

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Objek Penelitian

Objek yang menjadi sasaran pada penelitian kali ini yaitu *memes* yang berpotensi untuk menjadi *hate speech* ataupun tidak, yang didapat dari *facebook competition* sebanyak 9.540 data dengan nama “*hateful memes dataset*” dan sudah dilabeli dengan label angka 1 atau berarti “*hateful*” dan angka 0 atau berarti “*non-hateful*”. Definisi dari *hateful* yang terdapat pada label data adalah adanya penyerangan secara langsung maupun tidak langsung yang membahas tentang ras, kebangsaan, agama, kasta sosial, *gender*, kecacatan atau penyakit, dan juga membandingkan manusia dengan hal-hal yang bukan manusia seperti contohnya binatang, sedangkan definisi dari *non-hateful* yang terdapat pada label data adalah sebaliknya. Data diambil dengan cara langsung melakukan unduh dari situs penyimpanan data yang diberikan dari *facebook competition*.

Setelah data berhasil diunduh, label dari data akan dimasukkan kedalam *model* yang dibuat dengan Bahasa pemrograman *python* dan akan dilakukan *testing* menggunakan data *testing* yang juga berasal dari *hateful memes dataset* untuk melakukan evaluasi *model* yang dibuat dengan metrik *accuracy*.

### 3.2 Metode Penelitian

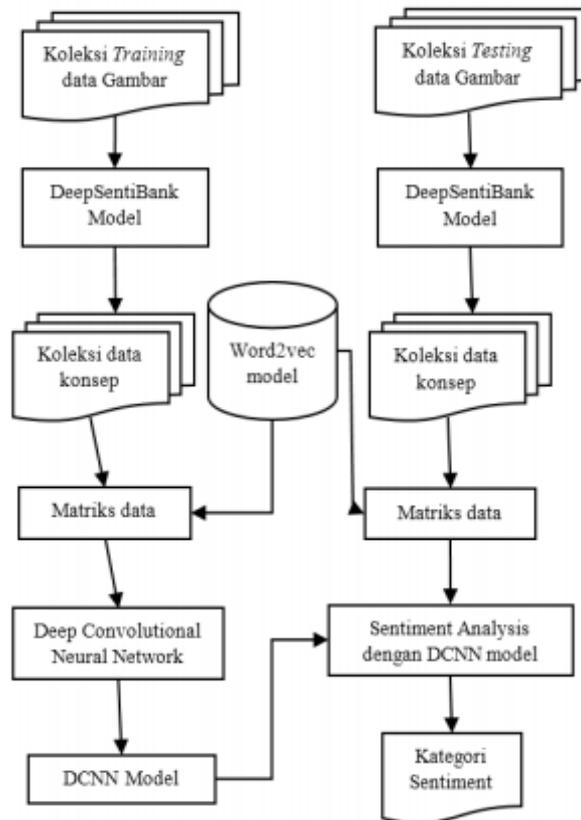
**Tabel 3. 1 Tabel perbandingan metode**

Metode yang digunakan	Keuntungan	Kerugian
<i>Support Vector Machine</i>	<ul style="list-style-type: none"> <li>- Memberikan solusi yang unik.</li> <li>- Efisien.</li> <li>- Menghindari overfitting.</li> </ul>	<ul style="list-style-type: none"> <li>- Algoritma dengan tingkat kompleksitas yang tinggi.</li> <li>- Berjalan dengan lambat</li> </ul>
<i>Artificial Neural Network</i>	<ul style="list-style-type: none"> <li>- Dapat melakukan <i>training</i> dengan <i>dataset</i> yang memiliki <i>noise</i>.</li> <li>- Efisien untuk ukuran data yang besar.</li> </ul>	<ul style="list-style-type: none"> <li>- Tingkat komputasi yang tinggi.</li> <li>- <i>Lazy learner</i>.</li> </ul>
<i>Decision Tree</i>	<ul style="list-style-type: none"> <li>- Tidak perlu mengeluarkan effort yang besar.</li> <li>- Mudah untuk menafsirkan hasilnya.</li> </ul>	<ul style="list-style-type: none"> <li>- Tingkat <i>Classification error</i> yang tinggi.</li> <li>- Sangat sensitif mengenai <i>splitting data</i>.</li> </ul>
<i>Convolutional Neural Network.</i>	<ul style="list-style-type: none"> <li>- <i>Model</i> yang sangat populer untuk <i>image classification</i>.</li> <li>- Komputasi yang efisien.</li> <li>- Dapat mendeteksi fitur yang penting dengan sendirinya.</li> <li>- <i>State-of-the-art model</i> dalam <i>ImageNet challenge</i>.</li> </ul>	<ul style="list-style-type: none"> <li>- Operasi yang memakan waktu dikarenakan terbagi menjadi beberapa proses.</li> <li>- Tidak cocok untuk perangkat yang kurang memadai</li> <li>- Membutuhkan dataset yang besar untuk melatih <i>model</i>.</li> </ul>

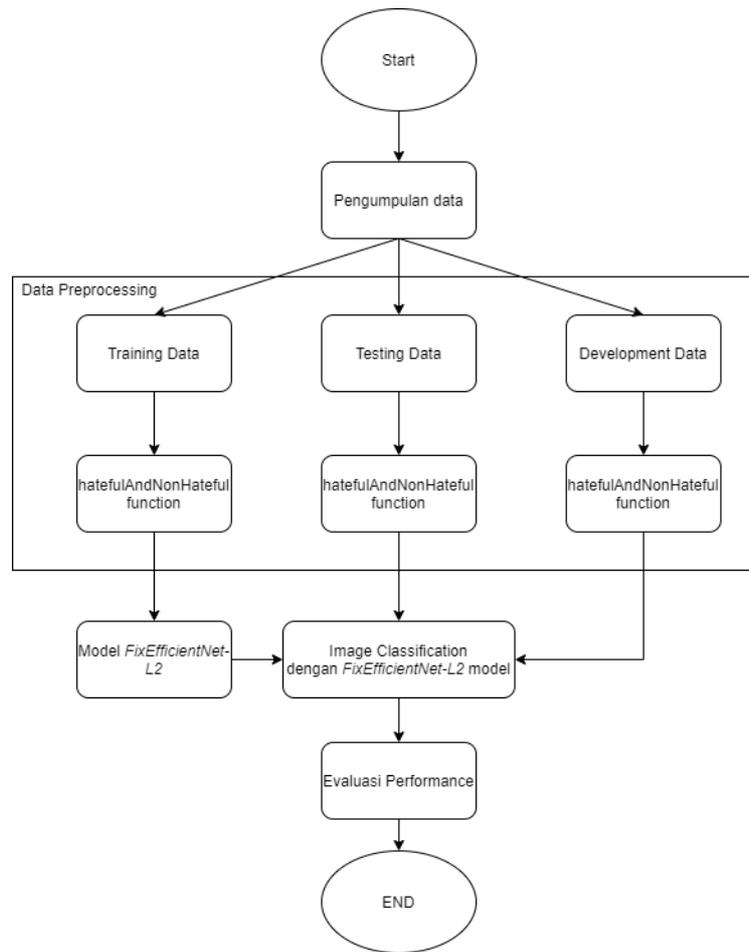
Berdasarkan perbandingan dari tabel diatas penelitian ini akan menggunakan algoritma *Convolutional neural network* dengan arsitektur *FixEfficientNet-L2*. Pemilihan arsitektur *FixEfficientNet* didasarkan pada beberapa hal:

1. Merupakan salah satu arsitektur terbaru dalam *Convolutional Neural Network* (2020).
2. Mendapatkan akurasi *state-of-the-art* pada *ImageNet Challenge* pada awal tahun 2020 dengan akurasi 88.5%
3. Merupakan gabungan metode dari *FixRes* dan *EfficientNet*. Metode *FixRes* terbukti dapat meningkatkan performa dari arsitektur CNN, dapat dilihat pada Gambar 2. 5 Akurasi dari *FixResNeXt* pada *ImageNet Challenge*, dan pada Gambar 2. 6 Hasil peningkatan *EfficientNet* dibanding dengan *FixEfficientNet*

Proses yang dilakukan pada penelitian ini merupakan modifikasi dari penelitian Hans Juwiatho, Esther Irawati Setiawan, dan Joan Santoso dapat dilihat pada Gambar 3. 1 Kerangka pikir menggunakan DCNN [28], dimana mereka juga membuat *model image classification* menggunakan *deep convolutional neural network*. Proses penelitian ini dibagi menjadi empat bagian yaitu pengumpulan *dataset*, *data preprocessing*, *deep convolutional network model*, *evaluasi performance*. Untuk melihat detail dari urutan proses yang akan dilakukan dapat dilihat pada Gambar 3. 2 Modifikasi kerangka pikir menggunakan *FixEfficientNet-L2*.



**Gambar 3. 1 Kerangka pikir menggunakan *DCNN* [28]**



**Gambar 3. 2** Modifikasi kerangka pikir menggunakan *FixEfficientNet-L2*

### 3.2.1 Pengumpulan data

Pengumpulan data pada Gambar 3. 2 Modifikasi kerangka pikir menggunakan *FixEfficientNet-L2* dilakukan dengan cara langsung melakukan unduh pada *page* yang disediakan oleh *facebook competition* untuk mengunduh *hateful memes dataset*. *Hateful memes dataset* pada awalnya berjumlah 162.000 data, setelah melalui beberapa tahapan *filtering* yang dilakukan dari pihak *facebook competition* agar data yang dipakai sesuai dengan kebutuhan dan pada akhirnya jumlah dari data menjadi 9.540 data dengan dua kelas yaitu “*hateful*” dan “*non-hateful*”.

Data dibagi menjadi 3 bagian yaitu sebagai berikut:

1. *Training Data* (8500 data).
2. *Validation Data* (500 data).
3. *Testing Data* (540 data).

### 3.2.2 *Data preprocessing*

Pada sub-proses *data preprocessing* terdapat dua tugas yang dilakukan yaitu sebagai berikut:

1. *Splitting data*

Data yang telah diunduh, akan dibagi menjadi tiga set data yaitu pertama adalah *training data* (8500 gambar) yang akan digunakan untuk melatih *model* yang akan dibuat, kedua adalah *testing data* (540 gambar) yang akan digunakan untuk melakukan evaluasi terhadap *model* yang telah dibuat, dan yang terakhir adalah *development data* (500 gambar) yang akan digunakan untuk melakukan evaluasi terhadap *model* dengan *new data*. Tahap *splitting data* telah dilakukan oleh tim *Facebook AI*.

2. *hatefulAndNonHateful function*

Setelah dilakukan *splitting* data dalam program data akan dimasukkan kedalam *function* yang akan melakukan perubahan pada data gambar, dan juga menambahkan label yang sesuai pada masing-masing gambar tersebut sehingga *model* yang dibuat dapat mempelajari data tersebut.

### 3.2.3 Model *FixEfficientNet-L2*

Algoritma yang digunakan untuk membuat *model* pada penelitian kali ini adalah *deep convolutional neural network*. Model DCNN yang akan dibuat bernama “*FixEfficientNet-L2*”. *FixEfficientNet-L2* adalah salah satu dari sekian banyak *model* dari *deep convolutional neural network* yang mendapatkan akurasi *state-of-the-art* pada *ImageNet Challenge*. *FixEfficientNet-L2* dibuat berdasarkan *architecture EfficientNet*, dan menambahkan sebuah metode bernama *FixRes*. *FixRes* adalah sebuah metode untuk mengoptimisasi resolusi pada saat *training* dan *testing* [19]. Model dari *FixEfficientNet-L2* akan dilatih menggunakan *training* data yang telah dibagi pada tugas sebelumnya. Model *FixEfficientNet* yang dilatih, akan dilatih dengan beberapa konfigurasi yaitu sebagai berikut:

**Tabel 3. 1 Konfigurasi Model *FixEfficientNet-L2***

<i>No</i>	<i>Epoch</i>	<i>Train size</i>	<i>Test Size</i>	<i>Batch Size</i>
1	11	16	32	16
2	11	16	64	16
3	11	16	128	16
4	11	16	256	16
5	11	16	512	16
6	11	32	64	16
7	11	32	128	16

<i>No</i>	<i>Epoch</i>	<i>Train Size</i>	<i>Test Size</i>	<i>Batch Size</i>
8	11	32	256	16
9	11	32	512	16
10	11	64	128	16
11	11	64	256	16
12	11	64	512	16
13	11	128	256	16
14	11	128	512	16
15	11	256	512	16

### 3.2.4 Evaluasi *performance*

Evaluasi *performance* adalah proses pengukuran seberapa baik atau buruknya sebuah *model* algoritma berjalan. Tolak ukur yang akan digunakan untuk menjadi evaluasi *performance* pada penelitian kali ini adalah nilai *accuracy* dalam Bahasa pemrograman *python*.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

**Rumus 3. 1 Accuracy [29]**

Catatan:

TP: True Positive

TN: True Negative

FP: False Positive

FN: False Negative

### **3.3 Variabel Penelitian**

Terdapat dua jenis variable yang terdapat pada penelitian kali ini yaitu variabel independen dan variabel dependen. Pertama, terdapat variabel independen yang dapat diartikan sebagai suatu hal yang mempengaruhi atau menyebabkan terjadinya variabel dependen [30], dalam penelitian kali ini variabel independen tersebut adalah gambar *memes*. Kedua, terdapat variabel dependen yang dapat diartikan sebagai variabel yang nilainya dapat dipengaruhi atau disebabkan oleh variabel independen [30], dalam penelitian ini variabel dependen tersebut adalah target kelas dari klasifikasi *memes*.

### **3.4 Teknik Pengumpulan Data**

Teknik pengumpulan data adalah cara yang digunakan untuk mendapatkan data yang dibutuhkan untuk melakukan penelitian. Pengumpulan data merupakan salah satu tahapan yang paling penting dalam penelitian. Data dikumpulkan dengan cara langsung mengunduh dari *page* yang disediakan oleh *facebook competition* ([31], harus mempunyai akun).

### **3.5 Teknik Pengambilan Sampel**

Teknik pengambilan sampel yang digunakan untuk mengambil sampel pada penelitian ini adalah *non-probability sampling* dengan Teknik *purposive sampling*. *Non-probability sampling* adalah teknik pengambilan sampel yang tidak memberikan kemungkinan yang sama rata untuk setiap sampel yang ada

dalam populasi [30]. *Purposive sampling* adalah salah satu teknik pengambilan sampel *non-probability sampling* yang mempunyai berbagai kriteria tertentu untuk dipenuhi yang sejalan dengan tujuan penelitian [32]. Kriteria tertentu yang harus dipenuhi sampel yang akan digunakan dalam penelitian ini adalah [33]:

1. Gambar merupakan *memes*.
2. Gambar tidak duplikasi.
3. *Memes* menggunakan Bahasa Inggris.
4. *Memes* yang asli dapat dikonstruksi ulang dengan membuat *memes* yang baru tanpa harus mengganti arti dari *memes* yang asli.

### **3.6 Teknik Analisis Data**

Teknik analisis data yang akan digunakan dalam penelitian kali ini yaitu analisis data kualitatif karena akan menentukan bagus atau tidaknya *model* yang dibuat berdasarkan akurasi dari *model* tersebut. Analisis data juga akan menggunakan *programming language* yang sudah sangat *populer* untuk kasus *deep learning* yaitu *python*.