

Pendeteksi Sederhana Ungkapan Bullying menggunakan Algoritma Regular Expression

Pendeteksian ungkapan bullying pada aplikasi messaging LINE menggunakan LINE chatbot

Ardysatrio Fakhri Haroen - 13517062

Institut Teknologi Bandung
Sekolah Teknik Elektro dan Informatika
Bandung, Indonesia
ardyharoen@gmail.com

Abstrak—Salah satu isu sosial yang paling marak sekarang adalah *bullying*. Termasuk diantaranya *cyber bullying*, yakni *bullying* yang dilakukan menggunakan teknologi dan media sosial. Isu ini makin marak karena teknologi makin berkembang seiring waktu. Pada makalah ini, penulis mengusulkan sebuah metode pendeteksian ungkapan *bullying* pada media *messenger* LINE yang dideteksi menggunakan algoritma *regex* yang diimplementasikan pada sebuah LINE *chatbot* yang dapat menerima pesan dari pengguna.

Keywords—*bullying, detection, string matching, chatbot, regex*

I. PENDAHULUAN

Salah satu dampak dari perkembangan teknologi adalah munculnya berbagai media sosial online. Seiring waktu, jumlahnya makin bertambah banyak dan jenisnya makin bervariasi. Selagi masing-masing dari mereka berkompetisi untuk menciptakan fitur-fitur yang paling baru dan menjangkau pasar-pasar yang belum tersentuh. Hal ini tentu menimbulkan perkembangan pesat jumlah orang yang menggunakan media sosial tersebut, didukung pula dengan makin mudahnya akses kepada gawai dan internet seiring waktu. Salah satu konsumen utama dari perkembangan teknologi tersebut adalah para pemuda. Hal ini dikarenakan mereka adalah generasi yang paling aktif mencari tahu tentang hal-hal baru dan generasi yang memiliki pikiran yang cukup dinamis untuk menyesuaikan diri terhadap cepatnya perubahan.

Sebagaimana segala hal, perkembangan yang pesat tersebut memiliki sisi problematis pula. Salah satunya, terbentuknya berbagai lingkungan virtual di media sosial yang tidak terawasi dengan baik, yang perbuatan-perbuatan

di lingkungan tersebut tidak terikat secara kuat dengan hukum, sehingga memungkinkan sebagian orang, termasuk para pemuda, untuk bertindak semena-mena disana. Telah banyak kasus amoral yang terjadi; penggunaan media sosial untuk penyebaran pornografi, diskriminasi besar-besaran terhadap suatu golongan, dsb. Salah satu tindak amoral yang marak terutama di kalangan para pemuda adalah *cyber bullying*. Istilah tersebut merujuk pada tindakan penindasan/*bullying* yang dilakukan pada lingkup media sosial. Aksi ini telah memakan banyak sekali korban, dan dampak yang ditimbulkan pada korban bisa sangat besar, terutama dari sisi mentalitas. Depresi, trauma, gangguan kecemasan, gangguan mental lain, dan bahkan aksi bunuh diri sangat mungkin menjadi akhir bagi seorang korban *cyber bullying*.

Belum ada penelitian yang sudah terbuka di Indonesia mengenai aksi *cyber bullying*, sehingga statistik yang terpercaya masih sulit untuk didapatkan. Akan tetapi, sudah ada penelitian yang dilakukan di Amerika Serikat mengenai tindakan ini:

1. Menurut studi yang dilakukan *American Pew Research Centre*, 9 dari 10 remaja telah menyaksikan kekejian atau tindak *bullying* di jaringan media sosial[1]
2. Studi yang dilakukan lembaga non profit *DoSomething.org* mengungkapkan beberapa statistik mengenai tindak *cyber bullying*[1]:
 - a. Sekitar 43% anak telah menjadi korban *bullying* di dunia maya. 1 dari 4 anak telah mengalaminya lebih dari sekali.
 - b. hanya 1 dari 10 korban berniat untuk memberitahukan keadaannya kepada

orangtua atau orang dewasa yang terpercaya

- c. 75% siswa mengakui pernah mengunjungi suatu website untuk mengecek sesamanya.

Statistik diatas tentunya sangat mengkhawatirkan, dan tidak ada jaminan bahwa Indonesia tidak berada dalam resiko yang sama, karena makin banyak remaja Indonesia yang mendapatkan akses kepada teknologi.

Sebelumnya, telah terdapat makalah yang membahas topik serupa, baik itu tindak *cyber bullying* dengan bahasa Indonesia maupun bahasa Inggris, yang dapat dilihat pada tautan-tautan berikut:

1. https://www.researchgate.net/publication/277568369_Online_Social_Network_Bullying_Detection_Using_Intelligence_Techniques
2. http://simki.unpkediri.ac.id/mahasiswa/file_artikel/2018/14.1.03.03.0101.pdf

Makalah diatas menggunakan data yang didapat dari media sosial *Twitter*, sehingga data teks yang didapatkan dalam bentuk *posting*, sedangkan penulis menggunakan data yang didapat dari aplikasi pesan LINE, sehingga data yang didapat berupa ungkapan percakapan yang diharapkan dapat memberikan data yang lebih alami. Selain itu, makalah diatas menggunakan algoritma-algoritma pembelajaran mesin untuk mengklasifikasi apakah sebuah teks mengandung ungkapan *bullying* atau tidak, sedangkan pada makalah ini hanya digunakan pemrosesan *string* sederhana untuk menyaring kata-kata yang mungkin mengandung ungkapan *bullying*.

Pada makalah ini, penulis akan membahas sebuah *chatbot* pada aplikasi LINE yang akan menerima berbagai contoh masukan percakapan, yang kemudian akan diproses, seberapa mungkin suatu ungkapan termasuk ungkapan *bullying*. Penulis memilih aplikasi LINE karena inilah yang banyak dipakai oleh pemuda dan karena *Messaging API*-nya yang mudah diimplementasikan

II. DASAR TEORI

A. API

API merupakan singkatan dari *Application Programming Interface*, yang merupakan kumpulan *routine* dan *protocol* yang digunakan untuk membangun aplikasi perangkat lunak. Secara garis besar, API mendefinisikan aturan-aturan mengenai interaksi antar perangkat lunak satu dengan yang lainnya.[2]

Salah satu contoh API adalah yang dimiliki oleh sistem operasi, yang membolehkan kita untuk

meng-copy paste informasi berupa teks atau gambar dari satu aplikasi ke aplikasi lainnya.

Awalnya, API hanya didesain sebagai metode interaksi antara satu program dengan program lain, tidak fleksibel. Namun, seiring perkembangan zaman, program dan aplikasi berada di semakin banyak tempat. Sehingga tidak cukup apabila hanya membuat API antara program A dan B secara spesifik, sehingga dibuatlah standar pembuatan API bernama RESTful API yang membuat berbagai API universal sehingga dapat dipakai secara lebih fleksibel.[3]

B. Chatbot

Chatbot adalah program komputer yang didesain untuk menyimulasikan percakapan dengan satu atau lebih orang. *Chatbot* ada yang dapat menerima dan menyampaikan pesan dalam bentuk teks, ada yang dalam bentuk suara, ada pula yang keduanya. *Chatbot* biasa digunakan dalam otomatisasi pelayanan manusia, sehingga perusahaan-perusahaan yang membutuhkan *Customer Service* yang selalu berjalan, biasanya sudah mengaplikasikan *Chatbot* sebagai pelayan operatornya.

C. Line Messaging API

LINE Messaging API adalah API yang dibuat oleh perusahaan *messaging app* bernama LINE, yang membolehkan kita mendapatkan berbagai informasi pada suatu *instance* pengiriman pesan oleh pengguna.

Diantara informasi yang bisa didapatkan adalah:

1. Pesan yang dikirimkan pengguna, baik itu berupa gambar, teks, suara, atau bahkan sticker *emoticon*.
2. ID pengguna, sehingga kita bisa mengetahui siapa yang mengirimkan pesan
3. waktu disaat pengguna mengirimkan pesan

API ini biasa digunakan ketika seseorang ingin membuat sebuah *chatbot* di LINE, karena seseorang membutuhkan informasi *instance* pesan untuk mengetahui pesan apa yang paling tepat untuk membalas pesan user.

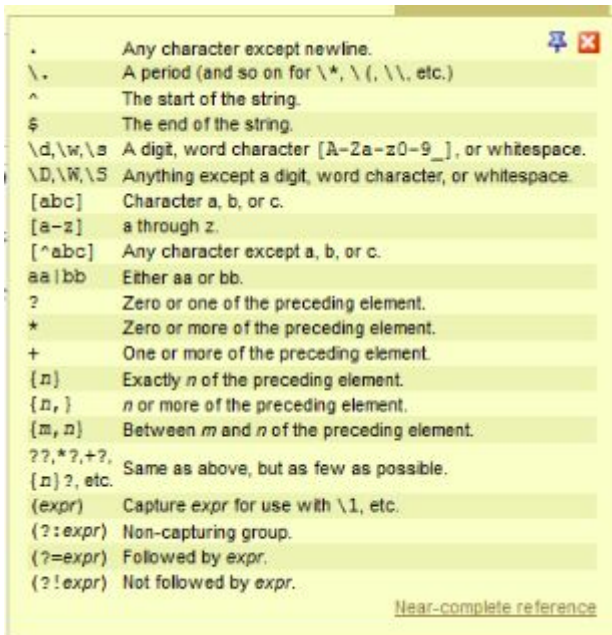
Selain mendapatkan informasi, LINE API juga membolehkan sebuah bot untuk:

1. Mengirimkan sebuah pesan, baik dalam bentuk gambar, sticker *emoticon*, atau teks.
2. Bergabung atau Keluar dari Grup percakapan.

D. Algoritma Regular Expression

Algoritma *Regular Expression*, atau yang biasa disebut dengan *regex*, adalah salah satu algoritma yang digunakan dalam pencocokan string. Algoritma ini bekerja dengan mengaplikasikan sebuah pola pencarian (*search pattern*) kepada suatu string, kemudian melihat apakah ada string atau substring yang memenuhi pola pencarian tersebut.

Pola pencarian yang digunakan *regex* melibatkan penggunaan beberapa karakter khusus yang mempunyai makna-makna tertentu. Dibawah ini daftar karakter-karakter khusus yang digunakan pada *regex*.



.	Any character except newline.
\.	A period (and so on for *, \{, \}, etc.)
^	The start of the string.
\$	The end of the string.
\d,\w,\s	A digit, word character [A-Za-z0-9_], or whitespace.
\D,\W,\S	Anything except a digit, word character, or whitespace.
[abc]	Character a, b, or c.
[a-z]	a through z.
[^abc]	Any character except a, b, or c.
aa bb	Either aa or bb.
?	Zero or one of the preceding element.
*	Zero or more of the preceding element.
+	One or more of the preceding element.
{n}	Exactly n of the preceding element.
{n,}	n or more of the preceding element.
{m,n}	Between m and n of the preceding element.
??,*?,+?, {n}?, etc.	Same as above, but as few as possible.
(expr)	Capture expr for use with \1, etc.
(?:expr)	Non-capturing group.
(?=expr)	Followed by expr.
(?!expr)	Not followed by expr.

[Near-complete reference](#)

Gambar 2.1

Karakter-karakter khusus yang mempunyai makna tertentu pada *regex*.

Sumber: Slide Kuliah IF2211 mengenai *regex*. Diakses 26 April 2019, pukul 09.05

Berikut ini beberapa contoh search pattern *regex* beserta maknanya:

1. "hi|hello" akan cocok dengan string yang mempunyai substring "hello" atau "hi"
2. "a(bc)*" akan cocok dengan string yang mempunyai substring "a", diikuti dengan 0 atau lebih pengulangan string "bc".
3. "ab{3,5}" akan cocok dengan string yang mempunyai substring "a", diikuti persis dengan string "b" yang diulang sebanyak 3 sampai 5 kali (inklusif).

E. Bullying

Bullying adalah aksi yang agresif, dan tidak dikehendaki pada anak-anak atau remaja berusia sekolah, yang melibatkan ketidakseimbangan kekuatan baik secara fisik maupun nonfisik. Perbuatan ini terulang, atau sangat berpotensi untuk terulang seiring waktu. Kedua belah pihak, baik yang melakukan tindakan *bullying* maupun korbannya terkena resiko besar atas masalah jangka panjang.[4]

Ada batasan agar tindakan-tindakan tertentu dapat disebut *bullying*, yakni tindakan tersebut harus agresif dan melibatkan:

1. Ketidakseimbangan kekuatan
ketidakseimbangan ini terjadi antara pelaku dan korban. Pelaku mempunyai kekuatan yang jauh lebih besar, bisa dalam bentuk kekuatan fisik, informasi (akses terhadap informasi rahasia korban), atau popularitas, yang menyebabkannya dapat menindas korban tanpa menghiraukan penilaian sosial dari teman sebayanya.
2. Pengulangan
Tindakan *bullying* terjadi lebih dari sekali, atau berpeluang untuk terjadi lebih dari sekali.

Bentuk bentuk tindakan *bullying* meliputi:

1. *bullying* verbal, yakni menulis atau mengatakan hal-hal keji, dapat berupa:
 - a. Mengganggu
 - b. Mengancam dengan kekerasan
 - c. Membuat panggilan buruk
2. *bullying* sosial, yakni merusak reputasi atau hubungan sosial seseorang, dapat berupa:
 - a. Mengucilkan seseorang secara sengaja
 - b. Menyebarkan rumor-rumor tentang seseorang
 - c. Mempermalukan seseorang di depan umum
3. *bullying* fisik, yakni menyakiti seseorang atau barang kepemilikannya, dapat berupa:
 - a. Menendang/memukul
 - b. Mengambil atau merusak barang seseorang
 - c. Meludahi

Tindakan *bullying* berdampak sangat besar terhadap kehidupan korban dan pelakunya.

1. Dampak pada korban: meningkatkan kemungkinan terjadinya depresi, gangguan kecemasan, dan gangguan mental lainnya, selain itu juga dapat menimbulkan problematika kesehatan dan menurunkan prestasi.
2. Dampak pada pelaku: pelaku *bullying* akan sangat mungkin terlibat penyalahgunaan alkohol dan

narkotika dewasanya, juga meningkatkan kemungkinan terlibat tindak kriminal.

Salah satu jenis bullying berdasarkan medianya adalah *cyber bullying*, yakni tindakan bullying baik berupa verbal atau sosial yang dilakukan menggunakan media elektronik, atau dilakukan dalam dunia maya. Bisa menggunakan gawai, media sosial, forum, dsb. Jenis bullying inilah yang akan diangkat pada makalah ini.

III. IMPLEMENTASI

Pada implementasi ini, penulis hanya akan membahas ungkapan-ungkapan *bullying* secara verbal saja, karena yang dilakukan hanya *text analysis* dari pesan yang dikirim lewat media LINE.

Selain itu, karena berbagai keterbatasan, penulis juga berasumsi sebuah ungkapan makian bisa dianggap sebagai ungkapan *bullying*, walaupun ungkapan makian tidak selalu diniatkan untuk *bullying*. Oleh karena itu, yang akan dicari oleh algoritma regex adalah berupa ungkapan makian.

Berikut ini penjabaran implementasinya:

A. Search Pattern yang digunakan pada Regex

Untuk membuat search pattern, penulis menggunakan beberapa asumsi yang memodelkan sebuah ungkapan yang kiranya bisa dianggap sebagai ungkapan makian, diantaranya:

1. Pengguna yang berniat mengutarakan ungkapan *bullying* akan setidaknya memuat dua buah kata dalam ungkapannya, yaitu satu buah kata kasar dan satu buah kata objek (Sebagai objek makiannya).
2. Kedua buah kata yang ada pada point no.1 dapat dibalik, yakni bisa saja kata kasar terlebih dahulu baru kata objek, atau bisa jadi kata objek terlebih dahulu baru kata kasar.
3. Karena kedua kata tersebut hanyalah batas minimal ungkapan, maksimalnya ungkapan tersebut dapat disisipi kata kata lain diantara kedua kata tersebut (kata objek dan kata kasar), baik itu disisipkan di depan, di tengah-tengah, maupun di belakang

Berdasarkan asumsi-asumsi yang dibuat diatas, dapat ditarik kesimpulan bahwa search pattern yang harus dibuat adalah sebagai berikut:

pattern 1:

```
"(.*( "+forb+" )).**(
)* (" +name+" ).*) | (.*( "+name+" )
).**( )* (" +forb+" ).*)" "
```

variabel name pada pattern diatas adalah objek yang berupa nama orang, sedangkan variabel forb adalah merupakan kata-kata kotor yang digunakan.

pattern 2:

```
"(.*( "+forb+" ) ).*(
)* (" +sec+" ).*) | (.*( "+sec+" ) ).*(
)* (" +forb+" ).*)" "
```

variabel sec pada pattern diatas adalah objek yang berupa kata ganti orang kedua, sedangkan variabel forb adalah merupakan kata-kata kotor yang digunakan.

Penjelasan pola regex:

1. Di tengah-tengah string, terdapat karakter “[|]” yang bermakna “atau”. Karakter ini ditambahkan agar asumsi nomor 2 dapat tercakup. Yakni bisa kata kotor diungkapkan dahulu diikuti kata objek, atau sebaliknya
2. Sebelum variabel (forb, sec, name) dan pada akhir pattern, terdapat string “.*”, yang artinya ungkapan dapat didahului dan diikuti ungkapan kata-kata lain. Hal ini untuk memenuhi asumsi nomor 3.
3. Diantara variabel (antara forb & sec dan forb & name) terdapat string “(+)” yang artinya setelah suatu ungkapan (entah kata kasar atau objek), akan diikuti minimal satu buah spasi. Kemudian diikuti dengan string “.*”, yang artinya antara kedua ungkapan (kata kasar dan objek) dapat disisipkan berbagai kata. Kemudian diikuti dengan “)*”, yang artinya sebelum ungkapan kedua (entah kata kotor atau objek) bisa ada spasi. Hal ini untuk memenuhi asumsi ketiga.

Karena ungkapan yang dicari hanya ungkapan makian minimal (hanya terdapat satu buah kata kasar), sedangkan pada suatu ungkapan bisa jadi terdapat dua atau lebih kata kasar (yang menunjukkan makin besarnya makian), penulis membuat juga sistem poin bagi tiap ungkapan yang diberikan pengguna. Akan dijelaskan pada poin *app.py* dibawah.

tiap kata kasar yang ditemukan, points bertambah 3.

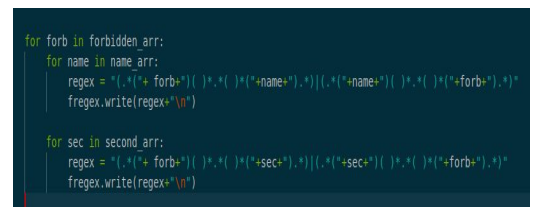
Bila tidak ditemukan ungkapan makian pada ungkapan yang diberikan pengguna, maka *chatbot* tidak akan mencetak apa-apa ke *bubblenya*.

Untuk pencocokan dengan regex, penulis menggunakan library *re* milik python. Sehingga penulis harus meng-*importnya* terlebih dahulu.

Untuk menemukan substring yang ingin dicari menggunakan regex, penulis menggunakan method yang terdapat pada *library re* yang bernama *search*. Yang argumennya : *re.search(<regex pattern>, <string text>, <flags>)*.

Flags dapat dikatakan sebagai opsi yang dapat diatur dalam pencarian. Penulis menggunakan opsi *re.IGNORECASE* untuk mengabaikan huruf besar dan huruf kecil dalam *regex matching*.

2. makeregex.py



```
for forb in forbidden_arr:
    for name in name_arr:
        regex = "(.*{+ forb+} (.*) (.*){+name+}.*)|(.*{+name+} (.*) (.*){+forb+}.*)"
        regex.write(regex+"\n")

for sec in second_arr:
    regex = "(.*{+ forb+} (.*) (.*){+sec+}.*)|(.*{+sec+} (.*) (.*){+forb+}.*)"
    regex.write(regex+"\n")
```

Gambar 3.2

Loop utama untuk memproduksi regex yang akan ditulis pada file *regexes.txt*

Merupakan source code python yang digunakan untuk membuat regex dari database string yang sudah didapat.

Pertama, program ini akan memasukkan semua kata-kata yang ada di *database* kedalam 3 buah array. Kemudian akan diiterasi pada tiap array untuk menghasilkan semua kombinasi kata kotor dengan objek yang memungkinkan

Aturan pembuatan *search pattern regex* sudah dijelaskan pada poin sebelumnya.

3. second.txt

Merupakan file *.txt* yang berisi kata ganti orang kedua dan ketiga yang umum penulis dengar.

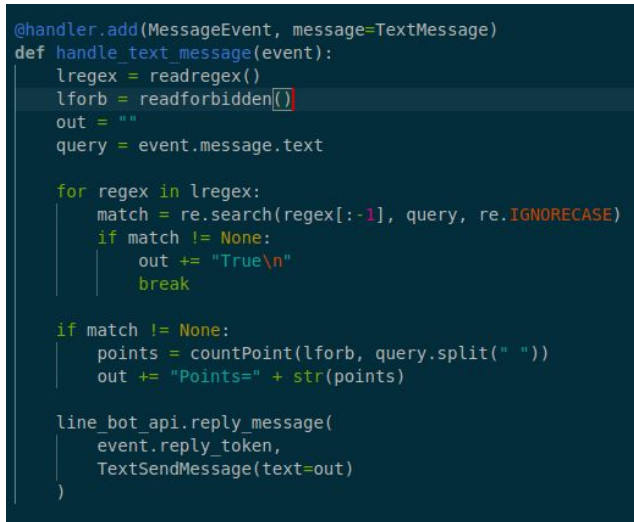
4. names.txt

Merupakan file *.txt* yang berisi 100 buah nama-nama orang Indonesia yang umum, yang penulis ambil dari website <http://faktamuharam.blogspot.com/2012/11/top-1-000-nama-populer-di-indonesia.html>

B. Arsitektur Aplikasi

files yang paling utama pada aplikasi ada beberapa, yakni:

1. app.py



```
@handler.add(MessageEvent, message=TextMessage)
def handle_text_message(event):
    lregex = readregex()
    lforb = readforbidden()
    out = ""
    query = event.message.text

    for regex in lregex:
        match = re.search(regex[:-1], query, re.IGNORECASE)
        if match != None:
            out += "True\n"
            break

    if match != None:
        points = countPoint(lforb, query.split(" "))
        out += "Points=" + str(points)

    line_bot_api.reply_message(
        event.reply_token,
        TextSendMessage(text=out)
    )
```

Gambar 3.1

Fungsi utama pada program *app.py*, *handle_text_message*, untuk menangani semua input yang masuk dari user

Merupakan *source code* python utama, yang mengandung main program, juga merupakan sebagai entry point dari webhook yang ditautkan pada LINEbot yang dibuat.

Program ini bertugas mendapatkan masukan yang diberikan pengguna, untuk kemudian diproses oleh chatbot.

Setelah tiap masukan dari pengguna, jika ungkapan yang diutarakan terdeteksi mengandung makian, chatbot mencetak dua hal ke *chat bubble* miliknya:

1. True, bila ditemukan ungkapan yang sebelumnya diutarakan pengguna mengandung makian
2. Points, yang memberitahukan seberapa parah makian yang diberikan pengguna. Nilai points ditentukan dengan menghitung jumlah kata-kata kasar yang terdapat pada ungkapan pengguna. Untuk

Untuk tiap nama yang penulis ambil, penulis juga mencantumkan di file .txt nya nama panggilan yang mungkin untuk nama tersebut. Yakni berupa dua string terakhir dari nama tersebut. Misalkan nama seseorang "Budi", maka penulis juga cantumkan "di" pada file names.txt ini.

5. forbiddens.txt

Merupakan file .txt yang berisi berbagai kata-kata kotor dalam bahasa Indonesia yang penulis dapat dari tautan berikut:

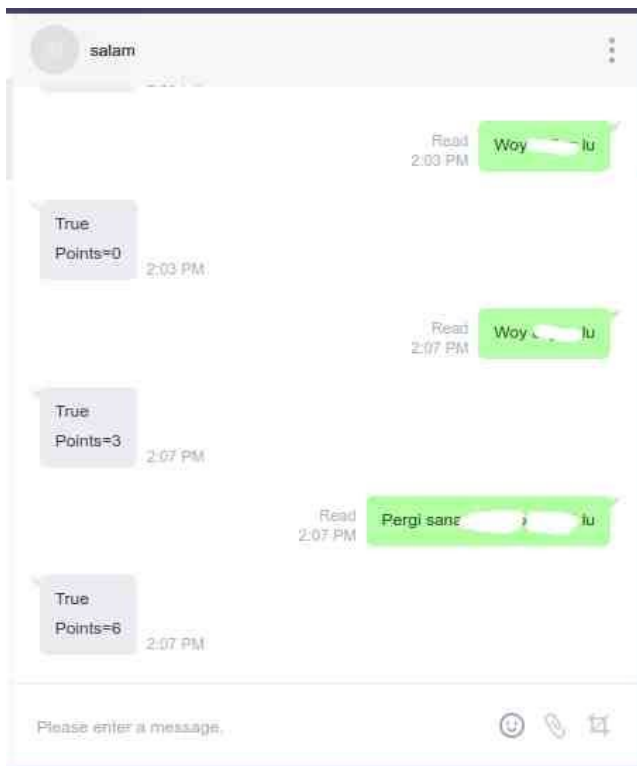
<https://www.kaskus.co.id/thread/54d98d18118b468a558b4567/daftar-kata-kata-kotor-di-indonesia-yang-sering-diucapkan/>

6. regexes.txt

Merupakan file .txt yang berisi kumpulan regex yang akan dicocokkan dengan input pengguna, untuk mengetahui apakah input pengguna mengandung ungkapan makian atau tidak.

IV. EKSPERIMEN

Berikut adalah eksperimen yang penulis lakukan pada chatbot. Demi menjaga norma, penulis menyensor kata-kata kotor yang sengaja dimasukkan.



KESIMPULAN

Bullying merupakan satu hal kritis yang sangat perlu ditangani, terlebih lagi cyberbullying yang makin lama makin marak seiring berkembangnya teknologi. Dengan algoritma regex kita dapat dimudahkan untuk mencari suatu informasi dari suatu string, sehingga berbagai ungkapan dapat kita cari dengan lebih mudah, termasuk didalamnya ungkapan *bullying*. Harapannya di masa depan teknologi dapat lebih menangani masalah-masalah sosial seperti ini.

REFERENSI

- [1] <https://theorganicagency.com/blog/life-death-consequences-cyber-bullying/> (Diakses tanggal 26 April 2019 pukul 06.51)
- [2] <https://www.webopedia.com/TERM/A/API.html> (Diakses tanggal 26 April 2019 pukul 08.07)
- [3] <https://www.mulesoft.com/resources/api/what-is-an-api> (Diakses tanggal 26 April 2019 pukul 08.30)
- [4] <https://www.stopbullying.gov/what-is-bullying/index.html> (Diakses tanggal 26 09.32)

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 29 April 2012

A handwritten signature in black ink on a light grey background. The signature is stylized and appears to read 'Ardysatrio Fakhri Haroen'.

Ardysatrio Fakhri Haroen
13517062

