

## **BAB IV PERANCANGAN**

### **4.1. Analisis Sistem Terdahulu**

Analisis sistem terdahulu memiliki tujuan untuk mengetahui kelebihan, kekurangan serta cara pembuatan maupun cara kerja dari sistem yang telah dibuat sebelumnya. Dengan melakukan analisis sistem terdahulu diharapkan dapat memperbaiki kekurangan yang dimiliki oleh sistem terdahulu, serta dapat mengembangkan hal – hal yang belum dimiliki oleh sistem terdahulu. Selain itu juga dapat mengetahui kebutuhan apa saja yang harus dimiliki oleh sistem yang akan dikembangkan.

Sistem terdahulu yang dikembangkan oleh Nathania Elvina, Andre Rusli, dan Seng Hansun tahun 2019, meneliti mengenai *chat bot* pada aplikasi line yang dikembangkan menggunakan bahasa pemrograman java serta algoritma *Naïve Bayes* untuk mengklasifikasi antara pesan yang penting dan pesan yang tidak penting. *Chatbot* ini kemudian dipasang di dalam sebuah *group chat* komite di dalam aplikasi line. Dataset yang digunakan adalah *history chat* di dalam *group* komite tersebut.

Aplikasi yang dibangun dalam penelitian ini merupakan *chatbot* pada aplikasi Discord menggunakan algoritma *Naïve Bayes* serta bahasa pemrograman python. *Chatbot* ini nantinya akan dipasang di dalam sebuah *server chat* discord untuk dapat melakukan filter *bad word* pada *chat* yang dikirimkan anggota *server*. Dataset yang digunakan adalah *Dataset bad word* beserta *Dataset* dari *history chat* di dalam server sebagai data latih.

### **4.2. Spesifikasi Kebutuhan Sistem Baru**

Setelah melakukan analisis sistem terdahulu menghasilkan berupa spesifikasi yang dibutuhkan untuk sistem baru yaitu sistem implementasi filter *bad word* pada *chatbot* discord. Untuk membangun sistem diperlukannya spesifikasi yang meliputi spesifikasi kebutuhan input, spesifikasi proses, dan spesifikasi kebutuhan output.

#### 4.2.1. Spesifikasi Kebutuhan Input

Data yang digunakan dalam penelitian diambil dari beberapa sumber yaitu :

1. Dataset *id-abusive-language-detection* milik ibrohim
2. Dataset *izzan Indonesia text toxicity* yang di *scraped* dari *social media* Instagram, Twitter, dan Kaskus. Dengan 4 kategori label: pornografi, *hate speech*, *racism*, *radicalism*.
3. Dataset Rizal komentar Instagram *cyberbullying*
4. Dataset *Swear Words* berbahasa Indonesia

Data – data tersebut kemudian digabungkan dan diberikan label untuk melatih model dalam mengenali ungkapan negatif (*bad words*) dan ungkapan positif. Sehingga *chatbot* dapat membaca Input yang nantinya berupa pesan teks, dan bisa membedakannya. sehingga penting bagi *chatbot* untuk mengetahui data apa saja yang termasuk bagian *bad words*.

Untuk mengetahui dataset yang digunakan sudah memiliki kriteria data yang cukup untuk memfilter *bad word*, maka perlu diperhatikan hal – hal berikut.

##### 1. Jumlah data

Berdasarkan *rule of thumb* menyatakan *supervised deep learning algorithm* dapat memiliki performa yang baik dengan 5000 data yang sudah diberikan label per kategorinya (Goodfellow et al., 2017).

##### 2. Pengkategorian

Data yang dikumpulkan mencakup dalam berbagai kategori yang termasuk dalam *bad word*, contoh kategorinya adalah :pornografi, *hate speech*, *racism*, dan *radicalism*. Menurut Ibrohim dan Budi kata kasar bisa dibagi menjadi beberapa kategori yaitu : kondisi, binatang, makhluk astral, objek, bagian tubuh, anggota keluarga, aktivitas seksual, dan pekerjaan (Ibrohim & Budi, 2018).

##### 3. Pemahaman Konteks

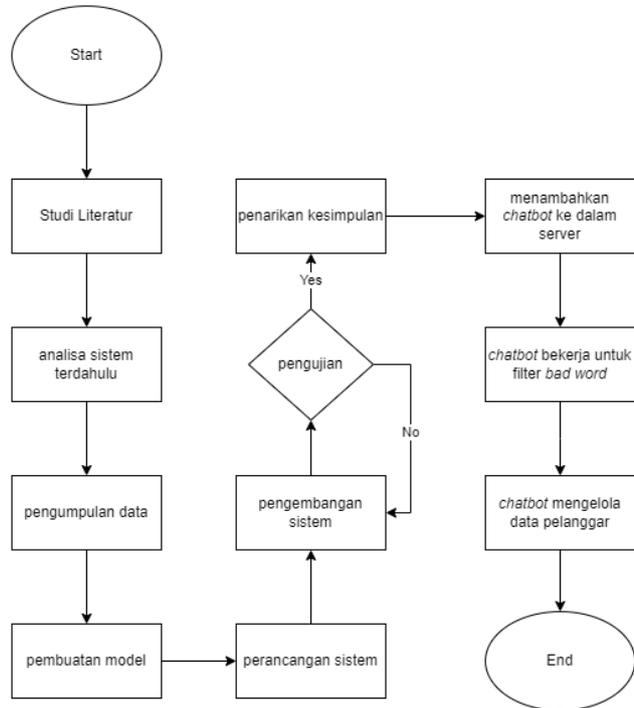
Data yang dikumpulkan terdiri dari berbagai macam konteks, dalam konteks tertentu kata kata kasar tidak menjadi kata kasar.

#### 4. Pemantauan dan Evaluasi

Data kata-kata kasar sewaktu – waktu dapat berubah seiring perkembangan jaman serta terbawanya budaya dari daerah lain, karena itu perlu adanya pemantauan serta evaluasi dan pengupdaten data.

##### 4.2.2. Spesifikasi Proses

Rincian proses untuk implementasi filter *bad word* pada *chatbot* discord mencakup pada studi literatur, analisa sistem terdahulu, pengumpulan data, pembuatan model, penerapan algoritma pada model, perancangan sistem, pengembangan sistem, dan terakhir pengambilan kesimpulan. Hasil dari proses



Gambar 4. 1 flowchart proses

tersebut *chatbot* akan diintegrasikan dengan platform *chat discord* yang kemudian dapat digunakan di dalam server-server di discord seperti yang terlihat pada Gambar 4.1.

*Chatbot* memiliki fungsi utama untuk memfilter kata – kata dari obrolan server yang diindikasikan sebagai *bad word*. Adapun fitur yang dimiliki *chatbot* antara lain:

- a. Dapat memfilter *chat bad word* dan mengirimkan peringatan kepada user yang mengirimkannya.
- b. Dapat mengelola data pelanggar yaitu orang yang melanggar aturan *chat* karena mengirimkan *bad word* di *chat*. Dengan melakukan *mute*, *kick*, dan *ban*.
- c. Dapat menampilkan list pelanggar
- d. Dapat menampilkan fitur yang dimiliki *chatbot*.

*Chatbot* yang dikembangkan dapat dimasukkan ke dalam server oleh admin server di discord.

#### **4.2.3. Spesifikasi Kebutuhan Output**

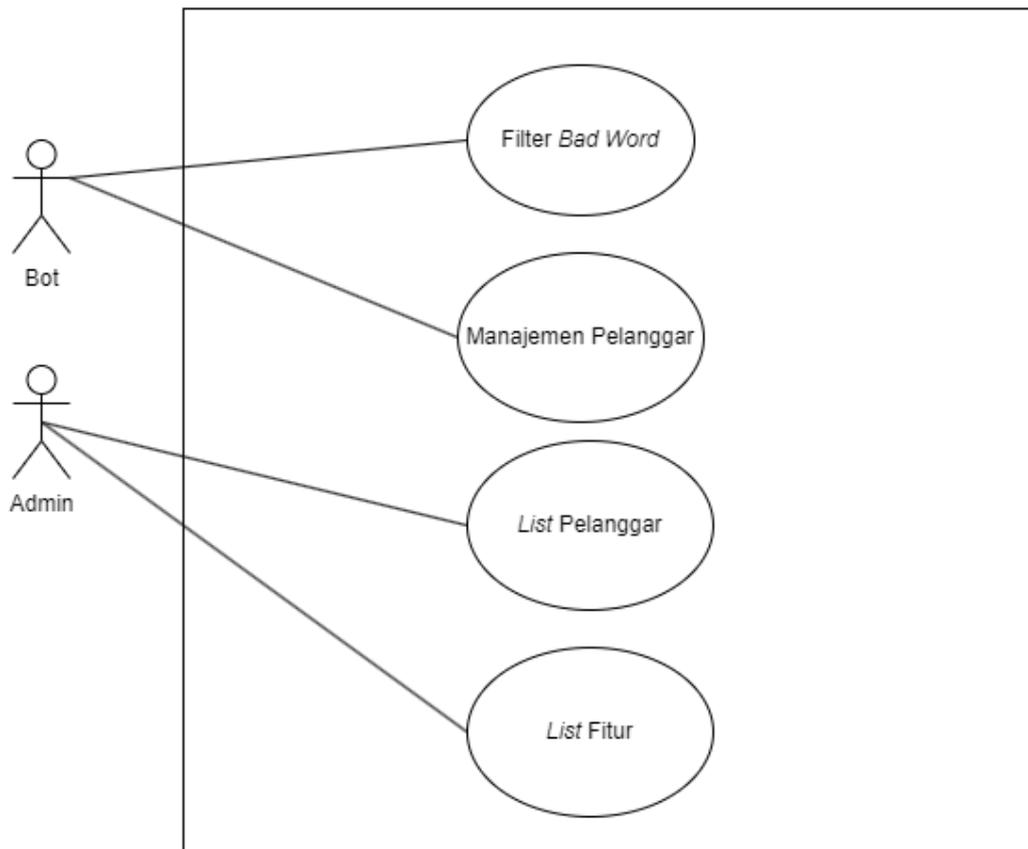
Output yang dihasilkan oleh proses berupa model yang diintegrasikan dengan *bot discord*, sehingga dinamakan *chatbot*. *Chatbot* dapat mendeteksi pesan teks dari pengguna discord di dalam server apakah mengandung *bad words*, jika iya maka *chatbot* akan mengirimkan peringatan kepada pengirim pesan *chat*.

#### **4.3. Perancangan Sistem**

Untuk mengembangkan *chatbot filter bad word* ini, dibutuhkan sebuah rancangan yang bisa menggambarkan secara keseluruhan bagaimana cara kerja sistem. Rancangan tersebut dapat membantu pengembangan menjadi lebih detail dan lebih cepat dalam prosesnya. Dalam merancang sistem ini dibutuhkan sebuah gambaran yang disebut *Unified Modeling Language (UML)*. UML dapat membantu pengembang agar fitur yang dibuat berfungsi sesuai dengan kebutuhan pengguna yang telah tergambar di UML sebelumnya, sehingga tidak ada lagi yang

tertinggal ataupun terlewat. Selanjutnya akan dibahas mengenai UML yang diperlukan untuk mengembangkan *chatbot* ini.

#### 4.3.1. Use Case



Gambar 4. 2 Use Case Chatbot

*Use case* diagram menggambarkan hubungan antara aktor yaitu pengguna aplikasi dengan fitur – fitur yang ada pada sistem. Gambar 4.2 adalah *use case* diagram yang menggambarkan secara visual tentang aktifitas yang dapat dilakukan aktor pada fitur – fitur yang tersedia dalam sistem/*chatbot*.

Untuk mendapatkan detail dari *use case* diagram dibutuhkanlah sebuah *scenario use case* yang menggambarkan detail dari *use case* dalam bentuk tabel. Tabel *scenario* memuat beberapa informasi, yaitu; Nama *scenario*, aktor, deskripsi singkat, langkah – langkah *scenario* yang dilakukan, dan langkah – langkah *scenario* alternatif untuk menghadapi *case* yang tidak diharapkan.

Tabel 4. 1 Scenario Usecase Filter Bad Word

Nama Use Case	Filter Bad Word	
Deskripsi	Admin/User dapat melihat chatbot memberikan peringatan kepada user yang mengirim chat mengandung bad word	
Aktor	Bot	
Skenario Normal	Aksi Aktor	Respon Sistem
	Step 1: Admin/user memasukkan bad word	
		Step 2: Memberikan peringatan kepada user yang mengirim bad word
	Step 3: Admin/user dapat melihat peringatan yang diberikan chatbot	
Skenario Alternatif	Aksi Aktor	Respon Sistem
	Step 1: Admin/user memasukkan kata tidak mengandung bad word	
		Step 2: Chatbot tidak melakukan response apapun

Tabel 4. 2 Scenario Usecase Manajemen Pelanggar

Nama Use Case	Manajemen Pelanggar	
Deskripsi	Bot mengelola data pelanggar di server secara otomatis, berdasarkan frekuensi pelanggar mengirim bad word	
Aktor	Bot	
Skenario Mute	Aksi Aktor	Respon Sistem
	Step 1: Chatbot mendeteksi ada user yang mengirimkan bad words lebih dari 3 kali	
		Step 2: Memberikan perintah mute selama 1 hari kepada chatbot untuk user tersebut
	Step 3: Chatbot melakukan mute terhadap user selama 1 hari	
Skenario Kick	Step 1: Chatbot mendeteksi setelah	

	user yang ter-mute selama 1 hari, mengirim <i>bad words</i> lagi	
		<i>Step 2:</i> Memberikan perintah kick kepada <i>chatbot</i> untuk user tersebut
	<i>Step 3:</i> <i>Chatbot</i> melakukan <i>kick</i> terhadap user	
Skenario Ban	<i>Step 1:</i> Mendeteksi user yang telah di kick masuk kembali ke dalam server dan mengirimkan <i>bad words</i> lagi	
		<i>Step 2:</i> Memberikan perintah ban kepada <i>chatbot</i> untuk user tersebut
	<i>Step 3:</i> <i>Chatbot</i> melakukan ban terhadap user	

Tabel 4. 3 Scenario Usecase List Fitur

Nama Use Case	List Fitur	
Deskripsi	Admin/User dapat menampilkan fitur <i>chatbot</i> yang ada di server mereka	
Aktor	Admin / User	
Skenario Normal	Aksi Aktor	Respon Sistem
	<i>Step 1:</i> Admin/user membuka setting di discord lalu ke bagian manage integration	
		<i>Step 2:</i> Menampilkan list fitur yang ada pada <i>chatbot</i>

Tabel 4. 4 Scenario Usecase List Pelanggar

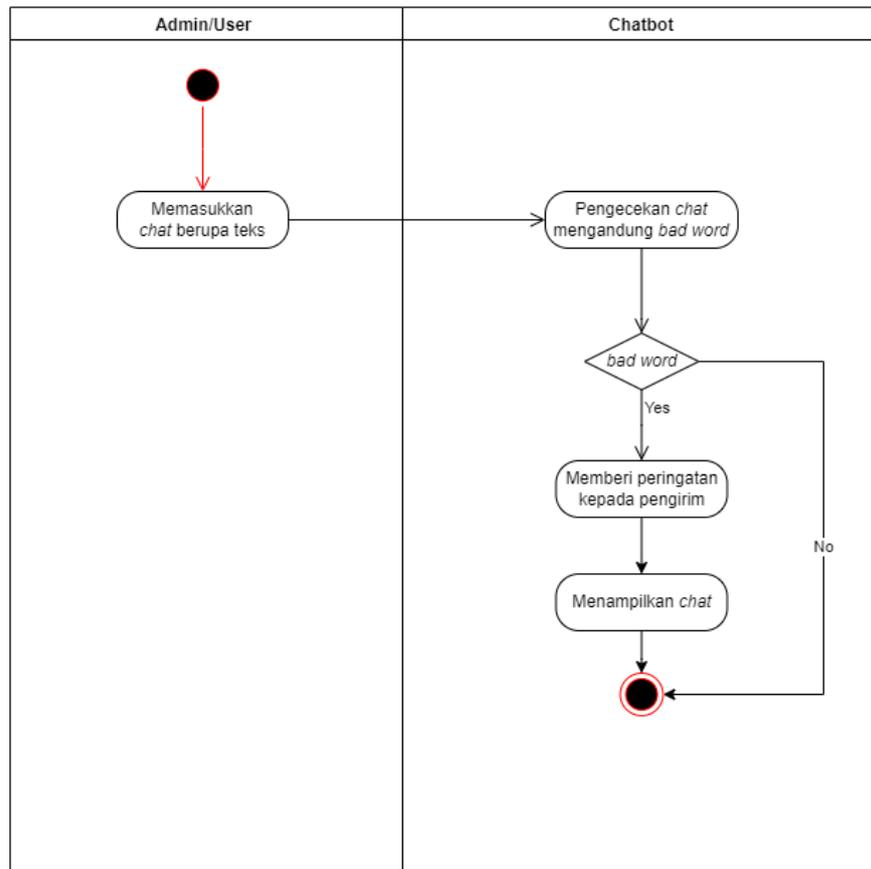
Nama Use Case	List Pelanggar	
Deskripsi	Admin/ <i>User</i> dapat menampilkan list pelanggar yang mengirimkan <i>bad words</i>	
Aktor	Admin / <i>User</i>	
Skenario Normal	Aksi Aktor	Respon Sistem
	<i>Step 1:</i> Admin/ <i>user</i> mengetikkan perintah /users di dalam <i>chat</i>	
	<i>Step 2:</i> Menampilkan list pelanggar yang mengirimkan <i>bad words</i>	

#### 4.3.2. Activity Diagram

*Activity* diagram merupakan sebuah diagram yang menggambarkan aktivitas serta alur kerja sistem secara terstruktur.

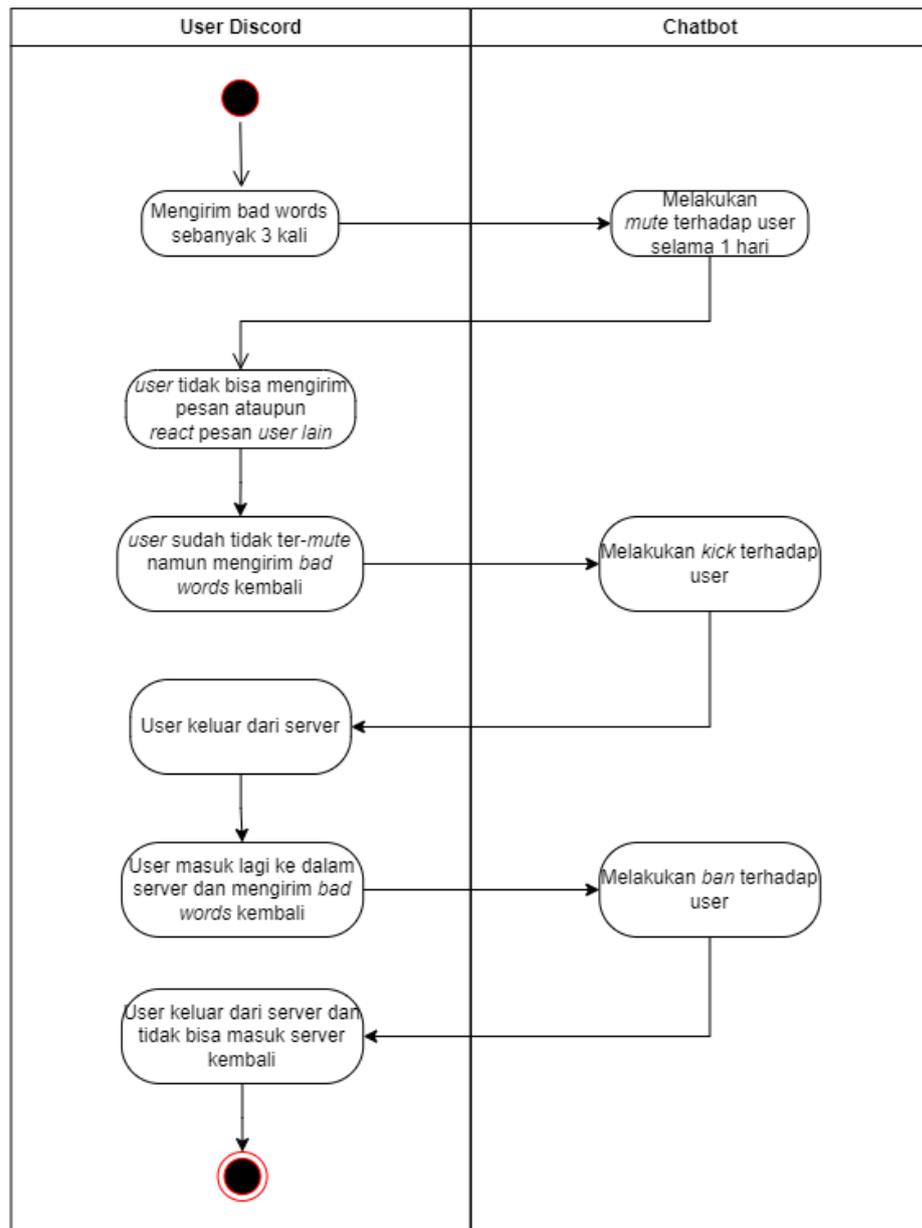
##### a) *Activity Diagram Filter Bad Word*

Gambar 4.3 merupakan tampilan *activity diagram* untuk proses alur kerja sistem pada saat filter *bad word*. Diagram memiliki 2 entitas yaitu Admin/*user* di discord dan *chatbot*. Admin/*user* memasukkan input *chat* berupa teks, selanjutnya *chatbot* akan menganalisa apakah di dalam *chat* tersebut mengandung *bad word* di dalamnya atau tidak. Bila terdapat *bad word* maka *chatbot* akan memberikan peringatan kepada user yang mengirimkan *bad word* tersebut. Namun bila tidak terdapat *bad word*, *chatbot* tidak memberikan *response* apapun.



Gambar 4. 3 Activity Diagram Filter Bad Word

b) Activity Diagram Manajemen Pelanggar



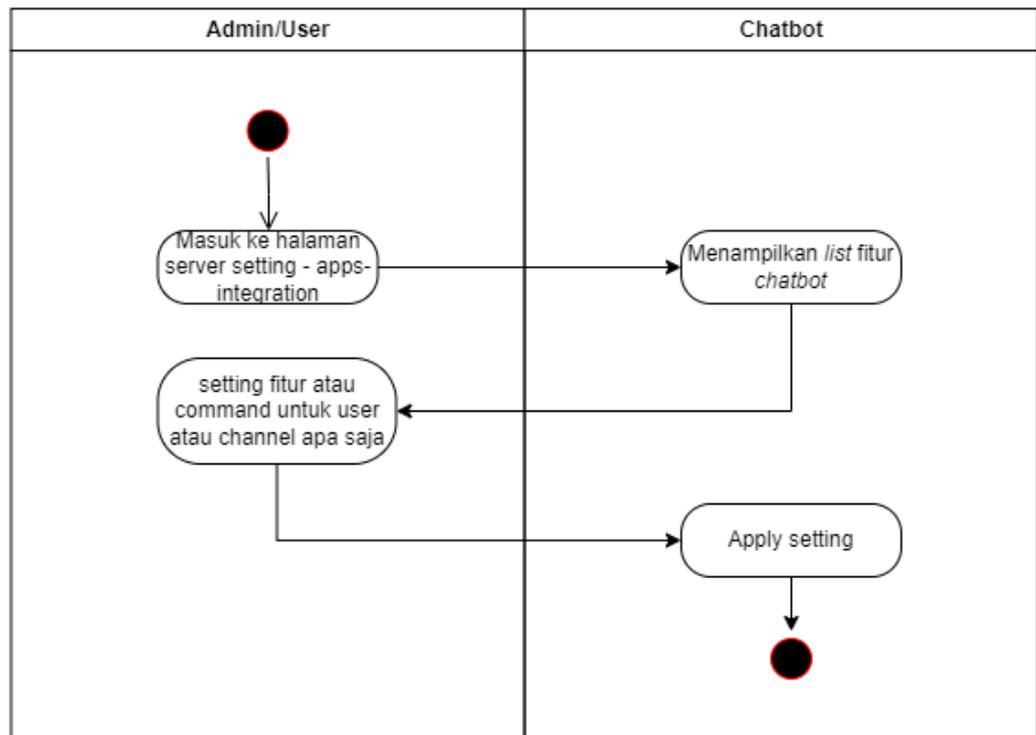
Gambar 4. 4 Activity Diagram Kelola Data Pelanggar

Pada gambar 4.4 ditampilkan *activity diagram* bagaimana bot mengelola data pelanggar. Jika user mengirimkan *bad words* sebanyak 3 kali, maka *chatbot* akan melakukan *mute* terhadap *user* tersebut selama 1 hari. User yang ter-*mute* tidak dapat memberikan pesan dan me-*react* pesan *user* lain. Setelah 1 hari, user tersebut sudah tidak ter-*mute*, namun jika dia mengirim *bad words* lagi, maka *chatbot* akan melakukan *kick*, sehingga user tersebut keluar dari server. User yang ter-*kick*, jika ia masuk

kembali ke dalam server dan mengirim *bad words* lagi, maka *chatbot* akan melakukan *ban* terhadap *user* sehingga *user* tidak akan bisa masuk kembali ke dalam server.

c) *Activity Diagram List Fitur*

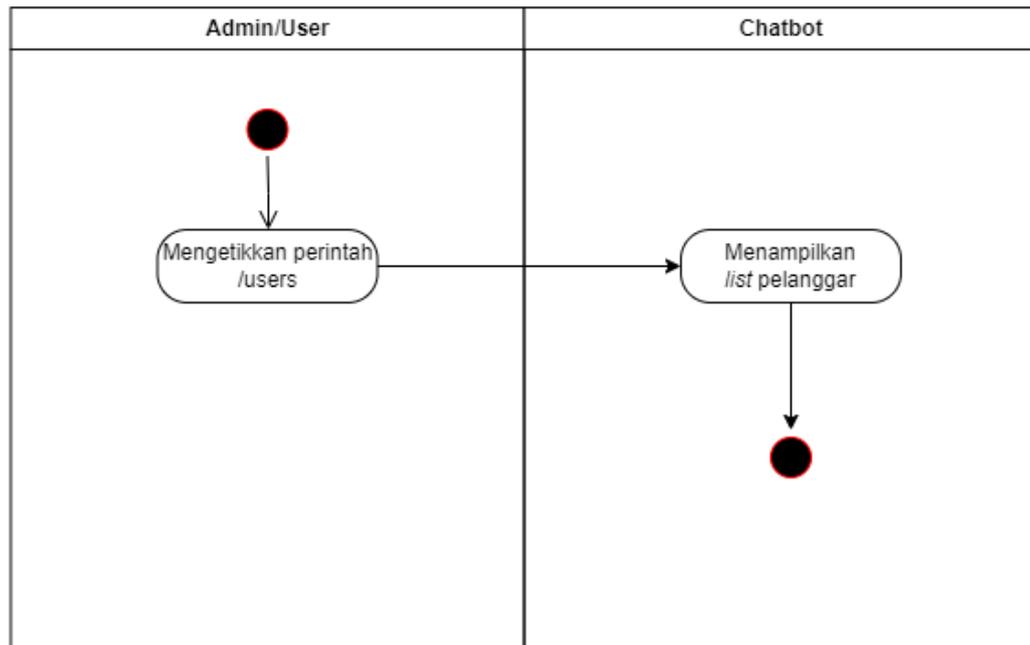
Untuk dapat menampilkan fitur yang ada pada *chatbot* admin/*user* harus memasuki halaman server *setting* discord terlebih dahulu dan memilih *chatbot*. *Chatbot* kemudian akan menampilkan list fitur.



Gambar 4. 5 Activity Diagram List Fitur

d) *Activity Diagram List Pelanggar*

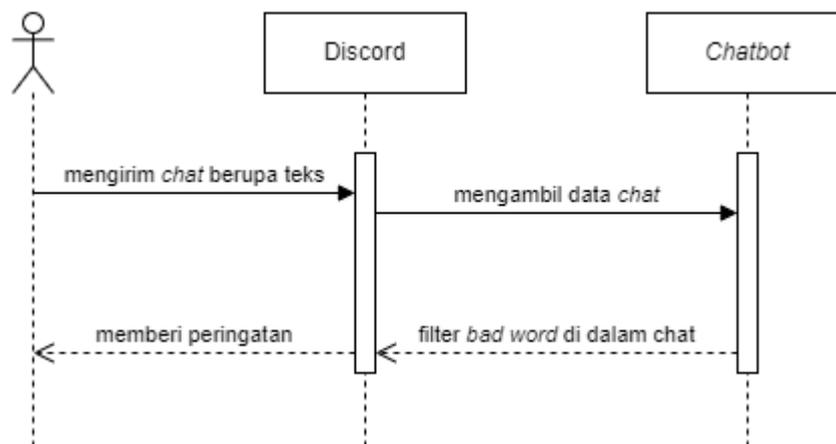
Untuk dapat menampilkan *list* pelanggar yang telah mengirimkan *bad words*, *user* harus mengetikkan perintah */users* di dalam *chat*. *Chatbot* kemudian akan meresponse dengan menampilkan list pelanggar.



Gambar 4. 6 Activity Diagram List Pelanggar

### 4.3.3. Sequence Diagram

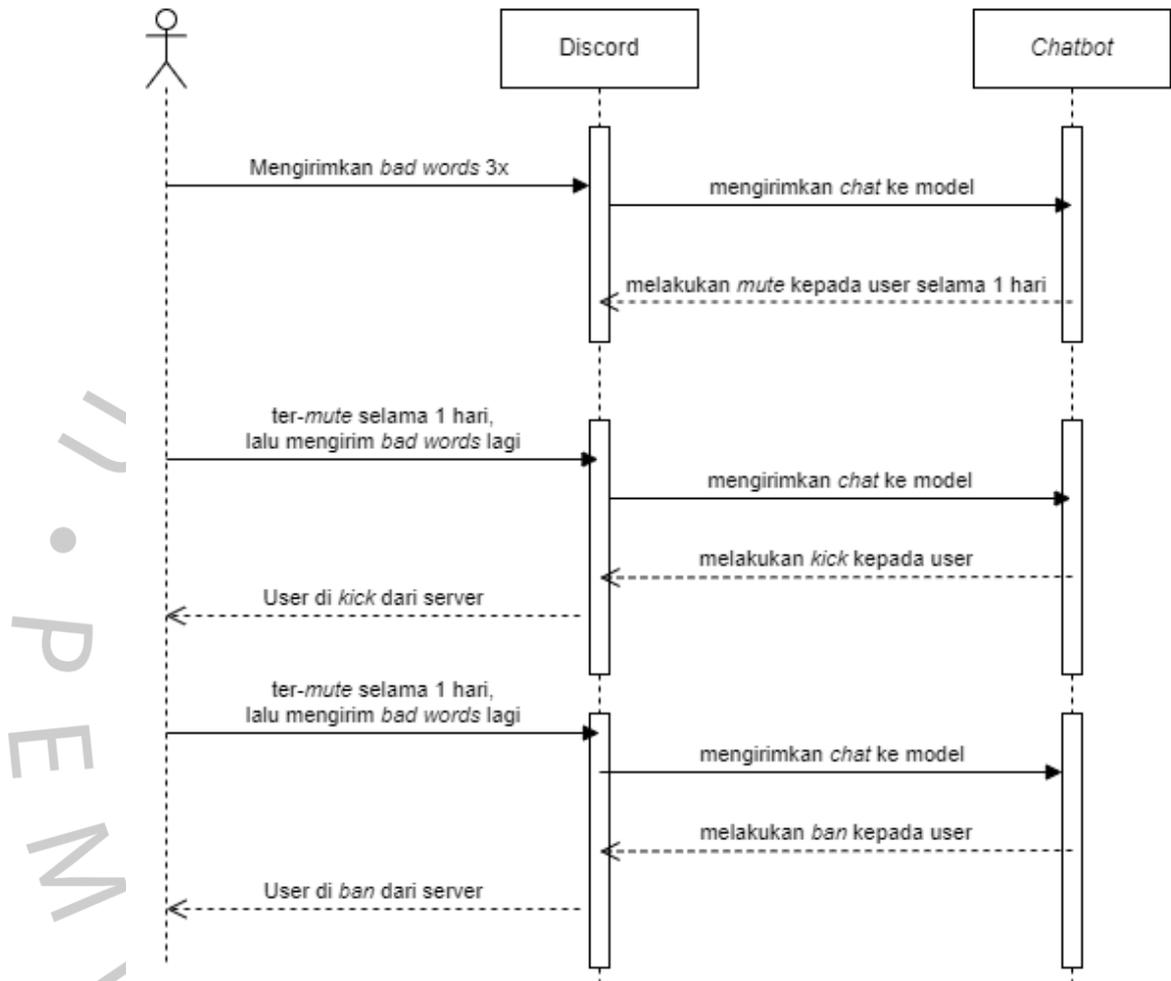
*Sequence* diagram merupakan diagram yang menggambarkan perjalanan data dari komponen sistem sehingga interaksi tersebut dapat membantu dalam memudahkan identifikasi masalah serta perbaikan masalah di waktu mendatang.



Gambar 4. 7 Sequence Diagram Filter Bad Word

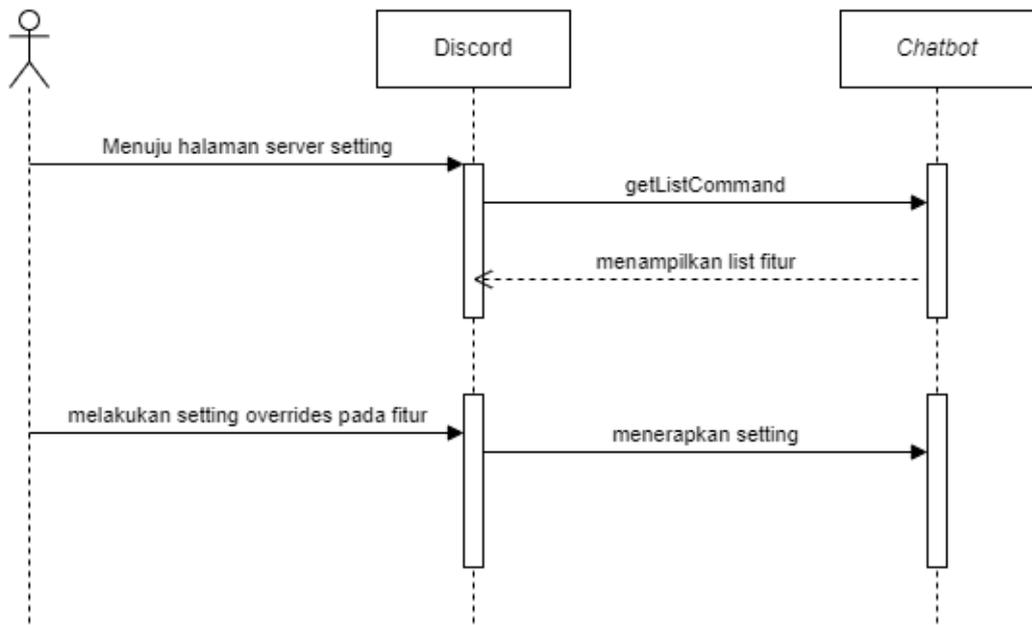
Gambar 4.7 menggambarkan pergerakan data pada sistem dari halaman *chat* server discord sampai kepada model *chatbot*. *Chatbot* akan melakukan

pengecekan apakah *chat* tersebut mengandung *bad word* jika iya *chatbot* akan mengklasifikasikan *chat* tersebut sebagai *bad word* dan memberi peringatan kepada user yang mengirimkannya.



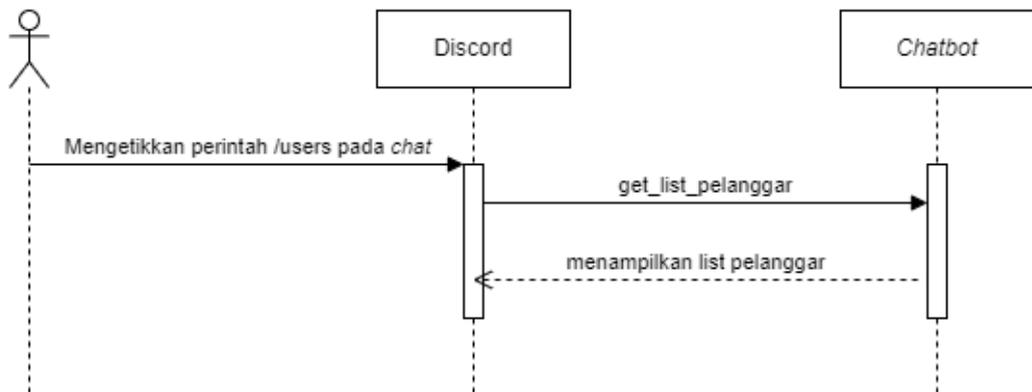
Gambar 4. 8 Sequence Diagram Kelola Data Pelanggar

Gambar 4.8 merupakan *sequence diagram* yang menampilkan proses manajemen pelanggar, proses pertama yang dilakukan adalah melakukan *mute* selama 1 hari, jika masih mengirimkan *bad words* maka akan di *kick* dari server, dan jika masuk ke dalam server lalu mengirimkan *bad words* lagi, maka akan di *ban* dari server.



Gambar 4. 9 Sequence Diagram List Fitur

Sequence Diagram pada gambar 4.9 menunjukkan proses untuk menampilkan list fitur *chatbot* pada server discord. Pengguna diharuskan menuju ke halaman server setting discord terlebih dahulu untuk mendapatkan list fitur pada *chatbot*.



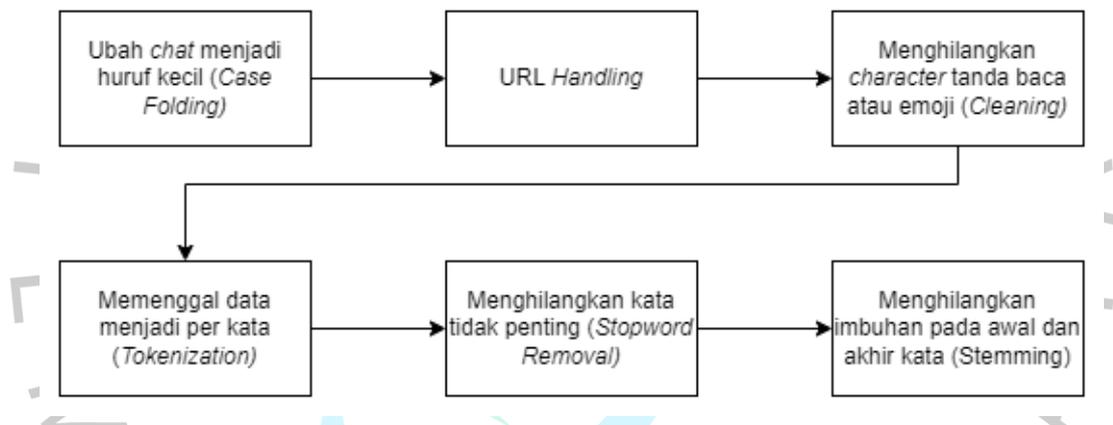
Gambar 4. 10 Sequence Diagram List Pelanggar

Sequence Diagram pada gambar 4.10 menunjukkan proses untuk menampilkan list pelanggar. *User* harus mengirimkan perintah `/users` ke dalam *chat*, lalu bot akan memberikan *response* list pelanggar yang mengirimkan *bad words* di server tersebut.

#### 4.3.4. Pembuatan Model

##### 4.3.4.1 Data Preprocessing

Sistem pengklasifikasi terdiri dari dua subsistem, yaitu Subsistem Utama dan Subsistem Bot. Subsistem Utama bertugas untuk melatih dan menguji data, sementara Subsistem Bot adalah bot Discord yang bertujuan untuk mengklasifikasikan pesan yang mengandung kata-kata kasar dalam grup obrolan serta menyediakan fitur tambahan untuk melatih model berdasarkan karakteristik masing-masing grup. Model dilatih dengan data pesan, untuk dapat melatih model dengan baik, pesan yang masuk harus melewati pra – pemrosesan seperti Gambar 4.11.



Gambar 4. 11 Data Preprocessing

##### 4.3.4.2 Simulasi Naïve Bayes

Pada simulasi perhitungan algoritma Naïve Bayes secara manual pada klasifikasi *bad word*, hal pertama yang harus dilakukan adalah menentukan dataset serta *bad words* yang akan kita klasifikasikan. Untuk itu akan digunakan dataset sederhana dengan beberapa kalimat dan klasifikasi *bad words*.

Tabel 4.5 Contoh Data Bad Words

No.	Bad Words
1.	Kurang ajar
2.	Bodoh

Tabel 4. 6 Contoh Pengklasifikasian Data

No	Data Kalimat	Class
1.	Dia sangat pintar	Bukan <i>bad word</i>
2.	Dia kurang ajar	<i>Bad word</i>
3.	Kamu bodoh	<i>Bad word</i>
4.	Dia hebat	Bukan <i>bad word</i>
5.	Kamu sangat bodoh	<i>Bad word</i>

Pada tahap ini akan diklasifikasikan kalimat “Dia Kurang Pintar”

Langkah – langkahnya adalah sebagai berikut :

1. Hitung probabilitas prior ( $P(\text{BadWord})$ ) dan ( $P(\neg\text{BadWord})$ )
  - a)  $P(\text{BadWord}) = \text{Jumlah kalimat bad word} / \text{Jumlah total kalimat} = 3/5$
  - b)  $P(\neg\text{BadWord}) = \text{Jumlah kalimat bukan bad word} / \text{Jumlah total kalimat} = 2/5$
2. Hitung probabilitas likelihood ( $P(X/\text{BadWord})$  dan  $P(X/\neg\text{BadWord})$ )
  - a)  $P(X/\text{BadWord}) = P(\text{"Dia"}/\text{BadWord}) * P(\text{"kurang"}/\text{BadWord}) * P(\text{"pintar"}/\text{BadWord})$
  - b)  $P(X/\neg\text{BadWord}) = P(\text{"Dia"}/\neg\text{BadWord}) * P(\text{"kurang"}/\neg\text{BadWord}) * P(\text{"pintar"}/\neg\text{BadWord})$

Hitung Frekuensi kata dalam masing – masing kelas

Tabel 4. 7 Kata dalam Class

Kata	Bad Word	¬ Bad Word
Dia	1	2
Kurang	1	0
pintar	0	1
ajar	1	0
Kamu	2	0
Bodoh	2	0
Hebat	0	1
Sangat	1	1
Total	8	5

Hitung probabilitas kata dengan Laplace Smoothing ( $\alpha = 1$ )

$$P("Dia"/BadWord) = (1+1)/(8+8) = 2/16$$

$$P("Dia"/¬BadWord) = (2+1)/(5+8) = 3/13$$

$$P("kurang"/BadWord) = (1+1)/(8+8) = 2/16$$

$$P("kurang"/¬BadWord) = (0+1)/(5+8) = 1/13$$

$$P("pintar"/BadWord) = (0+1)/(8+8) = 1/16$$

$$P("pintar"/¬BadWord) = (1+1)/(5+8) = 2/13$$

$$P(X/BadWord) = (2/16 * 2/16 * 1/16) = 0.00024414062$$

$$P(X/¬BadWord) = (3/13 * 1/13 * 2/13) = 0.00007565910$$

### 3. Hitung probabilitas posterior

a) Menghitung P(X):

$$\begin{aligned} P(X) &= (0.00024414062 * 3/5) + (0.00007565910 * 2/5) \\ &= 0.00015837788 \end{aligned}$$

b) Probabilitas posterior *Bad Word* ( $P(\text{BadWord}|X)$ )

$$\begin{aligned} P(\text{BadWord}|X) &= (0.00024414062 * 3/5) / 0.00015837788 \\ &= 0.462621 \end{aligned}$$

c) Probabilitas posterior  $\neg$ *BadWord* ( $P(\neg\text{BadWord}|X)$ )

$$\begin{aligned} P(\neg\text{BadWord}|X) &= (0.00007565910 * 2/5) / 0.00015837788 \\ &= 0.537379 \end{aligned}$$

Berdasarkan hasil perhitungan diatas menunjukkan bahwa data “Dia Kurang Pintar” termasuk dalam kelas bukan *bad word* ( $\neg\text{BadWord}$ ) karena probabilitas posterior lebih tinggi untuk kelas “ $\neg\text{BadWord}$ ” (0.537379) daripada untuk kelas “*BadWord*” (0.462621).

Contoh lainnya kita akan mengklasifikasikan kalimat “kurang ajar”

1. Hitung probabilitas prior ( $P(\text{BadWord})$ ) dan ( $P(\neg\text{BadWord})$ )

a)  $P(\text{BadWord}) = \text{Jumlah kalimat bad word} / \text{Jumlah total kalimat} = 3/5$

b)  $P(\neg\text{BadWord}) = \text{Jumlah kalimat bukan bad word} / \text{Jumlah total kalimat} = 2/5$

2. Hitung probabilitas likelihood ( $P(X|\text{BadWord})$  dan  $P(X|\neg\text{BadWord})$ )

a)  $P(X|\text{BadWord}) = P(\text{"kurang"}|\text{BadWord}) * P(\text{"ajar"}|\text{BadWord})$

b)  $P(X|\neg\text{BadWord}) = P(\text{"kurang"}|\neg\text{BadWord}) * P(\text{"ajar"}|\neg\text{BadWord})$

Hitung Frekuensi kata dalam masing – masing kelas

Tabel 4. 7 Kata dalam Class

Kata	Bad Word	$\neg$ Bad Word
Dia	1	2
Kurang	1	0
pintar	0	1
ajar	1	0
Kamu	2	0
Bodoh	2	0
Hebat	0	1
Sangat	1	1
Total	8	5

Hitung probabilitas kata dengan Laplace Smoothing ( $\alpha = 1$ )

$$P(\text{"kurang"}|\text{BadWord}) = (1+1) / (8+8) = 2/16$$

$$P(\text{"kurang"}|\neg\text{BadWord}) = (0+1) / (5+8) = 1/13$$

$$P(\text{"ajar"}|\text{BadWord}) = (1+1) / (8+8) = 2/16$$

$$P(\text{"ajar"}|\neg\text{BadWord}) = (0+1) / (5+8) = 1/13$$

$$P(X|\text{BadWord}) = (2/16 * 2/16) = 0.015625$$

$$P(X|\neg\text{BadWord}) = (1/13 * 1/13) = 0.005917$$

### 3. Hitung probabilitas posterior

d) Menghitung P(X):

$$\begin{aligned} P(X) &= (0.015625 * 3/5) + (0.005917 * 2/5) \\ &= 0.024168 \end{aligned}$$

e) Probabilitas posterior *Bad Word* ( $P(\text{BadWord}|X)$ )

$$\begin{aligned} P(\text{BadWord}|X) &= (0.015625 * 3/5) / 0.024168 \\ &= 0.387909 \end{aligned}$$

f) Probabilitas posterior  $\neg$ *Bad Word* ( $P(\neg\text{BadWord}|X)$ )

$$\begin{aligned} P(\neg\text{BadWord}|X) &= (0.005917 * 2/5) / 0.024168 \\ &= 0.024482 \end{aligned}$$

Berdasarkan hasil perhitungan diatas menunjukkan bahwa kata “kurang ajar” termasuk dalam kelas *bad word* (*BadWord*) karena probabilitas posterior lebih tinggi untuk kelas "*BadWord*" (0.387909) daripada untuk kelas " $\neg$ *BadWord*" (0.024482).

#### **4.3.4.3 Pelatihan Model**

Menggunakan Algoritma Naïve Bayes untuk melatih model klasifikasi menggunakan set data latih. Model akan belajar mengenali pola dalam teks yang mengindikasikan keberadaan kata-kata kasar.

#### **4.3.4.4 Implementasi Model pada Chat Bot Discord**

Mengintegrasikan model yang telah dilatih pada chat bot Discord. Chat bot akan menggunakan model untuk mengidentifikasi dan memfilter pesan yang mengandung kata-kata kasar secara real-time pada aplikasi Discord.

#### **4.3.5. Perancangan Pengujian**

Untuk memastikan sistem telah berfungsi sesuai yang diharapkan, diperlukanlah sebuah pengujian, seperti yang dijelaskan pada Bab 3, akan dilakukan pengujian dengan dua metode, yaitu pengujian kotak hitam (*black box testing*) dan pengujian kotak putih (*white box testing*).

##### **4.3.5.1 Pengujian Kotak Hitam (*Black Box Testing*)**

Pengujian *black box testing* berfokus pada fungsionalitas sistem dengan membuat skenario – skenario yang mungkin akan terjadi pada saat sistem digunakan. Skenario dibuat dengan format *input* yang diberikan dan *output* yang dikeluarkan apakah sudah sesuai dengan ekspektasi baik dalam skenario positif, ataupun skenario negatif.

Tabel 4. 8 Scenario Black Box Testing

No	Fitur	Skenario	Hasil yang Diharapkan
1	Filter <i>bad word</i>	User menginput <i>chat</i> teks dengan <i>bad word</i> di dalamnya User menginput <i>chat</i> teks tanpa <i>bad word</i> di dalamnya	<i>Chatbot</i> memberikan peringatan kepada <i>user</i> <i>Chatbot</i> tidak memberikan response apapun
2	Manajemen Pelanggar	User mengirimkan <i>bad words</i> selama 3 kali User yang sudah di <i>mute</i> mengirim <i>bad words</i> lagi User yang di <i>kick</i> dari server, join kembali ke dalam server, dan mengirimkan <i>bad words</i>	User ter- <i>mute</i> tidak dapat mengirim pesan dan tidak dapat <i>me-react</i> pesan orang lain selama 1 hari User ter- <i>kick</i> dari server User ter- <i>ban</i> dari server, dan tidak akan bisa join kembali ke dalam server
3	List Fitur	User mengakses halaman server setting di Discord	Menampilkan list fitur pada <i>chatbot</i>
4	List Pelanggar	User mengetikkan perintah <i>/users</i>	Menampilkan list pelanggar yang mengirimkan <i>bad words</i>

#### 4.3.5.2 Pengujian Kotak Putih (*White Box Testing*)

Pengujian *white box testing* berfokus pada algoritma dan kode program yang akan dijalankan. Kode program nantinya dipecah berdasarkan fungsinya dan dilakukan pengetesan apakah kode tersebut sudah berjalan sesuai fungsinya atau belum.

Tabel 4. 9 Scenario White Box Testing

No	Skenario Pengujian	Hasil yang diharapkan
1	Cleansing Data	Data yang masuk ke dalam bot sudah bersih dan dapat dibaca oleh bot
2	Integrasi model dengan Bot Discord	Model dapat dimuat dalam discord bot
3	Filter <i>bad word</i>	Bot dapat memfilter <i>chat bad word</i>
4	List User yang mengirimkan <i>bad word</i>	Bot dapat memberikan list user pelanggar pengirim <i>bad word</i>
5	Update data dan model	Bot terupdate dengan data baru dari perintah yang dimasukkan Admin/ <i>User</i>