BAB III
METODOLOGI PENELITIAN

3.1. Subyek dan Obyek Penelitian

Subjek penelitian merupakan data yang diamati. Subjek penelitian ini merupakan dataset dari Tugas Akhir bab I sampai II seluruh prodi pada mahasiswa angkatan tahun 2014 – 2018 bersumber dari *https://repository.ittelkom-pwt.ac.id/*. Objek penelitian adalah permasalahan yang akan diteliti. Objek penelitian ini adalah **Kombinasi Algoritme LSTM dan GRU dalam Pembuatan Teks Generator Tugas Akhir 1.**

3.2. Alat Dan Bahan Penelitian

3.2.1. Alat

A. Perangkat keras

Perangkat keras yang digunakan untuk membuat penelitian ini :

1. *Processor* : *AMD Ryzen* 3 3200U
2. *Memory* : 8 *GB RAM DDR 4*
3. *Graphic Card* : *AMD Radeon Vega 3 Graphics*
4. *Operating System* : *MX Linux 21.1 64 bit*
5. *Harddisk* : 1000 *GB*
6. *SSD* : 128 *GB*

B. Perangkat Lunak

Perangkat lunak yang digunakan untuk membuat penelitian ini :

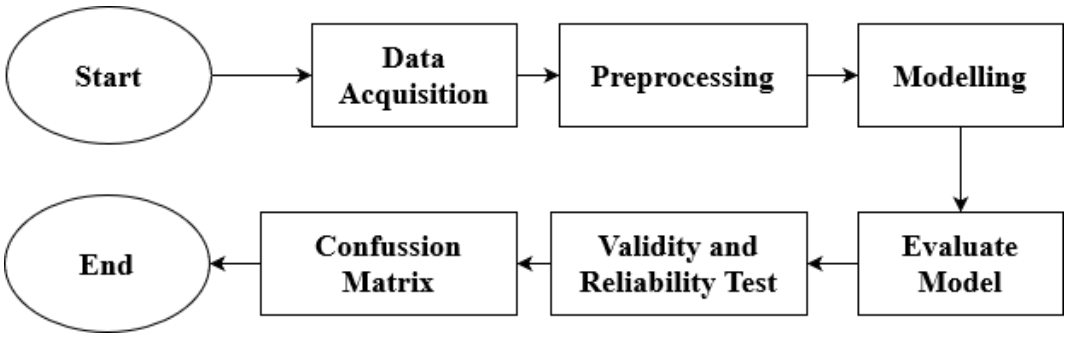
1. *Colab* digunakan untuk melakukan *developing* model.
2. *Visual Studio Code* digunakan untuk *developing* website.
3. *Git* digunakan untuk *push* file dalam server *github*.

3.2.2. Bahan Penelitian

1. *Dataset* TA 1 bab I sampai II mahasiswa angkatan 2014 – 2018 bersumber dari *https://repository.ittelkom-pwt.ac.id/*.
2. *Dataset* yang dikategorikan terhadap kelompok keahlian yang terdapat pada prodi S1 Teknik Informatika, yaitu SC, TI dan RPLM. Peneliti menggunakan seluruh data yang ada dan memasukan ke dalam kelas yang berhubungan atau mendekati.

3.3. Diagram Alir Penelitian

Bagian ini menjelaskan mengenai tahapan yang dilakukan pada penelitian Kombinasi Algoritme *LSTM* dan *GRU* dalam Pembuatan Teks Generator Tugas Akhir 1. Tahap ini disusun secara sistematis agar memudahkan peneliti untuk mencapai tujuan. Penelitian dimulai dari perumusan masalah, menentukan tujuan penelitian, melakukan kajian pustaka, melakukan *data acquisition*, *preprocessing*, *modelling*, dan *evaluate model*. Berikut adalah diagram alur pada Gambar 3.1.



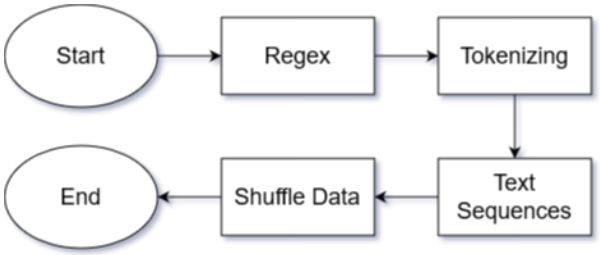
Gambar 3. 1. Diagram alir penelitian

3.3.1. *Data Acquisition*

Data acquisition dilakukan dengan download pada *https://repository.ittelkom-pwt.ac.id/*. *Dataset* yang diambil adalah data skripsi bab 1 – 2 mahasiswa angkatan 2014 - 2018 dari seluruh prodi yang di ITTP. Berdasarkan awal berdirinya ITTP, Program Studi S1 Teknik Telekomunikasi adalah program studi pertama yang ada di ITTP. Sehingga, hal ini akan selaras dengan semakin banyak jumlah data dari program studi tersebut.

3.3.2. *Preprocessing*

Dalam *preprocessing* melalui beberapa proses. Dimulai dengan menentukan pola teks *regex* sehingga mendapatkan *dataset* yang bersih. *Tokenizing* untuk merubah data menjadi representatif angka. *Text sequences* untuk membagi jumlah kata menjadi *feature* dan *target*. Diakhiri dengan *shuffle data* untuk melakukan *random* pada urutan data. Proses tersebut digambarkan pada Gambar 3.2.



Gambar 3. 2. Diagram *preprocessing dataset*

3.3.2.1. *Regex*

Regex merupakan *library* yang digunakan untuk melakukan mengambil data dengan pola yang cocok *(pattern matching)* terhadap data teks. Daripada memilah data secara manual, implementasi regex dapat mempercepat pemilahan data. Dengan menggunakan pola tertentu, regex dapat menghasilkan teks yang sesuai dengan kebutuhan peneliti. Sehingga, penggunaan regex akan mengoptimalkan proses *preprocessing*.

3.3.2.2. *Tokenizing*

Model tidak dapat melakukan proses *training* terhadap data teks secara langsung, sehingga memerlukan proses *tokenizing*. *Tokenizing* melakukan konversi kata menjadi token – token. Token akan merepresentasikan setiap kata yang ada pada dataset. Setelah tahapan tokenizing, data sudah bisa diolah dalam tahap *modelling*.

3.3.2.3. *Text Sequences*

Text sequences adalah teknik untuk menentukan panjang kata. Teknik ini berfungsi untuk membagi kata menjadi fitur dan target sesuai dengan panjang

seq_length. Teknik ini berguna untuk menghasilkan teks baru berdasarkan teks yang ada, tetapi panjang teks keseluruhannya terlalu besar untuk diproses sekaligus oleh model. Alih-alih melakukan proses training data teks sebagai satu blok input, kita dapat membaginya menjadi sejumlah *seq_length* yang lebih kecil.

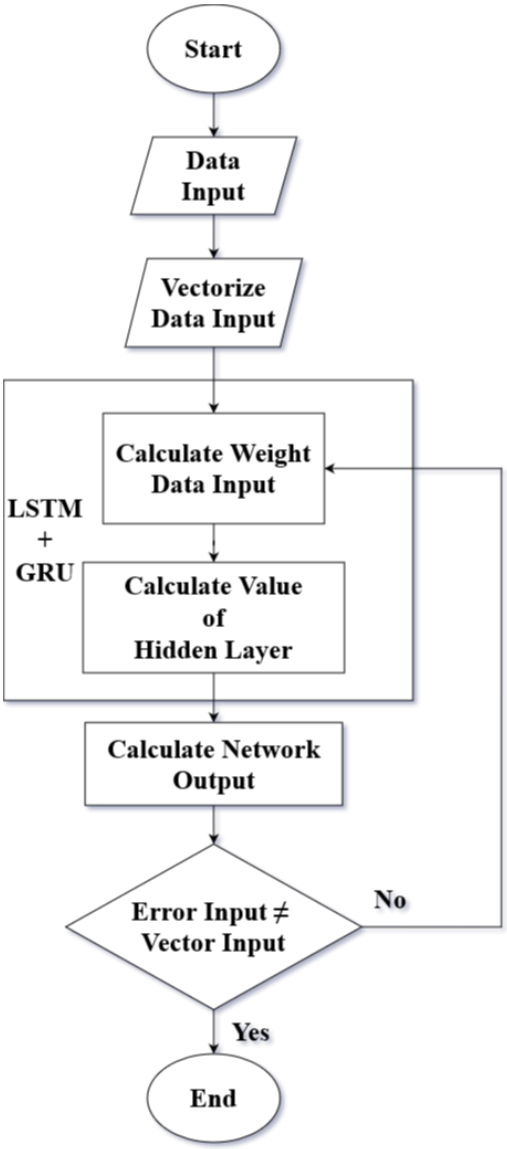
3.3.2.4. *Shuffle Data*

Shuffle data adalah melakukan pengacakan dalam dataset yang sudah siap sebelum proses modelling. Tujuan dari melakukan *shuffle* adalah untuk memastikan bahwa dataset tidak memiliki pola tertentu atau urutan yang terstruktur, yang dapat mempengaruhi kinerja dan generalisasi model. Ketika data diurutkan atau terstruktur dengan baik, model dapat "menghafal" urutan tersebut dan mengandalkan pola urutan tersebut saat memproses data. Akibatnya, model tidak dapat mempelajari keterkaitan antara fitur - fitur yang sesungguhnya dan target yang ingin diprediksi, melainkan hanya mengandalkan pola urutan itu sendiri.

Dengan mengacak data, kita memastikan bahwa setiap sampel data muncul dalam urutan acak, dan model tidak bisa lagi mengandalkan pola urutan untuk memahami hubungan di dalam data. Ini membantu model untuk lebih fokus pada fitur-fitur aktual yang berhubungan dengan tugas yang ingin diprediksi dan memperbaiki kemampuan generalisasinya pada data yang belum pernah dilihat sebelumnya.

3.3.3. Implementasi Kombinasi LSTM dan GRU pada RNN.

Proses kerja RNN dengan melakukan implementasi pada data input menjadi sebuah vektor yang bertujuan untuk merepresentasikan data yang akan diproses menggunakan RNN. Data input tersebut, selanjutnya dilakukan proses pembobotan dan penghitungan berdasarkan *hidden layer* sesuai dengan arsitektur yang dibuat. Terakhir, model akan menghasilkan sebuah nilai, jika nilai tersebut masih tidak sesuai, dilakukan proses *backpropagation* untuk mengevaluasi bobot sehingga menghasilkan nilai yang sesuai. Proses *backpropagation* yang panjang menghasilkan efek *vanishing gradient* pada proses inilah kombinasi *LSTM* dan *GRU* digunakan untuk mengatasi permasalahan tersebut.



Gambar 3. 3 Implementasi LSTM dan GRU pada RNN

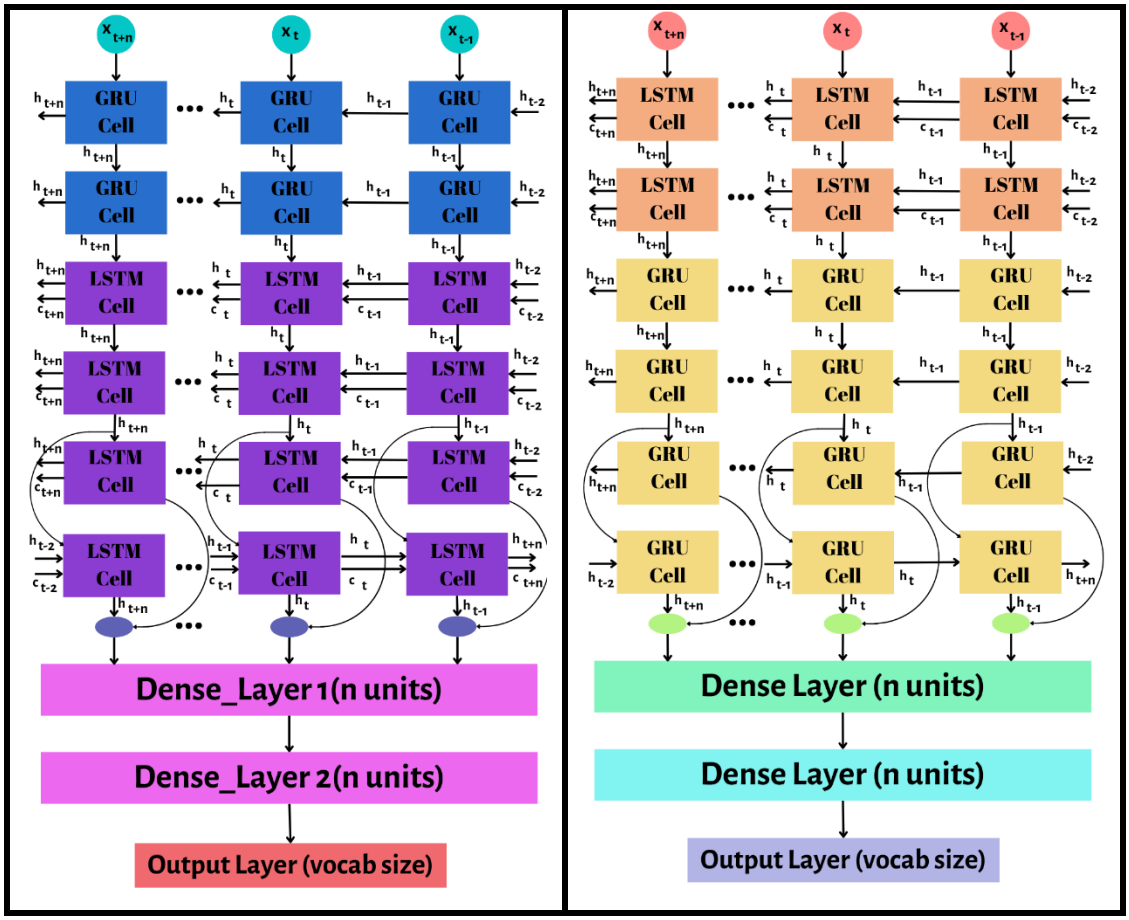
3.3.4. *Modelling*

Tahap pembuatan model menggunakan algoritma *LSTM* dan *GRU*, serta arsitektur *bidirectional*, dalam penelitian ini terdapat 3 skema model. Masing - masing skema memiliki 2 model, yaitu LSTM-biGRU dan GRU-biLSTM. skema 1 dan 2 dapat dilihat pada Tabel 3.1, sedangkan skema 3 terdapat pada Tabel 3.2. Tabel tersebut akan dijelaskan secara spesifik penggunaan layer dari masing masing model pada masing masing skema.

Tabel 3. 1. Struktur skema 1 dan 2

Model 1	Model 2
Embedding (n dims)	*Embedding (n dims)*
LSTM_1 (n units)	*GRU_1 (n units)*
LSTM_2 (n units)	*GRU_2 (n units)*
GRU_1 (n units)	*LSTM_1 (n units)*
GRU_2 (n units)	*LSTM_2 (n units)*
Bidirectional(GRU_2)	*Bidirectional(LSTM_2)*
Dense_1(n units)	*Dense_1 (n units)*
Dense_2 (n units)	*Dense_2 (n units)*
Dense_output (vocab_size)	*Dense_output (vocab_size)*

Berdasarkan Tabel 3.1 merupakan bentuk visual dari model *LSTM-biGRU* dan *GRU-biLSTM*, perbedaan utama pada skema 1 dan 2 adalah banyaknya jumlah units pada *LSTM*, *GRU*, *Dense layer* dan jumlah dimensi pada *embedding layer*.



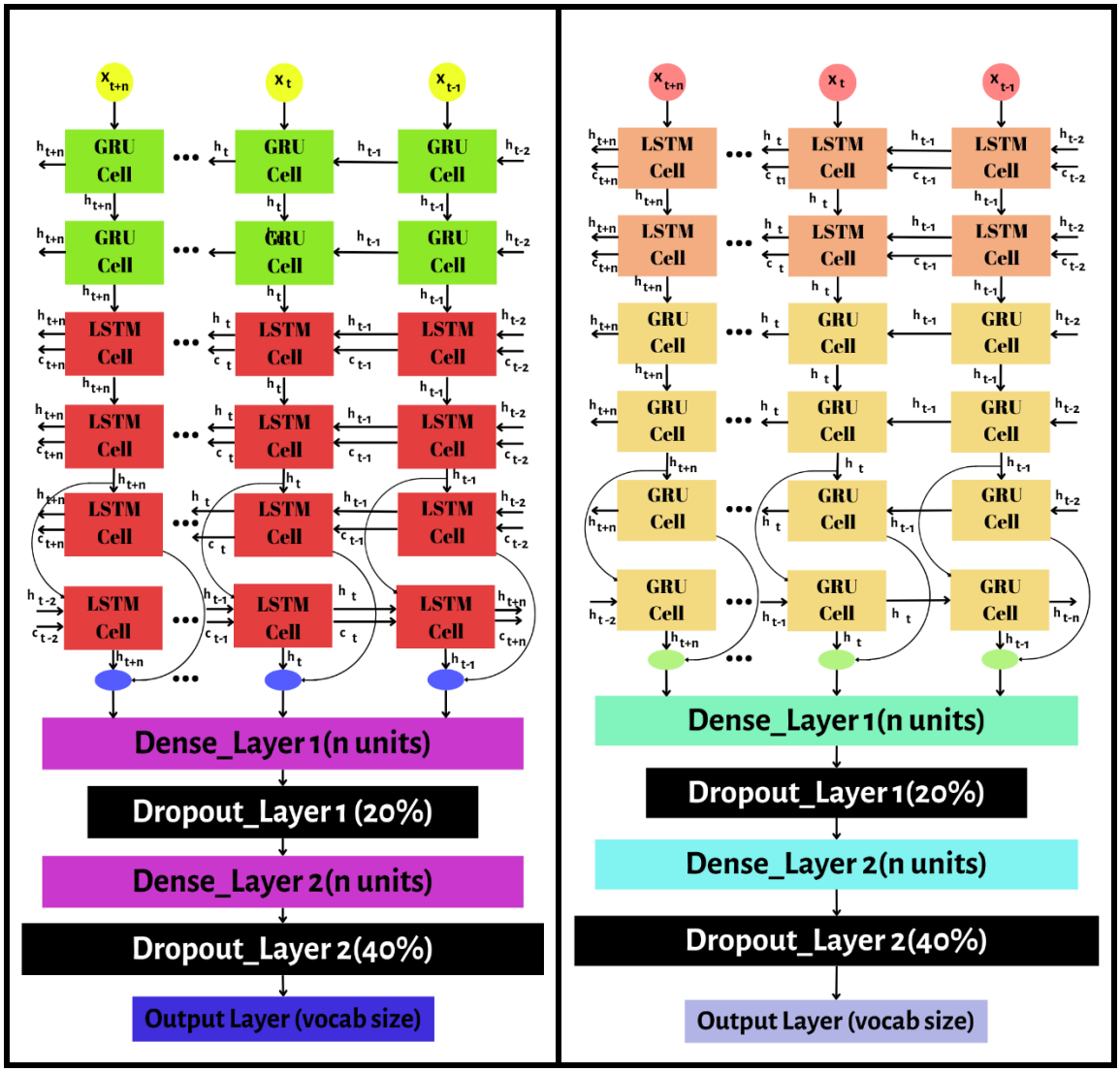
Gambar 3.4. Arsitektur model skema 1 dan 2

Skema 3 memiliki perbedaan pada jumlah *layer*, skema ini melakukan penambahan 2 *dropout layer* setelah *dense layer* pertama dan kedua. *Dropout layer* pertama sebesar 20% dan *dropout layer* kedua sebesar 40%.

Tabel 3. 2. Struktur skema 3

Model 1	Model 2
Embedding (n dims)	*Embedding (n dims)*
LSTM_1 (n units)	*GRU_1 (n units)*
LSTM_2 (n units)	*GRU_2 (n units)*
GRU_1 (n units)	*LSTM_1 (n units)*
GRU_2 (n units)	*LSTM_2 (n units)*
Bidirectional(GRU_2)	*Bidirectional(LSTM_2)*
Dense_1(n units)	*Dense_1 (n units)*
Dropout_1(0.2)	*Dropout_1(0.2)*
Dense_2 (n units)	*Dense_2 (n units)*
Dropout_2(0.4)	*Dropout_2(0.4)*
Dense_output (vocab_size)	*Dense_output (vocab_size)*

Berdasarkan Tabel 3.2 dibandingkan dengan skema 2 memiliki perbedaan yang jelas terutama pada tambahan *dropout layer*, selain itu model yang ada dalam skema ini memiliki perbedaan pada jumlah dimensi pada *embedding layer*, serja jumlah *units* pada *dense*, *lstm*, dan *gru layer*.



Gambar 3. 4. Arsitektur model skema 3

3.3.5. Model *Evaluation*

Penelitian ini menggunakan evaluasi model terhadap besaran bias variabel *loss* dan *val_loss*, semakin tinggi bias yang dihasilkan model maka performa model dinyatakan buruk, sebaliknya semakin konvergen model maka model dinyatakan memiliki performa yang baik. Model yang memiliki performa *loss* dan *val_loss* konvergen selanjutnya akan diuji melalui uji kuesioner dengan membandingkan hasil tulisan model dan tulisan penulis. Kuesioner tersebut akan menghasilkan akurasi model melalui perbandingan data teks sampel buatan model dan buatan penulis.

3.3.5.1. Total kata masuk akal

Seluruh model pada masing – masing skema akan diuji lebih mendalam yang dapat diakses pada tautan https://s.id/dokumentasi-model. *Spreadsheet* tersebut terdapat kolom "total kata" yang diambil dari kolom "kata masuk akal" dengan menggunakan rumus (3.1) sebagai dibawah ini. Total kata masuk akal diambil dari seberapa banyak hasil sampel generated text yang masih memiliki korelasi terhadap inputan.

$$\text{total kata} = \text{X1} + \text{X2} + \text{X3} \ldots \text{Xn} \quad (3.1)$$

X = Kata yang masuk akal

n = Jumlah kalimat

3.3.5.2. Rata - rata kata masuk akal

Kolom rata – rata diambil dari jumlah total kata masuk akal dibagi dengan total data dengan rumus (3.2) di bawah ini,

$$rata - rata = \frac{X1 + X2 + X3 \ldots Xn}{n} \quad (3.2)$$

$\text{X} = \text{Kata yang masuk akal}$

$\text{n} = \text{Jumlah kalimat}$

Sehingga semakin banyak jumlah kata masuk akal maka berbanding lurus dengan tingginya jumlah rata rata, jika nilai rata – rata tinggi maka model memiliki rata rata yang bagus dari seluruh sampel.

3.3.5.3. Standar Deviasi Sampel

Kolom standar deviasi sampel merupakan kolom penghitungan menggunakan rumus (3.3) di bawah ini,

$$s = \sqrt{\frac{\sum(X - \bar{X})^2}{n - 1}}$$
$$\text{s} = \text{Standar deviasi}$$
$$\text{X} = \text{Kata masuk akal} \qquad (3.3)$$
$$\bar{X} = \text{Rata rata kata masuk akal}$$
$$\text{n} = \text{Jumlah kalimat}$$

Rumus tersebut akan menghasilkan nilai yang digunakan untuk mencari sebaran data sampel dan seberapa titik data individu ke rata – rata nilai sampel, sehingga jika hasil mendekati 0 maka dapat disimpulkan bahwa model tersebut memiliki stabilitas yang baik.

3.3.6. Uji Validitas dan Reliabitas

Uji validitas dan reliabilitas bertujuan untuk menguji kuesioner dapat dinyatakan valid dan reliabel, dengan menggunakan rumus r tabel (3.4) jika nilai signifikansi lebih tinggi daripada r tabel maka data dinyatakan valid.

$$\text{df} = \text{N} - 2$$
$$\text{df} = \text{degree of freedom} \qquad (3.4)$$
$$\text{N} = \text{Jumlah responden}$$

Reliability test bertujuan untuk menguji reliabilitas kuesioner, pengujian reliabilitas dapat menggunakan perbandingan dengan nilai Alpha Cronbach, jika nilai Alpha Cronbach > 0.6 maka data dapat dinyatakan reliabel.

3.3.7. *Confussion Matrix*

Setelah proses pembagian kuesioner, akan didapatkan suatu nilai, di mana dalam kuesioner tersebut hanya dapat memilih dengan 2 variabel, jika responden memilih **“ya”** maka teks tersebut merupakan representatif buatan **“manusia”** sebaliknya jika responden memilih **“tidak”** maka merepresetasikan buatan **“model”**. Dari hasil responden akan dihitung untuk dan menghasilkan confussion matrix seperti pada Tabel 3.3.

Tabel 3. 3. *Confussion Matrix*

Hasil Responden			
		Tidak (model)	Ya (Manusia)
True Label	Model	True Model (TM)	False Model (FM)
	Human	False Human (FH)	True Human (TH)
		Model	Human
	Predicted Label		

Berdasarkan tabel di atas akan menghasilkan rebuah akurasi mengenai manusia yang dapat membedakan manusia dan model serta sebaliknya, berdasarkan rumus (3.5) di bawah ini akan menghasilkan individu yang dapat membedakan antara hasil *generated text* buatan model dan teks buatan penulis.

$$X = \frac{TM + TH}{(TM + TH + FM + FH)} \quad (3.5)$$

X = Akurasi responden yang **dapat** membedakan tulisan model dan tulisan manusia

Selain menghasilkan akurasi responden yang dapat membedakan tulisan buatan model dan buatan penulis, berdasarkan Tabel 3.3 akan dihitung untuk membandingkan akurasi manusia yang tidak dapat membandingkan teks buatan model dan buatan penulis melalui rumus berikut (3.6).

$$Y = \frac{FM + FH}{(TM + TH + FM + FH)} \quad (3.6)$$

Y = Akurasi responden yang **tidak dapat** membedakan tulisan model dan tulisan manusia