom *to*. Kolom *to* adalah posisi akhir dari kata target pada kalimat. Tabel 3.4 menunjukkan pembuatan kolom *to* pada dataset.
- c. Selanjutnya adalah tahap word embedding pada data konteks dan target menggunakan GloVe [25].

Tabel 3. 4 Dataset dengan *shift data to*.

<i>Content</i>	<i>Aspect</i>	<i>Sentiment</i>	<i>From</i>	<i>To</i>
hahah yeah i know but the also good thing is that im good friends with britney spears ! lol   : p	britney spears	1	71	85
lawmakers challenge obama 's libya authority	obama	1	20	25
and they were tryna kill us all and everyone was like shit run ! then it went normal again : l and nick jonas was sitting on a chair all sad	nick jonas	0	99	109

### 3.5 Rancangan Solusi



Gambar 3. 2 Arsitektur *Interactive Attention Network*.

Pada sub bab ini dijelaskan rancangan solusi menggunakan arsitektur *interactive attention network* (IAN) yang ditunjukkan pada Gambar 3.2. Seperti yang ditunjukkan pada Gambar 3.2 diambil contoh konteks yaitu "woowww how easy to use iphone" dan target pada konteks tersebut adalah "iphone", untuk masukan pada lapisan *word embedding* kalimat pada dataset terlebih dahulu diberikan nilai indeks nya pada tiap kata, contoh dari indeks kata pada target dan konteks dapat dilihat pada Tabel 3. 5.

Tabel 3. 5 Representasi Indeks Kata.

Jenis Kalimat	Kalimat	Representasi Indeks
Konteks	woowww how easy to use iphone	[5622, 8357, 5142, 11966, 9432, 9432, 978]
Target	iphone	[154]

### 3.5.1 Representasi Indeks *Vocab*

Untuk mengatur panjang urutan vektor digunakan jumlah maksimal banyaknya kata dalam suatu konteks dan target pada keseluruhan dataset, karena setiap target dan konteks memiliki jumlah kata yang berbeda-beda, maka untuk mengisi panjang vektor agar semua sama panjang digunakan *pad\_sequences* yang ada pada *keras* untuk mengisi urutan kata dengan angka 0 secara otomatis. Sehingga jika terdapat kalimat yang pendek maka makin banyak pula angka 0 pada *feature vektor* tersebut. Pada penelitian ini jumlah maksimal panjang kata pada konteks adalah 73 dan pada target adalah 3. Contoh representasi *pad\_sequence* pada salah satu konteks dan target dapat dilihat pada Tabel 3.6 Struktur data sudah dapat diolah untuk selanjutnya menjadi input data pada lapisan *word embedding*.

Tabel 3. 6 Representasi *Pad\_Sequence*.

Jenis Kalimat	Kalimat	Representasi Indeks	Representasi Pad_Sequence	Panjang
Konteks	wooww how easy to use iphone	[5622, 8357, 5142, 11966, 9432, 978]	[0, 0, ... , 5622, 8357, 5142, 11966, 9432, 978 ]	73
Target	iphone	[154]	[ 0, 0, 154 ]	3

### 3.5.2 *Word Embedding*

Pada proses untuk merepresentasikan sebuah kata, Ma dkk. [9] menyimpan tiap kata menjadi sebuah *vektor real-value low-dimentional*, atau yang disebut juga *word embedding* [22]. Pada tahap *word embedding* dimasukkan beberapa parameter yaitu *input\_dim* yang bernilai 13519 pada konteks dan 178 pada target dan *output\_dim* yaitu jumlah *neuron* yang pada penelitian ini bernilai 300. Selanjutnya didapatkan  $w^k \in R^d$  dari  $M^{v \times d}$ , dimana  $k$  adalah index kata pada konteks atau target,  $d$  adalah dimensi *embedding* dan  $v$  adalah ukuran dari *vocabulary*. Sehingga pada penelitian ini didapatkan variabel input berupa *word embedding tensor* konteks dan target yang selanjutnya dimasukkan ke lapisan LSTM. Gambar 3.3 adalah isi dari vektor *word embedding* konteks dan target.

```

array([[ 0.0000000e+00,  0.0000000e+00,  0.0000000e+00, ...,
         0.0000000e+00,  0.0000000e+00,  0.0000000e+00],
       [ 2.3511000e-01, -2.1643000e-02,  1.3349000e-01, ...,
        -4.3313000e-01,  3.8047999e-01,  2.7798000e-01],
       [-8.2003552e-04,  1.3818503e-04, -2.3892927e-03, ...,
        2.5631068e-03,  6.4901303e-04, -1.3273649e-03],
       ...,
       [ 4.7389001e-01, -2.7373999e-01,  2.7880999e-01, ...,
        -1.2738000e-01, -3.5618001e-01, -4.2798002e-03],
       [-2.1952000e-01,  2.0062999e-01, -2.4699999e-01, ...,
        2.3988000e-01,  7.7310002e-01, -2.1836001e-01],
       [ 0.0000000e+00,  0.0000000e+00,  0.0000000e+00, ...,
        0.0000000e+00,  0.0000000e+00,  0.0000000e+00]], dtype=fl
oat32)

array([[ 0.          ,  0.          ,  0.          , ...,  0.          ,  0.
',
       [ 0.          ],
401 [ -0.18849   ,  0.064902  ,  0.087744  , ...,  0.12711   ,  0.28
',
       [ -0.35852   ],
914 [ -0.59174   ,  0.46521   , -0.29661   , ...,  0.17906   ,  0.32
',
       [ 0.46797   ],
       ...,
354 [ -0.090702  ,  0.44452   , -0.74577   , ..., -0.51422   , -0.34
',
       [ 0.20713   ],
132 [ 0.32344   ,  0.38108   , -0.39391   , ..., -0.2093   ,  0.22
',
       [ -0.0035063 ],
       [ 0.          ,  0.          ,  0.          , ...,  0.          ,  0.
',
       [ 0.          ]], dtype=float32)

```

Gambar 3. 3 *Word Embedding* Konteks dan Target.

### 3.5.3 Pembangkitan *Hidden States*

Selanjutnya, Ma dkk. [9] menggunakan LSTM *network* untuk melakukan *learning* dari *hidden word semantic*. Penggunaan metode tersebut pada penelitian [9] dikarenakan kata-kata dalam kalimat memiliki ketergantungan yang kuat satu sama lain, dan LSTM cukup bagus untuk melakukan *learning* pada *long-term dependencies* dan bisa menghindari *vanishing gradient problem* dan *expansion problem* [9]. Secara formal, nilai masukan *word embedding*  $w^k$ , *previous cell state*  $c^{k-1}$  dan *previous hidden state*  $h^{k-1}$ , *current cell state*  $c^k$  dan *current hidden state*  $h^k$  di jaringan LSTM diperbarui dengan Persamaan 2.10, 2.11, 2.12, 2.13, 2.14 dan

2.15. Pada proses ini didapatkan *hidden states*  $[h_c^1, h_c^2, h_c^3, \dots, h_c^n]$  sebagai representasi dari konteks dan *hidden states* dari target  $[h_t^1, h_t^2, h_t^3, \dots, h_t^n]$ . Contoh representasi dari *hidden states* target dan konteks dapat dilihat pada Gambar 3. 4 dan 3. 5.

```
array([[2.3481000e-01, -0.13488000e-01, 0.01481000e-01, -
0.13988000e-01, ..., 0.1321000e-01, -0.3988000e-01],
[0.03481000e-01, -0.12388000e-01, 1.34381000e-01, -
0.41988000e-01, ..., 0.3411000e-01, -0.4988000e-01],
...
[1.3431000e-01, -0.1188000e-01, 0.1381000e-01, -0.7148000e-
01, ..., 1.4281000e-01, -0.3131000e-01]], dtype=float32)
```

Gambar 3. 4 Contoh *Hidden States* Konteks.

```
array([[2.3481000e-01, -0.3988000e-01, 2.3481000e-01,
-0.1982000e-01, ..., 1.3481000e-01, -0.1981000e-01],
[1.2291000e-01, -0.1428000e-01, 1.4141000e-01, -
0.1948000e-01, ..., 2.3411000e-01, -0.1488000e-01],
[2.1484000e-01, -1.3918000e-01, 1.1481000e-01, -
0.3188000e-01, ..., 2.3431000e-01, -0.3981000e-01]],
dtype=float32)
```

Gambar 3. 5 Contoh *Hidden States* Target.

### 3.5.4 Pembangkitan Vektor *Attention*

Setelah mendapatkan *hidden states* konteks dan target, tahap selanjutnya adalah mendapatkan representasi target dan konteks menggunakan persamaan 2.16 dan 2.17. Berdasarkan *hidden states* target dan konteks maka diambil nilai rata-rata nya menggunakan persamaan 2.16 dan 2.17, sehingga diperoleh nilai  $c_{avg}$  dan  $t_{avg}$  yang merupakan representasi dari target dan konteks. Representasi ini digunakan untuk pembangkitan *attention*.

Tahap berikutnya yaitu mempertimbangkan nilai target  $c_{avg}$  dan konteks  $t_{avg}$  untuk pembangkitan *attention* yang digunakan untuk memilih informasi penting yang berkontribusi untuk menetapkan polaritas sentimen. Ma dkk. [9]

mempertimbangkan pengaruh pada konteks dari target dan pengaruh pada target dari konteks, yang dapat memberikan lebih banyak petunjuk untuk memperhitungkan fitur sentimen terkait seperti yang dinyatakan pada Gambar 3.2.

Pembangkitan *attention* dimulai dengan membangkitkan vektor *attention*  $\alpha_i$  menggunakan representasi target  $t_{avg}$  yang sudah didapatkan dengan Persamaan 2.17. Pembangkitan vektor *attention*  $\alpha_i$  dilakukan dengan menerapkan persamaan 2.18. Selanjutnya dibangkitkan vektor *attention*  $\beta_i$  menggunakan representasi konteks  $c_{avg}$  yang sudah didapatkan dengan persamaan 2.16. Untuk membangkitkan vektor *attention*  $\beta_i$  diterapkan persamaan 2.20.

Setelah mendapatkan vektor *attention*  $\alpha_i$  dan  $\beta_i$ , tahap selanjutnya adalah membangkitkan representasi target  $c_r$  dan konteks  $t_r$  menggunakan *attention*  $\alpha_i$  dan  $\beta_i$  menggunakan persamaan 2.21 dan 2.22. Pada akhirnya, Setelah mendapatkan representasi target  $c_r$  dan konteks  $t_r$ , dilakukan mengabungkan representasi target  $c_r$  dan konteks  $t_r$  dalam bentuk vektor yang digunakan sebagai *classifier*.

### 3.5.5 Probabilitas Sentimen

Representasi vektor hasil penggabungan representasi target  $c_r$  dan konteks  $t_r$  diproyeksikan menggunakan *non-linear layer* ke ruang  $C$  yang ditargetkan dengan perhitungan yang ditunjukkan pada Persamaan 2.23. Akhir dari arsitektur IAN adalah menghitung probabilitas sentimen menggunakan persamaan 2.24 dimana dipilih label positif, negatif atau netral dengan memilih probabilitas terbesar diantara nilai dari tiga label tersebut dan digunakan sebagai hasil akhir.

## 3.6 Parameter Pada Arsitektur Interactive Attention Network

Dapat dilihat pada Gambar 3.6 merupakan ringkasan model pada arsitektur Interactive Attention Network (IAN). *Param #* pada ringkasan model menunjukkan jumlah parameter yang dimiliki oleh lapisan tertentu, parameter disini adalah bobot dan atau bias pada tiap layer. Pada *Layer* model terdapat 5.731.576 parameter. *Layer* model terdiri dari *embedding layer*, *LSTM layer* dan *attention layer*. Pada arsitektur IAN terdapat dua *dense layer* dengan *dense layer* pertama memiliki

parameter 76.928 dan dense layer kedua memiliki parameter 387. Detail perhitungan pada tiap layer terdapat pada Tabel 3.7.

Layer (type)	Output Shape	Param #	Connected to
input_text (InputLayer)	(None, 73)	0	
input_aspect_text (InputLayer)	(None, 3)	0	
model_1 (Model)	(None, 600)	5731576	input_text[0][0] input_aspect_text[0][0]
dense_1 (Dense)	(None, 128)	76928	model_1[1][0]
dense_2 (Dense)	(None, 3)	387	dense_1[0][0]
Total params: 5,808,891			
Trainable params: 1,699,791			
Non-trainable params: 4,109,100			

Gambar 3. 6 Ringkasan Model.

Dapat dilihat pada Gambar 3.6 terdapat *trainable params* dan *non-trainable params*, *trainable params* adalah parameter yang dapat dilakukan optimasi nilainya pada saat *training*, sedangkan *non-trainable params* adalah parameter yang tidak dapat dioptimasi nilainya pada saat *training*, pada penelitian ini *non-trainable params* adalah *embedding layer*.

Tabel 3. 7 Perhitungan Parameter IAN.

Embedding Konteks	vocab * dim_embedding	13519 * 300	4055700
Embedding Target	vocab * dim_embedding	178 * 300	53400
LSTM Target	$g \times [h(h+i) + h]$	$4 * (300 * (300 + 300) + 300)$	721200
LSTM Konteks	$g \times [h(h+i) + h]$	$4 * (300 * (300 + 300) + 300)$	721200
Attention Konteks	(dim_attention * dim_attention) + bias	$(300 * 300) + 73$	90073
Attention Target	(dim_attention * dim_attention) + bias	$(300 * 300) + 3$	90003
Dense Layer 1	(dim_attention concat * dim_dense 1) + bias	$(600 * 128) + 128$	76928
Dense Layer 2	(dim_dense 1 * dim_dense 2) + bias	$(128 * 3) + 3$	387
Parameter Model			5731576
Parameter Dense			77315
Total Parameter			5808891
<i>Non-Trainable</i> Parameter (Embedding)			4109100
Trainable Parameter (Total Parameter - Non-Trainable Parameter)			1699791

### 3.7 Desain Eksperimen Interactive Attention Network

Pada penelitian ini, digunakan sejumlah parameter yang perlu dicari pengaturan terbaiknya, dimana parameter serta penggunaan rentang/pilihan nilainya dapat diamati pada Tabal 3.8. Pada penelitian ini Seluruh *word embedding* dari konteks dan target diinisialisasi menggunakan *300-dimension word vectors pre-trained GloVe* [25]. Semua kata *out-of-vocabulary* diinisiasi dengan *sampling* dari *uniform distribution*  $U(-0.1, 0.1)$ . Semua matriks bobot diberikan nilai awal dengan

sampling dari *uniform distribution*  $U(-0.1, 0.1)$ , dan semua bias diatur dengan nilai nol.

Selain itu, dimensi untuk *word embeddings*, *vektor attention*, dan *LSTM hidden states* diatur dengan nilai 300 [12]. Untuk melatih semua parameter *interactive attention network* (IAN), Ma dkk. [9] menggunakan Momentum [26], yang menambahkan *fraction*  $\gamma$  dari vektor pembaruan pada langkah sebelumnya ke vektor pembaruan saat ini. Penggunaan *pooling function* pada interaksi antara konteks dengan target. Koefisien normalisasi  $L_2$  dalam fungsi objektif diatur ke  $10^{-5}$ , *dropout rate* diatur dengan nilai 0,5.

Pada penelitian ini, untuk mendapatkan hasil yang terbaik dari model arsitektur IAN, proses *training* digunakan parameter yang berbeda-beda agar mendapatkan model yang terbaik. Dari sejumlah model yang dihasilkan dibandingkan serta dianalisis dari nilai akurasi serta *macro-f1*. Adapun ketentuan nilai parameter yang digunakan untuk pembuatan model arsitektur IAN sebagai berikut.

Pada Tabel 3.8 terdapat beberapa parameter yang dianalisis pengaruhnya terhadap model pada penelitian ini, yaitu *embedding tuning*, *pooling function* dan *sequence model*. Pada *embedding tuning* digunakan *fixed word embedding* yang mana tidak memperbarui bobot *embedding* selama pelatihan dan *fine tuned word embedding* yang mana memperbarui bobot *embedding* selama pelatihan.

*Pooling function* pada penelitian ini digunakan untuk mendapatkan representasi awal dari target dan konteks, pada penelitian ini dianalisis variasi dari empat *pooling function*, yaitu *max* dimana diambil nilai terbesar untuk representasi awal dari target dan konteks, *min* dimana diambil nilai minimal untuk representasi awal dari target dan konteks, *avg* dimana diambil nilai rata-rata untuk representasi awal dari target dan konteks, dan *avg + max* dimana dikombinasi nilai rata-rata dan nilai maksimal untuk representasi awal dari target dan konteks.

Tabel 3. 8 Parameter pengujian model arsitektur IAN.

Parameter	Keterangan
<i>Uniform Distribution</i>	U (-0.1, 0.1)
<i>Word Embedding Dimention</i>	300
<i>Vektor Attention Dimention</i>	300
<i>LSTM Hiddent States Dimention</i>	300
<i>Dropout Rate</i>	0.5
<i>Embedding Tuning</i>	<i>fixed embeddings</i> atau <i>fine-tuned embeddings</i>
<i>Pooling Function</i>	Min, Avg, Max, Avg + Max
<i>Sequence Models</i>	LSTM, GRU, BiLSTM

Pada penelitian ini juga dianalisis efek dari perbedaan model sekuensial pada arsitektur IAN, ada tiga model sekuensial yang dicobakan pada penelitian ini, yaitu *long short-term memory* (LSTM), *gated recurrent unit* (GRU), dan *bi-directional long short-term memory* (Bi-LSTM).

Tabel 3. 9 Tahapan pengujian parameter.

Tahap ke	Parameter
1	<i>Embedding Tuning</i>
2	<i>Pooling Function</i>
3	<i>Sequence Models</i>

Pengujian dilakukan pada setiap parameter, dimana hasil pengujian parameter yang telah didapat digunakan untuk pengujian berikutnya sehingga mendapatkan model yang terbaik. Tabel 3.9 adalah tahapan pengujian parameter.

### 3.8 Evaluasi dan Hasil Klasifikasi

Tahapan evaluasi berfungsi untuk menghitung seberapa baik sistem dalam melakukan klasifikasi sentimen berbasis target. Pada tahap evaluasi, penelitian ini menggunakan skema penelitian Dong dkk. [10], Ma dkk. [9], Wang dkk. [12] digunakan dua langkah evaluasi yaitu, *accuracy* dan *macro-average F1-Score*. *Accuracy* didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual yang didefinisikan pada Persamaan 3.1. *Macro-average F1-Score* merupakan perbandingan rata-rata *precision* dan *recall* pada tiap *class*. *Precision* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem jika dianalogikan dengan penelitian ini, *precision* dapat menjawab pertanyaan "Jika diketahui suatu *tweet* ditandai sentimen positif oleh *classifier*, berapa peluang *tweet* tersebut benar memiliki sentimen positif?". *Recall* adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi, *recall* dapat menjawab pertanyaan "Jika diketahui suatu *tweet* memiliki sentimen positif, berapa peluang *classifier* dapat memprediksi *tweet* tersebut memiliki sentimen positif juga?". *Class* pada penelitian ini adalah sentimen positif, negatif dan netral. Perhitungan *precision*, *recall*, *f1-score* serta *macro-f1* didefinisikan pada Persamaan 3.2, 3.3, 3.4, 3.5

$$Accuracy = \frac{T}{N} \quad (3.1)$$

Dimana T adalah jumlah sampel yang diprediksi secara benar, N adalah total jumlah sampel. Hasil klasifikasi menampilkan polaritas sentimen dari kalimat terkait.

$$Precision = \frac{T_p}{T_p + F_p} \quad (3.2)$$

$$Recall = \frac{T_p}{T_p + F_n} \quad (3.3)$$

$$F1-Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \quad (3.4)$$

$$MacroF1 = \frac{\sum f1score}{jumlah\ class} \quad (3.5)$$

Untuk mendapatkan *precision*, *recall* dibutuhkan paramater sebagai berikut:

1. *True positives* ( $T_p$ ): jumlah sentimen yang dipilih dengan benar.
2. *False negatives* ( $F_n$ ): jumlah sentimen yang benar yang tidak dipilih.
3. *False positives* ( $F_p$ ): jumlah sentimen yang salah yang dipilih.
4. *True negatives* ( $T_n$ ): jumlah sentimen yang salah yang tidak dipilih.

### 3.9 Jadwal Penelitian

Pelaksanaan penelitian ini dilaksanakan selama tujuh bulan dari bulan Oktober 2019 sampai dengan April 2020 dengan rincian kegiatan dan waktu pelaksanaan pada Tabel 3.9.

Tabel 3. 10 Jadwal Penelitian.

No.	Kegiatan	Waktu (Bulan)															
		Oktober			November			Desember		Januari		Februari		Maret		April	
1	Analisis Masalah	■	■	■													
2	Studi Literatur			■	■	■	■	■									
3	Analisis Arsitektur				■	■	■	■	■								
4	Pengumpulan Data						■	■	■								
5	Penelusuran Proposal				■	■	■	■	■								
6	Pelaksanaan Penelitian								■	■	■	■	■	■			
7	Penulisan Laporan Hasil Penelitian										■	■	■	■	■		
8	Seminar Hasil													■	■	■	