PENAMBANGAN KICAUAN DI TWITTER: SENTIMEN, KATA POPULER DAN POHON TAGAR

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Teknik Informatika



Oleh:
Ian Febriyanto Adhijaya
10523272

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

2014

LEMBAR PENGESAHAN DOSEN PEMBIMBING

PENAMBANGAN KICAUAN DI TWITTER: SENTIMEN, KATA POPULER DAN POHON TAGAR

TUGAS AKHIR

Disusun Oleh:

Nama : Ian Febriyanto Adhijaya

No. Mahasiswa : 10523272

Yogyakarta, 13 Mei 2014

Pembimbing

Fathul Wahid, ST., M.Sc., Ph.D.

ii

LEMBAR PERNYATAAN KEASLIAN

HASIL TUGAS AKHIR

Saya yang bertanda tangan di bawah ini:

Nama : Ian Febriyanto Adhijaya

No. Mahasiswa : 10523272

Menyatakan seluruh komponen dan isi dalam laporan Tugas Akhir ini adalah hasil karya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, maka saya siap menanggung resiko dan konsekuensi apapun.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagimana mestinya.

Yogyakarta, 13 Mei 2014

Ian Febriyanto Adhijaya

LEMBAR PENGESAHAN PENGUJI

PENAMBANGAN KICAUAN DI TWITTER: SENTIMEN, KATA POPULER DAN POHON TAGAR

Disusun Oleh:

Nama : Ian Febriyanto Adhijaya

No. Mahasiswa : 10523272

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Teknik Informatika

Yogyakarta, 8 Juli 2014

Tim Penguji

Fathul Wahid, ST., M.Sc., Ph.D.
Ketua

M. Andri Setiawan, ST.,M.Sc.
Anggota 1

Beni Suranto, ST., M.SoftEng.
Anggota 2

Mengetahui

Ketua Jurusan Teknik Informatika

Universitas Islam Indonesia

Yudi Prayudi, S.Si., M.Kom.

iv

HALAMAN MOTTO

Nalar hanya akan membawa anda dari A menuju B, namun imajinasi mampu membawa anda dari A ke manapun.
(Albert Einstein)

Dua hal yang tak terbatas: alam semesta dan kebodohan manusia, dan saya tidak yakin tentang alam semesta.
(Albert Einstein)

Belajar bukan tentang mengejar nilai, di bangku sekolah ataupun kuliah, semua yang salah pada hari ini dapat diperbaiki besok, hal yang sangat berbeda apabila terjadi di luar sana.
(Penulis)

Hutan bukanlah sekumpulan pohon dan binatang di dalamnya. Namun hutan adalah sekumpulan gedung dengan orang di dalamnya, mereka liar bukan karena jahat, tapi karena mereka tidak peduli, mereka tidak dibuat untuk bercanda, apalagi merasakan cinta.
(Penulis)

KATA PENGANTAR

Assalamu'alaikum Warohmatullahi Wabarokatuh

Syukur alhamdulilah saya panjatkan kepada Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga Tugas Akhir ini yang berjudul "**Penambangan Kicauan di Twitter: Sentimen, Kata Populer dan Pohon Tagar**" telah selesai dengan baik. Berbagai rintangan dan masalah di luar zona nyaman, dengan usaha, kerja keras, bimbingan dan dukungan oleh banyak pihak, sehingga dapat teratasi dengan baik dalam penyusunan Tugas Akhir ini.

Segala hal ilmu yang saya tuntut di Universitas Islam Indonesia selama kurang lebih 3,5 tahun sangat bermanfaat dalam penyususunan Tugas Akhir ini, tak lepas dalam proses penyusunannya sangat memberikan banyak sekali pelajaran di luar bangku perkuliahan, walaupun masih banyak ilmu yang belum saya pahami sepenuhnya, namun tidak mematahkan semangat saya untuk terus belajar.

Banyak pihak yang telah memberikan inspirasi dan memantik api imajinasi dalam Tugas Akhir ini dan mereka tak patah arang memberikan semangat terus menerus dalam proses penyelesaiannya. Oleh karena itu penyelesaian Tugas Akhir ini merupakan wujud terima kasih yang sedalam-dalamnya kepada:

1. Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga penyelesaian Tugas Akhir ini dapat terselesaikan dengan baik.
2. Bapak Fathul Wahid, ST., M.Sc., Ph.D yang telah ikhlas membagi banyak ilmu dan membimbing saya dengan sabar dan baik.
3. Kedua orang tua saya, Bambang Riyadi dan Erning Triwinarsih, yang tak henti-hentinya mendoakan dan membakar semangat saya, seperti menyulut bensin dengan api.
4. Kakak saya, Erwin Adhi Sanjaya yang memberikan dorongan dalam menyelesaikan Tugas Akhir ini.
5. Teman-teman sekaligus sahabat-sahabat saya yang telah banyak membantu tanpa jemu, yang tidak bisa disebutkan satu per satu.

Semoga apa yang telah mereka berikan mendapatkan balasan yang lebih baik dari Allah SWT. Walaupun banyak sekali kekurangan dalam penulisan Tugas

Akhir ini, semoga pembaca tidak sungkan memberikan kritik yang kritis dan saran yang membangun, sehingga Tugas Akhir ini dapat menjadi lebih baik dan bermanfaat kedepannya, baik bagi penulis maupun pembaca.

Wassalamu'alaikum Warohmatullahi Wabarokatuh

Yogyakarta, Mei 2014

Ian Febriyanto Adhijaya

ABSTRAK

Twitter merupakan media sosial yang sangat populer saat ini, tercatat lebih 169 milyar posting kicauan sejak 2012 dan terus bertambah hingga sekarang. Indonesia sendiri merupakan negara tersebasar kelima pengguna Twitter, tercatat 19,5 juta orang pengguna pada tahun 2011 dan terus bertambah mengingat pesatnya pertumbuhan pengguna Internet di Indonesia dari usia 10-20 tahun. Kicauan dalam Twitter bisa merupakan wujud ekspresi atau pendapat seseorang mengenai suatu hal yang berisi penilaian positif, negatif ataupun netral. Oleh karena itu penambangan opini publik melalui Twitter dirasa efektif karena banyaknya data yang tersedia. Untuk mengetahui opini publik, maka perlu dilakukannya proses analisis sentimen yang berfungsi untuk melakukan klasifikasi kicauan tersebut ke dalam kategori positif, negatif atau netral. Pada dasarnya proses klasifikasi menggunakan metode string matching, di mana melakukan pencocokan kata dalam kicauan dengan daftar kata positif, negatif dan netral untuk proses klasifikasi. Selain hasil dari klasifikasi kicauan, keluaran dari aplikasi adalah berupa kata populer, yang merupakan kata yang sering muncul dalam kicauan. Selain itu, pohon tagar yang berisi tagar apa saja yang sering muncul dalam kicauan. Hasil dari pembuatan aplikasi untuk Tugas Akhir adalah untuk mengetahui topik atau isu yang berkembang dalam masyarakat umum di Indonesia melalui Twitter yang dicari berdasarkan tagar, dan dilakukan analisis sentimen untuk mengetahui penilaian positif, negatif, atau netral oleh masyrakat tentang topik tersebut.

Kata kunci: Analisis sentimen, Twitter, Pohon Tagar, Kata Populer

TAKARIR

Tweet	Kicauan atau pesan singkat 140 karakter.
Microblogging	Suatu bentuk blog yang yang memiliki pembaruan teks yang lebih sedikit.
Hashtag	Simbol '#' yang digunakan untuk menandai kata kunci atau topik dalam kicauan.
Array	Merupakan larik yang dapat menyimpan nilai dari sebuah data ke dalam indeks dari larik tersebut.
Use case diagram	Diagram yang menggambarkan interaksi pengguna dengan aplikasi.
User interface	Antarmuka aplikasi
Sequence diagram	Diagram yang menggambarkan interaksi antar obyek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu.
Class diagram	Diagram yang menggambarkan kelas-kelas dalam sebuah sistem dan hubungannya antara satu dengan yang lainnya.
Representational State Transfer	Beberapa gaya arsitekur berbasis jaringan dan dikombinasikan dengan batasan-batasan tambahan yang mendefinisikan antarmuka konektor.
Application Programming Interface	Merupakan bentuk pemrograman yang memungkinkan dua aplikasi berkomunikasi, baik berbeda *platform*, sistem operasi dan bahasa pemrograman.
Web service	Perangkat lunak sistem yang dirancang untuk interoperabilitas dari mesin ke mesin melalui jaringan.
Server	Sistem komputer yang menyediakan sebuah layanan pada suatu jaringan.

Request	Bentuk mengirimkan sebuah permintaan untuk mengakses sumber daya Twitter melalui API.
Default	Penyetelan tetap saat menjalankan sebuah program atau aplikasi.
Storage	Tempat penyimpanan data atau sumber daya.
Consumer key	Kunci unik yang digunakan untuk mengidentifikasi aplikasi ketika membuat permintaan pada API.
Consumer secret	Kunci unik yang digunakan untuk tanda keamanan saat melakukan permintaan ke Twitter.
Access token	Kunci unik yang digunakan oleh pengguna untuk mengakses sumber data dari akun Twitter pengguna sendiri.
Access token secret	Kunci unik yang digunakan untuk tanda keamanan saat mengakses sumber data dari akun Twitter pengguna sendiri.
Request access token	Melakukan permintaan kunci *access token* dan *access token secret.*
Login	Proses untuk masuk ke dalam aplikasi.
Link	Alamat yang digunakan untuk membuka suatu situs tertentu.
Retweet	Pengulangan sebuah kicauan.
Third party application	Aplikasi yang dikembangkan oleh pihak ketiga.
Short link	Alamat suatu *website* yang telah dipersingkat.
Posting	Menulis dan menerbitkan kicauan dalam Twitter.
Real time	Kondisi pengoperasian suatu sistem atau aplikasi yang dibatasi oleh rentang waktu tertentu.
Redo	Mengembalikan suatu keadaan seperti setelah dilakukan sebuah aksi tertentu.

Undo	Mengembalikan suatu keadaan seperti keadaan yang sebelumnya.
Node	Sebuah simpul yang terhubung dengan jalur atau garis.
String matching	Pencocokan kata
Parsing	Proses untuk mengurai atau mengubah struktur teks sintaksis.
Method	Sebuah blok pernyataan yang dapat digunakan berulang kali dalam sebuah program.
String	Kumpulan dari karakter, bilangan, spasi dan lain sebagainya yang berada dalam tanda petik.
Destroy session	Menghancurkan sesi pengguna yang telah terbentuk
Regular Expression	Sebuah metode untuk mencari suatu pola dalam sebuah string.

DAFTAR ISI

LEMBAR PENGESAHAN DOSEN PEMBIMBING ii
LEMBAR PERNYATAAN KEASLIAN iii
LEMBAR PENGESAHAN PENGUJI iv
HALAMAN MOTTO v
KATA PENGANTAR vi
ABSTRAK viii
TAKARIR ix
DAFTAR ISI xii
DAFTAR TABEL xiv
DAFTAR GAMBAR xv
DAFTAR RUMUS xviii
BAB I 1
1.2 Latar Belakang 1
1.3 Rumusan Masalah 2
1.4 Batasan Masalah 2
1.5 Tujuan dan Manfaat Penelitian 2
1.6 Langkah-langkah Penyelesaian 2
1.7 Sistematika Penulisan 3
BAB II 4
2.1 REST API Twitter 4
2.2 Pengambilan Data dari Twitter 6
2.3 Penambangan Opini Publik 7
2.4 Analisis Sentimen 8
BAB III 10
3.1 Pengumpulan Kicauan Sebagai Data 10
3.2 Analisa Pola Kicauan Sebagai Data 10
3.3 Pembersihan Kicauan 11
3.4 Proses Analisis 12
3.4.1 Daftar Kata 12

3.4.2 Proses Perhitungan 14
3.5 Pohon Tagar 16
3.6 Kata Populer 17
3.7 Analisa Kebutuhan 18
3.7.1 Kebutuhan Input 18
3.7.2 Kebutuhan Proses 18
3.7.3 Kebutuhan Output 20
3.8 Perancangan 21
3.8.1 Use Case Diagram 21
3.8.2 Sequence Diagram 22
3.8.3 Class Diagram 30
3.8.4 Perancangan *Array* 34
3.8.5 Perancangan *Interface* 38
BAB IV 41
4.1 Batasan Implementasi 41
4.2 Implementasi 41
4.2.1 Implementasi Program 41
4.2.2 Implementasi Antarmuka 55
BAB V 60
5.1 Kesimpulan 60
5.2 Keterbatasan dan Saran 60
DAFTAR PUSTAKA 63
LAMPIRAN 64

DAFTAR TABEL

Tabel 3.1 Contoh pembersihan kicauan 11
Tabel 3.2 Contoh daftar kata positif 12
Tabel 3.3 Contoh daftar kata negatif 13
Tabel 3.4 Contoh daftar kata netral 13
Tabel 3.5 Contoh daftar kata awalan 14
Tabel 3.6 Contoh daftar kata diabaikan 14

DAFTAR GAMBAR

Gambar 2.1 Web API 4
Gambar 2.2 Contoh JSON 5
Gambar 2.3 Contoh kicauan 7
Gambar 2.4 Illustrasi kicauan yang mengandung sentimen 9
Gambar 3.1 Contoh kicauan sebagai data 10
Gambar 3.2 Contoh hasil analisis sentimen untuk pengguna belum teraktivasi ... 16
Gambar 3.3 Contoh hasil analisis sentimen untuk pengguna teraktivasi 16
Gambar 3.4 Contoh pohon tagar 17
Gambar 3.5 Contoh grafik kata populer 17
Gambar 3.6 *Use case diagram* 21
Gambar 3.7 *Sequence diagram* untuk melakukan pencarian kicauan berdasarkan tagar untuk pengguna belum teraktivasi 22
Gambar 3.8 Sequence diagram untuk mendaftar ke sistem 23
Gambar 3.9 *Sequence diagram* untuk melakukan login 24
Gambar 3.10 *Sequence diagram* untuk melakukan pencarian kicauan berdasarkan tagar untuk pengguna teraktivasi 25
Gambar 3.11 *Sequence diagram* untuk melakukan hapus data kicauan 26
Gambar 3.12 *Sequence diagram* untuk menghapus, menambah dan mengubah daftar kata 26
Gambar 3.13 *Sequence diagram* untuk melakukan aktivasi pengguna 27
Gambar 3.14 *Sequence diagram* untuk melakukan deaktivasi pengguna 27
Gambar 3.15 *Sequence diagram* untuk melakukan ubah *password* pengguna teraktivasi 28
Gambar 3.16 *Sequence diagram* untuk melakukan hapus pengguna 29
Gambar 3.17 *Class diagram* untuk pewarisan kelas 30
Gambar 3.18 *Array* data pengguna terdaftar 34
Gambar 3.19 Contoh *array* daftar kata positif 35
Gambar 3.20 *Array* untuk proses *login* 35
Gambar 3.21 *Array* untuk proses analisis sentimen 36

Gambar 3.22 *Array* untuk *string matching* 37
Gambar 3.23 Halaman muka..... 38
Gambar 3.24 Hasil pencarian untuk pengguna tidak teraktivasi..... 38
Gambar 3.25 Halaman muka pengguna teraktivasi 39
Gambar 3.26 Halaman daftar kata 39
Gambar 3.27 Halaman kelola pengguna 40
Gambar 4.1 Deklarasi *consumer key* dan *consumer secret*..... 42
Gambar 4.2 *Script* instansiasi TwitterOauth 42
Gambar 4.3 *Script* untuk proses pencarian kicauan berdasarkan tagar..... 43
Gambar 4.4 *Script* Proses menampung kicauan..... 44
Gambar 4.5 *Script* untuk proses menyimpan tagar hasil pencarian oleh pengguna teraktivasi 45
Gambar 4.6 *Script* untuk proses mengambil kicauan pengguna yang tersimpan . 46
Gambar 4.7 *Script* untuk menghitung probabilitas awal..... 46
Gambar 4.8 *Script* untuk proses pembersihan kicauan 47
Gambar 4.9 *Script* untuk proses pemecahan kicauan..... 48
Gambar 4.10 *Script* untuk proses klasifikasi 49
Gambar 4.11 *Script* untuk proses pembentukan data grafik sentimen pengguna teraktivasi 50
Gambar 4.12 *Script* untuk proses pembentukan data grafik sentimen untuk pengguna belum teraktivasi..... 51
Gambar 4.13 *Script* pembentukan data kata populer 52
Gambar 4.14 *Script* untuk proses pengecekan status aktivasi pengguna 52
Gambar 4.15 *Script* untuk proses pembentukan data pohon tagar..... 53
Gambar 4.16 *Script* penyimpanan data pengguna..... 54
Gambar 4.17 Data pengguna yang telah ter-*serialize* 54
Gambar 4.18 *Script* penyimpanan data daftar kata 55
Gambar 4.19 Data daftar kata awalan yang telah tersimpan..... 55
Gambar 4.20 Halaman muka..... 55
Gambar 4.21 Halaman hasil analisis sentimen dan kata populer untuk pengguna tidak teraktivasi 56

Gambar 4.22 Hasil pohon tagar untuk pengguna tidak teraktivasi 56
Gambar 4.23 Halaman muka pengguna teraktivasi 57
Gambar 4.24 Grafik kata populer pengguna teraktivasi 57
Gambar 4.25 Pohon tagar pengguna teraktivasi 58
Gambar 4.26 Halaman untuk mengelola daftar kata..... 58
Gambar 4.27 Halaman admin untuk mengelola pengguna 59

DAFTAR RUMUS

Rumus 3.1 *Prior* (Probabilitas Awal) 15
Rumus 3.2 *Likelihood* (Kemungkinan) 15
Rumus 3.3 *Posterior* (Probabilitas Akhir) 15

BAB I
PENDAHULUAN

1.2 Latar Belakang

Twitter merupakan *microblogging* yang memungkinkan penggunannya untuk membaca dan mengirim pesan berbasis teks hingga 140 karakter yang kemudian banyak dikenal dengan sebutan kicauan (*tweet*). Twitter menjadi sangat populer karena penggunaannya yang relatif sangat mudah dan sederhana, sehingga pengguna bisa menulis kicauan dengan berbagai hal yang dianggapnya menarik. Sejak Twitter muncul pada Maret 2006, terhitung hingga Juli 2012 total kicauan mencapai 163 milyar kicauan (Basch, 2012).

Indonesia tercatat sebagai pengguna Twitter terbesar kelima di dunia dengan total pengguna sebanyak 19,5 juta orang. Hal ini juga dikarenakan pengguna Internet di Indonesia mencapai 55 juta orang (terakhir tercatat pada Desember 2011) dan bahkan banyak pengguna Internet merupakan anak muda mulai dari usia 15-20 tahun dan 10-14 tahun dan terus akan bertambah (Kominfo, 2011). Melihat hal ini, untuk di Indonesia sendiri tidak bisa dipungkiri jumlah kicauan akan terus meningkat. Oleh karena itu, opini publik mengenai suatu hal dapat ditambang dari Twitter dengan lebih mudah karena tingkat kicauan yang sangat tinggi sehingga memungkinkan pengumpulan banyak data dengan mudah.

Untuk mengetahui opini publik tentang suatu hal, maka diperlukan analisis sentimen. Analisis sentimen itu sendiri adalah menganalisis opini mengenai pendapat seseorang mengenai topik tertentu atau suatu hal yang lain yang berisi penilaian negatif, positif atau netral. Analisis sentimen terhadap kicauan pengguna Twitter akan sangat bermanfaat untuk mengetahui opini publik terhadap berbagai hal, sebagai contoh adalah perusahaan dapat mengetahui bagaimana pendapat masyarakat tentang produk atau jasa mereka, demikian juga partai politik ataupun politikus dapat mengetahui bagaimana pandangan masyarakat tentang citra mereka dan lain sebagainya.

Satu hal yang sering tercantum dalam kicauan pengguna Twitter adalah tanda pagar yang kemudian sering disingkat menjadi tagar (*hashtag*), di mana tagar

berfungsi untuk mengelompokkan kicauan menurut topik sehingga pengguna Twitter dapat mengelompokkan kicauannya dengan mencantumkan tagar di setiap kicauannya (Twitter Inc, 2013). Oleh karena itu maka pengumpulan data akan dilakukan berdasarkan tagar yang kemudian dilakukan analasis sentimen.

1.3 Rumusan Masalah

Membuat aplikasi yang menggunakan kicauan dari pengguna Twitter untuk menambang opini publik berdasarkan tagar yang dicari, kemudian melakukan analisis sentimen, melihat kata populer yang sering muncul dalam kicauan pengguna Twitter dan melihat pohon tagar yang terdiri dari tagar apa saja yang sering muncul dalam kicauan bersamaan dengan tagar yang dicari.

1.4 Batasan Masalah

Pada pembuatan aplikasi ini memiliki beberapa batasan, yaitu:

1. Sumber data merupakan hasil pencarian kicauan berdasarkan tagar.
2. Sumber data yang diolah menggunakan bahasa Indonesia.

1.5 Tujuan dan Manfaat Penelitian

Aplikasi dapat mengambil data dari Twitter berupa kicauan berdasarkan tagar yang dicari, kemudian dilakukan analisis sentimen untuk setiap kicauan tersebut agar dapat diketahui penilaian positif, negatif atau netral dari pandangan publik terkait suatu topik berdasarkan tagar tersebut.

1.6 Langkah-langkah Penyelesaian

1. Menganalisis pola kicauan secara manual mengenai kicauan yang berisi penilaian positif, negatif ataupun netral berdasarkan tagar yang dicari.
2. Analisis kebutuhan
   a. Kebutuhan *input*
   b. Kebutuhan proses
   c. Kebutuhan *output*
3. Merancang aplikasi yang terdiri dari tahapan:

a. *Use case diagram*
b. *Sequence diagram*
c. *Class diagram*
d. Perancangan *array* untuk penyimpanan data
e. Perancangan *user interface*

4. Implementasi sistem meliputi:
   a. Penjelasan mengenai proses pengambilan data hingga analisis sentimen
   b. Penjelasan penyimpanan data pengguna dan daftar kata
   c. *User interface*
5. Pengujian sistem meliputi:
   a. Pengumpulan data berupa kicauan dari Twitter berdasarkan tagar yang dicari
   b. Analisis sentimen untuk melakukan klasifikasi

1.7 Sistematika Penulisan

Laporan Tugas Akhir ini disusun dalam lima bab dan berurutan sesuai tahapan-tahapannya. Setelah Bab 1 ini, dilanjutkan Bab 2 berisi landasan teori yang digunakan dalam pembuatan Tugas Akhir yaitu API Twitter, pengambilan data dari Twitter dan analisis sentimen. Bab 3 berisi metodologi yang akan digunakan yaitu terkait pengumpulan data, analisis dan perancangan aplikasi.

Kemudian Bab 4 terkait hasil dan pembahasan yaitu berisi hasil dari implementasi dan pengujian aplikasi yang dibuat. Terakhir adalah Bab 5 terkait dengan kesimpulan dan saran. Bab ini berisi kesimpulan dan hasil implementasi aplikasi serta saran-saran yang perlu diperhatikan berdasarkan keterbatasan yang ditemukan dalam aplikasi yang telah dikembangkan.

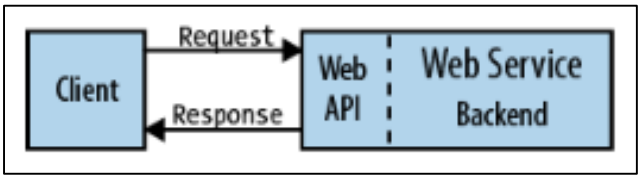
BAB II
LANDASAN TEORI

2.1 REST API Twitter

Representational State Transfer (REST) merupakan *hybrid style* yang berasal dari beberapa gaya arsitekur berbasis jaringan dan dikombinasikan dengan batasan-batasan tambahan yang mendefinisikan antarmuka konektor (Fielding, 2000), sedangkan *Application Programming Interface* (API) merupakan bentuk pemrograman yang memungkinkan dua aplikasi berkomunikasi, baik berbeda *platform*, sistem operasi dan bahasa pemrograman.

API merupakan pengembangan dari *web service* (layanan web) di mana layanan web merupakan perangkat lunak sistem yang dirancang untuk interoperabilitas dari mesin ke mesin melalui jaringan (W3C, 2004). API digunakan untuk meningkatkan fitur dan fungsionalitas pada salah satu aplikasi atau keduanya. Seperti yang diungkapkan oleh Masse (2011) tentang API yaitu:

"Layanan web dibangun bertujuan untuk mendukung kebutuhan dari sebuah situs atau aplikasi lainnya. Program klien menggunakan API untuk berkomunikasi dengan layanan web. Secara umum, API menghadapkan satu set data dan fungsi untuk memfasilitasi antara program komputer dan memungkinkan mereka untuk bertukar informasi" yang dapat digambarkan pada gambar 2.1.



Gambar 2.1 Web API

API diimplementasikan menggunakan metode HTTP (*Hyper Text Transfer Protocol*) yang merupakan pondasi komunikasi data untuk WWW (*World Wide Web*) di mana standar pengembangan HTTP termasuk dalam naungan IETF (*Internet Engineering Task Force*) dan W3C (*World Wide Web Consortium*). Masse (2011) menjelaskan beberapa *request method* standar yang digunakan dalam API, yaitu:

1. Metode GET, digunakan untuk mengambil atau membaca representasi sumber daya.
2. Metode POST, digunakan untuk membuat sumber daya baru.
3. Metode PUT, digunakan untuk melakukan pembaruan sumber daya, namun juga dapat digunakan untuk membuat sumber daya baru apabila belum tersedia.
4. Metode DELETE, digunakan untuk menghapus sumber daya yang ada.

Untuk keluaran dokumen hasil *request* yang direspon oleh *server* Twitter menggunakan API Twitter secara *default* berbentuk JSON (*JavaScript Object Notation*) di mana format ini lebih mudah untuk dibaca oleh manusia. Penulisan JSON lebih singkat dari XML (*Extensible Markup Language*) yang di mana format penulisan XML diawali dan diakhiri menggunakan tag yang dirasa kurang efektif dan membutuhkan *storage* yang lebih besar dari JSON. Penulisan JSON terbilang mudah, karena menggunakan "{" dan "}" sebagai tanda sintaksnya, contoh JSON dapat dilihat pada gambar 2.2.

```
{
    "firstName": "John",
    "lastName": "Smith",
    "age": 25,
    "address": {
        "streetAddress": "21 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": 10021
    },
    "phoneNumbers": [
        {
            "type": "home",
            "number": "212 555-1234"
        },
        {
            "type": "fax",
            "number": "646 555-4567"
        }
    ]
}
```

Gambar 2.2 Contoh JSON

Untuk melakukan *parsing* atau menghasilkan data berformat JSON sudah tersedia dalam berbagai macam bahasa pemrograman. Adapun kekuranganya adalah penggunaan JSON lebih terfokus pada layanan web.

API yang dikembangkan Twitter bertujuan untuk memudahkan para pengembang aplikasi untuk mengakses sumber daya Twitter. Penggunaan API Twitter versi terbaru, yaitu v1.1 telah mengacu pada hal yang dijabarkan diatas, namun untuk menggunakan API Twitter, pengembang perlu mendaftarkan aplikasi yang dikembangkan pada Twitter, kemudian pengembang mendapatkan *consumer_key* dan *consumer_secret* yang merupakan kunci untuk menghubungkan aplikasi pada API Twitter. Pengembang juga akan mendapatkan *access_token* dan *access_token_secret*, di mana ini akan sangat bermanfaat apabila aplikasi merupakan aplikasi satu pengguna ataupun penggunaan aplikasi hanya untuk pribadi dan membutuhkan hak akses akun Twitter pribadi sehingga tidak perlu melakukan *request access token*. Apabila aplikasi dikembangkan untuk banyak pengguna yang membutuhkan *login* menggunakan akun Twitter-nya, maka tiap pengguna yang melakukan *login*, akan melakukan *request access token* saat *login* sukses dan mengijinkan aplikasi untuk mengakses akun pengguna.

2.2 Pengambilan Data dari Twitter

Setiap kicauan memiliki konten yang berbeda, namun dalam setiap kicauan terdapat beberapa karakteristik yang secara umum dapat dilihat sebagai berikut:

1. Panjang maksimum kicauan hanya 140 karakter, namun pengguna dapat berbagi foto, video maupun *link* suatu *website* di dalamnya.
2. Penggunaan bahasa dalam setiap kicauan sangat bervariasi, namun memiliki kecenderungan menggunakan bahasa yang tidak baku dan lebih sering menggunakan bahasa sesuai daerahnya masing-masing.
3. Pengguna dapat melakukan *retweet* (melakukan *update* ulang kicauan orang lain dengan menambahkan opini sendiri atau tidak) kicauan orang lain dan dimungkinkan terjadinya kelebihan jumlah karakter (lebih dari 140 karakter sesuai ketentuan). Secara umum apabila terjadi kelebihan karakter maka pengguna tidak dapat melakukan *posting* kicauan tersebut, namun pengguna dapat menggunakan *third party application* (aplikasi dari pihak ketiga) yang mampu menangani hal tersebut yaitu dengan meringkas kelebihan karakter menjadi sebuah *short link* yang di mana *short link* tersebut akan tercantum

dalam akhir kicauan setelah pengguna melakukan *posting*. *Short link* tersebut apabila dibuka akan menampilkan isi kicauan pengguna secara penuh.

4. Pengguna dapat mencantumkan tagar dalam kicauan mereka. Tagar seringkali dicantumkan pengguna sesuai topik dalam kicauan mereka.
5. Pengambilan kicauan pengguna dilakukan menggunakan aplikasi yang telah terhubung dengan Twitter melalui Twitter API berdasarkan tagar.

Gambar 2.3 merupakan contoh kicauan yang menggunakan tagar.



Keterangan:
1. Gambar atau foto pengguna
2. Nama pengguna
3. *Username* atau ID nama pengguna
4. Isi kicauan
5. Tagar
6. Waktu *posting* kicauan

Gambar 2.3 Contoh kicauan

Tagar dalam kicauan tersebut yang akan digunakan sebagai kata kunci kecocokan dengan tagar yang dicari, kemudian ditampung sebagai data yang akan dilakukan analisis sentimen.

2.3 Penambangan Opini Publik

Arti kata "opini" menurut KBBI (Kamus Besar Bahasa Indonesia) adalah pendapat, pikiran atau pendirian, sedangkan "opini publik" memiliki arti yaitu pendapat umum atau pendapat sebagian besar masyarakat. Opini dapat diungkapkan dengan banyak cara seperti melalui sosial media, surat, buku dan lain sebagainya.

Seiring kemajuan jaman, banyak opini yang diungkapkan melalui sosial media, hal ini dikarenakan sosial media lebih sederhana dan mudah untuk mengaksesnya. Salah satunya adalah sosial media Twitter yang merupakan mikroblog sangat populer saat ini. Tentang opini dalam mikroblog, Agarwal (et al, 2011) mengatakan:

“Situs mikroblog telah berkembang menjadi sumber berbagai jenis informasi. Hal ini dikarenakan sifat mikroblog yang melakukan *post* secara *real time* pesan orang tentang berbagai topik, membahas isu-isu saat ini, keluhan atau ekspresi sentimen tentang produk yang mereka gunakan sehari-hari”.

Menggunakan Twitter sangat dimungkinkan terjadinya banjir informasi, karena pada dasarnya mikroblog ini melakukan *post* kicauan atau informasi secara *real time* di mana hal tersebut belum tentu dibutuhkan oleh orang lain. Untuk meminimalisir pengambilan data yang tidak perlu maka pengambilan dilakukan berdarkan tagar yang dicari. Fungsi tagar itu sendiri adalah untuk menandai kata kunci atau topik di kicauan, tagar dibuat oleh pengguna Twitter sebagai cara menggolongkan kicauan (Twitter Inc, 2013). Mendapatkan kicauan sebagai data berdasarkan tagar untuk dilakukan penambangan opini terbilang cukup spesifik, mengingat fungsi tagar itu sendiri.

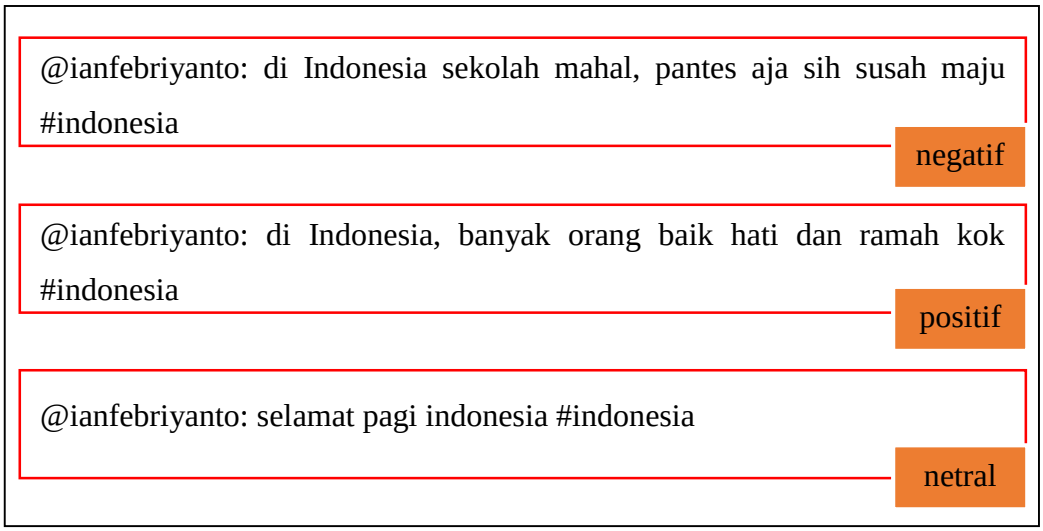
Opini dalam Twitter dapat memiliki sentimen. Arti kata sentimen menurut KBBI adalah pendapat atau pandangan yang didasarkan pada perasaan yang berlebih-lebihan terhadap sesuatu (bertentangan dengan pertimbangan pikiran). Aplikasi ini secara dasar mengidentifikasi sentimen yang dibagi menjadi tiga kategori, yaitu:

- Negatif, opini yang mengandung sentimen negatif maka memiliki kecenderungan berisi kurang baik atau tidak senang.
- Positif, opini yang mengandung sentimen positif maka memiliki kecenderungan berisi baik atau senang.
- Netral, opini yang tidak mengandung sentimen positif ataupun negatif.

Sentimen ini yang kemudian menjadi dasar untuk mengetahui bagaimana opini publik apakah cenderung negatif, positif atau netral tentang suatu topik yang dicari berdasarkan tagar.

2.4 Analisis Sentimen

Analisis sentimen yang dilakukan adalah dengan mengelompokkan kicauan yang didapatkan dari Twitter ke dalam tiga kategori yaitu negatif, positif dan netral. Contoh kicauan yang mengandung sentimen dapat dilihat pada gambar 2.4.



Gambar 2.4 Illustrasi kicauan yang mengandung sentimen

Setiap kicauan yang dianalisis akan memiliki nilai seberapa besar kicauan tersebut mengandung konten negatif, positif dan netral yang kemudian akan diambil nilai tertinggi dari ketiganya, sehingga menjadi dasar untuk melakukan pengkategorian di mana kicauan tersebut akan dikelompokkan. Apabila ketiga kategori memiliki nilai yang sama, maka kicauan tersebut akan diabaikan.

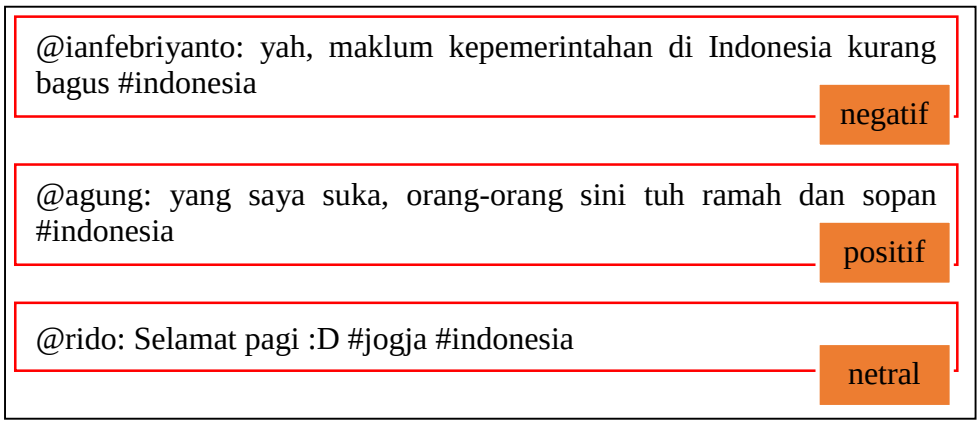
BAB III
METODOLOGI

3.1 Pengumpulan Kicauan Sebagai Data

Data yang berupa kicauan hasil pencarian berdasarkan tagar akan dikumpulkan dan ditampung berdasarkan tanggal kapan data tersebut diambil. Pengumpulan data akan dilakukan secara berkala agar mendapatkan lebih banyak variasi opini dalam kicauan dari waktu ke waktu. Hal ini juga dikarenakan batas maksimum kicauan yang dapat diambil dalam sekali pencarian adalah 100 kicauan. Batasan ini merupakan batasan dari API Twitter itu sendiri, terkadang jumlah kicauan hasil pencarian pun tidak mencapai 100 buah kicauan dalam sekali melakukan pencarian.

3.2 Analisa Pola Kicauan Sebagai Data

Untuk melakukan pencarian menggunakan tagar, tidak ada kekangan secara khusus yang dibutuhkan. Pengguna dapat melakukan pencarian tentang apa saja yang dianggapnya menarik untuk dicari berdasarkan tagar. Kicauan-kicauan hasil pencarian tersebut akan memiliki beberapa sentimen, dimana sentimen tersebut yang akan digunakan untuk mengelompokkan kicauan tersebut ke dalam kategori positif, negatif ataupun netral dengan contoh seperti pada Gambar 3.1.



Gambar 3.1 Contoh kicauan sebagai data

3.3 Pembersihan Kicauan

Sebelum melakukan pemecahan kicauan menjadi per kata maka perlu dilakukan pembersihan untuk mengubah kata menjadi lebih jelas dan mampu dibaca oleh sistem. Langkah pembersihan pertama adalah mengganti karakter yang kurang lazim dan angka yang digunakan dalam menggantikan beberapa karakter dalam bahasa Indonesia menjadi karakter yang lazim dan mudah dibaca. Metode yang digunakan adalah dengan memanfaatkan fungsi *strtr()* dalam bahasa pemrograman PHP, fungsi tersebut digunakan untuk menerjemahkan karakter tertentu dalam sebuah *string*, di mana telah didefinisikan karakter dan huruf apa saja yang secara umum digunakan untuk menggantikan beberapa karakter. Kedua adalah mengubah kicauan menjadi huruf kecil.

Ketiga adalah membersihkan dari karakter yang berulang, yaitu program akan mengidentifikasi karakter yang diulang lebih dari satu kali maka akan diganti dengan karakter yang sama sebanyak satu kali. Metode yang digunakan adalah dengan menggunakan *regular expression* (Regex) dalam bahasa pemrograman PHP, di mana fungsi ini digunakan untuk mengenali pola perulangan karakter dalam kata yang kemudian menghilangkan perulangan karakter tersebut. Ketiga proses tersebut dilakukan karena dengan menyeragamkan ukuran dan format karakter dapat menyederhanakan daftar kata yang tersimpan, sehingga untuk kata yang sama tidak terlalu bervariasi. Contoh pembersihan kicauan dapat dilihat pada tabel 3.1.

Tabel 3.1 Contoh pembersihan kicauan

Proses	Input	Output
Membersihkan karakter dan simbol	gÀgàl!	gagal
Mengganti angka yang terdapat dalam kata menjadi karakter	S3m4n64t	semangat
Membersihkan karakter yang diulang	beerhasiiiiiiiiiiiiiiiiiil	berhasil

3.4 Proses Analisis

3.4.1 Daftar Kata

Daftar kata digunakan untuk menampung kata-kata yang akan digunakan untuk proses analisis. Daftar kata dapat diubah, dihapus ataupun ditambah secara dinamis oleh pengguna melalui aplikasi. Berikut daftar kata yang digunakan dalam aplikasi antara lain:

1. Daftar kata mengandung sentimen

Setiap kicauan akan berisi kata kunci yang dapat digunakan untuk analisis. Kata kunci tersebut yang akan digunakan sebagai dasar untuk menentukan kategori masing-masing kicauan. Hal ini dihitung berdasarkan kemunculan kata kunci tersebut, di mana setiap kemunculan kata kunci akan dicocokan untuk daftar kata masing-masing kategori yaitu daftar kata positif, daftar kata negatif dan netral. Hal ini sebagai dasar proses analisis kicauan untuk mengkategorikannya dalam kategori positif, negatif ataupun netral.

Untuk beberapa frasa akan mengandung makna positif atau negatif, seperti ‘tidak bagus’ dan ‘kurang rapi’ akan mempunyai makna negatif dan ‘tidak jelek’ akan mempunyai makna positif. Untuk mengatasi hal ini diperlukan daftar kata awalan yang berisi ‘tidak’, ‘kurang’ dan lain sebagianya, sehingga apabila terdapat kata yang dikenai dengan awalan tersebut akan diproses dan digabungkan menjadi satu. Contoh daftar kata yang mengandung sentimen, dapat dilihat pada tabel 3.2, tabel 3.3 dan tabel 3.4.

Tabel 3.2 Contoh daftar kata positif

No.	Kata
1	bagus
2	indah
3	cantik
4	ramah
5	sopan
6	tidakjelek
7	rapi

Tabel 3.3 Contoh daftar kata negatif

No.	Kata
1	jelek
2	buruk
3	ancur
4	galak
5	marah
6	tidakbagus
7	kurangrapi

Tabel 3.4 Contoh daftar kata netral

No.	Kata
1	rata-rata
2	sedang
3	percaya
4	lengkap
5	akhirnya
6	cukup
7	biasa

2. Daftar kata awalan

Agar proses analisis lebih akurat dalam perhitungannya maka diperlukan daftar kata awalan. Daftar kata awalan digunakan untuk menganalisa makna kata yang apabila dikenai awalan tersebut akan mempunyai makna negatif atau positif, sebagai contoh kalimat ‘rumah ini tidak bagus’ yang seharusnya mempunyai makna negatif, namun dalam proses pemecahan kicauan untuk perhitungan, maka frasa ‘tidak bagus’ akan menjadi dua buah kata yang berbeda yaitu ‘tidak’ dan ‘bagus’, sehingga kata ‘tidak’ menjadi kata yang tidak dihitung, dan ‘bagus’ mempunyai makna positif karena terdapat dalam daftar kata positif, sehingga proses analisis akan kurang akurat dan kalimat diatas menjadi bermakna positif.

Untuk mengatasi hal ini maka dibutuhkan daftar kata awalan seperti ‘tidak’, ‘kurang’ dan lainnya, sehingga dalam pemecahan kicauan akan menggabungkan kata ‘tidak’ dan ‘bagus’ menjadi ‘tidakbagus’ yang dimana ‘tidakbagus’ sendiri akan dideklarasikan dalam daftar kata negatif, sehingga contoh kalimat di atas tetap bermakna negatif. Contoh daftar kata awalan dapat dilihat pada tabel 3.5.

Tabel 3.5 Contoh daftar kata awalan

No.	Kata
1	tidak
2	kurang

3. Daftar kata diabaikan

Daftar kata yang diabaikan digunakan untuk mempersingkat proses perhitungan dalam analisis kicauan dimana apabila terdapat kata yang tidak memiliki makna negatif, positif ataupun netral maka kata tersebut tidak perlu diproses. Sepertihalnya kata-kata yang menunjukkan obyek benda, orang, tempat atau lainnya yang memang tidak mempunyai makna sentimen. Contoh daftar kata yang diabaikan dapat dilihat pada tabel 3.6.

Tabel 3.6 Contoh daftar kata diabaikan

No.	Kata
1	aku
2	kamu
3	bagaimana
4	mengapa
5	siapa
6	entah
7	kapan

3.4.2 Proses Perhitungan

Pada dasarnya proses klasifikasi kicauan ke dalam kategori positif, negatif, atau netral, dilakukan dengan konsep *string matching*, di mana melakukan pencocokan daftar kata dengan kata dalam kicauan yang telah diambil. Setiap terjadi kecocokan dengan salah satu daftar kata positif, negatif atau netral maka akan menghasilkan nilai satu, yang ditampung dalam bentuk skor untuk setiap kategori dalam kicauan tersebut, dan akan terus bertambah apabila ditemukan kecocokan dalam iterasi berikutnya.

Untuk menghitung masing-masing nilai kemungkinan untuk setiap kategori yaitu, positif, negatif dan netral. Hal pertama yang dilakukan adalah menghitung

prior (probabilitas awal) untuk masing-masing kategori positif, negatif, dan netral. Menghitung probabilitas awal dapat dilakukan dengan rumus:

$$probabilitas\ awal(kategori\ x) = \frac{total\ kategori\ x}{total\ seluruh\ kategori} \quad (3.1)$$

Setelah memformulasikan probabilitas awal, untuk mengklasifikasikan kicauan dalam kategori positif, negatif atau netral diperlukan perhitungan jumlah kandungan kata tiap kategori tersebut yang terdapat dalam kicauan berdasarkan daftar kata yang telah ditetapkan. Hal ini merupakan awal untuk menghitung *likelihood* (kemungkinan) masing-masing kategori dari kicauan tersebut, sehingga dapat dihitung dengan rumus:

$$kemungkinan\ kategori\ x = \frac{jumlah\ kata\ kategori\ x\ dalam\ kicauan}{total\ kata\ kategori\ x\ dalam\ daftar\ kata} \quad (3.2)$$

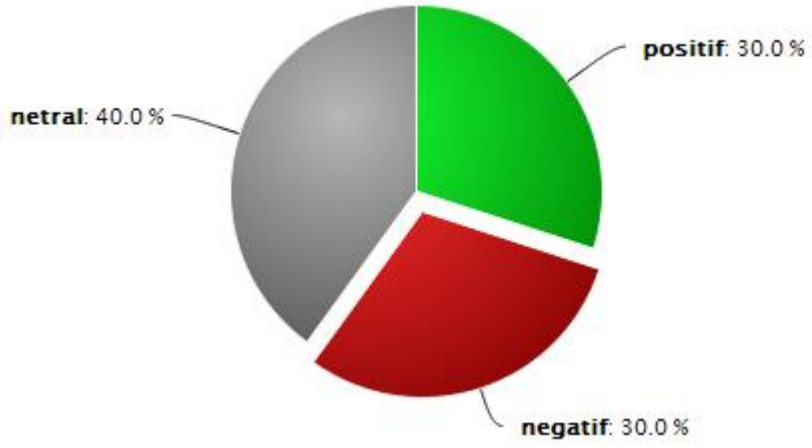
(StatSoft, Inc; 2013)

Untuk mendapatkan *posterior* (probabilitas akhir) dari masing-masing kategori dapat menggunakan rumus:

$$probabilitas\ akhir\ kategori\ x = probabilitas\ awal\ kategori\ x\ \times kemungkinan\ kategori\ x \quad (3.3)$$

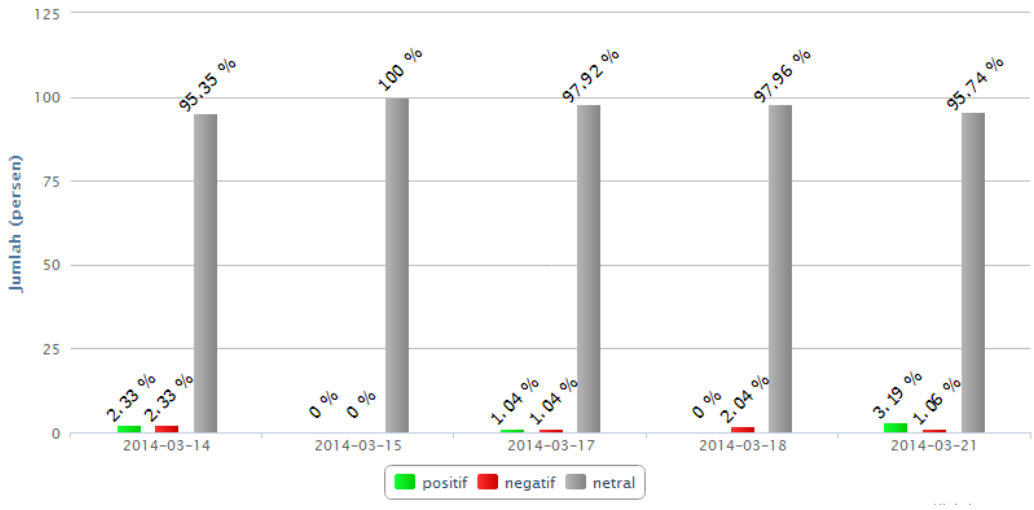
(StatSoft, Inc; 2013)

Setiap kicauan akan menghasilkan probabilitas akhir untuk masing-masing kategori, yaitu probabilitas akhir kategori negatif, probabilitas akhir kategori positif dan probabilitas akhir kategori netral. Masing-masing nilai probabilitas akhir tersebut digunakan untuk pembanding dalam mengkategorikan kicauan berdasarkan nilai probabilitas akhir terbesar. Data akan dipresentasikan dalam bentuk grafik menurut prosentase dari masing-masing kategori. Contoh hasil analisis sentimen untuk pengguna belum teraktivasi dapat dilihat pada gambar 3.2.



Gambar 3.2 Contoh hasil analisis sentimen untuk pengguna belum teraktivasi

Salah satu fitur dari pengguna teraktivasi adalah dapat melakukan pencarian tagar secara kontinyu dan dipresentasikan dalam bentuk grafik berdasarkan tanggal data tersebut diambil. Contoh hasil analisis sentimen untuk pengguna teraktivasi dapat dilihat pada gambar 3.3.

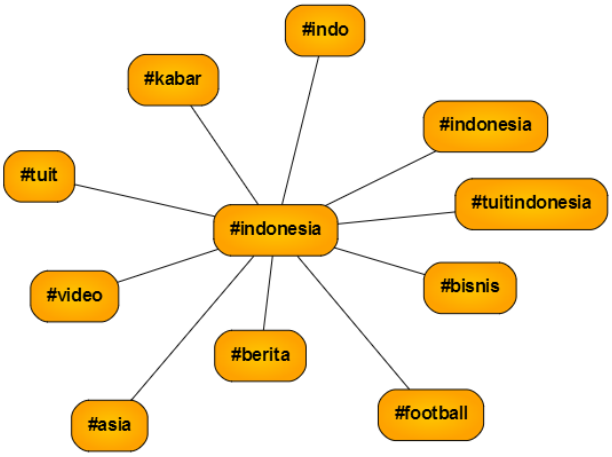


Gambar 3.3 Contoh hasil analisis sentimen untuk pengguna teraktivasi

3.5 Pohon Tagar

Pohon tagar merupakan hasil dari pengumpulan tagar yang tercantum atau muncul dalam kicauan bersamaan dengan tagar yang dicari. Tagar tersebut yang kemudian ditampung dan dipresentasikan dalam bentuk pohon di mana induk

utama dari pohon adalah tagar yang dimasukkan dalam pencarian. Gambar 3.4 merupakan contoh pohon tagar yang telah terbentuk.

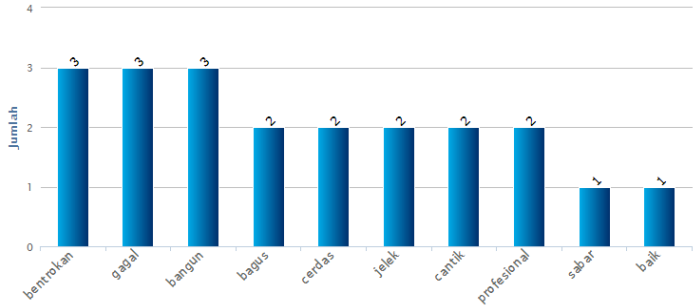


Gambar 3.4 Contoh pohon tagar

3.6 Kata Populer

Kata populer merupakan kata yang sering muncul dalam kicauan. Kata populer didapatkan dari kicauan hasil pencarian berdasarkan tagar, di mana kata tersebut mengandung sentimen dan terdapat dalam daftar kata positif, negatif ataupun netral, yang hasilnya kemudian dihitung jumlah kemunculannya dan di presentasikan dalam bentuk grafik.

Fungsi dari kata populer adalah sebagai indikator dari hasil analisis sentimen, di mana pengguna dapat melihat kata apa saja yang menjadi dasar mengapa hasil analisis sentimen menunjukkan kecenderungan untuk mengelompokkannya ke dalam kategori negatif, positif atau netral. Gambar 3.5 merupakan contoh grafik kata populer yang telah terbentuk.



Gambar 3.5 Contoh grafik kata populer

3.7 Analisa Kebutuhan

3.7.1 Kebutuhan Input

Data yang dibutuhkan dari aplikasi ini meliputi:

1. Kicauan dari Twitter

   Kicauan didapatkan dari Twitter berdasarkan tagar yang dicari.

2. Daftar kata

   Kumpulan kata yang akan digunakan sebagai dasar untuk mengelompokkan kicauan dalam kategori. Aplikasi akan menggunakan tiga daftar kata, yaitu:

   a. Positif, untuk kumpulan kata positif.
   b. Negatif, untuk kumpulan kata negatif.
   c. Netral, untuk kumpulan kata netral.

3. Data pengguna

   Data pengguna untuk pengguna yang teraktivasi meliputi *username* dan *password* yang digunakan untuk *login* ke sistem.

3.7.2 Kebutuhan Proses

Proses yang dibutuhkan dalam aplikasi ini terdiri dari beberapa kelompok proses, antara lain:

a. Proses analisis

Proses dalam kelompok ini, merupakan proses utama dalam aplikasi, dari proeses pengambilan data, hingga presentasi hasil pemrosesan data, meliputi sentimen, kata populer dan pohon tagar. Berikut proses yang dibutuhkan:

1. Proses untuk mengambil kicauan dari Twitter berdasarkan tagar yang dicari.
2. Proses untuk mengubah format tanggal pembuatan kicauan yang diambil.
3. Proses untuk menyimpan kicauan hasil pencarian berdasarkan tagar.
4. Proses untuk menyimpan tagar yang tercantum dalam kicauan hasil pencarian tagar.
5. Proses untuk mengambil kicauan yang tersimpan.
6. Proses untuk mengambil tagar yang tersimpan.
7. Proses untuk menghitung probabilitas awal.
8. Proses untuk pembersihan kicauan.

9. Proses untuk pemecahan kicauan menjadi per kata.
10. Proses untuk klasifikasi kicauan ke dalam kategori negatif, positif atau netral.
11. Proses untuk menghitung kata populer atau kata yang sering muncul dalam kicauan.
12. Proses untuk mempresentasikan hasil analisis sentimen dalam bentuk grafik.
13. Proses untuk mempresentasikan hasil kata populer dalam bentuk grafik.
14. Proses untuk mempresentasikan tagar dalam bentuk pohon tagar.
15. Proses untuk mereset data kicauan dan tagar yang tersimpan untuk pengguna teraktivasi.

b. *Login*

Proses dalam kelompok *login* akan menangani proses login pengguna teraktivasi. Berikut proses yang diperlukan:

1. Proses untuk memuat data pengguna meliputi *id*, *username* dan *password* yang digunakan untuk *login*.
2. Proses untuk *login*.
3. Proses untuk pengecekan status *login* yang terdapat dalam session pengguna teraktivasi.
4. Proses untuk pengecekan admin.
5. Proses untuk *logout*.

c. Manajemen daftar kata

Proses dalam daftar kata akan menangani proses pembentukan daftar kata sebagai kamus untuk proses analisis sentimen hingga mengelola daftar kata yang tersimpan oleh pengguna teraktivasi. Berikut proses yang dibutuhkan:

1. Proses untuk menangani penambahan, perubahan dan penghapusan pada daftar kata oleh pengguna teraktivasi.
2. Proses untuk melakukan penyimpanan daftar kata dalam bentuk non-*array*.
3. Proses pengambilan daftar kata non-*array*.

4. Proses penyimpanan daftar kata dalam bentuk *array*.
5. Proses pengambilan data dalam bentuk *array*.
6. Proses pembentukan kamus untuk proses klasifikasi kicauan.

Fungsi dari penyimpanan data dalam bentuk non-*array* adalah memungkinkan admin untuk melalukan perubahan, penambahan dan penghapusan secara langsung ke dalam *file* penyimpanan daftar kata. Fungsi dari penyimpanan daftar kata dalam bentuk *array* adalah untuk mempermudah proses pembentukan kamus yang digunakan untuk proses analisis sentimen dan menampilkan daftar kata ke pengguna teraktivasi.

d. Manajemen pengguna

Proses dalam kelompok pengguna, merupakan proses yang menangani tentang pengelolaan data pengguna teraktivasi itu sendiri dan pengelolaan pengguna oleh admin. Berikut proses yang dibutuhkan:

1. Proses daftar yang menangani pengguna baru saat mendaftar ke sistem.
2. Proses menampilkan kepada admin seluruh pengguna yang telah teraktivasi maupun belum teraktivasi.
3. Proses aktivasi pengguna untuk menangani admin saat melakukan aktivasi pengguna yang belum teraktivasi.
4. Proses deaktivasi pengguna untuk menangani admin saat melakukakan deaktivasi pengguna yang telah teraktivasi.
5. Proses hapus pengguna untuk menangani admin saat menghapus pengguna.
6. Proses ubah *password* untuk menangani pengguna teraktivasi melakukan perubahan *password* .

3.7.3 Kebutuhan Output

Keluaran dari aplikasi ini meliputi:

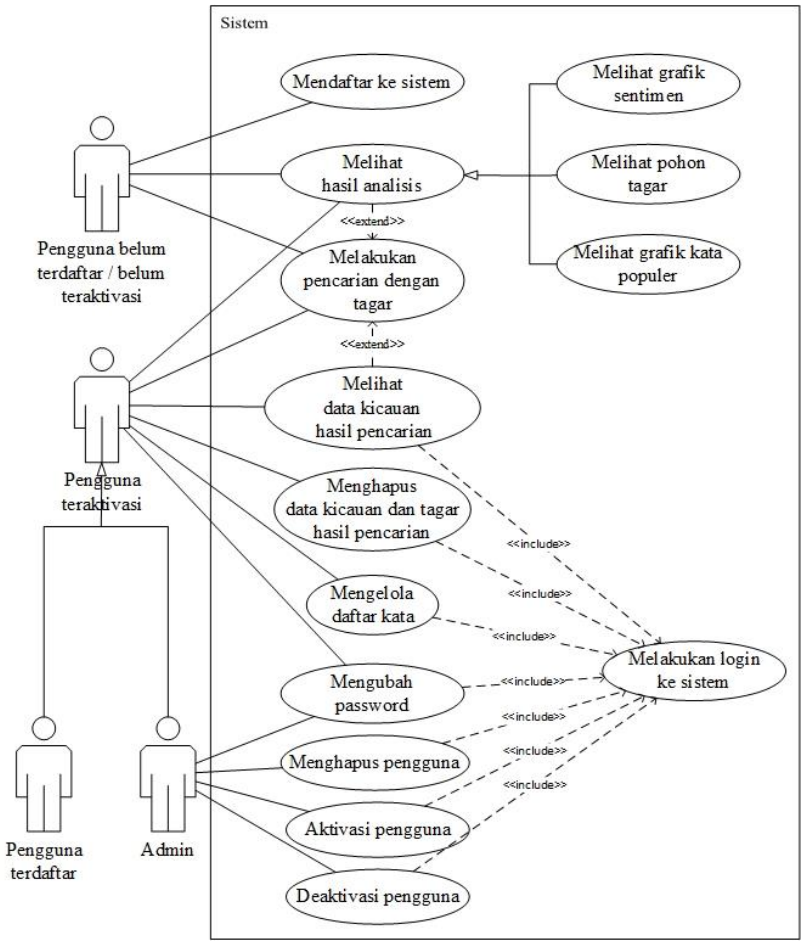
1. Grafik setimen hasil analisis negatif, positif dan netral berdasarkan tanggal kicauan tersebut diambil.
2. Grafik kata populer yang sering digunakan dalam kicauan.

3. Pohon tagar, tagar yang sering muncul bersamaan dengan tagar yang digunakan dalam pencarian.
4. Hasil tampungan kicauan saat pencarian berdasarkan tagar oleh pengguna teraktivasi.

3.8 Perancangan

3.8.1 Use Case Diagram

Pada Gambar 3.6 menunjukkan bagaimana peran setiap aktor yang menggunakan aplikasi ini. *Use case diagram* digunakan untuk mendiskripsikan bagaimana interaksi aktor (pengguna) dengan sistem. Terdapat tiga aktor, yaitu pengguna belum teraktivasi, pengguna yang telah teraktivasi dan admin, di mana setiap pengguna mempunyai tingkat akses aplikasi yang berbeda. Gambar 3.6 adalah *use case diagram* yang menjelaskan interaksi aktor dan aplikasi.

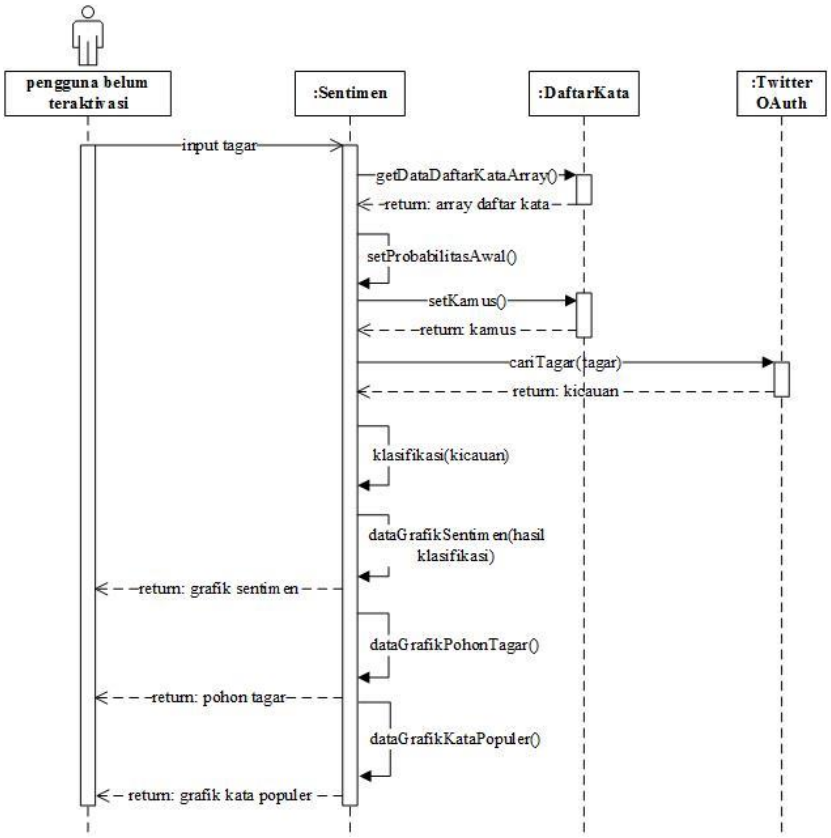


Gambar 3.6 *Use case diagram*

3.8.2 Sequence Diagram

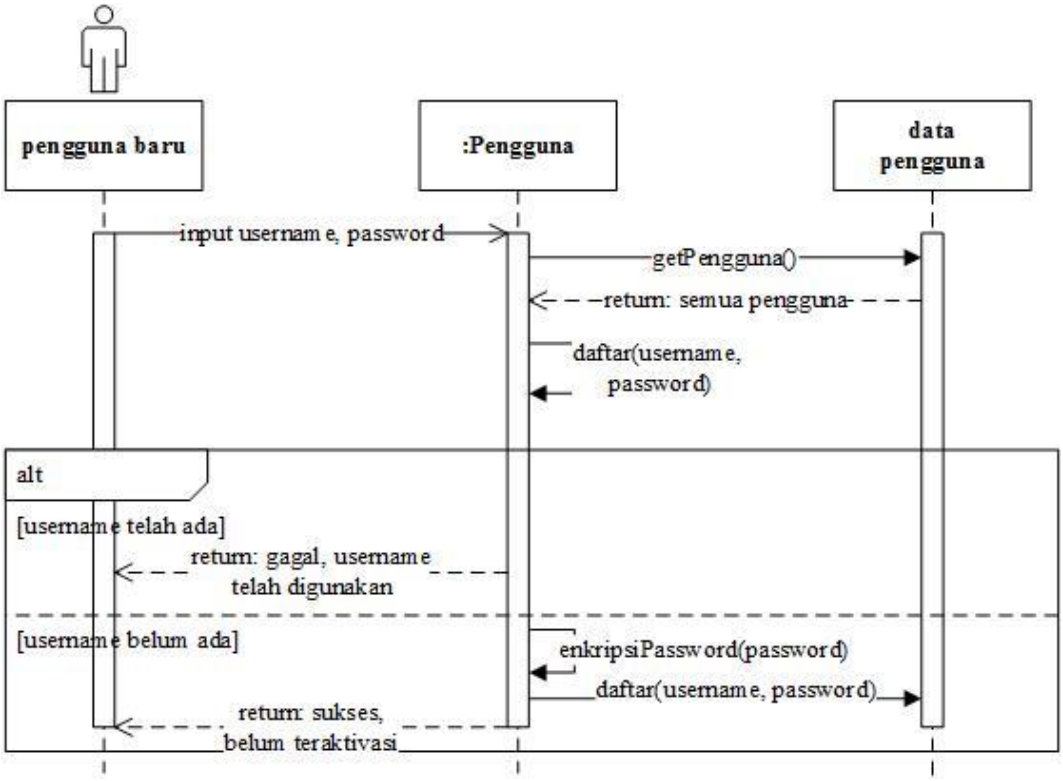
Sequence diagram digunakan untuk menggambarkan interaksi antar obyek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. *Sequence diagram* terdiri dari dimensi vertikal (waktu) dan horizontal (obyek-obyek terkait). Secara garis besar pengguna dalam aplikasi ini dibedakan menjadi dua yaitu pengguna teraktivasi dan pengguna belum teraktivasi.

Gambar 3.7 merupakan *sequence diagram* untuk melakukan pencarian kicauan bedasarkan tagar bagi pengguna belum teraktivasi, aplikasi akan melakukan *load* daftar kata dan melakukan perhitungan probabilitas awal untuk setiap kategori positif, negatif dan netral terlebih dahulu, kemudian melakukan pembentukan kamus untuk melakukan proses *string matching*, yang selanjutnya melakukan *request* ke API Twitter, setelah mendapatkan *response* berupa kacauan hasil pencarian, kemudian dilakukan proses analisis sentimen, kata populer dan pohon tagar.



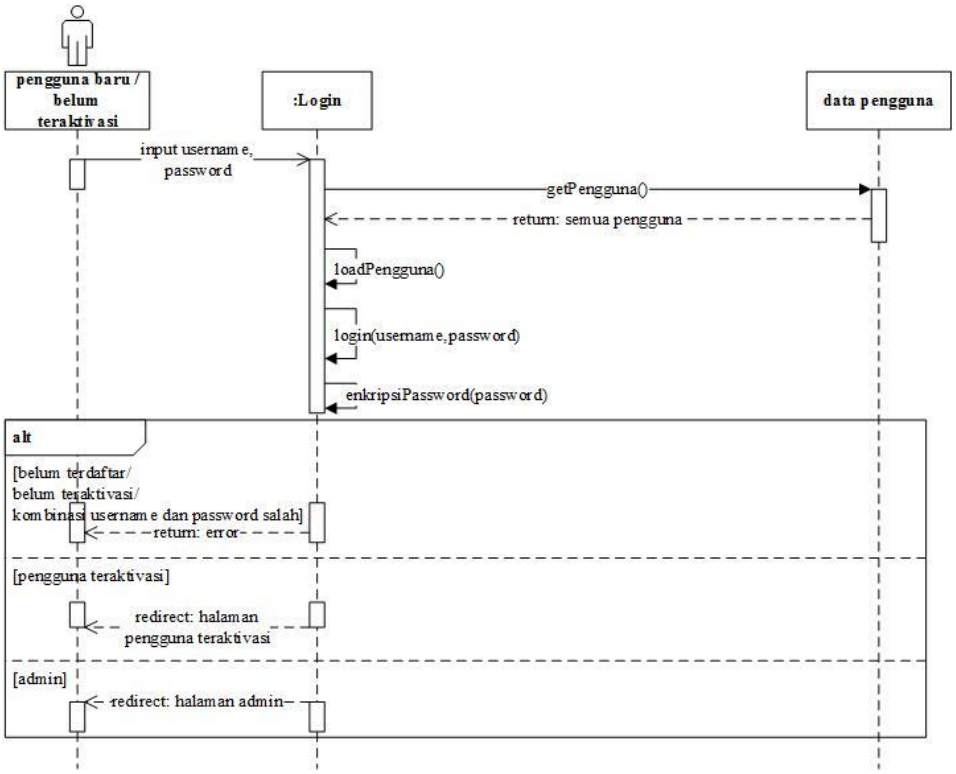
Gambar 3.7 *Sequence diagram* untuk melakukan pencarian kicauan berdasarkan tagar untuk pengguna belum teraktivasi

Gambar 3.8 merupakan *sequence diagram* untuk melakukan pendaftaran ke aplikasi, di mana pengguna baru melakukan pendaftaran dengan memasukkan *username* dan *password*, kemudian sistem melakukan pengecekan *username* ke semua pengguna yang telah terdaftar agar tidak terjadi duplikasi *username* pengguna. Apabila sudah tidak ditemukan *username* yang sama maka pengguna dapat menunggu untuk diaktivasi oleh admin.



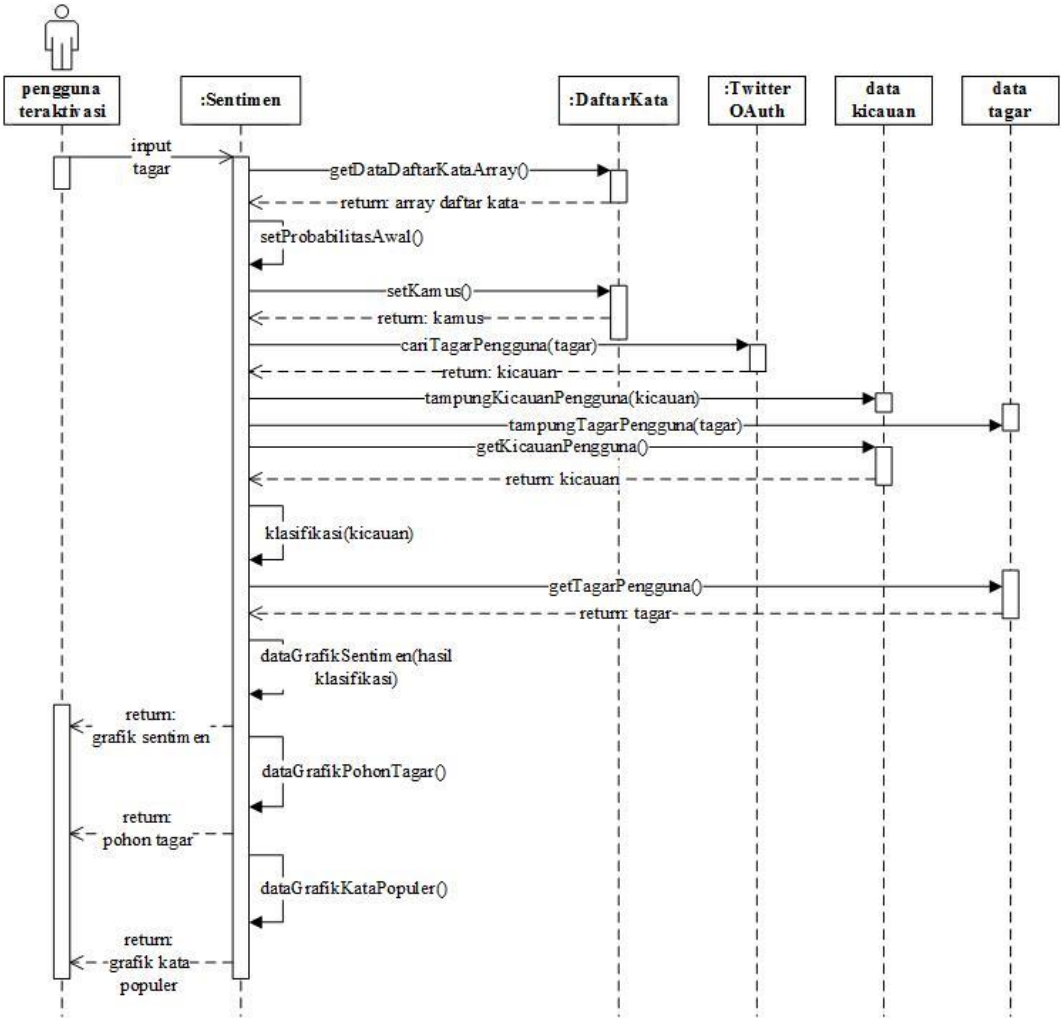
Gambar 3.8 *Sequence diagram* untuk mendaftar ke sistem

Gambar 3.9 merupakan *sequence diagram* untuk melakukan *login* ke dalam aplikasi, di mana sistem melakukan pengecekan *username* dan *password* pengguna yang telah tersimpan. Selain itu sistem akan melakukan pengecekan *level* pengguna dan status aktivasi pengguna. Apabila level merupakan admin, apabila login sukses akan langsung ditujukan ke halaman admin begitu juga dengan pengguna biasa. Sedangkan untuk status aktivasi, apabila belum teraktivasi maka tidak dapat melakukan *login.*



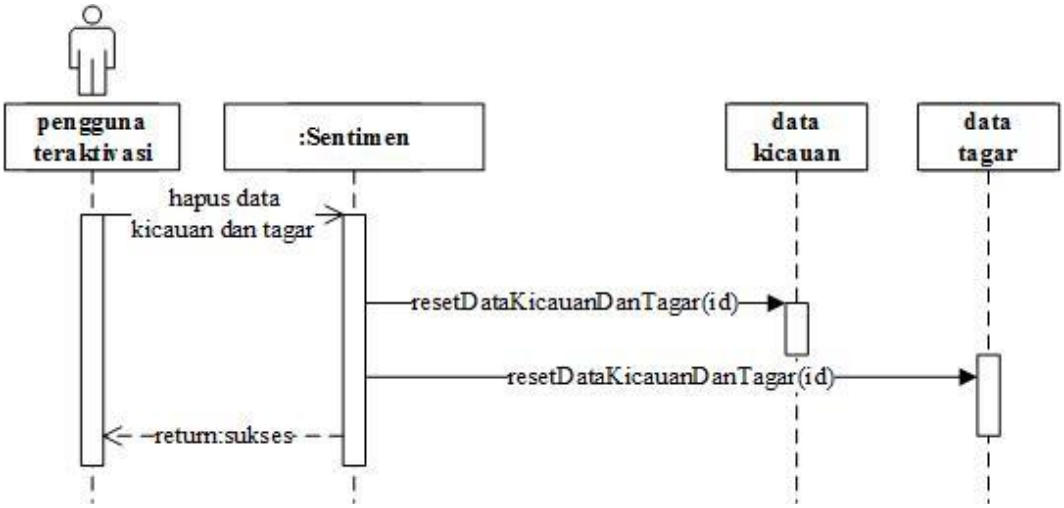
Gambar 3.9 *Sequence diagram* untuk melakukan *login*

Gambar 3.10 merupakan *sequence diagram* untuk pencarian kciauan berdasarkan tagar untuk pengguna teraktivasi. Pertama adalah melakukan *load* daftar kata untuk menghitung probabilitas awal untuk masing-masing kategori positif, negatif dan netral. Kedua melakukan pembentukan kamus untuk melakukan proses *string matching*. Disaat pengguna melakukan pencarian, kicauan dan tagar yang tercantum dalam kicauan akan disimpan dalam sebuah *file* penyimpanan. Kemudian untuk melakukan proses analisis sentimen, kata populer dan pohon tagar akan mengambil data yang telah tersimpan.



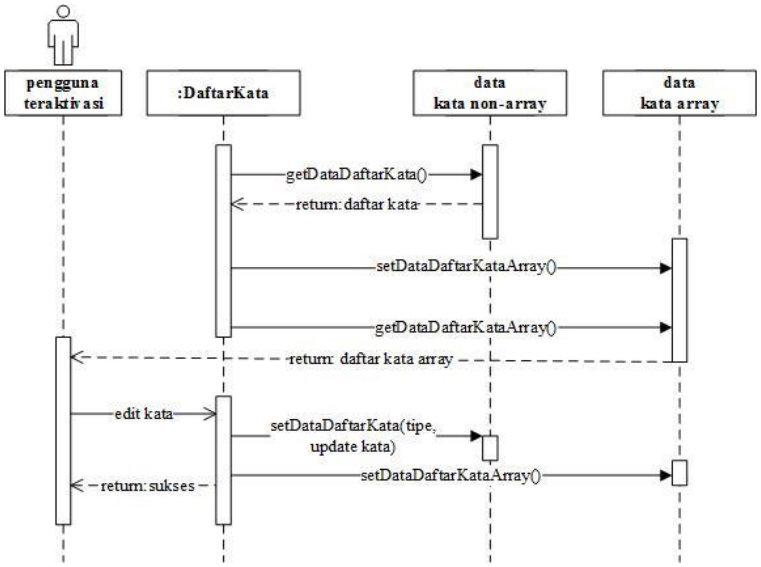
Gambar 3.10 *Sequence diagram* untuk melakukan pencarian kicauan berdasarkan tagar untuk pengguna teraktivasi

Gambar 3.11 merupakan *sequence diagram* untuk melakukan *reset* atau menghapus semua data hasil pencaraian kicauan yang tersimpan. Fitur ini hanya dapat dilakukan oleh pengguna yang telah teraktivasi



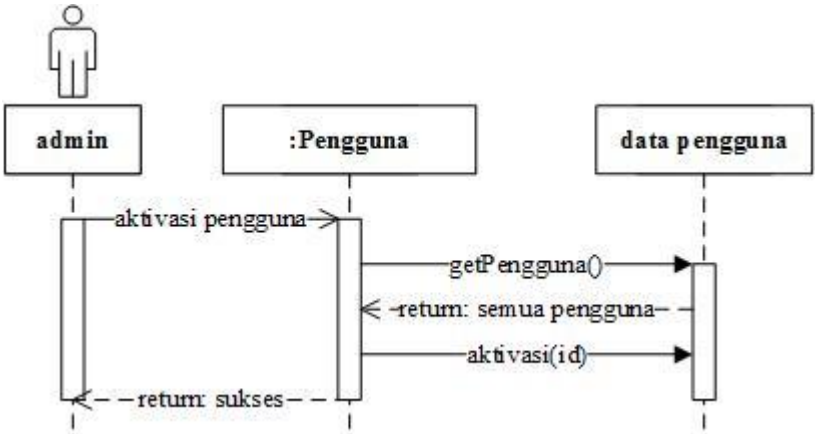
Gambar 3.11 *Sequence diagram* untuk melakukan hapus data kicauan

Gambar 3.12 merupakan *sequence diagram* untuk melakukan pengelolaan data daftar kata oleh pengguna teraktivasi. Daftar kata yang tersimpan berupa daftar kata non-*array* dan daftar kata berupa *array*. Penyimpanan daftar kata non-*array* digunakan untuk memudahkan admin melakukan pengelolaan daftar kata tanpa harus masuk ke dalam aplikasi.



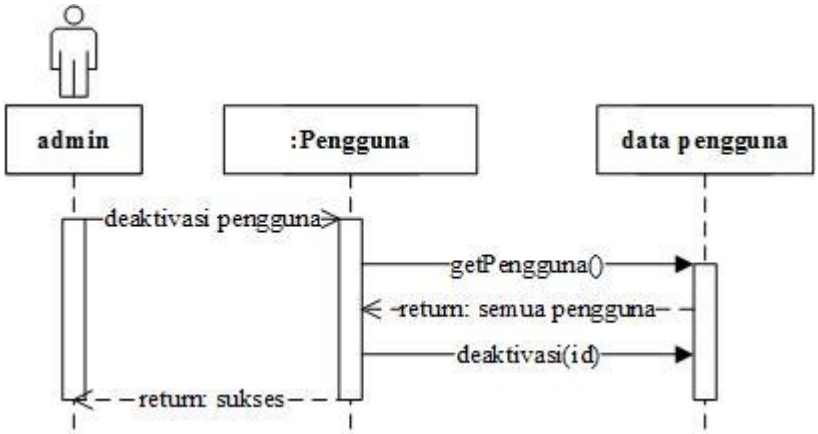
Gambar 3.12 *Sequence diagram* untuk menghapus, menambah dan mengubah daftar kata

Gambar 3.13 merupakan *sequence diagram* untuk melakukan aktivasi pengguna oleh admin, di mana melakukan *load* semua pengguna dan menggunakan *id* pengguna sebagai parameter dalam melakukan aktivasi.



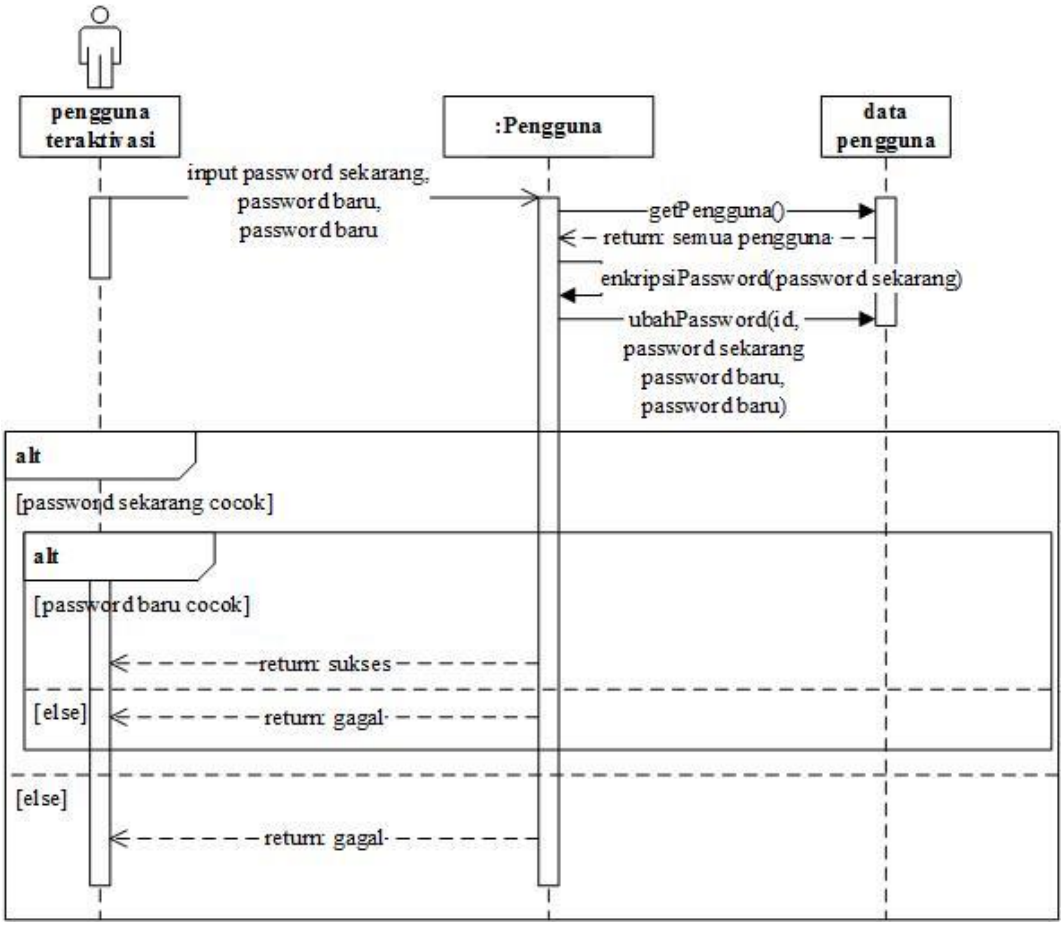
Gambar 3.13 *Sequence diagram* untuk melakukan aktivasi pengguna

Gambar 3.14 merupakan *sequence diagram* untuk melakukan deaktivasi pengguna oleh admin, di mana melakukan load semua pengguna dan menggunakan *id* pengguna sebagai parameter untuk melakukan deaktivasi.



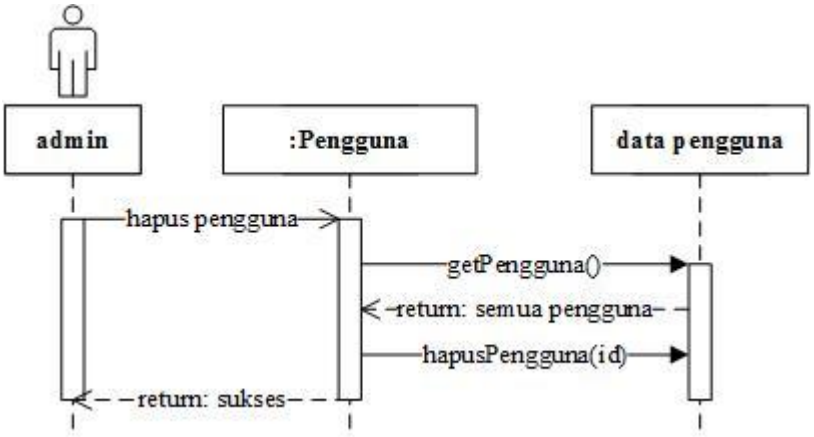
Gambar 3.14 *Sequence diagram* untuk melakukan deaktivasi pengguna

Gambar 3.15 merupakan *sequence diagram* untuk melakukan ubah *password*, di mana pengguna melakukan *input* password lama yang akan diganti dan melakukan *input* password baru sebanyak dua kali untuk dilakukan pencocokan agar meghindari kesalahan pengetikan dalam penggantian *password* baru.



Gambar 3.15 *Sequence diagram* untuk melakukan ubah *password* pengguna teraktivasi

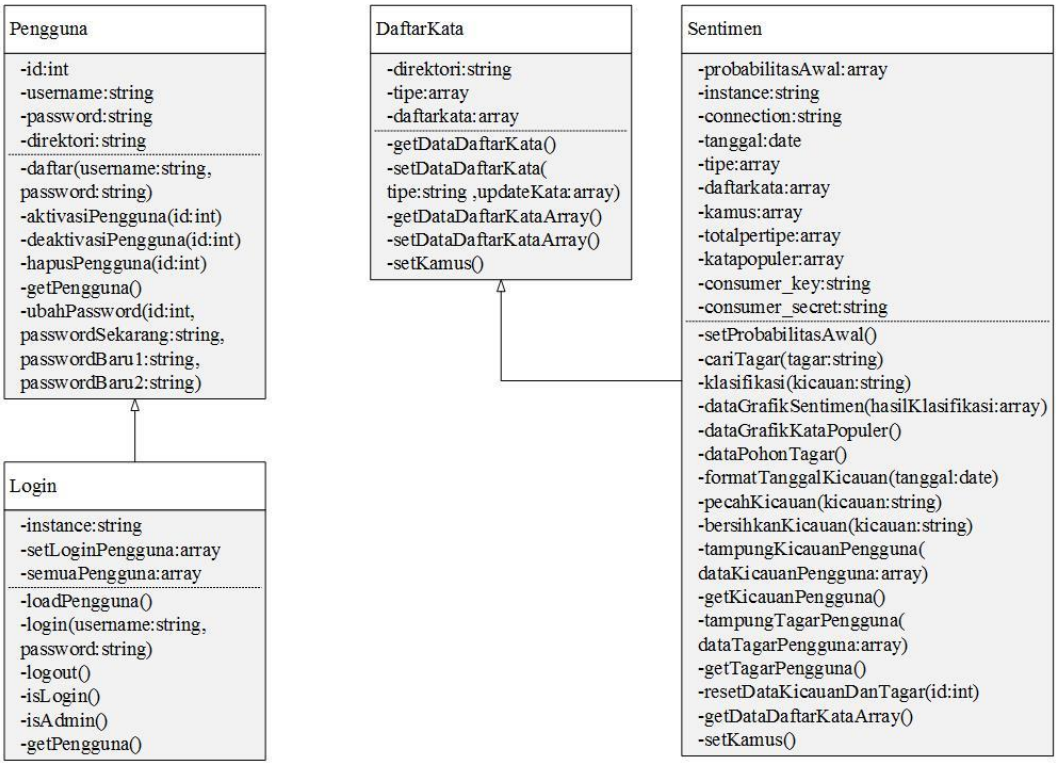
Gambar 3.16 merupakan *sequence diagram* untuk menghapus pengguna yang telah terdaftar, yaitu dengan memanfatkan *id* pengguna sebagai parameter saat menghapus.



Gambar 3.16 *Sequence diagram* untuk melakukan hapus pengguna

3.8.3 Class Diagram

Class diagram digunakan untuk menggambarkan kelas-kelas yang dibentuk dalam sebuah sistem serta menjelaskan bagaimana hubungan antara satu kelas dengan kelas yang lainnya. *Class diagram* juga menjelaskan struktur dan detail tiap kelas di dalam model desain dari suatu sistem. Gambar 3.17 adalah *class diagram* yang digunakan untuk menggambarkan strukur kelas-kelas yang terdapat dalam aplikasi ini.



Gambar 3.17 *Class diagram* untuk pewarisan kelas

Berikut penjelasan masing-masing *method* tiap kelas dari *class diagram* gambar 3.17:

a. Kelas Pengguna

1) daftar(*username*:*string*, *password*:*string*), digunakan untuk menangani pendaftaran pengguna baru. *Input* berupa *username* dan *password*, namun apabila *username* sudah ada maka proses pendaftaran tidak dilanjutkan.
2) aktivasiPengguna(*id*:*int*), digunakan untuk melakukan aktivasi pengguna oleh admin. Parameter berupa id pengguna yang akan diaktivasi.

3) deaktivasiPengguna(*id*:*int*), digunakan untuk deaktivasi pengguna oleh admin. Parameter berupa *id* pengguna yang akan dideaktivasi.
4) hapusPengguna(*id*:*int*), digunakan untuk menghapus penggun oleh admin. Parameter berupa *id* pengguna yang akan dihapus.
5) getPengguna(), digunakan untuk mengambil daftar pengguna yang tersimpan. Keluaran berupa *array* daftar semua pengguna.
6) ubahPassword(*id*:*int*, passwordSekarang:*string*, passwordBaru1:*string*, passwordBaru2:*string*), digunakan untuk menangani pengguna teraktivasi melakukan perubahan *password*. Pada mulanya dilakukan pencocokan dengan *password* yang sekarang pengguna gunakan, kemudian dilakukan pencocokan *password* yang pengguna akan gunakan, proses akan gagal apabila salah satu pencocokan bernilai salah.

b. Kelas Login

1) loadPengguna(), digunakan untuk melakukan memuat semua pengguna untuk pencocokan *username* dan *password* saat proses *login*. Keluaran *method* ini berupa *array* daftar semua pengguna.
2) login(*username*:*string*, *password*:*string*), digunakan untuk menangani proses *login* pengguna. Proses yang dilakukan berupa pencocokan *username* dan *password* dalam *array* keluaran loadPengguna(), keluaran dari *method* ini berupa *id* pengguna apabila berhasil melakukan *login*.
3) logout(), digunakan untuk proses *logout*. Proses ini memberikan nilai *null* pada *session* yang terbentuk dan kemudian dilakukan *destroy* pada *session* yang terbentuk.
4) isLogin(), digunakan untuk pengecekan pengguna sudah *login* atau belum pada *session* yang telah terbentuk.
5) isAdmin(), digunakan untuk pengecekan admin atau bukan, hal ini terkait mengenai akses dan menu khusus untuk admin.
6) getPengguna(), merupakan *method* pewarisan dari kelas Pengguna.

c. Kelas DaftarKata

1) getDataDaftarKata(), digunakan untuk mengambil daftar kata yang tersimpan dalam bentuk non-*array*.
2) setDataDaftarKata(tipe:*string*, updateKata:*array*), digunakan untuk melakukan penyimpanan pada daftar kata non-*array*.
3) getDataDaftarKataArray(), digunakan untuk mengambil daftar kata yang telah tersimpan dalam bentuk *array*.
4) setDataDaftarKataArray(), digunakan untuk menyimpan daftar kata dalam bentuk *array*.
5) setKamus(), digunakan untuk membentuk daftar kata sesuai dengan kategorinya dalam bentuk *array*.

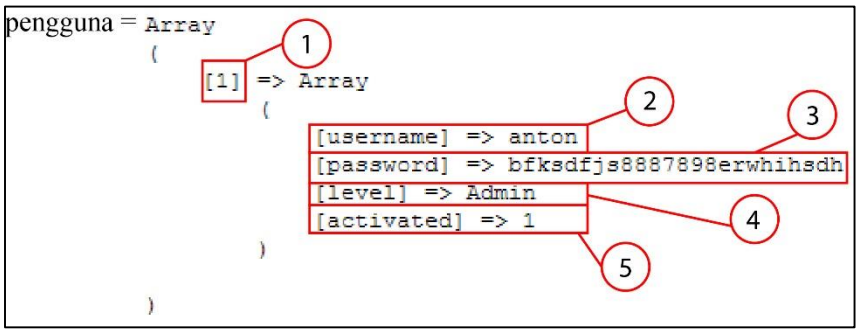
d. Kelas Sentimen

1) setProbabilitasAwal(), digunakan untuk menghitung probabilitas awal kategori positif, negatif dan netral berdasarkan jumlah daftar kata tiap kategori. Keluaran berupa probabilitas tiap kategori yang ditampung dalam *array*.
2) cariTagar(tagar:*string*), digunakan untuk melakukan pencarian tagar berdasarkan masukan pengguna. Untuk pengguna teraktivasi maka hasil pencarian akan disimpan dan dilanjutkan pemrosesan analsis sentimen, kata populer dan pohon tagar berdasarkan data yang tersimpan. Untuk pengguna belum teraktivasi, menghasilkan keluaran berupa kicauan yang kemudian dilakukan pemrosesan analisis sentimen, kata populer dan pohon tagar.
3) klasifikasi(kicauan:*string*), digunakan untuk melakukan klasifikasi kicauan dalam kategori positif, negatif atau netral, dimana keluaran berupa salah satu kategori tersebut.
4) dataGrafikSentimen(hasilKlasifikasi:*array*), digunakan untuk mengolah data hasil klasifikasi yang akan digunakan dalam grafik sentimen.
5) dataGrafikKataPopuler(), digunakan untuk mengolah data kata populer yang akan digunakan dalam grafik kata populer.

6) dataPohonTagar(), digunakan untuk mengolah data tagar yang kemudian data yang akan digunakan dalam presentasi pohon tagar.
7) formatTanggalKicauan(tanggal:*date*), *method* ini digunakan mengubah format tanggal kapan kicauan tersebut di *posting* menjadi lebih sederhana waktu disimpan.
8) pecahKicauan(kicauan:*string*), digunakan untuk memecah kicauan menjadi per kata. Hal ini dilakukan untuk mempermudah proses *string matching* berdasarkan daftar kata.
9) bersihkanKicauan(kicauan:*string*), digunakan untuk membersihkan kicauan dari kata yang kurang lazim dalam bahasa indonesia, seperti menghilangkan angka yang digunkan sebagai pengganti beberapa huruf tertentu, menghilangkan simbol atau tanda baca, mengubah simbol huruf yang tidak lazim digunakan dalam bahasa indonesia menjadi yang seharusnya dan menghilangkan huruf berulang dalam kata.
10) tampungKicauanPengguna(dataKicauanPengguna:*array*), digunakan menyimpan kicauan hasil pencarian untuk pengguan teraktivasi.
11) getKicauanPengguna(), digunakan untuk mengambil data kicauan pengguna teraktivasi yang tersimpan.
12) tampungTagarPengguna(dataTagarPengguna:*array*), digunakan untuk menyimpan tagar hasil pencarian oleh pengguna teraktivasi.
13) getTagarPengguna(), digunakan untuk mengambil data tagar pengguna teraktivasi yang tersimpan.
14) resetDataKicauanDanTagar(id:*int*), digunakan untuk menghapus semua data kicauan dann tagar yang telah tersimpan. Parameter *id* didapatkan dari *session* pengguna teraktivasi saat melakukan *login*.
15) getDataDaftarKataArray(), merupakan *method* hasil pewarisan dengan kelas DaftarKata.
16) setKamus(), merupakan *method* hasil pewarisan dengan kelas DaftarKata.

3.8.4 Perancangan *Array*

Karena aplikasi tidak menggunakan *database management system* (DBMS) terkait efiensi dan efektivitas program, maka alternatif yang digunakan untuk menyimpan data adalah dengan menyimpan data tersebut dalam bentuk *array* yang telah ter-*serialize*. Dalam bahasa pemrograman PHP, fungsi dari *serialize* itu sendiri adalah untuk menyimpan data tanpa harus kehilangan struktur atau jenis data tersebut. Gambar 3.18 merupakan perancangan *array* yang digunakan untuk penyimpanan data.

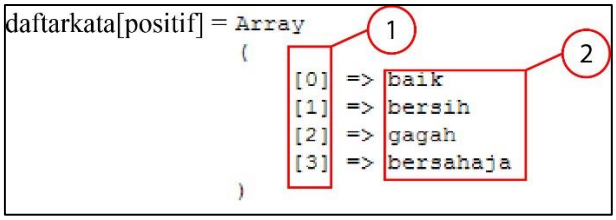


Keterangan:

1. Id pengguna
2. *Username* pengguna
3. *Password* pengguna yang telah terenkripsi
4. Level pengguna, level berisi “Admin” untuk admin, dan “Pengguna” untuk pengguna biasa
5. Status aktivasi pengguna, berisi “1” apabila sudah teraktivasi, dan “0” apabila belum teraktivasi

Gambar 3.18 *Array* data pengguna terdaftar

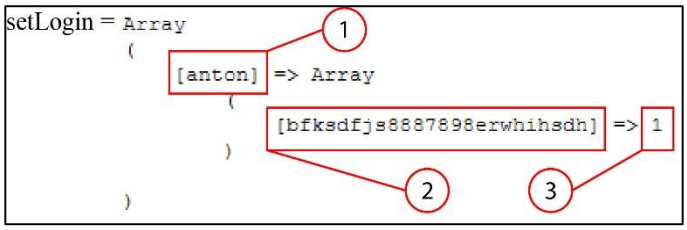
Gambar 3.19 merupakan rancangan untuk penyimpanan daftar kata. Daftar kata disimpan dalam setiap *file* sesuai kategorinya yaitu, positif, negatif, netral, diabaikan dan awalan.



Keterangan:
1. Indeks *array default* apabila array terbentuk
2. Isi daftar kata yang disimpan sesuai kategorinya

Gambar 3.19 Contoh *array* daftar kata positif

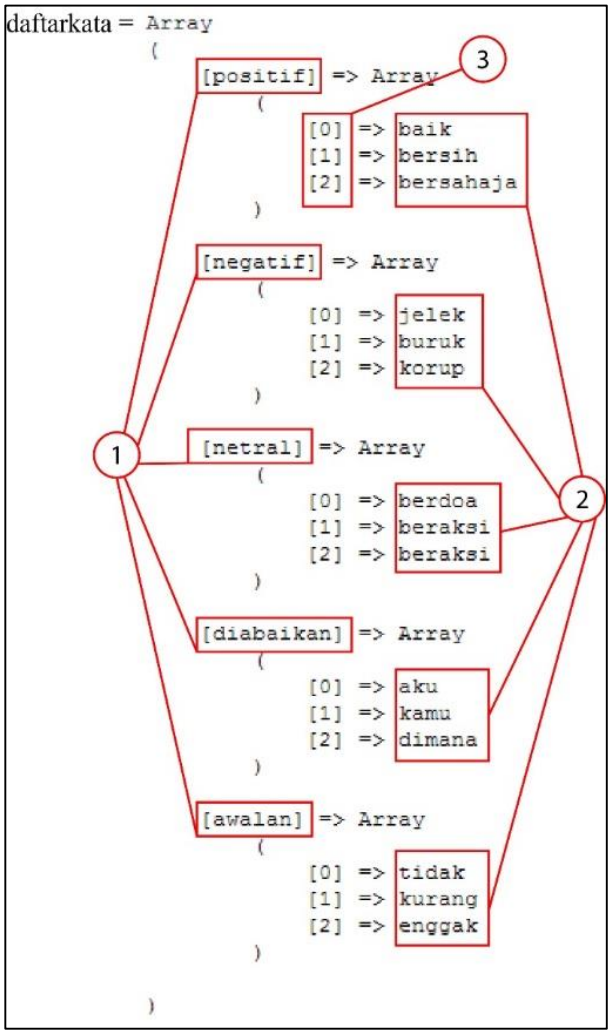
Selain rancangan rancangan *array* diatas, ada beberapa rancangan *array* yang digukanan untuk menunjang proses aplikasi, meliputi proses untuk *login* dan analisis sentimen. Gambar 3.20 merupakan rancangan *array* untuk proses *login*, dimana terdapat *array* dua dimensi terdiri dari indeks pertama berisi *username* pengguna dan indeks kedua berisi *password* pengguna yang telah terenkripsi. Apabila sukses melakukan *login* maka akan menghasilkan *Id* pengguna, apabila gagal *login* maka tidak akan memberikan hasil atau keluaran apapun.



Keterangan:
1. *Username* pengguna
2. *Password* pengguna yang telah terenkripsi
3. *Id* pengguna

Gambar 3.20 *Array* untuk proses *login*

Untuk proses analisis sentimen, hal pertama yang dilakukan adalah menghitung probabilitas awal untuk setiap kategori. Probabilitas awal didapatkan berdasarkan jumlah kata dalam daftar kata tiap kategori, hal ini mengacu pada rumus nomor 3.1, di mana rancangan *array* untuk daftar kata dapat dilihat pada gambar 3.21.

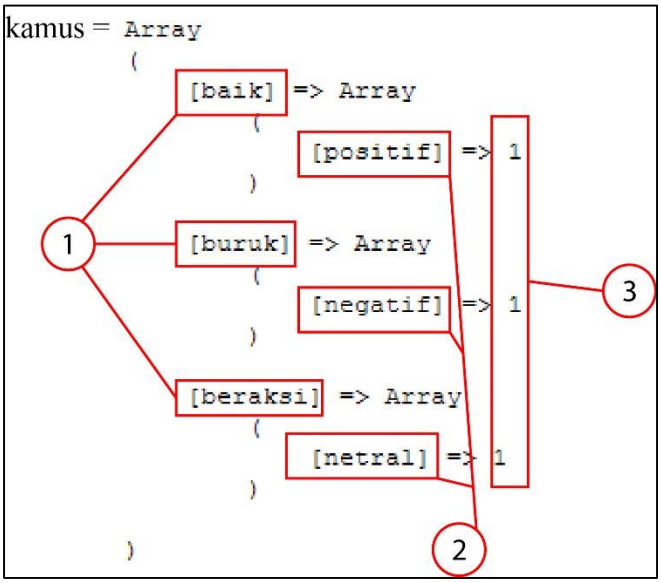


Gambar 3.21 *Array* untuk proses analisis sentimen

Keterangan:

1. Kategori daftar kata
2. Isi daftar kata sesuai kategori
3. Indeks *default* apabila *array* terbentuk

Untuk penghitungan probabilitas awal hanya menggunakan daftar kata kategori positif, negatif dan netral, sedangkan daftar kata awalan dan diabaikan akan digunakan dalam proses klasifikasi kicauan. Proses klasifikasi kicauan memerlukan sebuah kamus yang digunakan dalam proses *string matching*, apabila kata tersebut cocok dengan salah satu kata dalam kategori maka akan menghasilkan nilai satu, sedangkan apabila tidak ditemukan kecocokan maka tidak menghasilkan nilai apapun. Gambar 3.22 merupakan rancangan *array* yang digunakan untuk *string matching*.



Keterangan:

1. Kata dalam setiap kategori
2. Kategori daftar kata
3. Nilai keluaran apabila terjadi kecocokan kata

Gambar 3.22 *Array* untuk *string matching*

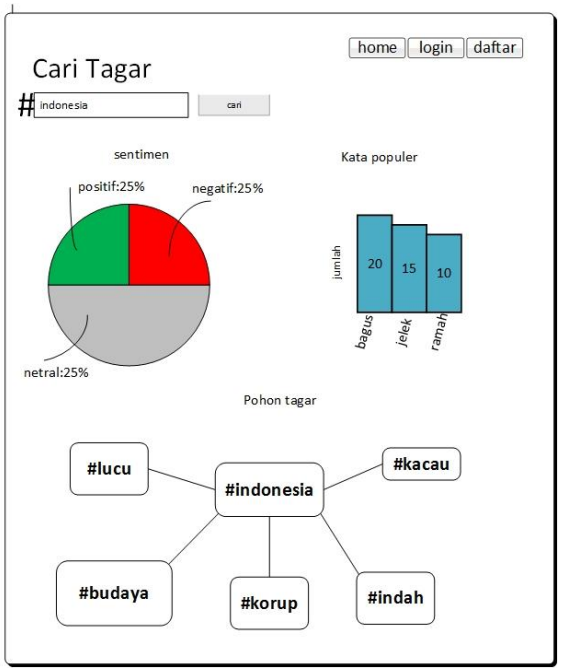
3.8.5 Perancangan *Interface*

Gambar 3.23 merupakan halaman yang pertama kali dibuka ketika mengakses aplikasi ini.



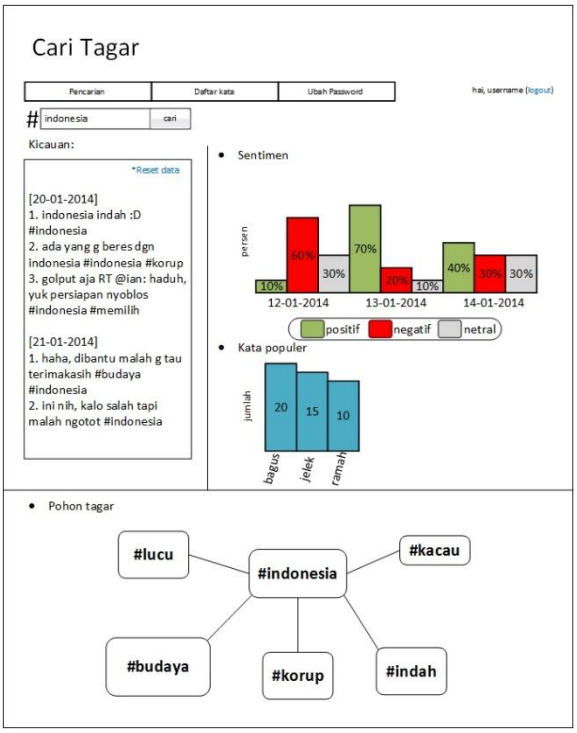
Gambar 3.23 Halaman muka

Gambar 3.24 menampilkan hasil analisis kicauan yang telah dicari yaitu berupa sentimen, pohon tagar dan kata populer untuk pengguna tidak teraktivasi.



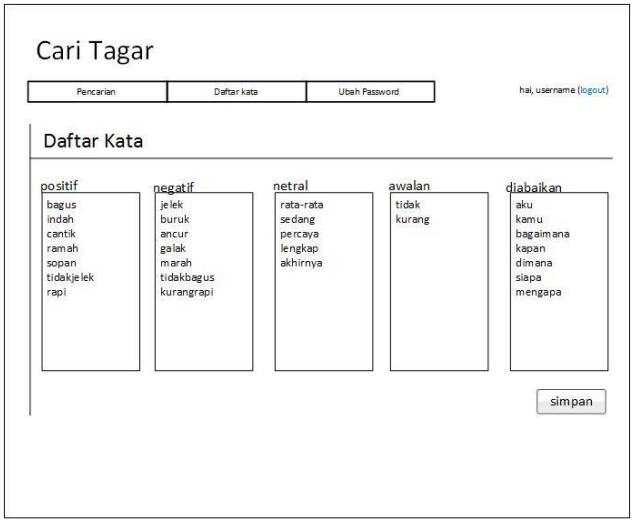
Gambar 3.24 Hasil pencarian untuk pengguna tidak teraktivasi

Gambar 3.25 merupakan rancangan halaman untuk pengguna yang telah terdaftar dan teraktivasi setelah melakukan *login*.



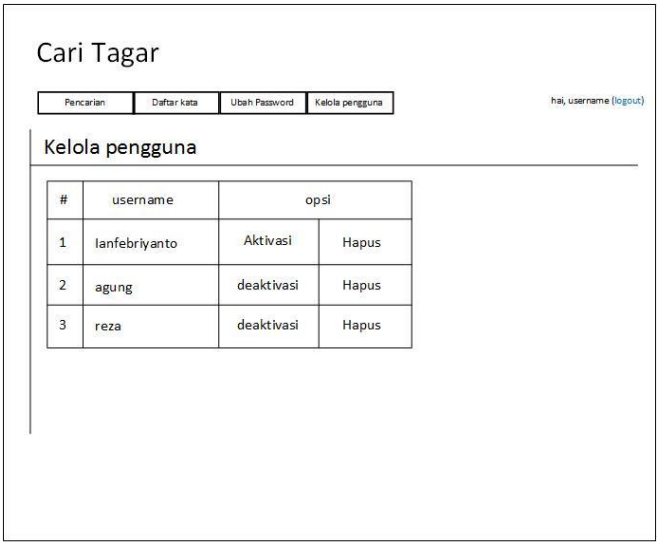
Gambar 3.25 Halaman muka pengguna teraktivasi

Gambar 3.26 merupakan halaman untuk mengelola daftar kata yang digunakan untuk analisis sentimen.



Gambar 3.26 Halaman daftar kata

Gambar 3.27 merupakan halaman untuk admin untuk mengelola pengguna yang terdaftar, meliputi melakukan aktivasi atau deaktivasi akun penguna terdaftar ataupun menghapus pengguna terdaftar.



Gambar 3.27 Halaman kelola pengguna

BAB IV
HASIL DAN PEMBAHASAN

4.1 Batasan Implementasi

Aplikasi ini memilki beberapa batasan, yaitu:

1. Kicauan hasil pencarian berdasarkan tagar hanya menggunakan bahasa Indonesia.
2. Secara *default* jumlah maksimal hasil kicauan dalam sekali pencarian adalah 100, namun sering kali hasil pencarian kurang dari 100.
3. Pengguna teraktivasi hanya dapat menyimpan kicauan dalam rentang tujuh hari, apabila melakukan pencarian di hari ke-8 maka kicauan di tanggal terakhir dalam penyimpanan data kicauan akan dihapus.
4. Jika hasil klasifikasi untuk sebuah kicauan memiliki nilai positif, negatif dan netral memiliki nilai yang sama, maka kicauan tersebut akan diabaikan.

4.2 Implementasi

Pada tahap implementasi ini akan dibagi menjadi dua bagian yaitu, implementasi program dan implementasi antarmuka.

4.2.1 Implementasi Program

Untuk menjaga fokus pembahasan, bagian ini akan menjelaskan bagaimana proses utama dalam aplikasi ini, yaitu mencakup melakukan pencarian kicauan berdasarkan tagar beserta hasil sentimen, kata populer dan pohon tagar. Proses tersebut antara lain:

1. Proses pencarian kicauan berdasarkan tagar
2. Proses menampung kicauan hasil pencarian
3. Proses menampung tagar hasil pencarian
4. Proses mengambil kicauan yang tersimpan
5. Proses menghitung probabilitas awal
6. Proses pembersihan kicauan

7. Proses analisis sentimen
8. Proses pembentukan data kata populer
9. Proses pembentukan data pohon tagar
10. Penyimpanan data pengguna dan data daftar kata berupa *array* yang telah dilakukan *serialize* ke dalam *file* penyimpanan

4.2.1.1 Proses Pencarian Kicauan Berdasarkan Tagar

Pertama adalah membuat koneksi dengan API Twitter, yang dibutuhkan adalah berupa *consumer key*, *consumer key secret*, *access token* dan *access token secret* yang telah didapatkan setelah mendaftarkan aplikasi pada Twitter. Namun karena aplikasi hanya digunakan untuk melakukan pencarian berdasarkan tagar maka cukup dengan mendeklarasikan *consumer key* dan *consumer key secret* saja pada kelas sentimen yang dapat dijelaskan dalam gambar 4.1.

```
private $consumer_key = 'Lm6w53PPn7mzoihr185BmA';
private $consumer_secret =
QkSt5sxNZaHSqc3v6nIIAPoZ4ZwUQzkMf4KvMGxHXQ';
```

Gambar 4.1 Deklarasi *consumer key* dan *consumer secret*

Kemudian melakuakn instansiasi *library* yang digunakan untuk melakukan koneksi dengan API Twiter yang dideklarasikan dalam konstruktor kelas sentimen yang dapat dilihat pada gambar 4.2.

```
$this->connection = new TwitterOAuth($this-
>consumer key,$this->consumer secret);
```

Gambar 4.2 *Script* instansiasi TwitterOauth

Hasil koneksi tersebut akan digunakan dalam proses pencarian tagar dalam kelas sentimen. Gambar 4.3 merupakan *script* untuk melakukan pencarian kicauan berdasarkan tagar. Terdapat pengecekan status aktivasi pengguna, apabila pengguna dengan status teraktivasi, kicauan dan tagar yang tercantum dalam kicauan hasil pencarian akan disimpan, sedangkan untuk pengguna tidak teraktivasi, kicauan dan tagar yang tercantum dalam kicauan hasil pencarian akan langsung dilakukan proses analisis.

```
function cariTagar($tag){
$aktivasi = $this->instance->session->userdata('activated');
if($tag != ""){
$tagar = "#";
$tagar .= $tag;
$tagar = strtolower($tagar);
$kicauan   =   $this->connection->get('search/tweets',   array('q'   =>
$tagar,'count'=>100,'result_type' => 'mixed','lang' => 'id' ));
if ($aktivasi) {
       $tanggal = $this->tanggal;
       $id = $this->instance->session->userdata('id');
       $dataTagarPengguna[$id]['input'] = $tagar;
       if(!empty($kicauan->statuses)){
       foreach ($kicauan->statuses as $status) {
              $tgl = $this->formatTanggalKicauan($status->created_at);
              if ($tanggal == $tgl) {
              $dataKicauanPengguna[$id][$tgl][] = $status->text;
              foreach ($status->entities->hashtags as $tagar) {
              $dataTagarPengguna[$id]['hasil'][$tgl][]   =   "#".$tagar-
>text;}}}
$this->tampungKicauanPengguna($dataKicauanPengguna);
$this->tampungTagarPengguna($dataTagarPengguna);
       }else{return false;}
}else{
       $this->tagar['input'] = $tagar;
       if(!empty($kicauan->statuses)){
       foreach ($kicauan->statuses as $status) {
       $dataKicauan[] = $status->text;
       foreach ($status->entities->hashtags as $tagar) {
       $this->tagar['hasil'][] = "#".$tagar->text;}}
return $dataKicauan;
       }else{return false;}
}
}
}
```

Gambar 4.3 *Script* untuk proses pencarian kicauan berdasarkan tagar

4.2.1.2 Proses Menampung Kicauan Hasil Pencarian

Gambar 4.4 merupakan *script* yang digunakan untuk menampung kicauan hasil pencarian berdasarkan tagar untuk pengguna teraktivasi. Data kicauan yang akan disimpan hanyalah kicauan hasil pencarian selama tujuh hari.

```
public function tampungKicauanPengguna($dataKicauanPengguna){
$dir = $this->direktori['general'] . "dataKicauanPengguna.php";
if(file_exists($dir)){
if(isset($dataKicauanPengguna) && $dataKicauanPengguna!=''){
   $backupData = file_get_contents($dir);
   $temp = unserialize($backupData);
foreach ($dataKicauanPengguna as $id => $value) {
  foreach ($value as $tanggal => $val) {
    foreach ($val as $isi) {
      $kicauan[] = $isi;
}
if (!empty($temp[$id][$tanggal])) {
  $temp[$id][$tanggal]                                                    =
array_merge($temp[$id][$tanggal],$kicauan);
  $temp[$id][$tanggal] = array_unique($temp[$id][$tanggal]);
}else{
  $temp[$id][$tanggal] = array_unique($kicauan);
}
  $temp[$id] = array_slice($temp[$id], -7,7, true);
}
}
$data = serialize($temp);
file_put_contents($dir, $data);
}
}else{
return 0;
}
}
```

Gambar 4.4 *Script* Proses menampung kicauan

4.2.1.3 Proses Menampung Tagar Hasil Pencarian

Tagar yang tercantum dalam kicauan hasil pencarian akan disimpan, hal ini untuk memudahkan dalam mengolah pohon tagar untuk pengguna teraktivasi. Gambar 4.5 merupakan script yang digunakan untuk menampung tagar yang tercantum dalam kicauan hasil pencarian berdasarkan tagar oleh pengguna teraktivasi. Sama halnya dengan menampung data kicauan, tagar yang akan disimpan hanyalah hasil pencarian selama tujuh hari. Untuk pengguna tidak teraktivasi maka tagar akan ditampung dalam variabel global dalam bentuk *array*.

```
public function tampungTagarPengguna($dataTagarPengguna){
$dir = $this->direktori['general'] . "dataTagarPengguna.php";
if(file_exists($dir)){
  if(isset($dataTagarPengguna) && $dataTagarPengguna!=''){
    $backupData = file_get_contents($dir);
    $temp = unserialize($backupData);
    foreach ($dataTagarPengguna as $id => $value) {
      $temp[$id]['input'] = $dataTagarPengguna[$id]['input'];
foreach ($dataTagarPengguna[$id]['hasil'] as $tanggal => $hasil) {
   foreach ($hasil as $tag) {
    $temp[$id]['hasil'][$tanggal][] = $tag;
   }
}
$temp[$id]['hasil'] = array_slice($temp[$id]['hasil'], -7,7, true);
}
$data = serialize($temp);
file_put_contents($dir, $data);
}
}else{
return 0;
}
}
```

Gambar 4.5 *Script* untuk proses menyimpan tagar hasil pencarian oleh pengguna teraktivasi

4.2.1.4 Proses Mengambil Kicauan yang Tersimpan

Sebelum melakukan analisis sentimen atau klasifikasi kicauan ke dalam kategori positif, negatif atau netral, untuk pengguna teraktivasi maka akan mengambil kicauan hasil pencarian yang telah tersimpan, *script* dapat dilihat pada gambar 4.6.

```
public function getKicauanPengguna(){
$dir = $this->direktori['general']."dataKicauanPengguna.php";
$id = $this->instance->session->userdata('id');
if(file_exists($dir)){
$kicauan = file_get_contents($dir);
$kicauan = unserialize($kicauan);
return $data = (isset($kicauan[$id])) ? $kicauan[$id] : null ;
}else{
return 0;}
}
```

Gambar 4.6 *Script* untuk proses mengambil kicauan pengguna yang tersimpan

4.2.1.5 Proses Menghitung Probabilitas Awal

Hal pertama yang perlu dilakukan sebelum melakukan proses analisis sentimen adalah menghitung probabilitas awal dari masing-masing kategori positif, negatif dan netral. Probabilitas awal dihitung berdasarkan jumlah kata dalam daftar kata masing-masing kategori positif, negatif dan netral, di mana *script* dapat dilihat pada gambar 4.7.

```
public function setProbabilitasAwal(){
$totaldata = 0;
$kata = $this->daftarKata;
foreach ($this->tipe as $tipe) {
   $this->totalPerTipe[$tipe] = sizeof($kata[$tipe]);
    $totaldata += $this->totalPerTipe[$tipe];}
foreach ($this->tipe as $tipe) {
  $this->prior[$tipe]    =    $this->totalPerTipe[$tipe]    /
$totaldata;}}
```

Gambar 4.7 *Script* untuk menghitung probabilitas awal

Daftar kata yang digunakan dalam *script* pada gambar 4.7, diperoleh dari kelas daftar kata yang telah *inheritance* dengan kelas sentimen.

4.2.1.6 Proses Pembersihan Kicauan

Gambar 4.8 merupakan *script* untuk pembersihan kicauan. Hal pertama yang dilakukan adalah dengan mengganti karakter-karakter kurang lazim digunakan dalam bahasa Indonesia, dan mengganti angka-angka yang digunakan untuk mengganti beberapa karakter dalam bahasa Indonesia. Kemudian menggunakan regex (*regular expresion*) untuk menghilangkan karakter yang berulang dalam kata. Apabila kata tersebut terdapat dalam daftar kata awalan, positif, negatif dan netral maka tidak akan di proses untuk menghilangkan perulangan karakter dalam kata.

```
public function bersihkanKicauan($kicauan){
$karakter =
/* A */      chr(192) . chr(193) . chr(194) . chr(195) . chr(196) .
chr(197) .
/* a */      chr(224) . chr(225) . chr(226) . chr(227) . chr(228) .
chr(229) .
/* O */      chr(210) . chr(211) . chr(212) . chr(213) . chr(214) .
chr(216) .
/* o */      chr(240) . chr(242) . chr(243) . chr(244) . chr(245) .
chr(246) . chr(248) .
/* E */      chr(200) . chr(201) . chr(202) . chr(203) .
/* e */      chr(232) . chr(233) . chr(234) . chr(235) .
/* C */      chr(199) .
/* c */      chr(231) .
/* I */      chr(204) . chr(205) . chr(206) . chr(207) .
/* i */      chr(236) . chr(237) . chr(238) . chr(239) .
/* U */      chr(217) . chr(218) . chr(219) . chr(220) .
/* u */      chr(249) . chr(250) . chr(251) . chr(252) .
/* y */      chr(255) .
/* N */            chr(209) .
/* n */            chr(241) .
/*angka*/   "1345690" .
/*simbol*/   "!.,?/"."@#$%^&*()-_+={}[]<>;:" .
/*\*/        chr(92) .
/*' "*/      chr(96) . chr(145) . chr(146) . chr(147) . chr(148);
$karakterbersih     =     strtolower(strtr($kicauan,     $karakter,
'AAAAAAaaaaaaOOOOOOooooooEEEEeeeeCcIIIIiiiiUUUUuuuuyNnieasggo
'));
$kata = explode(" ", $karakterbersih);
foreach ($kata as $pecah) {
if(!in_array($pecah, $this->daftarKata['awalan']) &&
   !in_array($pecah, $this->daftarKata['positif']) &&
   !in_array($pecah, $this->daftarKata['negatif']) &&
   !in_array($pecah, $this->daftarKata['netral'])){
$kataPerulangan[] = preg_replace("/(.)\\1+/", "$1", $pecah);
}else{$kataPerulangan[] = $pecah;}}
$bersih = implode(" ", $kataPerulangan);
return $bersih;
}
```

Gambar 4.8 *Script* untuk proses pembersihan kicauan

4.2.1.7 Proses Pemecahan Kicauan

Gambar 4.9 merupakan *script* untuk melakukan pemecahan kicauan menjadi per kata. Pertama dilakukan pengecekan apakah daftar kata awalan terdapat dalam kicauan tersebut. Apabila ada, maka spasi antara kata awalan dengan kata yang ada dibelakangnya akan dihilangkan. Hal ini guna untuk mengatasi apabila terdapat frasa yang memiliki makna positif atau negatif.

```
public function pecahKicauan($kicauan){
$kicauanBersih = " ".$this->bersihkanKicauan($kicauan);
foreach ($this->daftarKata['awalan'] as $awalan) {
if(strpos($kicauanBersih," ".$awalan." ")){
$kicauanBersih=str_replace(" ".$awalan." ", " ".$awalan,
$kicauanBersih);}}
$pecah = explode(" ",$kicauanBersih);
$pecah = array_filter($pecah);
return $pecah;
}
```

Gambar 4.9 *Script* untuk proses pemecahan kicauan

4.2.1.8 Proses Analisis Sentimen

Proses analisis sentimen dilakukan dengan mencocokkan tiap kata kicauan yang telah dipecah dengan daftar kata positif, negatif, atau netral yang apabila cocok maka akan bernilai 1, jumlah nilai ini yang akan terus bertambah menurut kategorinya apabila ditemukan kecocokan pada iterasi berikutnya. Setelah iterasi selesai maka akan dilakukan proses perhitungan kemungkinan tiap kategori dan dikalikan dengan probabilitas awal masing-masing kategori.

Untuk kata populer didapatkan dari proses klasifikasi, di mana kata yang terdapat dalam kicauan dan terdapat dalam daftar kata positif, negatif ataupun netral akan ditampung pada *array* yang menjadi variabel global. *Script* dapat dilihat pada gambar 4.10.

```
public function klasifikasi($kicauan){
$total_skor = 0;
$skor = array();
$pecah2 = $this->pecahKicauan($kicauan);
foreach ($this->tipe as $tipe) {
    $skor[$tipe] = 0;
  foreach ($pecah2 as $pecah) {
    if(!in_array($pecah, $this->daftarKata['diabaikan'])){
       if(isset($this->kamus[$pecah][$tipe])){
          $jum = $this->kamus[$pecah][$tipe];
          $this->katapopuler[] = $pecah;
       } else {
         $jum = 0;
       }
  $skor[$tipe] += $jum;
}}}
foreach ($this->tipe as $tipe) {
   $skor[$tipe] = $skor[$tipe] / $this->totalPerTipe[$tipe];
   $skor[$tipe] = $this->prior[$tipe] * $skor[$tipe];
}
arsort($skor);
if (count(array_unique($skor))==1) {
   return false;
}else{
   return key($skor);
}
}
```

Gambar 4.10 *Script* untuk proses klasifikasi

Fungsi klasifikasi ini akan memberikan keluaran berupa kategori positif, negatif atau netral yang merupakan hasil dari perhitungan diatas berdasarkan nilai tertinggi tiap kategori tersebut. Apabila tiap kategori mempunyai probabilitas akhir yang sama, maka akan memberikan keluaran *false* yang berarti kicauan tersebut tidak memiliki sentimen atau tidak dapat diklasifikasikan.

Script pada gambar 4.11 merupakan proses untuk menampilkan hasil analisis sentimen. Untuk pengguna teraktivasi akan melakukan pengecekan status aktivasi pengguna yang kemudian dilakukan penghitungan jumlah masing-masing kategori. Keluaran dari fungsi ini berupa JSON yang akan digunakan untuk menampilkan grafik sentimen berdasarkan tanggal kapan dilakukannya pencarian, karena pengguna teraktivasi mempunyai fitur di mana dapat melakukan pencarian selama tujuh hari.

```
public function dataGrafikSentimen($hasilKlasifikasi){
$aktivasi = $this->instance->session->userdata('activated');
if (is_array($hasilKlasifikasi)) {
$hasilKlasifikasi = array_filter($hasilKlasifikasi);
}
if ($aktivasi) {
if(!empty($hasilKlasifikasi) && $hasilKlasifikasi!=null){
foreach ($hasilKlasifikasi as $tanggal => $value) {
$total_skor = 0;
$hasilKlasifikasi[$tanggal]                                          =
array_count_values(array_filter($hasilKlasifikasi[$tanggal]));
foreach ($hasilKlasifikasi[$tanggal] as $tipe => $nilai) {
$total_skor += $nilai;
}
foreach ($this->tipe as $tipe) {
if (isset($hasilKlasifikasi[$tanggal][$tipe])) {
$hasilKlasifikasi[$tanggal][$tipe]                                   =
$hasilKlasifikasi[$tanggal][$tipe] / $total_skor * 100;
$hasilKlasifikasi[$tanggal][$tipe]                                   =
number_format($hasilKlasifikasi[$tanggal][$tipe],2);
} else {
$hasilKlasifikasi[$tanggal][$tipe] = number_format(0,2);
}
}
$data['tanggal'][] = $tanggal;
$data['positif'][]  =  (float)  $hasilKlasifikasi[$tanggal]['positif'];

$data['negatif'][] = (float) $hasilKlasifikasi[$tanggal]['negatif'];
$data['netral'][] = (float) $hasilKlasifikasi[$tanggal]['netral'];}
$data['tanggal'] = json_encode($data['tanggal']);
$data['positif'] = json_encode($data['positif']);
$data['negatif'] = json_encode($data['negatif']);
$data['netral']  = json_encode($data['netral']);
}else{
$data['tanggal'] = json_encode(array('tidak ada data'));
$data['positif'] = json_encode(array(0));
$data['negatif'] = json_encode(array(0));
$data['netral']  = json_encode(array(0));}
return $data;
}
```

Gambar 4.11 *Script* untuk proses pembentukan data grafik sentimen pengguna teraktivasi

Pengguna belum teraktivasi mempunyai tampilan grafik sentimen yang berbeda dengan pengguna teraktivasi karena data kicuan yang digunakan merupakan hasil dari satu kali pencarian, untuk itu *script* untuk pembentukan grafik hasil analisis sentimen untuk pengguna belum teraktivasi dapat dilihat pada gambar 4.12.

```
else{
if(!empty($hasilKlasifikasi) && $hasilKlasifikasi!=null){
$total_skor = 0;
$dataGrafik = array_count_values(array_filter($hasilKlasifikasi));
foreach ($dataGrafik as $tipe => $nilai) {
$total_skor += $nilai;}
foreach ($this->tipe as $tipe) {
if(isset($dataGrafik[$tipe])){
$dataGrafik[$tipe] = $dataGrafik[$tipe] / $total_skor * 100;
$dataGrafik[$tipe] = number_format($dataGrafik[$tipe] , 2);
}else{
$dataGrafik[$tipe] = number_format(0,2);}}
foreach ($this->tipe as $tipe) {
if ($tipe == 'negatif') {
$data['grafik'][] = array('name'=>$tipe,
'y'=> (float) $dataGrafik[$tipe],
'sliced'=>true,
'selected'=>true);
}else{
$data['grafik'][] = array('name'=>$tipe,
'y'=>(float) $dataGrafik[$tipe]);}}
$data['grafik'] = json_encode($data['grafik']);
}else{
$data['grafik'][] = array('name'=>'tidak ada data','y'=> 100);
$data['grafik'] = json_encode($data['grafik']);}
return $data;}}
```

Gambar 4.12 *Script* untuk proses pembentukan data grafik sentimen untuk pengguna belum teraktivasi

4.2.1.9 Implementasi Kata Populer

Seperti yang telah dijelaskan, daftar kata populer didapatkan dalam proses klasifikasi yang kemudian ditampung dalam variabel global berbentuk *array*. Kata populer yang diambil hanyalah sepuluh kata dengan frekuensi terbanyak, yang kemudian data tersebut diubah menjadi bentuk JSON sebagai keluarannya yang akan digunakan untuk pembentukan grafik. *Script* pembentukan kata populer dapat dilihat pada gambar 4.13.

```
public function dataGrafikKataPopuler(){
$katapopuler = $this->katapopuler;
$katapopuler = array_count_values($katapopuler);
arsort($katapopuler);
$katapopuler = array_slice($katapopuler, 0,10);
foreach ($katapopuler as $kata=>$jumlah) {
$data['kategori'][] = $kata;
$data['jumlah'][] = $jumlah;}
if (isset($data['jumlah']) && isset($data['kategori'])) {
$data['jumlah'] = json_encode($data['jumlah']);
$data['kategori'] = json_encode($data['kategori']);
}else{$data['kategori'][] = 'tidak ada data';
$data['jumlah'][] = 0;
$data['jumlah'] = json_encode($data['jumlah']);
$data['kategori'] = json_encode($data['kategori']);}
return $data}
```

Gambar 4.13 *Script* pembentukan data kata populer

4.2.1.10 Implementasi Pohon Tagar

Penggalan *script* pada gambar 4.14 merupakan pembentukan pohon tagar untuk pengguna teraktivasi, karena untuk pengguna teraktivasi mengambil tagar yang telah tersimpan. Apabila pengguna belum teraktivasi maka akan mengambil dari tagar yang telah tersimpan dalam variabel global.

```
function dataPohonTagar(){
$aktivasi = $this->instance->session->userdata('activated');
if ($aktivasi) {
$tagar = $this->getTagarPengguna();
if (isset($tagar['input'])) {
$tag['input'] = strtolower($tagar['input']);
}
if (isset($tagar['hasil'])) {
foreach ($tagar['hasil'] as $tanggal => $isi) {
foreach ($isi as $tag2) {
$tag['hasil'][]=$tag2;
}}}
}else{
$tag = $this->tagar;}
```

Gambar 4.14 *Script* untuk proses pengecekan status aktivasi pengguna

Selanjutnya dilakukan pengecekan apakah tagar yang akan diproses ada atau tidak, jika ada, maka seluruh tagar dibuah menjadi huruf kecil, hal ini dilakukan agar tagar yang sama, tidak terlalu bervariasi yang dikarenakan besar kecilnya karakter. Tagar yang akan ditampilkan merupakan sepuluh tagar yang mempunyai nilai terbesar. *Script* pada gambar 4.15 yang digunakan untuk pembentukan pohon tagar.

```
if(isset($tag['hasil'])){
$tag['hasil']                                                    =
array_count_values(array_map('strtolower',$tag['hasil']));
arsort($tag['hasil']);
$frekuensiTagInput = $tag['hasil'][$tag['input']];
unset($tag['hasil'][$tag['input']]);
$tag['hasil'] = array_slice($tag['hasil'], 0,10);
$pohon['tagar'] = $tag['input'];
$i = 0;
$pohon['pohon'][] = array('key'=>0, 'text' => $tag['input'], 'num' =>
'parent');
foreach ($tag['hasil'] as $tag => $nilai) {
$i++;
$pohon['pohon'][] = array('key'=> (int) $i,'text' => $tag, 'num' =>
(int) $nilai);
}
$j = 1;
foreach ($pohon['pohon'] as $index => $val) {
$pohon['link'][] = array('from'=>(int) $j, 'to' => (int) 0);
$j++;
}
$pohon['pohon'] = json_encode($pohon['pohon']);
$pohon['link'] = json_encode($pohon['link']);
return $pohon;
}else{
if (isset($tag['input'])) {
$pohon['tagar'] = $tag['input'];
}else{
$pohon['tagar'] ='';
}
if($aktivasi){
$pohon['pohon'][] = array('key'=>0, 'text' => 'tidak ada data', 'num'
=> 'parent');
}else{
$pohon['pohon'][] = array('key'=>0, 'text' => $tag['input'], 'num' =>
(int) $frekuensiTagInput);}
$pohon['link'][] = array('from'=>(int) 1, 'to' => (int) 1);
$pohon['pohon'] = json_encode($pohon['pohon']);
$pohon['link'] = json_encode($pohon['link']);
return $pohon;
}
}
```

Gambar 4.15 *Script* untuk proses pembentukan data pohon tagar

Keluaran *script* pada gambar 4.15 berupa JSON yang berisi data yang ditampilkan dalam bentuk grafik pada pengguna.

4.2.1.11 Implementasi Penyimpanan Data

Proses penyimpanan data yang dilakukan adalah mengubah *array* sebagai struktur data pengguna, yang kemudian dilakukan *serialize array* tersebut agar tetap menjaga struktur dan jenis data yang digunakan, yang kemudian disimpan dalam sebuah *file* penyimpanan. Gambar 4.16 menjelaskan bagaimana proses penyimpanan data pengguna.

```
public function daftar($username, $password){
$pengguna = $this->semuaPengguna;
foreach ($pengguna as $id => $data) {
$listUsername[] = $data['username'];
}
$username = trim($username);
if(!in_array($username, $listUsername)){
$id = count($this->semuaPengguna);
if($id == 0 || empty($id)){
$id = 1;
}else{
$id = end(array_keys($pengguna)) + 1;
}
$password = $this->enkripsiPassword($password);
$penggunaBaru = array( $id => array('username'=>$username,
                                    'password'=>$password,
                                    'level' => 'Pengguna',
                                    'activated'=>0));
!empty($pengguna) ? $pengguna += $penggunaBaru : $pengguna =
$penggunaBaru;
$pengguna = serialize($pengguna);
file_put_contents($this->direktori, $pengguna);
return true;
}else{
return false;
}
}
```

Gambar 4.16 *Script* penyimpanan data pengguna

Setiap pengguna yang melakukan pendaftaran ke aplikasi, maka data *username* dan *password* akan langsung tersimpan dengan status aktivasi "0" atau "*false*". Gambar 4.17 merupakan hasil penyimpanan data pengguna setelah ter-*serialize.*

```
a:3:{i:1;a:4:{s:8:"username";s:13:"ianfebriyanto";s:8:"password";s:3
2:"3c73a84a238196f5648695010a63cf78";s:5:"level";s:5:"Admin";s:9:"ac
tivated";i:1;}i:2;a:4:{s:8:"username";s:5:"anton";s:8:"password";s:3
2:"3c73a84a238196f5648695010a63cf78";s:5:"level";s:8:"Pengguna";s:9:
"activated";i:1;}i:3;a:4:{s:8:"username";s:4:"andi";s:8:"password";s
:32:"3c73a84a238196f5648695010a63cf78";s:5:"level";s:8:"Pengguna";s:
9:"activated";i:0;}}
```

Gambar 4.17 Data pengguna yang telah ter-*serialize*

Data daftar kata juga disimpan dalam bentuk *array* yang sudah ter-*serialize*, berikut gambar 4.18 bagaimana proses penyimpanan daftar kata.

```
public function setDataDaftarKataArray(){
$kata2 = $this->daftarKata;
foreach ($this->tipe as $tipe) {
$dir = $this->direktori['kamus']."kamus.".$tipe.".php";
                    $kata = serialize(array_filter($kata2[$tipe]));
                    file_put_contents($dir, $kata);}
}
```

Gambar 4.18 *Script* penyimpanan data daftar kata

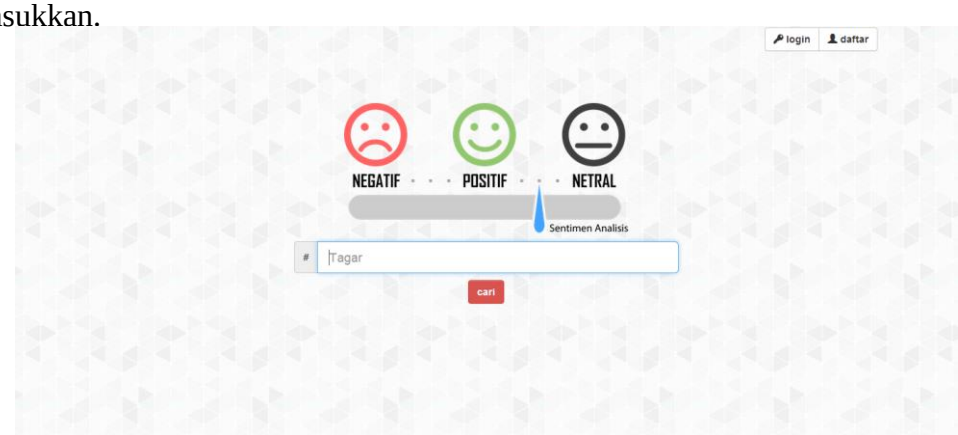
Gambar 4.18 menerangkan setiap daftar kata ditampung dalam *array* kemudian di-*serialize*, lalu disimpan dalam *file* penyimpanan yang berbeda sesuai nama kategorinya. Gambar 4.19 merupakan hasil penyimpanan daftar kata awalan.

```
a:7:{i:0;s:3:"bau";i:1;s:5
:"belum";i:2;s:6:"enggak";
i:3;s:6:"kurang";i:4;s:4:"
ndak";i:5;s:5:"nggak";i:6;
s:5:"tidak";}
```

Gambar 4.19 Data daftar kata awalan yang telah tersimpan

4.2.2 Implementasi Antarmuka

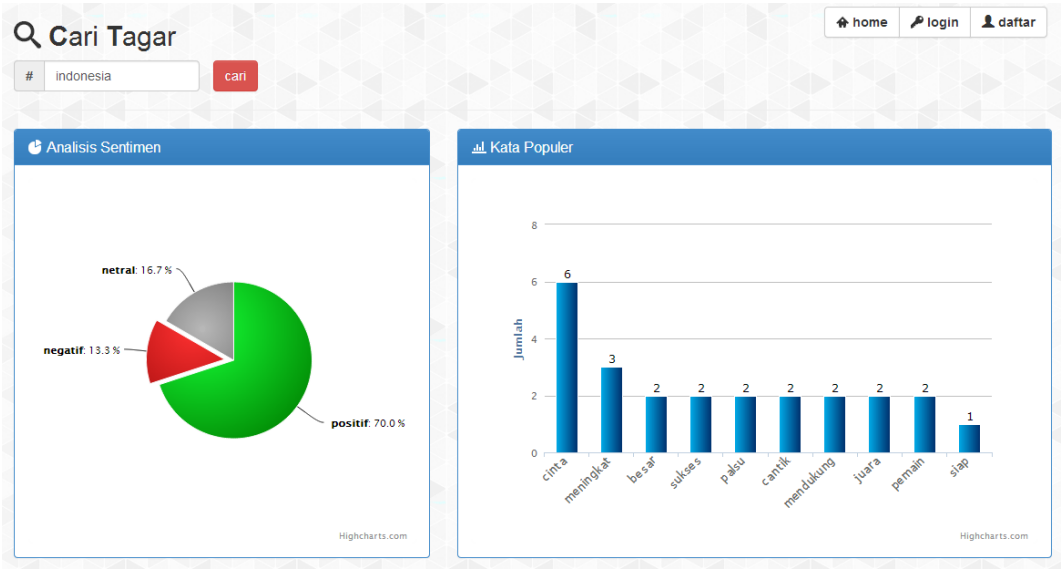
Gambar 4.20 merupakan halaman muka aplikasi, di mana pengguna tidak tidak teraktivasi dapat melakukan pencarian kicauan berdasarkan tagar yang di masukkan.



Gambar 4.20 Halaman muka

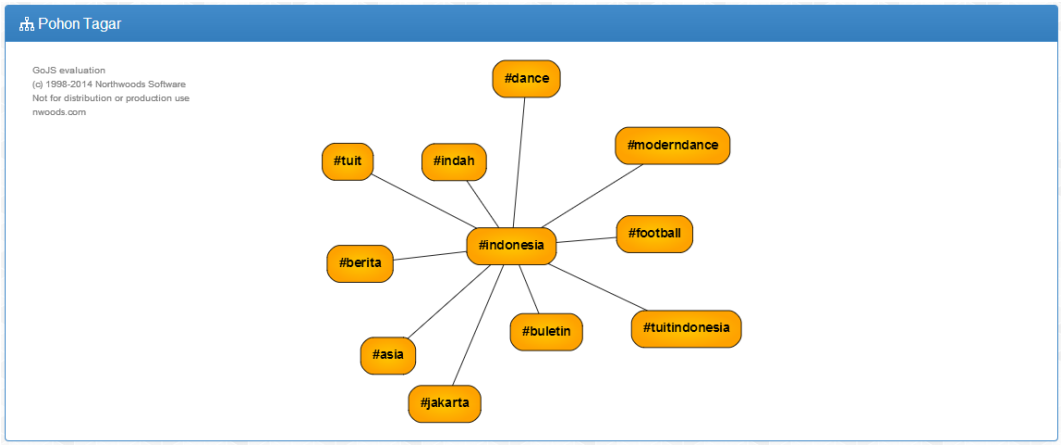
Gambar 4.21 merupakan hasil dari analisis senetimen dan kata populer untuk pengguna tidak teraktivasi dimana keduanya berbentuk grafik. Grafik analisis sentimen merupakan menampilkan prosentasi positif, negatif dan netral. Grafik kata

populer menampilkan sepuluh kata yang mempunyai jumlah kemunculan paling banyak.



Gambar 4.21 Halaman hasil analisis sentimen dan kata populer untuk pengguna tidak teraktivasi

Gambar 4.22 merupakan pohon tagar hasil pencarian, tagar dalam pohon merupakan tagar yang tercantum dalam kicauan hasil pencarian yang kemudian diambil sepuluh tagar dengan jumlah terbanyak.



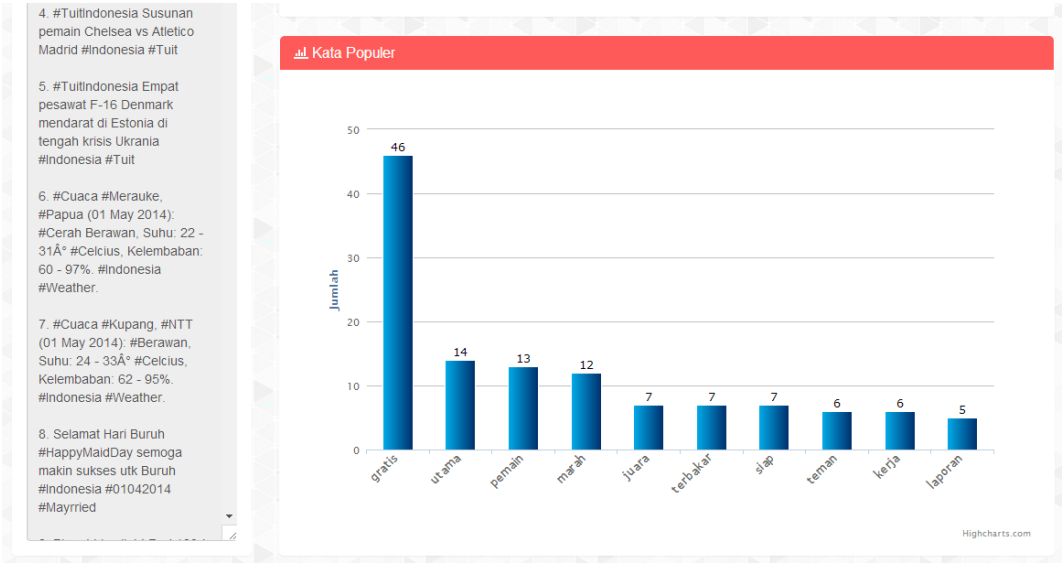
Gambar 4.22 Hasil pohon tagar untuk pengguna tidak teraktivasi

Gambar 4.23 merupakan halaman muka yang bagi pengguna teraktivasi. Setiap kicauan hasil pencarian yang tersimpan akan ditampilkan, dan hasil analisis sentimen akan ditampilkan dalam grafik berdasarkan tanggal pengambilan kicauan tersebut.



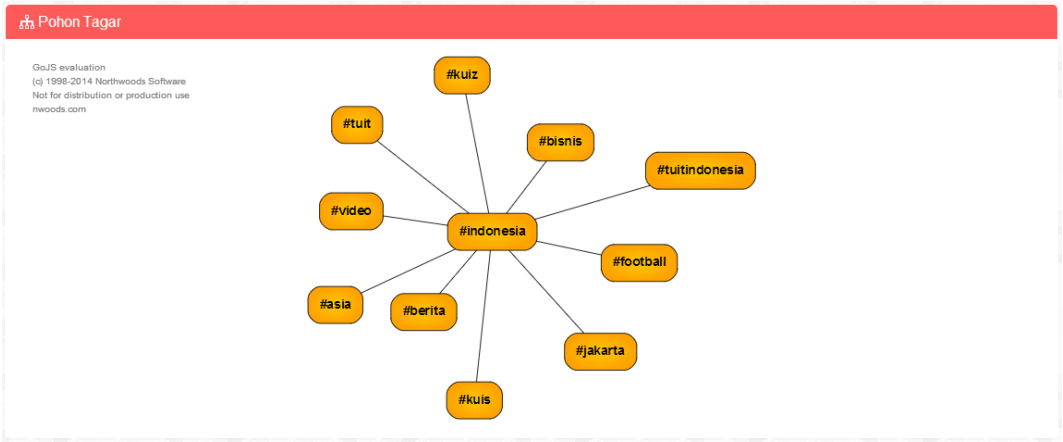
Gambar 4.23 Halaman muka pengguna teraktivasi

Gambar 4.24 merupakan grafik kata populer untuk pengguna teraktivasi. Kata yang ditampilkan merupakan sepuluh kata yang mempunyai jumlah terbanyak.



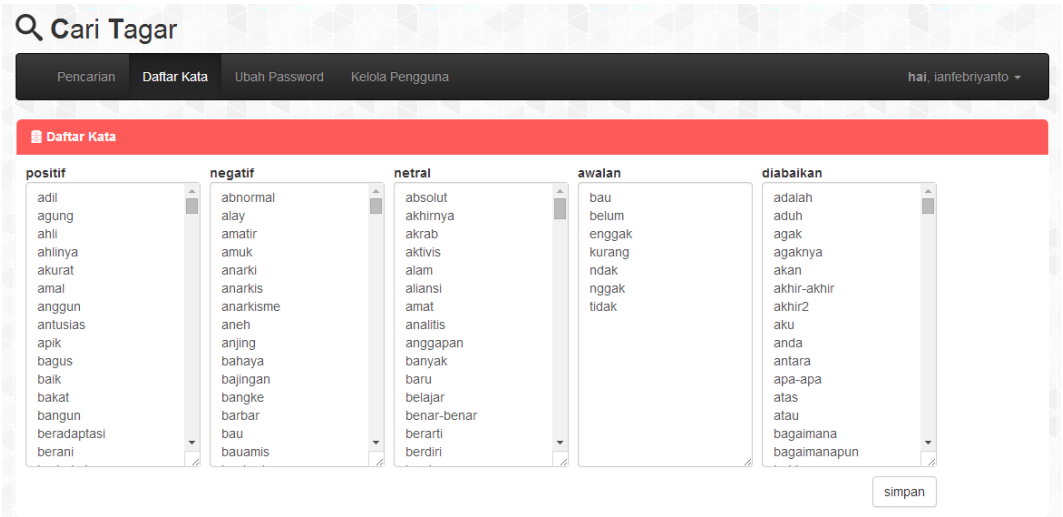
Gambar 4.24 Grafik kata populer pengguna teraktivasi

Gambar 4.25 merupakan pohon tagar untuk pengguna teraktivasi, tagar yang ditampilkan merupakan sepuluh tagar dengan jumlah paling banyak.



Gambar 4.25 Pohon tagar pengguna teraktivasi

Gambar 4.26 merupakan halaman untuk mengelola daftar kata bagi pengguna teraktivasi. Pengguna teraktivasi dapat menambah, mengurangi dan mengubah daftar kata.



Gambar 4.26 Halaman untuk mengelola daftar kata

Gambar 4.27 merupakan halaman admin untuk mengelola pengguna. Disini admin dapat melakukan aktivasi, deaktivasi dan menghapus pengguna yang telah terdaftar.



Gambar 4.27 Halaman admin untuk mengelola pengguna

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melalui tahap perancangan, pembuatan dan implementasi aplikasi, sehingga dapat ditarik beberapa kesimpulan, yaitu:

1. Aplikasi berhasil melakukan analisis terhadap kicauan yang telah dicari berdasarkan tagar, meliputi:
   a. Mampu menampung kicauan hasil pencarian berdasarkan tagar selama tujuh hari berdasarkan kapan pencarian tersebut dilakukan.
   b. Berhasil melakukan klasifikasi kicauan ke dalam kategori positif, negatif atau netral.
2. Aplikasi mampu mempresentasikan hasil analisis data kicauan, yaitu meliputi:
   a. Data hasil analisis sentimen selama tujuh hari dalam bentuk grafik bedasarkan tanggal kapan kicauan tersebut dicari.
   b. Sepuluh kata populer dengan jumlah terbanyak dalam bentuk grafik berdasarkan seluruh kicauan yang telah tersimpan.
   c. Sepuluh tagar dengan jumlah terbanyak dalam bentuk pohon tagar berdasarkan seluruh data tagar yang tersimpan.

5.2 Keterbatasan dan Saran

Karena aplikasi masih jauh dari sempurna dan dibutuhkan pengembangan lebih lanjut. Keterbatasan dan saran untuk aplikasi meliputi:

1. Aplikasi hanya mampu menyimpan kicauan selama tujuh hari pencarian. Kedepannya aplikasi dapat menyimpan lebih banyak kicauan berdasarkan waktu kapan pencarian berdasarkan tagar tersebut dilakukan.
2. Kata populer yang ditampilkan hanya sepuluh kata dengan jumlah terbanyak. Hal ini dikarenakan rata-rata kata yang memiliki frekuensi lebih dari sepuluh terbilang sedikit, selain itu juga agar menjaga tampilan grafik pada kata populer. Kedepannya agar kata populer yang ditampilkan diharapkan bisa lebih

dari sepuluh kata dan lebih spesifik berdasarkan arti dan hubungan sesuai topik berdasarkan tagar yang dicari dengan metode yang lebih baik lagi.

3. Tagar yang ditampilkan hanya sepuluh tagar dengan jumlah terbanyak. Hal ini dikarenakan banyaknya tagar yang tercantum dalam kicauan, selain itu juga untuk menjaga fokus tagar yang ditampilkan. Diharapkan kedepannya bisa lebih banyak dan mempunyai keterkaitan dengan tagar yang dicari dengan metode yang lebih baik.
4. Daftar kata yang digunakan untuk proses analisis atau klasifikasi masih terbilang sedikit dan aplikasi belum bisa tumbuh secara otomatis di mana belum dapat menambah daftar kata secara otomatis. Saran yang perlu diperhatikan adalah perlu adanya lebih banyak penambahan daftar kata positif, negatif juga netral agar klasifikasi lebih akurat dan kedepannya diharapkan mempunyai kemampuan untuk tumbuh secara otomatis, dimana dapat menambah daftar kata yang mengandung sentimen secara otomatis.
5. Aplikasi belum dapat menganalisa sentimen di suatu tempat yang lebih spesifik. Untuk kedepannya agar aplikasi dapat mendeteksi sentimen mengenai sebuah isu di suatu tempat dengan memanfaatkan koordinat yang tercantum dalam kicauan.
6. Aplikasi belum mempunyai metode klasifikasi yang mumpuni. Saran untuk pengembangan aplikasi, agar kedepannya mempunyai metode klasifikasi sentimen yang lebih bagus dan efektif untuk menambah keakuratan klasifikasi.
7. Tagar anak ataupun tagar induk tidak dapat diklik untuk memulai pencarian tagar baru, dan tidak terdapatnya fitur *redo* dan *undo* untuk mengembalikan data setelah dilakukannya pencarian. Untuk pengembangan selanjutnya agar tagar anak dan tagar induk dapat diklik untuk memulai pencarian tagar baru berdasarkan tagar tersebut. Tersedianya fitur *undo* dan *redo* di mana fungsi *undo* untuk mengembalikan kondisi data sebelum dilakukannya proses pencarian kicauan berdasarkan tagar, sedangkan *redo* digunakan untuk mengembalikan data setelah dilakukannya proses pencarian kicauan berdasarkan tagar.

8. *Node* dalam pohon tagar belum berbentuk lingkaran, di mana besar kecil *node* tersebut berdasarkan besarnya frekuensi tagar tersebut. Diharapkan kedepannya penggunaan *node* dapat berbentuk lingkaran pada presentasi pohon tagar, di mana ukuran besar dan kecil *node* tersebut berdasarkan banyaknya frekuensi tagar tersebut.

DAFTAR PUSTAKA

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011). Sentiment Analysis of Twitter Data. *Workshop on Language in Social Media* (hal. 30-38). Portland, Oregon, USA: The Association for Computational Linguistics.

Barber, I. (2010). *Bayesian Opinion Mining*. Dipetik Januari 15, 2014, dari http://phpir.com/bayesian-opinion-mining/

Basch, D. (2012). *Some Fresh Twitter Stats (as of Juli 2012, Dataset Included)*. Dipetik Desember 10, 2013, dari http://diegobasch.com/some-fresh-twitter-stats-as-of-july-2012

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. Dipetik Januari 4, 2014, dari http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

Kominfo. (2011). *Indonesia Peringkat Lima Pengguna Twitter*. Dipetik Desember 20, 2013, dari http://kominfo.go.id/index.php/content/detail/2366/Indonesia+Peringkat+Lima+Pengguna+Twitter/0/sorotan_media#.U20-AoF_vL1

Masse, M. (2011). *REST API Design Rulebook.* USA: O'Reilly Media, Inc.

StatSoft, Inc. (2013). *Electronic Statistics Textbook*. Dipetik Januari 20, 2014, dari http://www.statsoft.com/textbook/naive-bayes-classifier

Twitter Developer. (2013). *REST API v1.1 Resources*. Dipetik November 28, 2013, dari https://dev.twitter.com/docs/api/1.1

Twitter Inc. (2013). *About Twitter*. Dipetik Desember 10, 2013, dari https://about.twiiter.com/id

Twitter Inc. (2013). *Apa itu Hashtag*. Dipetik Desember 15, 2013, dari https://support.twitter.com/articles/20169360-apa-itu-hashtag-simbol

W3C. (2004). *Web Services Glossary*. Dipetik Januari 5, 2014, dari http://www.w3.org/TR/ws-gloss/#webservice

LAMPIRAN

Perbandingan Analisis Sentimen Oleh Manusia dan Aplikasi

Hasil perbandingan analisis sentimen yang dilakukan oleh manusia dan aplikasi, dengan beberapa ketentuan yang telah dilakukan, yaitu:

1. Kicauan hasil pencarian berdasarkan tagar ‘pilpres’.
2. Kicauan dicari pada tanggal 9 Juli 2014 pada waktu pukul 10.00 WIB dan 15.30 WIB.
3. Untuk analisis sentimen tidak memihak pada salah satu pihak atau ingin menjatuhkan salah satu pihak.

No.	Kicauan	Sentimen	
		Manusia	Aplikasi
1	Joko wi Pasti menang hehhe :) #Pilpres	Positif	Positif
2	warga indonesia yg baik. ini buktinya ðŸ™‹ðŸ™ŒðŸ˜³ðŸ˜• #pilpres #pemilu #beforework #coblos #InstaSize http://t.co/t17KQTtqPY	Netral	Positif
3	gw yakin ama no 2 :D #pilpres #kelingking #ink #likeforlike #jokowi http://t.co/XoVjhKVLFA	Netral	-
4	Hahahah dikala #Pilpres berkecamuk perang #Media semakin memanas SP vs ARB	Negatif	Negatif
5	Sudah pasti cuma satu ji yang menang masa dua... Hehe... #adamenangadakalah #Pilpres	Positif	Positif
6	Td ada yg baru dtg ke TPS nanya saya bisa nyoblos dsni ga? Tapi ga bawa apa2 ke TPS *tepokjidat #pilpres	Netral	Netral
7	Tapi tetap saja yg terpenting adalah bagaimana kita sebagai calon pemilih AKTIF mencari tau caranya agar dpt berpartisipasi #pilpres	Netral	-
8	Hasil quick count tiap channel gapodo #pilpres	Negatif	-
9	bener ya, siapa aja kan"@Infonesia_: Presiden SBY Siap Dukung Presiden Terpilih http://t.co/rrLnAD6Uwg #pilpres"	Positif	Positif
10	Adin lembaga survey sond bisa di percaya ni mah ! #Pilpres	Netral	-
11	Banyak-banyak istighfar Prof. @mohmahfudmd ini bulan Ramadhan, jangan kalap... #Pilpres	Netral	-

12	Nah loh skrg gmn ? Kalah kan , mau ngamuk atau mau diem , cie cie kalah cie cie , akibat FANATISME ,pdhl udh habis"an bgt dukungny #pilpres	Negatif	Negatif
13	Hahahahaa.. RT @spitndroll Udah adu penalti aja kalo gt #pilpres	Netral	-
14	Ketika dalih, "Jika ada 2 pilihan, pilih diantara kemudaratan terkecil" menjadi alasan berpartisipasi dlm #Pilpres	Netral	-
15	RT @jpnncom: Sebut Unggul 6 Persen, Ical Ngotot Prabowo Menang http://t.co/IoGH35jSmt #JPNN #Pilpres	Positif	Positif
16	Kita sdh berpartisipasi dg baik di #Pilpres ini, mari tetap jaga kondusivitas Indonesia & be sportive!	Positif	Positif
17	#OMG akhirnya dapet 2 presiden, alhamdulillaah..#pilpres	Netral	Netral
18	Menang #pilpres...mmhhh...bkn berarti smua org indonesia menyukai anda.....	Netral	Positif
19	warga indonesia yg baik. ini buktinya ðŸ™‹ðŸ™ŒðŸ˜³ðŸ˜• #pilpres #pemilu #beforework #coblos #InstaSize http://t.co/t17KQTtqPY	Netral	Positif
20	Potensi konfik dapat terjadi, ayo kita kawal dan amankan #Pilpres 2014	Positif	-
21	RT @_YLinx: Utk pastinya, mari kita tunggu hasil resmi dr KPU. Mari kita saling jg sikap n slg menghormati...#pilpres	Positif	Positif
22	PAK, menang ki beban. babanmu ki abot lho PAK. ojo sok nggampangke PAK. #GP #pilpres	Negatif	Positif
23	entah puasaku sudah batal apa belum :) #pilpres di kandang banteng	Netral	-
24	RT @reniyoheser: Saya"@Toraja_ID: Yang dukung Jokowi JK mana suaranya ? #Pilpres	Netral	-
25	sKENARIO #PILPRES CURANG MULAI DIJALANKAN @KPU_RI @Bawaslu_RI	Negatif	Negatif
26	RT @jpnncom: Hitung Cepat SMRC-LSI, Jokowi-JK Menang http://t.co/u3XZ1QsrJm #JPNN #pilpres	Netral	Positif
27	Sejak awal kampanye #Pilpres pemberitaan #Media sllu tdk berimbang,dan tdk bs dipercaya.	Negatif	Negatif

28	RT @seputarTA: setelah selesai #pilpres kita akan tau mana tukang survey yang asli dan mana yg iseng :)	Negatif	Negatif
29	Baru quickcount aja, kok sudah pada pesta di TVnya masing2 #pilpres	Negatif	Netral
30	Ngotot berohh "@jpnncom: Sebut Unggul 6 Persen, Ical Ngotot Prabowo Menang http://t.co/JOXJPN3jqp #JPNN #Pilpres"	Negatif	Positif
31	Ngeri liat antusias rakyat indonesia #pilpres tahun ini #JokowiJK_ForPresident	Positif	Positif
32	RT @SoalMalangan: Sak stasiun tipi sak lembaga surpe, sak lembaga surpe bedo asile. Hmm dadine kipa. #pilpres #kuikon	Negatif	-
33	Alay lo wkwk RT @bukan_fadil: Say no to golput #Pilpres #9July2014 http://t.co/KpXuLTFnvW	Negatif	Negatif
34	RT @empatsayap untuk Indonesia lebih baik :) #pilpres #KadoBuku http://t.co/wZoFkNkVTg	Positif	Positif
35	Wujud kgembiraan demokrasi,om sapi ngasi iced milk gratis,kuota 50 cup :brlaku dr jam 17.30 - 18.00 ayee,slm guyub rukun #pilpres @seputarTA	Positif	Positif
36	RT @GazaliMakmur: â€œ@jpnncom: Menang Versi Quick Qount, Jokowi: Saatnya Kita Mengawal Suara http://t.co/rB8vYmsoc3 #JPNN #PILPRESâ€•	Positif	Positif
37	RT @arsitekmenulis: Kalau benar @JokowiJusufK menang, saya benar-benar kagum dg rakyat Indonesia yg masih bisa berfikir jernih dalam kabut â€¦	Positif	Positif
38	Udah nyelup di TPS 5 :)) Apapun hasilnya, semoga jadi pemimpin yg amanah :) #demokrasi #pilpresâ€¦ http://t.co/VayHDNCEt9	Positif	Positif
39	"Jgn berikan hatimu kpd seseorang sebelum kamu yakin bahwa dia tahu bagaimana cara tuk memperlakukannya dgn baik. #PILPRES 2014 blm FINAL.."	Positif	Positif
40	Sudah jls yg terpilih siapa.... #pilpres	Netral	-
41	Ya Ø§ÙŽÙ„Ù„Ù‘Ù‡ • berikan yang terbaik untuk negrinya...aku mau nya Prabowo_hatta #Pilpres	Positif	Positif
42	Alhamdulilah RT @jpnncom Hitung Cepat SMRC-LSI, Jokowi-JK Menang	Positif	Positif

	http://t.co/5oI0ukZ2Fj #JPNN #pilpres #JokowiPresidenBaru		
43	RT @estiningsihdwi: Kalau sudah terjadi, baru bahas analisanya. Kalau bahas sekarang, nanti dianggap mendahului yg kuasa. :) #Psychologicalâ€¦	Netral	Netral
44	RT @jpnncom: Waspadai Penggiringan Opini Lewat Publikasi Hasil Survei http://t.co/2AbcBSzmvc #JPNN #pilpres	Negatif	-
45	Macam-macamnya lgi pada h2c ama pemenang #pilpres yee..	Netral	Positif
46	Selamatkan Indonesia sekarang juga! Menjaga Bhinneka Tunggal Ika dari mafia dan penguasa #pilpres	Positif	Positif
47	alhamdulillah tugas negara selesai ðŸ™ŒðŸ‘¼ #kpps #pilpres #2014 #Indonesia #instapict http://t.co/b6VtaH61SE	Netral	Netral
48	Sebenarnya dalam Quick count tersebut. siapa yang jadi pemenangnya #Pilpres 2014	Netral	-
49	Sangat berpotensi RT @morrisssey: Waduh, No. 1 udah declare memenangkan #Pilpres di TV One. Sementara No. 2 ... http://t.co/cEg5JfSiJZ	Netral	-
50	Okeh.. Sama2 deklarasi.. Gaada yg ngrasa kalah.. Anggaplah seri.. Adu penalti ajalah Pak... :D #pilpres	Netral	Negatif
51	Gara gara #pilpres nasib #Gaza seakan terabaikan,,mari jangan tutup mata saudarku,,#PrayForGazaPalestine @Deddy_Mizwar_ @aheryawan	Negatif	Negatif
52	Generasi Muda Demi Indonesia Baru #Partisipasi #kpps #Pilpres #Indonesia #2014 with @herlinriany http://t.co/qaMKkECeV0	Netral	-
53	RT @Abai_lawtama: Kok agak ganjil gitu, di TV-one punya ARB prabowo menang, di Metro-TV punya surya paloh, jokowi menang #pilpres	Negatif	Positif
54	dasar #pilpres yg aneh	Negatif	Negatif
55	Kita memilih presiden yg gaya bicaranya tidak sombong, tidak banyak janji & peduli rakyat. #Pilpres #JokowiDay #jokowijk #WeAreWinners	Positif	Positif
56	4 suara untuk kemenangan pilpres #Pilpres #TwoIsBetterThanOne #Mother #Brother #Me #Sisterâ€¦ http://t.co/51ydSpdJlD	Positif	-
57	Yang jadi presiden siapa ya kira* #pilpres	Netral	-

58	opo iki kok klaim menang kabeh -,- #piLpres	Negatif	Positif
59	RT @FeminaMagazine: RT @iamami0: kami sudah milih :) #pilpres #SuaraAndaBerharga..hope the best for Indonesia http://t.co/IDDMRB21y4	Netral	-
60	Hahaha @jpnncom: Hatta Rajasa Kalah Telak di Kampung Sendiri http://t.co/UdK6V9ybhh #JPNN #PILPRES"	Negatif	Negatif
61	#PrayForGaza. Org2 disana perang melawan Zionis. Indonesia jgn smpai perang melawan saudara sendiri karena #pilpres. https://t.co/sSLBwEcuXL	Negatif	Negatif
62	Ingat, kerja belum selesai. Proses #pilpres ini harus terus dikawal.	Netral	Positif
63	Pdhl kan sudah dipermudah. Tinggal bawa KTP/KK/Paspor ke alamat asal. Atau minta surat ket domisili dr lurah alamat tempat tinggal #pilpres	Positif	Positif
64	Mau 1 atau 2 terserah kalian! Kita hanya penikmat wkwks :D #Pilpres #PilCapres http://t.co/gcWjx4OCI9	Netral	-
65	Quick Count. Diambil sampel 2000 TPS seluruh indonesia. Jumlah TPS seluruh Indonesia 479.183 "kita tunggu hasil real dri KPU" #pilpres	Netral	-
66	kertas #PILPRES 2014 masyarakat sekitar kabupaten OKI masih banyak yg bekerja tampa mempedulikan PEMILU (Petani, Buruh, Nelayan)	Negatif	-
67	#PandeglangNyoblos di TPS 01 kel.sukaratu kec.majasari prabowo unggul 208 sedangkan jokowi 107 #pilpres #Pilpres2014	Positif	Positif
68	Media-media berbeda dgn Lembaga Survey yg berbeda pula. Only waitin for July, 22. Kepastian yg benar2 pasti. #Pilpres #Democracy	Netral	Positif
69	RT @NathaAgus: German sudah menang , skrg tinggal pak jokowi aja ! #jokowi9juli #jokowidodo #pilpres #9juliâ€¦ http://t.co/kh5jvewdUa	Netral	Positif
70	RT @estiningsihdwi: Belum ribut kan? Aihhh... Jangan lah.... :) #PsychologicalWarfare #Pilpres	Netral	-
71	Memang mungkin KPU nya juga kurang sosialisasi kepada masyarakat. Sehingga	Negatif	Negatif

	banyak yg masih bertanya2 bagaimana caranya #pilpres		
72	RT @WikuSuryomurti: apapun hsl akhinya, smg ini yg trbaik buat Indonesia. Apkh anugerah atau cobaan berikutnya. smg kita senantiasa diberi â€¦	Positif	Positif
73	RT @hilmanhin: Untuk Indonesia 5 tahun kedepan #selfie #pilpres @Kuningan_Asri http://t.co/g00pStPux2	Netral	-
74	Empat TPS di Meruyung Dimenangkan Prabowo-Hatta http://t.co/tgswNxrwcM #JPNN #pilpres	Positif	Positif
75	RT @501Awani: Dilema Indonesia Memilih #PILPRES #Pemilu2014 http://t.co/JNQZBCJQEn http://t.co/EeKbbCLJRr	Netral	Negatif
76	Jika Indonesia Cerdas, presiden dicalonkan 1 orang saja dan peringkat ke dua otomatis jadi wakilnya #piLpres	Negatif	Positif
77	Gunakan HAK SUARAMU!,untuk Indonesia yang lebih baikðŸ˜Š #letsvote #indonesia #pilpres #vespa #vespasuperâ€¦ http://t.co/I1YY4ELYUM	Positif	-
78	Tinggalkan Kebagusan, Jokowi Dikangenin Tetangga Mega http://t.co/5s2Y805QuZ #JPNN #Pilpres	Positif	-
79	RT @VictorySitohang: Kedua kubu saling klaim kemenangan. Sbg host piala dunia, sy senang dgr peluit berbunyi sbg tanda akhir pertandingan #â€¦	Netral	Positif
80	RT @AnitaSyn: Yang penting habis ini mari kita tuntut dan awasi janji2 presiden terpilih (copas mas ibal @mauulana_) #PILPRES PERDANA =]miâ€¦	Netral	Netral
81	Dan penting utk diingat, Jauhkan anggapan kita thadap capres tpilih sbg org yg sempurna. Tugas kita lah utk lengkapi kekurangannya #pilpres	Positif	Netral
82	Dua2nya deklarasi menang. Lihat siapa lembaga surveynyaa... #Pilpres	Netral	Positif
83	RT @jpnncom: Hatta Rajasa Kalah Telak di Kampung Sendiri http://t.co/e5ScHUYwlO #JPNN #PILPRES	Negatif	Negatif
84	TPS 18, Menteng, dipenuhi awak media #Pilpres 2014 http://t.co/YRaWkB4RUb	Netral	-

85	#pilpres | Kabar Gembira!!..donlot A5, isi, cari ketua PPS di TPSÂ² sepi jgn nyerah â†' A5 http://t.co/VCADAKdlsq http://t.co/lRSvF5ocCY	Positif	Negatif
86	Pemilihan Presiden Indonesia: Jokowi umum kemenangan, Prabowo enggan mengalah #PEMILU #PILPRES http://t.co/iEBzCwKEK1	Netral	-
87	Duh! "@jpnncom: Tim Prabowo-Hatta Diduga Main Uang http://t.co/H3pAzShGjM #JPNN #pilpres	Negatif	-
88	Tim Jokowi-JK Kantongi Kecurangan di Madura http://t.co/KaQZrJN2nE #JPNN #pilpres	Negatif	Negatif
89	kurang dari 12 jam final #worldcup #GER vs #ARG tapi di sosmed #pilpres lebih rame dengan iklim yang begitu-begitu saja	Negatif	-
90	Dg segala macam kelicikan, pencitraan n propaganda serta fanatisme yg berlebihan dr para pemujanya. Saya berharap Jokowi yg menang #pilpres	Positif	Positif
91	Siapapun yg menang, mari kita hargai semuanya.. jgn nyalahin yg udah kerja keras seperti KPU. #pilpres	Positif	Positif
92	Rakyat dgiring ke opini bkn ke fakta, satu sama lain mengklaim menang, demi "bapak senang". #pilpres	Negatif	-
93	Jadi rakyat yg mana ? Makanya berhati2lah menyikapi pilpres ini. #pilpres	Netral	-
94	kubu yg duluan deklarasi menang brdasar QC koq blm kliatan tabulasi data mrka,pengen tau menang dmn saja mrka #Pilpres #KawalKPU	Negatif	Positif
95	Kl yg ini kalah dbilang kecurangan akan rakyat, lah.. kok bs gt, la wong kl kata QC hampir 50% sama. #pilpres	Negatif	Negatif
96	Masih aneh nih dlm hasil pilpres... dsatu sisi kl yg itu kalah dsuruh legowo, kl yg ini kalah dbilang ada kecurangan. #pilpres	Negatif	Negatif
97	Ga habis fikir tentang quick count. Koq bisa pada deklarasi kemenangan. Duh, ... Ada apa dengan kecerdasan negeri Indonesia? #pilpres	Negatif	Positif
98	Kabar berita #pilpres agak mereda kalah pamor sama final #worldCup	Netral	Negatif

99	Panas sosmed gara" #pilpres.,indonesia emang pinter jelekin sana jelekin sini,bully sana bully sini.,demen bgt koar" di sosmed.!	Negatif	Negatif
100	Siapapun pemenang pilpres skrg, masih perlu membuktikan & meyakinkan hati rakyat selama 5 thn. Jadi skrg yg sdh menang ya Pak SBY. #Pilpres	Positif	Positif

Berdarkan data diatas, hasil perbandingan analsis data antara manusia dan aplikasi adalah:

1. Jumlah hasil analisis yang sama terdapat 45 buah.
2. Jumlah hasil analisis yang tidak sama terdapat 55 buah.