

BAB 3

METODE PENELITIAN

Penelitian berawal dari latar belakang permasalahan yang ada, memetakan proses-proses, mencari sumber permasalahan, dan akhirnya merancang dan mengembangkan suatu sistem yang dapat digunakan untuk mereduksi atau mengeliminasi permasalahan yang ada.

3.1 BAHAN DAN ALAT PENELITIAN

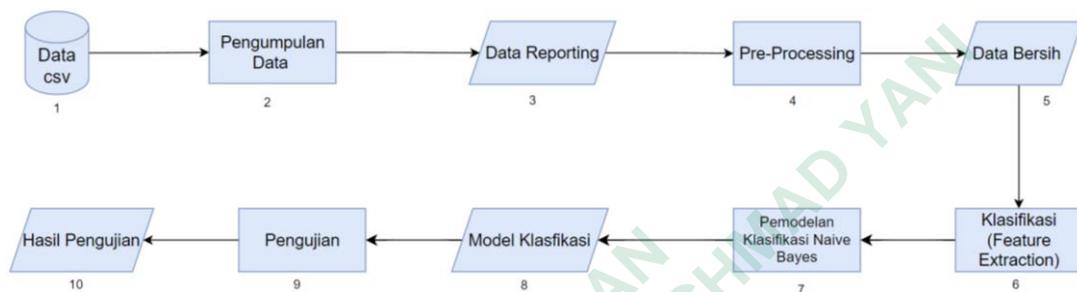
Bahan penelitian berisi bahan-bahan yang digunakan untuk melakukan penelitian pada sistem *bug reporting*. Bahan penelitian ini menggunakan data pelaporan masalah yang sudah pernah di report sebelumnya pada perusahaan dan menyimpan data dalam bentuk *csv* untuk diolah. Dari data yang sudah diolah akan mendapatkan sebuah analisa yang memudahkan untuk mengenali *dev type* dari *bug description*.

Alat penelitian yang digunakan berupa perangkat lunak untuk kebutuhan dalam mengerjakan sistem *bug reporting* seperti:

1. Sistem Operasi : Windows 11.
2. Prosesor : AMD Ryzen 5 5600H with Radeon Graphics
3.30GHz
3. RAM : 8 GB
4. IDE : Visual Studio Code versi 1.74.0
5. Pemrograman : Python v3.9.1 dan JavaScript v16.18.0
6. *Library* : Scikit-learn v1.2.2 dan nltk v3.8.1
7. *Framework* : Fastapi v0.96.0 dan Angular v16.0.4

3.2 JALAN PENELITIAN

Penelitian ini menggunakan pemrograman python dan metode klasifikasi *Naïve Bayes* untuk analisis sentimen pada data teks. Penelitian ini akan dimulai dari analisis kebutuhan, perancangan sistem, pengumpulan data, pra pemrosesan data, pengembangan sistem dan pengujian sistem yang ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Jalan Penelitian

3.3 PENGUMPULAN DATA

Pada tahap ini melakukan pengumpulan data *bug reporting* yang dimiliki oleh perusahaan. Data *bug reporting* yang dikumpulkan menggunakan bahasa Inggris dan telah memiliki tipe *developer* pada *bug reporting* sebelumnya. Dari data yang telah dikumpulkan mendapatkan 488 data *reporting* dengan tipe *developer front-end* dan 240 data *reporting* dengan tipe *developer back-end*. Maka total data yang didapatkan adalah 728 data. Data disimpan dalam format csv, yang merupakan format file untuk menyimpan data dalam bentuk tabel, yang didalam tabel memuat kolom untuk *dev type* dan *deskripsi bug*.

Data yang sudah dikumpulkan akan digunakan untuk melatih model klasifikasi *Naïve Bayes*, tetapi sebelum tahap melatih model perlu dilakukan tahap pra-pemrosesan atau *preprocessing* data. Pra-pemrosesan adalah langkah-langkah yang dilakukan untuk membersihkan dan mempersiapkan data sebelum dianalisis atau digunakan untuk melatih model.

Menampilkan contoh data yang telah dikumpulkan dari data *reporting* sebelumnya yang diambil dari *csv* dan sudah memiliki tipe *developer*. Contoh data ditunjukkan pada Tabel 3.1.

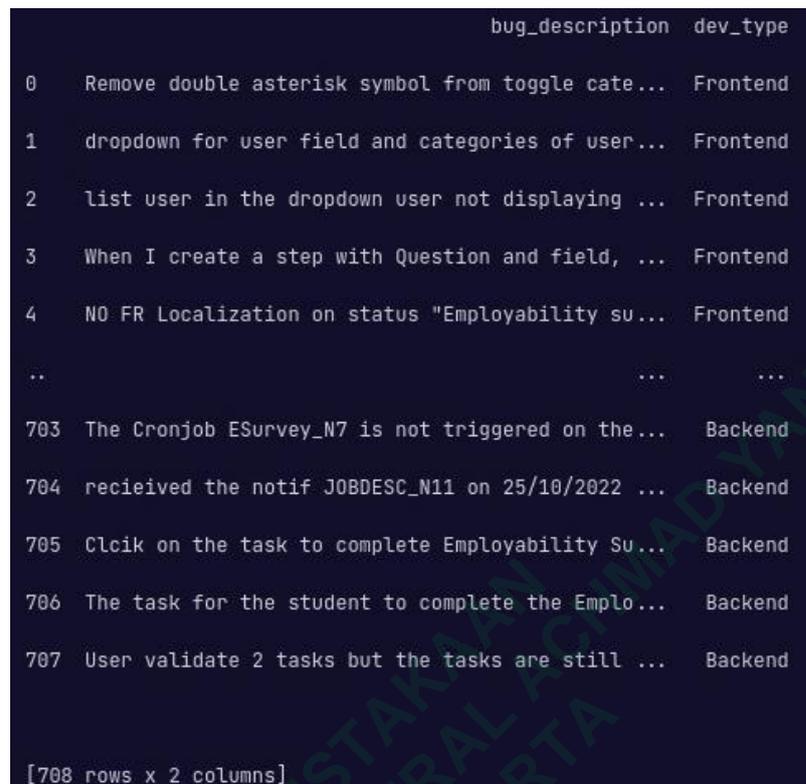
Tabel 3.1 Pengumpulan Data

No	Deskripsi	Tipe <i>Developer</i>
1	<i>Remove double asterisk symbol from toggle categories of users (check in academic admin and academic director).</i>	<i>Front-end</i>
2	<i>Dropdown for user field and categories of users no alphabetical order</i>	<i>Front-end</i>
3	<i>After sending one-time ES in Active Student Table, the selected template is not displayed in ES Employability Survey Parameters.</i>	<i>Back-end</i>
4	<i>There is a GraphQL error when trying to reject the ES of the student, which says "updatedFormProcess is not defined."</i>	<i>Back-end</i>
5	<i>ES Parameter for a sent ES must be non-editable for Date and Expiration date. Remove "ES" button for 2nd and 3rd ES when already sent to at least 1 student</i>	<i>Back-end</i>

Menampilkan potongan kode untuk memanggil semua data yang ada pada *csv* yang sudah memiliki tipe *developer* untuk semua *issue* yang pernah di-*report* sebelumnya pada perusahaan. Berikut potongan kode untuk pemanggilan data yang sudah dikumpulkan:

```
data =
pandas.read_csv('../classification/dat
```

Hasil dari pemanggilan data untuk semua data yang sudah dikumpulkan akan menampilkan hasil data yang ditampilkan pada Gambar 3.2 dengan kumpulan data yang belum di *preprocessing* dan data yang sudah memiliki tipe *developer* pada *reporting* sebelumnya.



bug_id	bug_description	dev_type
0	Remove double asterisk symbol from toggle cate...	Frontend
1	dropdown for user field and categories of user...	Frontend
2	list user in the dropdown user not displaying ...	Frontend
3	When I create a step with Question and field, ...	Frontend
4	N0 FR Localization on status "Employability su...	Frontend
..
703	The Cronjob ESurvey_N7 is not triggered on the...	Backend
704	recieived the notif JOBDESC_N11 on 25/10/2022 ...	Backend
705	Clcik on the task to complete Employability Su...	Backend
706	The task for the student to complete the Emplo...	Backend
707	User validate 2 tasks but the tasks are still ...	Backend

[708 rows x 2 columns]

Gambar 3.2 Hasil dari file csv

3.4 PRE-PROCESSING

Tahap *pre-processing* atau pra-pemrosesan data bertujuan untuk membersihkan, mengubah, dan menyiapkan data mentah untuk dapat diproses lebih lanjut untuk memastikan bahwa data siap digunakan dalam analisis atau pemodelan. Tahapan pra-pemrosesan data yang dilakukan meliputi *case folding*, *punctuation removal*, *tokenizing*, *stopword removal* dan *lemmatizing*:

3.4.1 Case Folding

Case folding merupakan salah satu tahap pra-pemrosesan yang umum dilakukan pada data teks dalam pemrosesan data. Tujuan dari *case folding* adalah untuk mengubah semua huruf dalam teks menjadi huruf kecil (*lowercase*) tanpa mengubah makna atau konteks dari teks. Hal ini dilakukan agar sistem atau model yang mengolah data teks dapat lebih mudah dalam mengenali setiap kata dan menghindari duplikasi informasi yang disebabkan oleh variasi huruf besar dan kecil.

Menampilkan contoh data *bug reporting* yang telah diubah menjadi huruf kecil pada kolom deskripsi. Langkah *case folding* telah diterapkan untuk mengurangi variasi dalam teks yang dianggap sama, sehingga data menjadi lebih terstruktur dan konsisten dalam representasi. Contoh data ditunjukkan pada Tabel 3.2. *Case folding* bertujuan untuk membersihkan data dan mengubah teks menjadi representasi yang lebih sederhana, sehingga dapat meningkatkan akurasi dan efisiensi analisis teks.

Tabel 3.2 *Case Folding*

No	Deskripsi	Tipe Developer
1	<i>remove double asterisk symbol from toggle categories of users (check in academic admin and academic director).</i>	<i>Front-end</i>
2	<i>dropdown for user field and categories of users no alphabetical order</i>	<i>Front-end</i>
3	<i>after sending one-time es in active student table, the selected template is not displayed in es employability survey parameters.</i>	<i>Back-end</i>
4	<i>there is a graphql error when trying to reject the es of the student, which says "updatedformprocess is not defined."</i>	<i>Back-end</i>
5	<i>es parameter for a sent es must be non-editable for date and expiration date. remove "es" button for 2nd and 3rd es when already sent to at least 1 student</i>	<i>Back-end</i>

Menampilkan potongan kode *case folding* yang bertujuan untuk mengubah teks pada data *bug_description* menjadi huruf kecil. Berikut potongan kode untuk *case folding*:

```
data['bug_description']=data['bug_description'].apply
(lambda text : text.lower())
```

Hasil dari *case folding* ditunjukkan pada Gambar 3.3 dengan menampilkan teks pada *bug_description* menjadi huruf kecil untuk menghilangkan antara huruf kapital dan huruf kecil.

	bug_description	dev_type
0	remove double asterisk symbol from toggle cate...	Frontend
1	dropdown for user field and categories of user...	Frontend
2	list user in the dropdown user not displaying ...	Frontend
3	when i create a step with question and field, ...	Frontend
4	no fr localization on status "employability su...	Frontend
..
703	the cronjob esurvey_n7 is not triggered on the...	Backend
704	recieived the notif jobdesc_n11 on 25/10/2022 ...	Backend
705	clcik on the task to complete employability su...	Backend
706	the task for the student to complete the emplo...	Backend
707	user validate 2 tasks but the tasks are still ...	Backend

[708 rows x 2 columns]

Gambar 3.3 Hasil dari *case folding*

3.4.2 Punctuation Removal

Punctuation removal merupakan tahapan penting dalam pra-pemrosesan data teks yang bertujuan untuk menghapus tanda baca pada deskripsi. Contoh dari *punctional* adalah titik (.), koma (,), semi-colon (;), colon(:), *question mark*(?), *exclamation mark*(!), *apostrophe*('), *underline*(-), dan masih banyak tanda baca yang lainnya seringkali tidak memberikan kontribusi signifikan terhadap analisis teks dan pemrosesan data, sehingga dianggap sebagai *noise* atau gangguan dalam data.

Menampilkan contoh data *bug reporting* yang telah menerapkan *punctuation removal* pada kolom deskripsi. Tanda baca seperti titik, koma, tanda seru, dan tanda tanya telah dihapuskan dari deskripsi *bug*. Dengan melakukan *punctuation removal*, deskripsi *bug* dalam tabel menjadi lebih terstruktur dan bebas dari gangguan tanda baca. Data pada tabel siap diproses lebih lanjut dalam tahap pra-pemrosesan lainnya yaitu *tokenizing* dan *stopword removal*. Contoh data ditunjukkan pada Tabel 3.3.

Tabel 3.3 *Punctuation Removal*

No	Deskripsi	Tipe Developer
1	<i>remove double asterisk symbol from toggle categories of users check in academic admin and academic director</i>	<i>Front-end</i>
2	<i>dropdown for user field and categories of users no alphabetical order</i>	<i>Front-end</i>
3	<i>after sending onetime es in active student table the selected template is not displayed in es employability survey parameters</i>	<i>Back-end</i>
4	<i>there is a graphql error when trying to reject the es of the student which says updatedformprocess is not defined</i>	<i>Back-end</i>
5	<i>es parameter for a sent es must be non editable for date and expiration date remove es button for 2nd and 3rd es when already sent to at least 1 student</i>	<i>Back-end</i>

Menampilkan potongan kode *punctuation removal* yang bertujuan untuk menghapus tanda baca yang ada pada data *bug_description*. Berikut potongan kode untuk *punctuation removal*:

```
data['bug_description']=data['bug_description'].apply
(lambda text : re.sub(r'^a-z|\s|', '', text))
print(data)
```

Hasil dari *punctuation removal* ditunjukkan pada Gambar 3.4 dengan menghapus tanda baca pada teks di *bug_description*.

	bug_description	dev_type
0	remove double asterisk symbol from toggle cate...	Frontend
1	dropdown for user field and categories of user...	Frontend
2	list user in the dropdown user not displaying ...	Frontend
3	when i create a step with question and field a...	Frontend
4	no fr localization on status employability sur...	Frontend
..
703	the cronjob esurveyn is not triggered on the prod	Backend
704	recieived the notif jobdescn on even the date...	Backend
705	clcik on the task to complete employability su...	Backend
706	the task for the student to complete the emplo...	Backend
707	user validate tasks but the tasks are still p...	Backend

[708 rows x 2 columns]

Gambar 3.4 Hasil dari *punctuation removal*

3.4.3 Tokenizing

Tokenizing merupakan salah satu langkah penting dalam pra-pemrosesan data teks yang dilakukan untuk memotong teks menjadi unit-unit yang lebih kecil, seperti kata atau token. Pada proses tokenizing, teks akan dipisahkan menjadi kata-kata dengan menghilangkan symbol, angka, dan spasi yang tidak diperlukan. Sehingga dapat menghasilkan daftar atau kumpulan kata yang dapat digunakan dalam analisis lebih lanjut.

Menampilkan contoh data deskripsi yang telah diubah menjadi daftar kata (token) setelah proses tokenizing. Setiap kata dalam deskripsi menjadi elemen dan memisahkan teks menjadi unit-unit kecil. Proses tokenizing merupakan langkah penting dalam pra-pemrosesan data teks karena mempersiapkan data untuk dianalisis lebih lanjut atau melatih model klasifikasi dengan cara membagi menjadi unit-unit yang lebih terstruktur dan dapat diolah. Data yang sudah dalam bentuk token akan lebih mudah dipahami oleh sistem atau model yang digunakan dalam pemrosesan sistem analisis. Contoh data ditunjukkan pada Tabel 3.4.

Tabel 3.4 *Tokenizing*

No	Deskripsi	Tipe Developer
1	<i>remove, double, asterisk, symbol, from, toggle, categories, of, users, check, in, academic, admin, and, academic, director</i>	<i>Front-end</i>
2	<i>dropdown, for, user, field, and, categories, of, users, no, alphabetical, order</i>	<i>Front-end</i>
3	<i>after, sending, onetime, es, in, active, student, table, the, selected, template, is, not, displayed, in, es, employability, survey, parameters</i>	<i>Back-end</i>
4	<i>there, is, a, graphql, error, when, trying, to, reject, the, es, of, the, student, which, says, updatedformprocess, is, not, defined</i>	<i>Back-end</i>
5	<i>es, parameter, for, a, sent, es, must, be, non, editable, for, date, and, expiration, date, remove, es, button, for, 2nd, and, 3rd, es, when, already, sent, to, at least, 1, student,</i>	<i>Back-end</i>

Menampilkan potongan kode *tokenizing* yang bertujuan untuk melakukan pemotongan kalimat menjadi perkata pada data *bug_description*. Berikut potongan kode untuk *tokenizing*:

```
data['bug_description']=data['bug_description'].apply
(lambda text : word_tokenize(text))
print(data)
```

Hasil dari *tokenizing* ditunjukkan pada Gambar 3.5 dengan membuat kalimat pada *bug_description* menjadi sebuah kata.

```

                                bug_description dev_type
0  [remove, double, asterisk, symbol, from, toggl... Frontend
1  [dropdown, for, user, field, and, categories, ... Frontend
2  [list, user, in, the, dropdown, user, not, dis... Frontend
3  [when, i, create, a, step, with, question, and... Frontend
4  [no, fr, localization, on, status, employabili... Frontend
..  .. ..
703 [the, cronjob, esurveyn, is, not, triggered, o... Backend
704 [recieived, the, notif, jobdescn, on, even, th... Backend
705 [clcik, on, the, task, to, complete, employabi... Backend
706 [the, task, for, the, student, to, complete, t... Backend
707 [user, validate, tasks, but, the, tasks, are, ... Backend

[708 rows x 2 columns]

```

Gambar 3.5 Hasil dari *tokenizing*

3.4.4 *Stopword Removal*

Membantu dalam memilih dokumen yang memenuhi kebutuhan, beberapa istilah yang umum digunakan tampaknya tidak banyak berguna. Dalam proses *stopword* menggunakan *library* NLTK. NLTK berguna untuk menghilangkan kata-kata umum dalam teks, seperti “*the*”, “*is*”, “*a*” dan sebagainya. Dengan menghilangkan, teks menjadi lebih sederhana dan fokus pada kata-kata yang lebih penting dalam analisis teks.

Menampilkan contoh data deskripsi yang sudah menerapkan *stopword removal* bertujuan untuk mengambil kata atau istilah terbanyak sering muncul yang akan dihapus karena tidak mempengaruhi pemrosesan data. Contoh data ditunjukkan pada Tabel 3.5.

Tabel 3.5 *Stopword Removal*

No	Deskripsi	Tipe Developer
1	<i>remove, double, asterisk, symbol, toggle, categories, user, academic, admin, academic, director</i>	<i>Front-end</i>
2	<i>dropdown, user, field, categories, user, alphabetical, order</i>	<i>Front-end</i>
3	<i>sending, onetime, es, active, student, table, selected, template, displayed, es, employability, survey, parameters</i>	<i>Back-end</i>
4	<i>graphql, error, trying, reject, es, student, says, updatedformprocess, defined</i>	<i>Back-end</i>
5	<i>es, parameter, sent, es, must, editable, date, and, expiration, date, remove, es, button, 2nd, 3rd, es, already, sent, least, 1, student</i>	<i>Back-end</i>

Menampilkan potongan kode *stopword removal* yang bertujuan untuk mengambil kata atau istilah terbanyak sering muncul. Pada data *bug_description* yang dikumpulkan akan dihapus karena tidak mempengaruhi pemrosesan data. Berikut potongan kode untuk *stopword removal*:

```
stop_words=set(stopwords.words('english'))
data['bug_description']=data['bug_description'].apply
(lambda tokens : [word for word in tokens if word not
in stop_words])
```

Hasil dari *stopword removal* ditunjukkan pada Gambar 3.6 dengan menghapus kata yang sering muncul atau kata yang tidak penting untuk dilakukan proses penghapusan kata.

	bug_description	dev_type
0	[remove, double, asterisk, symbol, toggle, cat...	Frontend
1	[dropdown, user, field, categories, users, alp...	Frontend
2	[list, user, dropdown, user, displaying, user, ...	Frontend
3	[create, step, question, field, change, toggle...	Frontend
4	[fr, localization, status, employability, surv...	Frontend
..
703	[cronjob, esurveyn, triggered, prod]	Backend
704	[recieived, notif, jobdescn, even, date, activ...	Backend
705	[clcik, task, complete, employability, survey, ...	Backend
706	[task, student, complete, employability, surve...	Backend
707	[user, validate, tasks, tasks, still, populate...	Backend

[708 rows x 2 columns]

Gambar 3.6 Hasil dari *stopwords removal*

3.5 FEATURE EXTRACTION

Dengan menggunakan metode *TF-IDF* untuk menghitung berapa kali sebuah kata muncul dalam sebuah dokumen, perhitungan dan perbandingan dilakukan pada tahap ini untuk mengklasifikasikan karakteristik suatu objek.

Tf- Idf merupakan gabungan dari *term frequency* dan *invers document frequency*, *term frequency* digunakan untuk menghitung banyaknya kata yang muncul di suatu topik sedangkan *invers document frequency* digunakan untuk menghitung kata yang jarang muncul untuk menilai pentingnya kata-kata dalam

analisis teks dalam pengambilan informasi pada deskripsi yang sudah di *preprocessing* yang ditunjukkan pada Tabel 3.6.

Tabel 3.6 Hasil *Pre-Processing*

No	Deskripsi	Jumlah Kata
1	<i>remove, double, asterisk, symbol, toggle, categories, user, academic, admin, academic, director</i>	11
2	<i>dropdown, user, field, categories, user, alphabetical, order</i>	7
3	<i>sending, onetime, es, active, student, table, selected, template, displayed, es, employability, survey, parameters</i>	13
4	<i>graphql, error, trying, reject, es, student, says, updatedformprocess, defined</i>	9
5	<i>es, parameter, sent, es, date, expiration, date, remove, es, button, 2nd, 3rd, es, sent, least, I, student</i>	17

Menghitung TF pada teks yang dicantumkan pada Tabel 3.7 dan akan dihitung berdasarkan setiap dokumen. Dari perhitungan tf dapat dilihat seberapa sering kata muncul untuk setiap dokumen. Untuk menghitung nilai TF dapat menggunakan formula (8).

Tabel 3.7 *Term Frequency (TF)*

No	Teks	D1	D2	D3	D4	D5
1	<i>user</i>	0,09	0,28	0	0	0
2	<i>es</i>	0	0	0,15	0,11	0,23
3	<i>student</i>	0	0	0,07	0,11	0,05
4	<i>remove</i>	0,09	0	0	0	0,05
5	<i>symbol</i>	0,09	0	0	0	0

Menghitung IDF pada teks yang dicantumkan pada Tabel 3.8 untuk menghitung seberapa jarang teks muncul pada setiap dokumen. Dengan melakukan perhitungan IDF yang ada pada formula (9).

Tabel 3.8 *Inverse Document Frequency (IDF)*

No	Teks	Perhitungan	IDF
1	<i>user</i>	$\log(5/2)$	0,91
2	<i>es</i>	$\log(5/3)$	0,47
3	<i>student</i>	$\log(5/3)$	0,47
4	<i>remove</i>	$\log(5/1)$	1,61
5	<i>symbol</i>	$\log(5/1)$	1,61

Menghitung TF-IDF pada teks yang dicantumkan pada Tabel 3.9 untuk menunjukkan kata yang dihitung memberikan informasi yang penting dari perhitungan TF dan IDF sebelumnya. Dengan melakukan perhitungan TF-IDF yang ada pada formula (10).

Tabel 3.9 Perhitungan TF-IDF

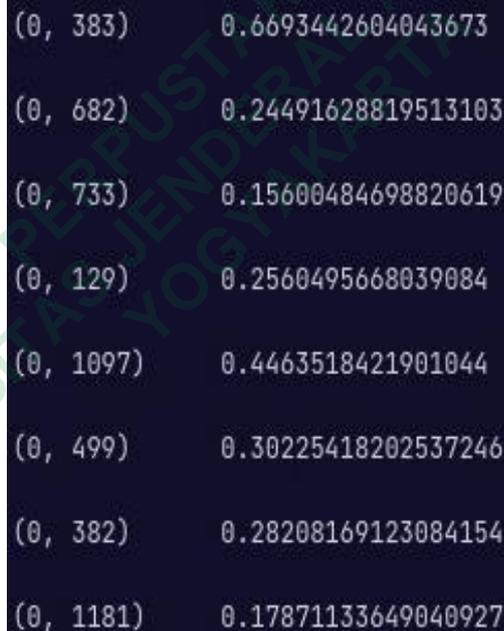
No	Teks	D1	D2	D3	D4	D5
1	<i>user</i>	0,0819	0,2548	0	0	0
2	<i>es</i>	0	0	0,0705	0,0517	0,1081
3	<i>student</i>	0	0	0,0329	0,0517	0,0235
4	<i>remove</i>	0,1449	0	0	0	0
5	<i>symbol</i>	0,1449	0	0	0	0

Dari kata “*user*” memberikan informasi bahwa kata termasuk kata yang unik karena hanya muncul di dua dokumen dari lima dokumen pada korpus. Kata “*es* dan *student*” yang muncul pada tiga dokumen dari lima dokumen dan kata “*student* dan *remove*” yang muncul pada satu dokumen dari lima dokumen. Maka, data “*user*, *remove* dan *symbol*” pada dokumen dianggap penting dan dapat memberikan informasi yang relevan dalam penentuan tipe *developer*. Sedangkan kata “*es* dan *student*” tidak dapat memberikan informasi yang jelas karena kata selalu berulang pada 3 dokumen yang dianggap kata tidak penting untuk digunakan dalam pemrosesan data.

Menampilkan potongan kode TF-IDF untuk menilai pentingnya kata-kata dalam analisis teks dalam pengambilan informasi deskripsi yang sudah di *preprocessing*. Berikut potongan kode untuk TF-IDF:

```
x_train_vector = vectorizer.fit_transform(x_train)
x_test_vector = vectorizer.transform(x_test)
print(x_train_vector, x_test_vector)
```

Menampilkan hasil dari TF-IDF untuk semua data csv yang telah di *preprocessing* dan sudah ditahap perhitungan dan perbandingan TF-IDF. Hasil dari TF-IDF di tunjukkan pada Gambar 3.7.



The image shows a screenshot of a terminal window displaying the output of a TF-IDF calculation. The output consists of eight lines, each representing a data point with an index and a corresponding TF-IDF value. The values are: 0.6693442604043673, 0.24491628819513103, 0.15600484698820619, 0.2560495668039084, 0.4463518421901044, 0.30225418202537246, 0.28208169123084154, and 0.17871133649040927. A large, semi-transparent watermark 'UNIVERSITAS YOGYAKARTA' is overlaid on the image.

(0, 383)	0.6693442604043673
(0, 682)	0.24491628819513103
(0, 733)	0.15600484698820619
(0, 129)	0.2560495668039084
(0, 1097)	0.4463518421901044
(0, 499)	0.30225418202537246
(0, 382)	0.28208169123084154
(0, 1181)	0.17871133649040927

Gambar 3.7 Hasil dari TF-IDF

3.6 PEMODELAN KLASIFIKASI NAÏVE BAYES

Pemodelan akan menggunakan metode klasifikasi *Naïve Bayes Multinomial* untuk mengimplmentasikan sistem *bug reporting* yang dapat mengklasifikasikan *bug* berdasarkan kata-kata dalam deskripsi *bug* yang telah di *preprocessing*. Sebelum melakukan pelatihan model, perlu menginisialisai dan mengkonfigurasi beberapa bagian terkait pemrosesan teks. Berikut adalah contoh kode-nya:

```

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
import joblib

# Inisialisasi model Naïve Bayes Multinomial
model = MultinomialNB()
# Inisialisasi TF-IDF Vectorizer
vectorizer = TfidfVectorizer()
# Fungsi untuk melatih model dengan data
def train_model(data):
    # Memisahkan bug_description (fitur) dan dev_type (label)
    dari data
    data_x = data['bug_description']
    data_y = data['dev_type']
    # Membagi data menjadi data pelatihan dan data pengujian
    x_train, x_test, y_train, y_test = train_test_split(data_x,
data_y, test_size=0.2, random_state=42)
    # Transformasi data TF-IDF
    x_train_vector = vectorizer.fit_transform(x_train)
    x_test_vector = vectorizer.transform(x_test)
    # Melatih model dengan data
    model.fit(x_train_vector, y_train)
    # Print akurasi model
    print('accuracy', model.score(x_test_vector, y_test))
    # Simpan model dan vectorizer ke file
    joblib.dump(model, 'classification/trained/model.pkl')
    joblib.dump(vectorizer,
'classification/trained/vectorizer.pkl')

```

Data yang sudah di *preprocessing* akan ketahap untuk melakukan pelatihan model dengan menggunakan fungsi “*train_model(data)*” yang telah didefinisikan pada kode diatas. Setelah model terlatih, model dan *vetorizer* akan disimpan dalam file “*model.pkl*” dan “*vectorizer.pkl*” yang dapat digunakan untuk mengklasifikasikan deskripsi pada sistem *bug reporting*.

3.7 PENGUJIAN MODEL

Pengujian model dengan menggunakan confusion matrix yang digunakan untuk mengevaluasi kinerja model klasifikasi pada data uji yang diketahui label sebenarnya. Matriks yang menyajikan hasil prediksi dari model secara rinci dan membandingkan dengan kelas sebenarnya dari setiap sampel data. Contoh dari confusion matrix pada sampel data yang ditunjukkan pada Tabel 3.10:

Tabel 3.10 Contoh *Confusion Matrix*

Tipe <i>Developer</i>	Prediksi <i>Front-End</i>	Prediksi <i>Back-End</i>
<i>Front-End</i>	85 (TP)	15 (FP)
<i>Back-End</i>	10 (FN)	150 (TN)

Dalam contoh diatas maka dapat disimpulkan bahwa:

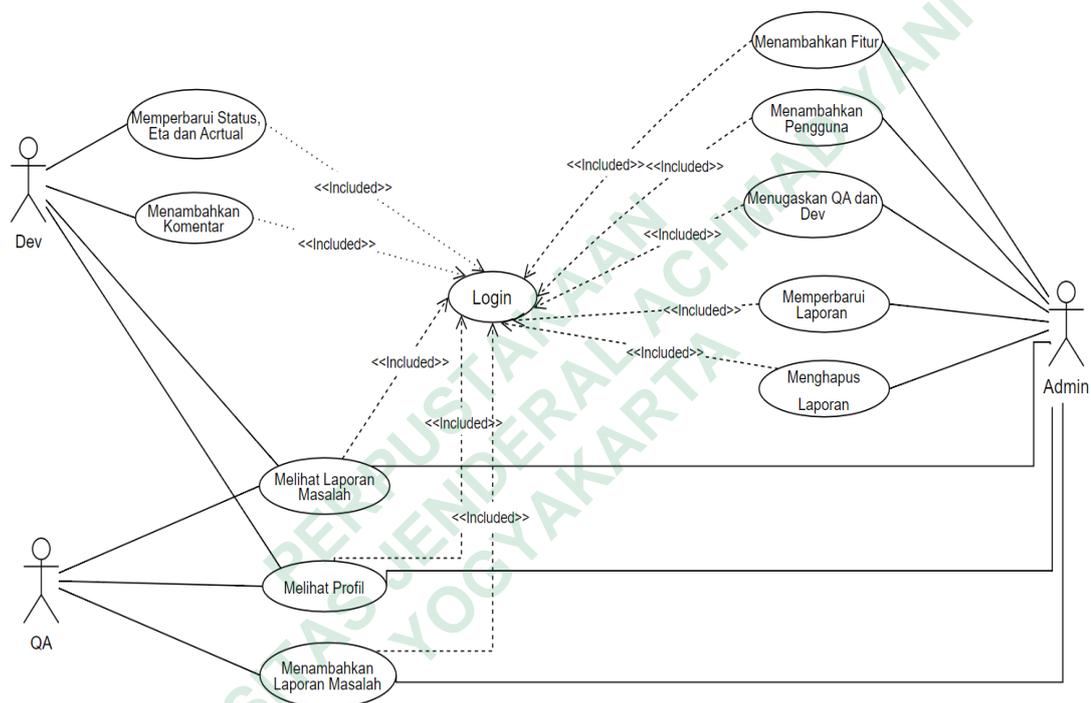
1. TP (*True Positive*) = 85, yaitu ada 85 sampel positif yang diprediksi benar sebagai positif.
2. TN (*True Negative*) = 150, yaitu adad 150 sampel negatif yang diprediksi benar sebagai negatif.
3. FP (*False Positive*) = 15, yaitu ada 10 sampel negatif yang salah di prediksi sebagai positif.
4. FN (*False Negative*) = 10, yaitu ada 10 sampel positif yang salah diprediksi sebagai negatif.

Confusion matrix memberikan informasi yang berguna tentang kinerja model klasifikasi. Dari matrix yang dihasilkan, dapat menghitung berbagai *metrix* evaluasi, seperti akurasi, *presisi*, *recall*, dan *F1-score*, untuk membantu mengukur seberapa baik model melakukan klasifikasi pada data uji.

3.8 PERANCANGAN SISTEM

3.8.1 Use Case Diagram

Pada tahap perancangan sistem dengan menggunakan *use case diagram* yang akan menggambarkan jalannya sistem yang akan dilakukan pengguna yang ditunjukkan pada Gambar 3.8.



Gambar 3.8 Use Case Diagram

Menjelaskan *use case diagram* kegiatan pengguna pada sistem pelaporan *bug* yang akan digunakan oleh admin, *quality assurance (qa)*, *developer* diantaranya *front-end* dan *back-end*. Penentuan tipe *developer* dilakukan pada tahap pengujian sistem dan memperbaiki masalah yang didapatkan oleh *quality assurance* yang bertugas untuk melakukan pengujian terhadap sistem yang sedang dikembangkan pada perusahaan. Terdapat beberapa fitur dan penjelasan terkait sistem *bug reporting* yang akan ditunjukkan pada Tabel 3.11.

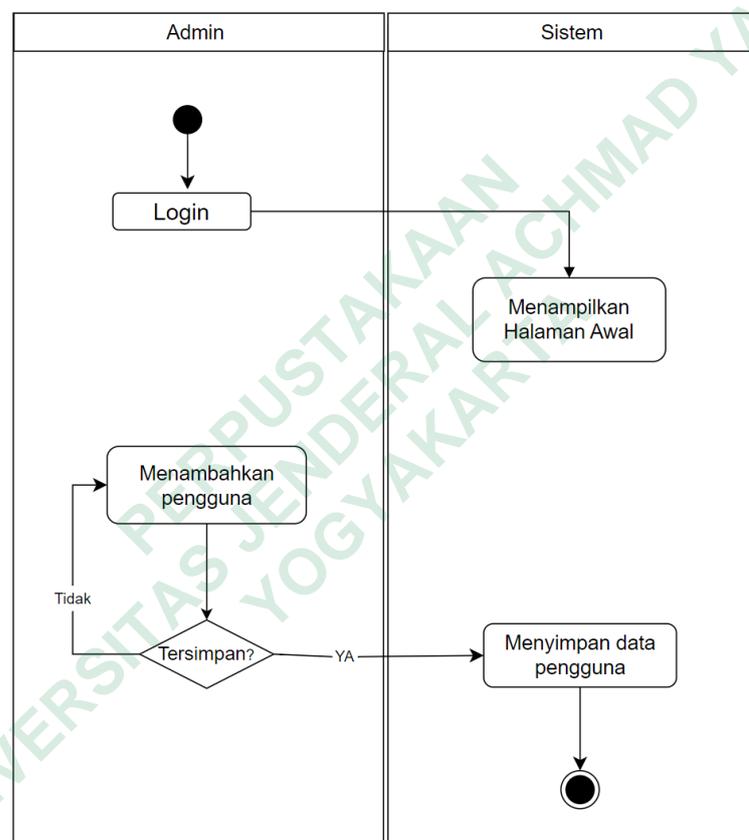
Tabel 3.11 Keterangan *Use Case* Diagram

Aktor	Use Case	Deskripsi
Admin, <i>Developer</i> (<i>front-end</i> & <i>back-end</i>) dan QA	Login dan melihat <i>profile</i>	Aktivitas yang menjelaskan <i>user</i> dapat mengakses <i>login</i> dan melihat <i>profile</i>
Admin	Menambahkan fitur	Aktivitas admin menambahkan fitur
	Menambahkan pengguna	Aktivitas admin menambahkan <i>developer</i> dan <i>qa</i>
	Melihat laporan fitur	Aktivitas admin dapat melihat laporan <i>bug</i> untuk semua fitur yang telah ditambahkan
	Menugaskan <i>developer</i> dan QA	Aktivitas admin menugaskan <i>developer</i> dan <i>qa</i>
	Menambahkan laporan masalah	Aktivitas admin dapat menambahkan laporan <i>bug</i>
	Mengubah laporan masalah	Aktivitas admin dapat mengubah laporan <i>bug</i>
	Menghapus laporan masalah	Aktivitas admin dapat menghapus laporan <i>bug</i>
	Mengubah kata sandi	Aktivitas admin dapat mengubah kata sandi
QA	Menambahkan laporan masalah	Aktivitas <i>qa</i> menambahkan laporan <i>bug</i> pada tabel
	Melihat laporan masalah berdasarkan fitur	Aktivitas <i>qa</i> hanya dapat melihat laporan <i>bug</i> untuk fitur yang diberikan izin akses oleh admin
	Mengubah kata sandi	Aktivitas <i>qa</i> dapat mengubah kata sandi
<i>Developer</i> (<i>Front-End</i> dan <i>Back-End</i>)	Mengubah status tugas	Aktivitas <i>developer</i> dapat mengubah status, <i>eta</i> dan <i>actual</i>
	Menambahkan komentar	Aktivitas <i>developer</i> dapat menambahkan komentar
	Mengubah kata sandi	Aktivitas <i>developer</i> dapat mengubah kata sandi

3.8.2 Activity Diagram

3.8.2.1 Activity Diagram Admin Menambahkan Pengguna

Aktivitas admin menambahkan pengguna berdasarkan *role* pengguna yaitu *front-end*, *back-end*, dan *quality assurance* dengan menambahkan dari menu pengguna untuk memasukkan nama pengguna (*username*) dan tipe pengguna yang ditunjukkan pada Gambar 3.9.



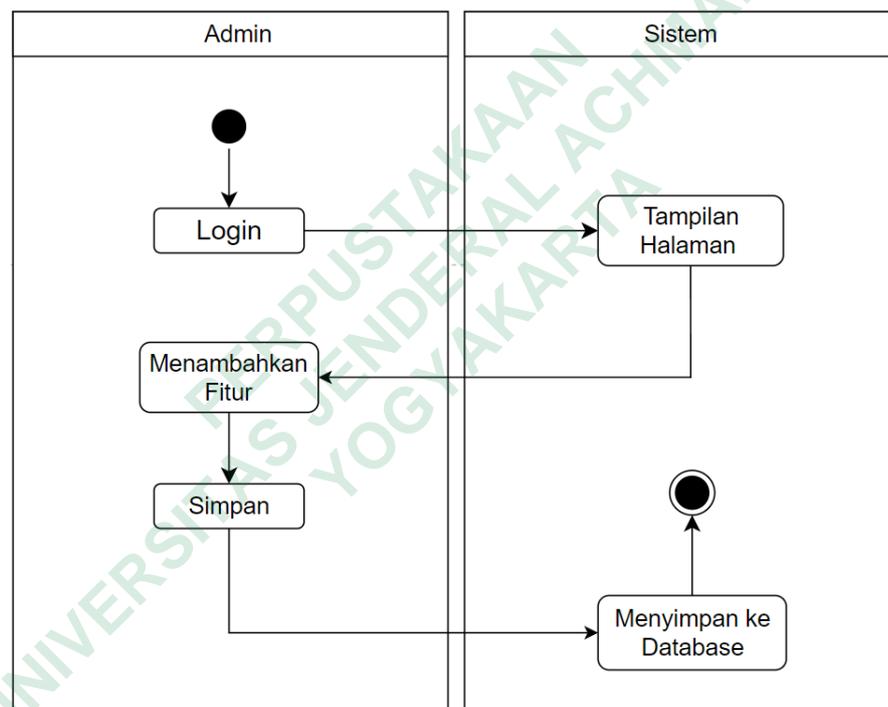
Gambar 3.9 Activity Menambahkan User

Admin terlebih dahulu *login* ke *platform* yang akan menampilkan menu pelaporan *bug*. Setelah itu admin akan menuju ke menu pengguna untuk dapat menambahkan pengguna, admin dapat menambahkan pengguna berdasarkan *role* dari pengguna yaitu *developer* (*front-end* atau *back-end*) dan *qa*. Jika pengguna yang dimasukkan sesuai dengan format maka pengguna akan disimpan ke database,

tetapi jika terjadi kesalahan saat memasukkan email maka admin akan kembali ke menu utama. *Activity* diagram menambahkan *user* ditunjukkan pada Gambar 3.3.

3.8.2.2 *Activity* Diagram Admin Menambahkan Fitur

Admin *login* terlebih dahulu menambahkan fitur dan akan menampilkan menu pelaporan *bug*. Setelah itu admin dapat menuju ke menu fitur dan dapat menambahkan fitur berdasarkan nomor referensi dari fitur yang diinginkan. Jika sudah mengisi data untuk fitur maka admin akan mengklik tombol kirimkan dan fitur akan ke simpan dalam database yang ditunjukkan pada Gambar 3.10.

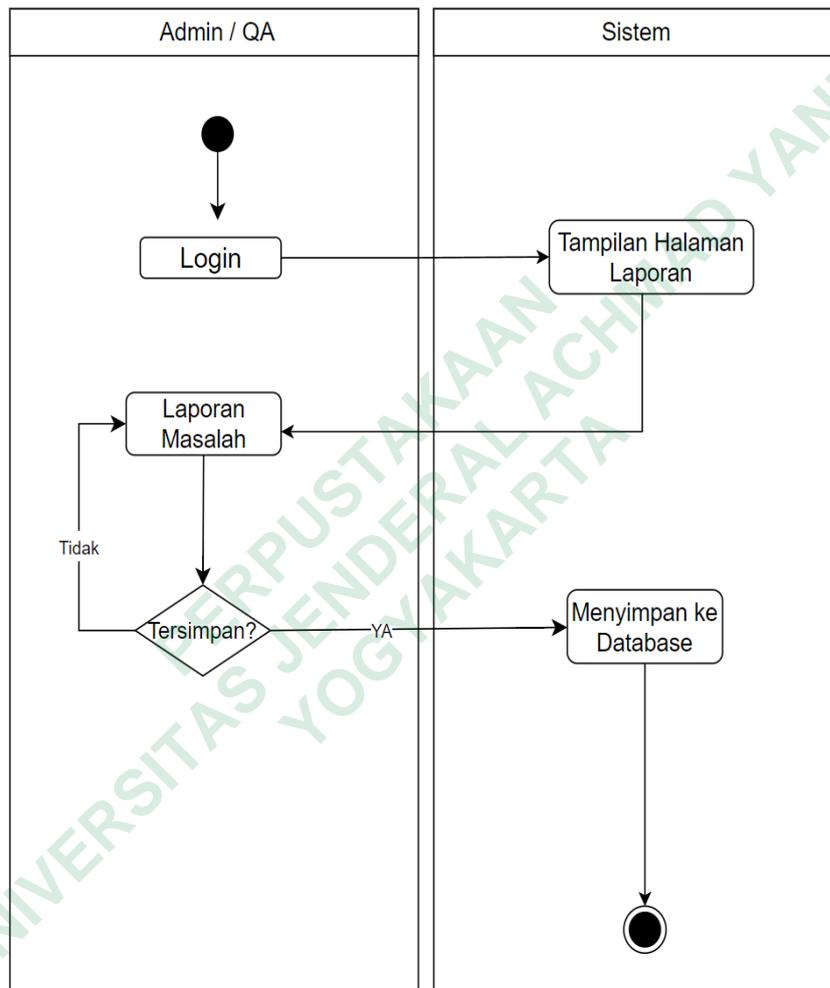


Gambar 3.10 *Activity* Menambahkan Fitur

3.8.2.3 *Activity* Diagram Menambahkan *Bug Reporting*

Activity diagram menambahkan laporan masalah yang di mulai dari pengguna melakukan *login* terlebih dahulu yang akan menuju ke halaman laporan *bug* dengan menampilkan *list* dari masalah yang sudah pernah dilaporkan sebelumnya. Qa dan admin dapat menambahkan laporan masalah dengan mengklik tombol “Add Issue”, maka akan menampilkan *popup* untuk mengisi data yang

dibutuhkan dalam laporan *bug*. Jika laporan *bug* berhasil maka akan menyimpan ke database dan jika laporan bug tidak berhasil atau masih ada beberapa bidang (*field*) yang belum diisi, maka akan tetap menampilkan *popup* untuk laporan *bug* yang ditunjukkan pada Gambar 3.11.



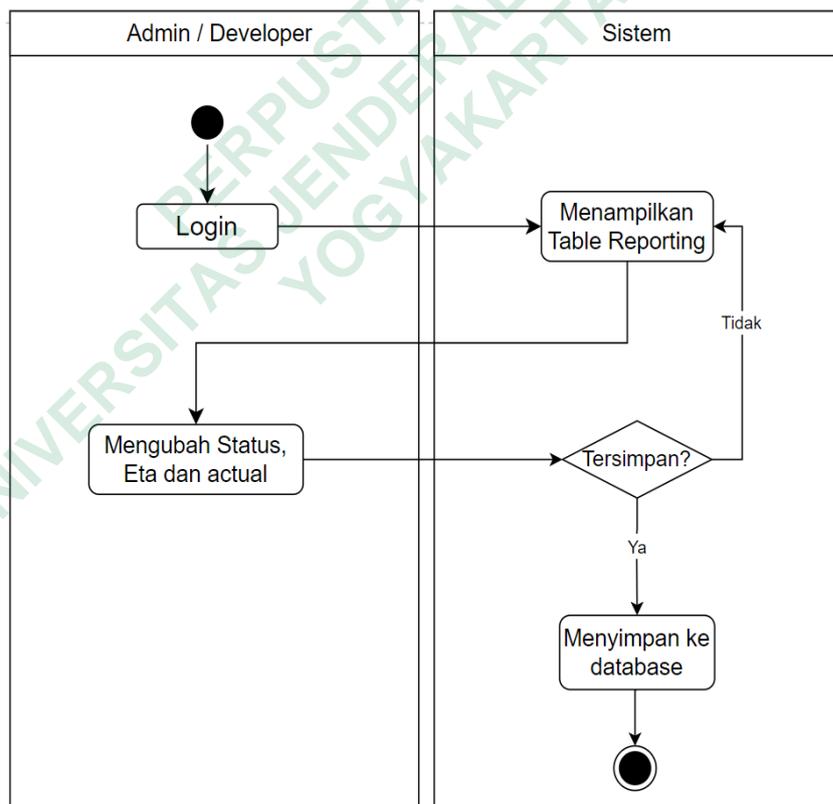
Gambar 3.11 Activity Menambah Laporan Masalah

3.8.2.4 Activity Diagram Developer Memperbarui Status

Activity diagram ini terdapat dua tipe *developer* yaitu *front-end* dan *back-end* yang berperan dalam memperbarui status, waktu *estimasi* (ETA) dan waktu *actual developer* pada bug reporting. Sebelum melakukan perubahan, developer diharuskan untuk melakukan login terlebih dahulu pada sistem. Setelah berhasil login, maka akan diarahkan ke halaman bug reporting yang ditunjukkan pada

Gambar 3.12. Setelah masuk ke halaman bug reporting, developer hanya memiliki akses untuk mengedit status, ETA, actual dan komentar. Developer dapat memilih status yang telah disediakan seperti “open”, “inprogress”, dev done” maupun “ready to test” untuk mengetahui status dari bug yang sedang diperbaiki. Selain itu, developer dapat memperbarui estimasi waktu yang diperlukan untuk menyelesaikan bug dan actual yang menunjukkan waktu sebenarnya saat bug berhasil diperbaiki.

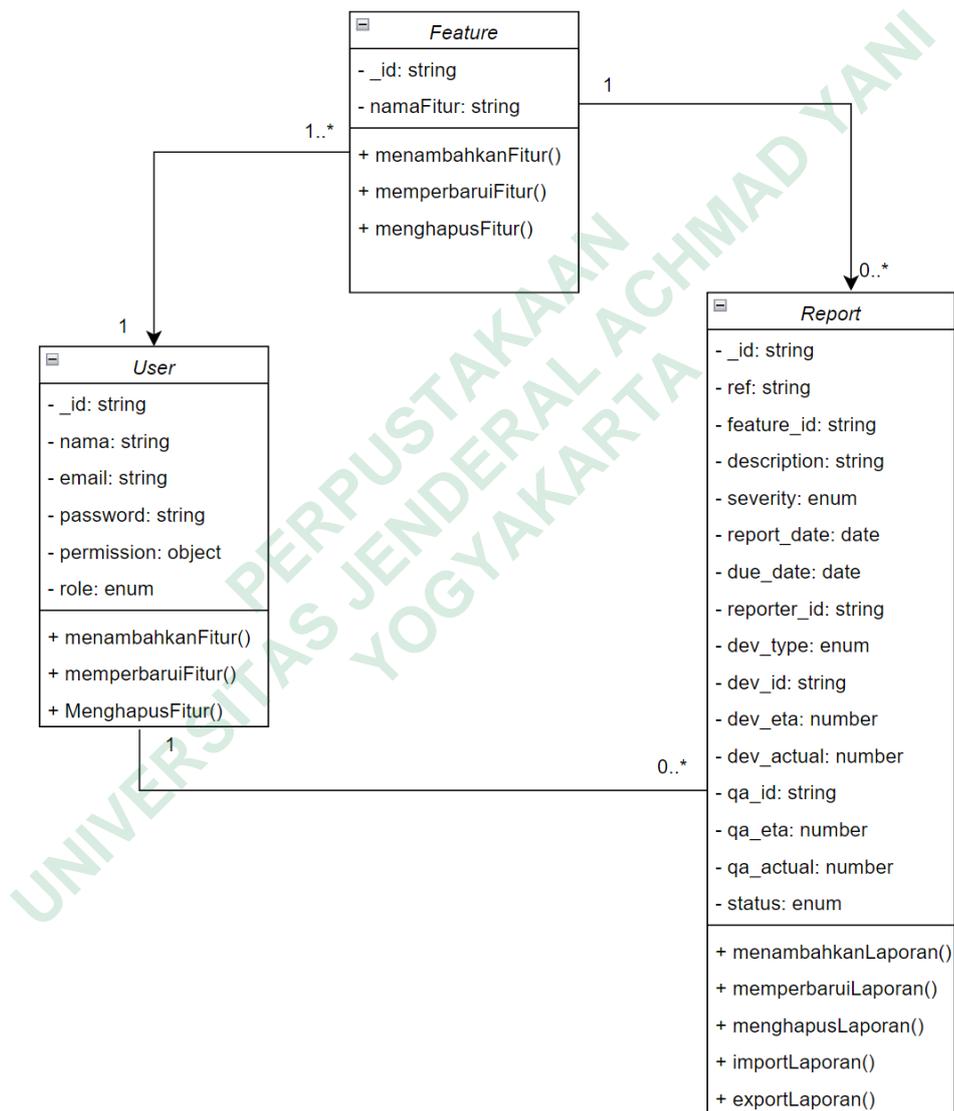
Selain itu, developer juga dapat memberikan komentar terkait bug yang berguna untuk memberikan informasi tambahan tentang perubahan yang telah diambil untuk memperbaiki bug. Data yang telah diubah akan disimpan kembali ke database. Hal ini memastikan bahwa informasi yang diperbarui akan tersimpan secara permanen.



Gambar 3.12 Activity Mengubah Status

3.8.3 Class Diagram

Class diagram digunakan untuk menunjukkan struktur dan atribut dari setiap kelas untuk merepresentasikan hubungan dari kelas-kelas pada Sistem *bug reporting* yaitu terdapat tabel *user*, *feature* dan *report* yang ditunjukkan pada Gambar 3.13.



Gambar 3.13 *Class Diagram*

3.8.4 Skema Basis Data

3.8.4.1 User (Pengguna)

Tabel *user* digunakan untuk menyimpan data dari pengguna yang dibagi berdasarkan dari *role* pengguna yaitu admin, fe, be dan qa. Serta id pengguna menjadi *primary key* yang ditunjukkan pada Tabel 3.12.

Tabel 3.12 Database *User*

<i>User</i>		
<i>Field</i>	<i>Type</i>	<i>Action</i>
id	object_id	PK
nama	string	
email	string	
password	string	
permission	object	
role	enum	

3.8.4.2 Feature (Fitur)

Tabel *feature* digunakan untuk menyimpan fitur dan hanya *role* admin yang dapat menambahkan fitur dan id menjadi *primary key* pada tabel fitur yang ditunjukkan pada Tabel 3.13.

Tabel 3.13 Tabel *Feature*

<i>Feature</i>		
<i>Field</i>	<u><i>Type</i></u>	<i>Action</i>
id	object_id	PK
nama	string	

3.8.4.3 Report (Pelaporan)

Tabel *report* digunakan untuk menyimpan data pelaporan *bug* yang dilaporkan berdasarkan fitur yang dikembangkan, maka id dari fitur akan menjadi *primary key* dan *feature_id*, *reporter_id*, *dev_id* dan *qa_id* akan menjadi

reference/fk untuk mengidentifikasi id yang bukan dari id tabel *report* yang ditunjukkan pada Tabel 3.14.

Tabel 3.14 Tabel *Report*

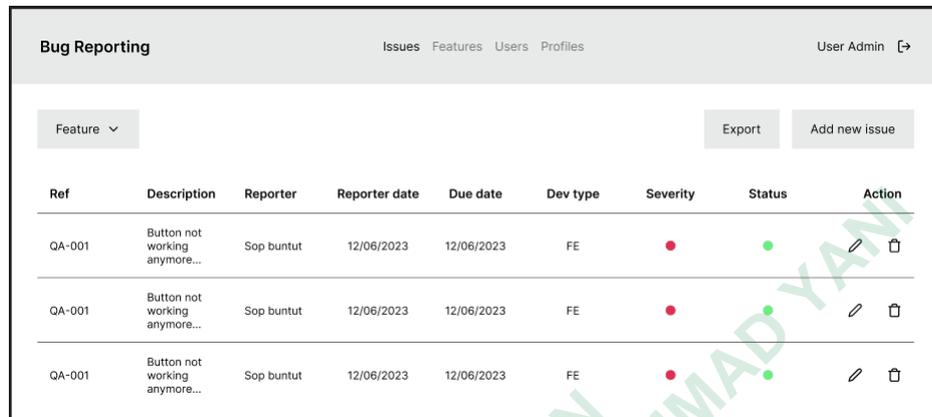
<i>Report</i>		
<i>Field</i>	<i>Type</i>	<i>Action</i>
id	object_id	PK
ref	string	
feature_id	object_id	Reference
description	string	
severity	enum	
report_date	date	
due_date	date	
reporter_id	object_id	Reference
dev_type	enum	
dev_id	object_id	Reference
dev_eta	number	
dev_actual	number	
qa_id	object_id	Reference
qa_eta	number	
qa_actual	number	
status	enum	

3.8.5 Desain Sistem

3.8.5.1 Desain Halaman *Issue*

Halaman tabel *issue* dapat diakses oleh admin, qa dan *developer* (*front-end* dan *back-end*) yang ditunjukkan pada Gambar 3.14. merupakan halaman semua pelaporan bug yang dilaporkan oleh penguj. Admin dapat melihat, menambahkan, memperbarui, dan menghapus *reporting*. Qa dapat melihat *reporting bug* yang diberikan akses oleh admin, menambahkan dan memperbarui. *Developer* (*front-end*

dan *back-end*) hanya dapat melihat tabel *reporting bug* yang diberikan akses oleh admin memperbarui kolom status, eta, aktual dan komen.



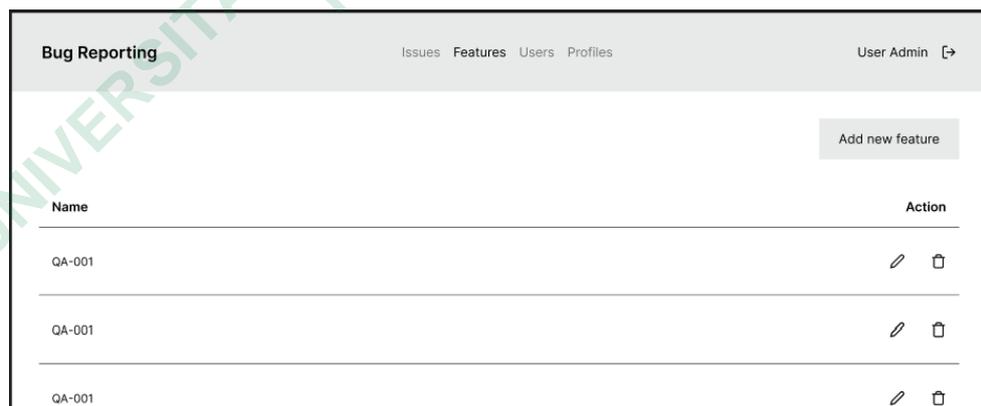
The screenshot shows the 'Bug Reporting' page with a navigation bar containing 'Issues', 'Features', 'Users', and 'Profiles'. The user is logged in as 'User Admin'. Below the navigation bar, there is a 'Feature' dropdown menu, an 'Export' button, and an 'Add new issue' button. The main content is a table with the following columns: Ref, Description, Reporter, Reporter date, Due date, Dev type, Severity, Status, and Action. The table contains three rows of data, all with the same values: Ref: QA-001, Description: Button not working anymore..., Reporter: Sop buntut, Reporter date: 12/06/2023, Due date: 12/06/2023, Dev type: FE, Severity: Red dot, Status: Green dot, and Action: Edit and Delete icons.

Ref	Description	Reporter	Reporter date	Due date	Dev type	Severity	Status	Action
QA-001	Button not working anymore...	Sop buntut	12/06/2023	12/06/2023	FE	●	●	
QA-001	Button not working anymore...	Sop buntut	12/06/2023	12/06/2023	FE	●	●	
QA-001	Button not working anymore...	Sop buntut	12/06/2023	12/06/2023	FE	●	●	

Gambar 3.14 Halaman *Issues*

3.8.5.2 Desain Halaman *Feature*

Halaman *feature* hanya dapat diakses oleh admin yang ditunjukkan pada Gambar 3.15. hanya dapat diakses oleh admin dan admin dapat menambahkan fitur baru, memperbarui dan menghapus fitur. Admin dapat memberikan hak akses kepada *role* pengguna lain seperti *qa* dan *developer* (*front-end* dan *back-end*) untuk dapat mengakses fitur yang diizinkan oleh admin.



The screenshot shows the 'Feature' page with a navigation bar containing 'Issues', 'Features', 'Users', and 'Profiles'. The user is logged in as 'User Admin'. Below the navigation bar, there is an 'Add new feature' button. The main content is a table with the following columns: Name and Action. The table contains three rows of data, all with the same values: Name: QA-001, and Action: Edit and Delete icons.

Name	Action
QA-001	
QA-001	
QA-001	

Gambar 3.15 Halaman *Feature*

3.8.5.3 Desain Halaman Tabel *User*

Halaman tabel *user* hanya dapat diakses oleh admin yang ditunjukkan pada Gambar 3.16. merupakan halaman untuk menambahkan pengguna berdasarkan *role* dari pengguna yaitu *qa* dan *developer* (*front-end* atau *back-end*) yang ditampilkan dalam bentuk tabel yang hanya dapat diakses oleh admin. Admin juga dapat menyunting dan menghapus pengguna, jika pengguna dihapus maka tidak akan menghapus laporan yang pernah dilaporkan sebelumnya.

First name	Last name	Email	Role	Action
Sop	Buntut	sop@buntut.com	Admin	
Sop	Buntut	sop@buntut.com	FE	
Sop	Buntut	sop@buntut.com	FE	

Gambar 3.16 Halaman *User*

3.8.5.4 Desain Halaman *Profile*

Halaman *profile* dapat diakses oleh admin, *qa* dan *developer* (*front-end* dan *back-end*) yang ditunjukkan pada Gambar 3.17 merupakan halaman untuk menyimpan data dari admin, *qa* dan *developer* (*front-end* dan *back-end*) dan dapat mengubah dan menghapus *field* atau gambar yang sudah dimasukkan sebelumnya.

Gambar 3.17 Halaman *Profile*