

Analisa Sentimen Menggunakan Data Twitter, Flume, Hive Pada Hadoop dan Java untuk Deteksi Kemacetan di Jakarta

Busman¹, Nurhayati², Nadika Sigit.S³, Tifani Shallynda.K⁴

¹Prodi Manajemen, Sekolah Tinggi Ilmu Ekonomi (STIE) Gotong Royong Jakarta

^{2,3,4} Prodi Teknik Informatika, Fakultas Sains dan Teknologi, UIN Syarif Hidayatullah Jakarta

¹busman.sjam@gmail.com, ²nurhayati@uinjkt.ac.id, ³nadikatrya@gmail.com, ⁴tyfany.kania@gmail.com

Abstract - Traffic congestion big cities in Indonesia is unavoidable, especially in Jakarta. The increasing number of vehicle and the lack of public transportation is the main cause of traffic congestion in Jakarta. It disturb people activities. Government already did various efforts to resolve congestion problem, however it needs high installation, maintenance cost and need time to be implemented. Peoples often complained about traffic congestion in Jakarta by posting in Twitter which called tweets. Every tweets post are saved in API Twitter and used for sentiment analysis. It analyzed emotion of the user. Based on the problems, we do research how to detect traffic congestion in Jakarta. Therefore, we try to makes Congestion Detection App. We design the app using UML diagrams. Congestion Detection App is connected with Hadoop, Flume, Hive and Derby. The app stream twitters data to collected by connecting with API Twitter. This app is Java-based application which can makes and view data tables. It performance searching tweets data by ID and analyze traffic condition on a certain region in Jakarta. The perform sentiment analysis to a certain tweet and display the result based on the data table. The result of research is comparing Data from Congestion Detection App with data from Google Maps. We make three valus categories which consist of three colors: green for less traffic congestion have a value of 1. Orange for medium-scale traffic congestion has value of 2 and Red for heavily traffic congestion has a value of 3. Based on three categories and value we use 4 regions for sample and comparing the values with value from Google Maps Data to get the accuracy. We got 81% average accuracy from the four samples. The result of Data from tweet sample compared with Google Maps Data. It have big detected congestion with Congestion Detection App.

Keywords- Sentiment Analysis, Big Data, Twitter Data, Congestion Detection

Abstrak - Kemacetan lalu lintas di kota-kota besar Indonesia seperti Jakarta tidak dapat dihindari dan berdampak pada kota disekitarnya. Banyaknya volume kendaraan dan kurangnya fasilitas kendaraan umum menjadi penyebab utama kemacetan dan menyebabkan aktifitas masyarakat terganggu. Pemerintah melakukan berbagai upaya untuk dapat menangani kemacetan, akan tetapi membutuhkan banyak biaya untuk instalasi, biaya perawatan dan waktu implementasi yang lama. Seringkali masyarakat mengeluhkan kemacetan di Jakarta melalui postingan Twitter yang biasa disebut tweets. Setiap postingan tweets masyarakat tersimpan dalam API Twitter yang bisa digunakan dan dianalisa. Salah satunya metode untuk menganalisa data twitter yaitu dengan menggunakan Analisa Sentimen yakni analisa terhadap emosi pengguna. Berdasarkan permasalahan tersebut penulis melakukan penelitian untuk dapat mendeteksi kemacetan di Jakarta. Penulis membuat Aplikasi Deteksi Kemacetan yang dirancang menggunakan beberapa diagram UML. Aplikasi Deteksi Kemacetan ini terkoneksi dengan Hadoop, Flume, Hive, Derby, dan Twitter API untuk dapat melakukan streaming data twitter secara *real-time* dalam pengumpulan datanya. Aplikasi ini dibuat dengan bahasa pemrograman Java, lengkap dengan fitur untuk melihat isi data, melakukan pencarian data tabel tweet berdasarkan ID, melakukan analisa keadaan lalu lintas suatu daerah, dan melakukan analisa sentimen terhadap suatu tweet pada tabel, dan menampilkan hasil analisa berdasarkan data pada tabel. Hasil penelitian ini dilakukan pengujian data dengan membandingkan data Aplikasi Deteksi Kemacetan dengan data Google Maps. Penulis memberikan nilai terhadap tiga kategori kemacetan berdasarkan warna di Google Maps. Ketiga kategori tersebut yaitu warna hijau untuk lancar dan diberikan nilai 1, warna orange untuk padat dan diberikan nilai 2, dan warna merah untuk macet dan diberikan nilai 3. Dari hasil ketiga nilai tersebut, penulis melakukan penilaian daerah kemacetan dengan menggunakan 4 daerah sebagai sample. Nilai dari sample tweet dibandingkan dengan nilai dari Data Google Maps untuk mendapatkan akurasi. Sehingga didapatkan akurasi rata-rata dari ke-empat sample yaitu sebesar 81%. Hasil perbandingan tersebut cukup besar dalam mendeteksi Kemacetan dengan Aplikasi Deteksi Kemacetan.

Kata kunci- Analisa Sentimen, Big Data, Data Twitter, Deteksi Kemacetan

I. PENDAHULUAN

Salah satu masalah utama yang dihadapi kota besar adalah kemacetan lalu lintas. Kemacetan lalu lintas menyebabkan peningkatan biaya transportasi dan juga

akibatnya pada rutinitas masyarakat. Permasalahan dari kemacetan lalu lintas menyebar dimana-mana, tetapi kota-kota besar yang mendapat dampak yang paling besar. Kemacetan yang semakin meningkat menyebabkan kesulitan untuk mengestimasi kepadatan jalanan secara

real time untuk membuat keputusan yang berhubungan dengan lalu lintas dan mengatur lalu lintas secara efisien [1].

Kemacetan di Jabodetabek umumnya disebabkan oleh peningkatan laju pertumbuhan jalan (termasuk jalan tol) sebesar 1% per tahun, yang tidak sebanding dengan laju pertumbuhan kendaraan yang mencapai 11% per tahun. Hal ini menyebabkan kemacetan yang parah pada jam-jam puncak [2].

Ada beberapa metode yang dapat digunakan untuk menyelesaikan kemacetan lalu lintas. Metode tersebut adalah pembangunan infrastruktur yang lebih baik seperti pelebaran jalan, jalan layang, dan jalan tol. Tetapi untuk negara berkembang, keuangan dan ketersediaan lahan untuk membangun infrastruktur tersebut merupakan persoalan yang serius [3].

Pendekatan yang lainnya adalah dengan menggunakan sensor dan kamera. Tetapi ada persoalan lagi jika menggunakan sensor dan kamera, seperti cakupan yang terbatas, implementasi yang mahal dan perawatan pada alat-alat tersebut. Metode lainnya menyarankan komunikasi antar kendaraan, tetapi metode ini membutuhkan pemasangan peralatan pada setiap kendaraan dan mempunyai kekhawatiran seperti keamanan dan privasi [3].

Namun dengan adanya big data terciptalah metode baru untuk mendeteksi kemacetan yaitu dengan menggunakan data dari sosial media twitter. Karena twitter menyediakan Application Programming Interface yang memungkinkan mengumpulkan tweet oleh pengguna pihak ketiga. API layanan gratis yang terbatas pada sample 1% dari semua tweet [4]. Sehingga data-data kemacetan mudah diperoleh, tanpa harus mengeluarkan banyak biaya.

Berdasarkan permasalahan di atas, penulis membangun Aplikasi yang dapat mendeteksi lokasi kemacetan yang ada di Jakarta dengan menggunakan data Twitter dan dianalisa menggunakan Analisa Sentimen.

Penelitian sebelumnya menggunakan hadoop dalam mengolah data untuk menentukan kepadatan lalu lintas menghasilkan total rata-rata akurasi sebesar 95.4% pada jurnal [5]. Sedangkan pada jurnal [6] melakukan sentimen analisis menggunakan twitter streaming API dari twitter dengan mengumpulkan tweets yang dipilih berdasarkan uji Chi-Square dan klasifikasi Naïve Bayes untuk mengklasifikasi tweet sebagai positif dan negatif.

Pada referensi jurnal [3] menunjukkan bahwa pemantauan lalu lintas dengan twitter sebagai media sosial membantu dengan hasil yang akurat, dengan model bahasa Naïve Bayes dapat dikombinasikan untuk mengidentifikasi lokasi kemacetan lalu lintas dan menampilkan statistik dan analisa sentimen menggunakan pie chart.

Kemudian pada jurnal [1] melaukan pendekatan semi-supervised untuk mendeteksi kejadian lalu lintas menggunakan twitter untuk membantu pengguna mendeteksi kejadian kecelakaan dan lalu lintas untuk

menghemat waktu. Dengan mengklasifikasikannya menjadi tweet terkait atau tidak terkait lalu lintas.

Berdasarkan metode-metode penelitian tersebut, penulis ingin mengetahui bagaimana menggunakan data twitter dan melakukan analisa datanya menggunakan analisa sentimen untuk mengetahui daerah di Jakarta yang sedang terjadi kemacetan, agar pengguna dapat menghindari kemacetan karna hal-hal tertentu.

II. METODE PENELITIAN

Big Data dapat didefinisikan sebagai sekumpulan data yang ukurannya melampaui kemampuan dari tool perangkat lunak basis data untuk mengambil, menyimpan, mengatur dan menganalisa. Kumpulan data tersebut secara umum dihasilkan melalui internet, perangkat mobile, sensor jaringan, sistem enterprise dan organisasi [7]. Big Data tidak hanya terfokus pada *volume*, *velocity* dan *variety* juga termasuk pada fokus Big Data. Hasil dari big data bisa terstruktur, tidak terstruktur dan semi terstruktur[8].

A. Hadoop

Apache Hadoop merupakan pilihan yang bagus untuk analisa Twitter karena Hadoop bekerja pada big data yang terdistribusi. Apache Hadoop merupakan sebuah framework perangkat lunak yang open source untuk penyimpanan terdistribusi dan pemrosesan terdistribusi skala besar pada data-sets di cluster. Hadoop menjalankan aplikasi menggunakan algoritma MapReduce, dimana data diproses secara paralel di node CPU yang berbeda[9].

Singkatnya, framework Hadoop mampu untuk mengembangkan aplikasi yang dapat berjalan pada cluster-cluster di komputer dan dapat melakukan analisa statistik menyeluruh untuk ukuran data yang sangat besar.

Adapun langkah-langkah untuk menjalankan Hadoop yaitu sebagai berikut:

1. Masuk ke terminal sebagai hduser, kemudian jalankan Hadoop dengan perintah start-all.sh dan periksa apakah sudah berjalan dengan memastikan kelengkapan Hadoop daemons dengan mengetikkan perintah jps.

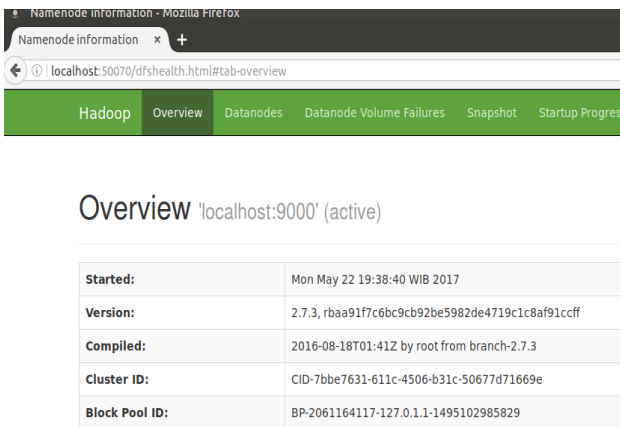
```

hduser@tifanshall-Lenovo-G405:~/home/tifanshall
File Edit View Search Terminal Help
hduser@tifanshall-Lenovo-G405:~$ sudo su hduser
[sudo] password for tifanshall:
hduser@tifanshall-Lenovo-G405:~/home/tifanshall$ start-all.sh
This script is deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-na
menode-tifanshall-Lenovo-G405.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser-da
tanode-tifanshall-Lenovo-G405.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hd
user-secondarynamenode-tifanshall-Lenovo-G405.out
Starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resource
manager-tifanshall-Lenovo-G405.out
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-n
odemanager-tifanshall-Lenovo-G405.out
hduser@tifanshall-Lenovo-G405:~/home/tifanshall$ jps
#993 secondarynamenode
#474 NameNode
#179 NodeManager
#610 DataNode
#044 ResourceManager
#223 Jps
hduser@tifanshall-Lenovo-G405:~/home/tifanshall$

```

Gambar 1. Tampilan Perintah untuk Menjalankan hadoop

2. Periksa Hadoop yang sudah berjalan dengan masuk ke browser dan mengetik localhost:50070 untuk masuk kedalam localhost hadoop.

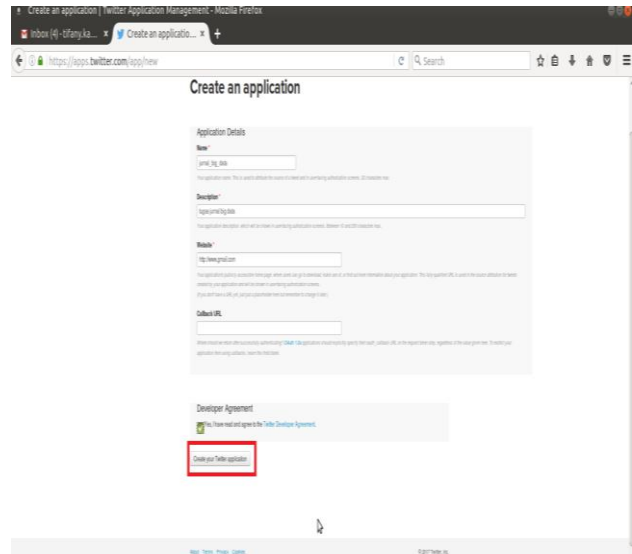


Gambar 2. Tampilan localhost Hadoop

B. Twitter API

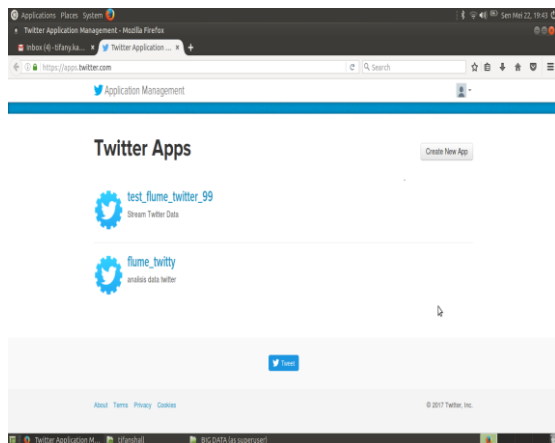
Twitter API merupakan sekumpulan URL yang mengambil parameter. URL ini mengizinkan pengguna mengakses fitur-fitur Twitter, seperti memposting tweet atau mencari tweet yang berisi suatu kata dan lain-lain[8].

Twitter API dibutuhkan untuk mengambil data di Twitter. Pertama yang harus dilakukan yaitu masuk ke <https://apps.twitter.com>.



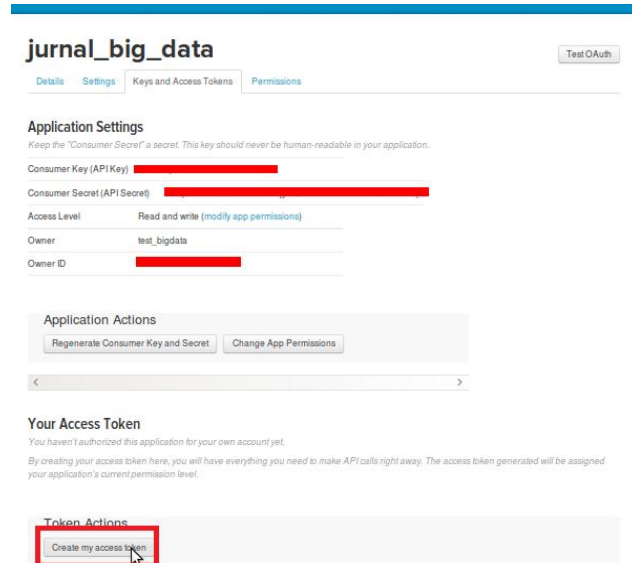
Gambar 4. Tampilan Form Create an Application pada Twitter Apps

Pada tampilan Keys and Access Tokens, buat accesstoken dengan mengklik button “Create my access token” tampilannya seperti gambar dibawah ini.



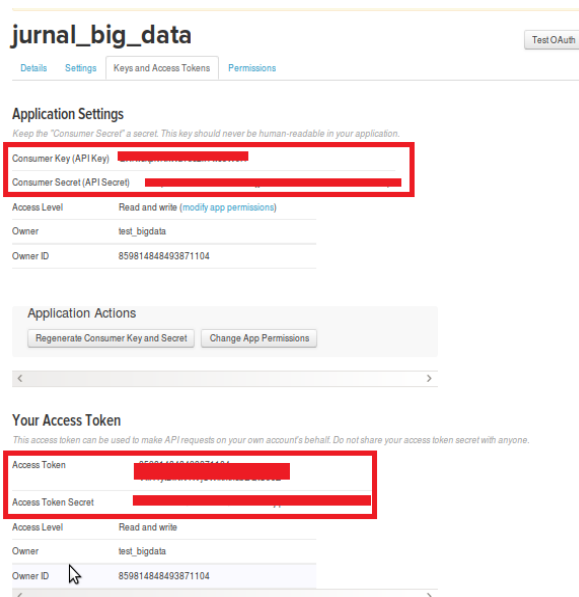
Gambar 3. Tampilan Twitter Apps

Langkah selanjutnya klik pada button “Create New App”. Kemudian isi Nama, deskripsi, website, dan ceklistdevelopera greement nya dan klik button “Create your Twitter application”.



Gambar 5. Tampilan Keys and Access Tokens pada Twitter Apps

Maka twitter API akan menampilkan consumer key, consumer secret, access token, dan access token secret. Hasilnya seperti gambar berikut:



Gambar 6. Tampilan Key dan Access Tokens

Pada gambar 6 merupakan tampilan key and access tokens pada twitter API. Consumer key, consumer secret, access token, dan access token secret dimasukkan ke file konfigurasi flume yaitu twitter.conf dan penulis memasukkan keywords yaitu kemacetan Jakarta, untuk mendapatkan data dari twitter API sesuai dengan keywords tersebut.

C. Flume

Apache Flume merupakan sebuah layanan yang terdistribusi, handal dan tersedia untuk mengumpulkan, menggabungkan dan memindahkan data streaming yang berjumlah besar ke HDFS. Flume dapat digunakan untuk menampung data twitter di HDFS[9].

```

hduser@tifanshall-Lenovo-G405:/home/tifanshall
File Edit View Search Terminal Help
hduser@tifanshall-Lenovo-G405:/home/tifanshall$ flume-ng agent -n TwitterAgent -f
/usr/local/flume/conf/twitter.conf
Warning: No configuration directory set! Use --conf <dir> to override.
Info: Including Hadoop libraries found via (/usr/local/hadoop/bin/hadoop) for HDFS
access
Info: Including Hive libraries found via (/usr/local/hive) for Hive access
./exec /usr/lib/jvm/java-8-openjdk-amd64/bin/java -Xmx20m -cp /usr/local/flume/li
b/*:/usr/local/hadoop/etc/hadoop:/usr/local/hadoop/share/hadoop/common/lib/*:/usr
/local/hadoop/share/hadoop/common/*:/usr/local/hadoop/share/hadoop/hdfs:/usr/local
/hadoop/share/hadoop/hdfs/lib/*:/usr/local/hadoop/share/hadoop/hdfs/*:/usr/local/ha
doo/share/hadoop/yarn/lib/*:/usr/local/hadoop/share/hadoop/yarn/*:/usr/local/ha
doo/share/hadoop/napreduce/lib/*:/usr/local/hadoop/share/hadoop/napreduce/*:/cont
rol/capacity-scheduler/*-jar:/usr/local/hive/lib/*-ojava.library.path=/usr/local
/hadoop/lib/native org.apache.flume.node.Application -n TwitterAgent -f /usr/local
/flume/conf/twitter.conf
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/flume/lib/slf4j-log4j12-1.6.1.jar!or
g/slf4j/inpl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-
log4j12-1.7.10.jar!org/slf4j/inpl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
17/05/23 07:12:30 INFO node.PollingPropertiesFileConfigurationProvider: Configurati
on provider starting

```

Gambar 7. Tampilan Perintah untuk Menjalankan Flume

Pada gambar 7 merupakan tampilan yang berisi perintah untuk menjalankan flume. Setelah flume dijalankan maka akan muncul tweet di HDFS, kemudian unduh data di HDFS yang berformat JSON dan buka <https://jsonlint.com> sebagai validator untuk mengubah data tweet agar bisa terbaca.

D. Hive

Hive merupakan data warehouse infrastructure tool untuk memproses data terstruktur di Hadoop. Hive digunakan untuk meringkaskan Big Data dan membuat pen-query-an dan analisa menjadi mudah. Apache Hive (HiveQL) dengan HDFS digunakan untuk menganalisa data. Hive menyediakan interface seperti SQL untuk memproses data yang tersimpan di HDP[9].

E. Analisa Sentimen

Analisa sentimen dapat diartikan sebagai klasifikasi sentimen dari data atau teks yang kebanyakan tidak terstruktur menggunakan kombinasi dari NLP (Natural Language Processing) dan teknik komputasional[3].

F. Bahasa Pemrograman Java

Java adalah sebuah bahasa pemrograman yang populer dikalangan para akademisi dan praktisi komputer. Java pertama kali dikembangkan untuk memenuhi kebutuhan akan sebuah bahasa komputer yang ditulis satu kali dan dapat dijalankan di banyak system komputer berbeda tanpa perubahan kode berarti. Pada umumnya, para pakar pemrograman berpendapat bahwa bahasa Java memiliki konsep yang konsisten dengan teori pemrograman objek dan aman untuk digunakan[10].

Java sampai saat ini masih merupakan bahasa pemrograman yang masih sangat di minati dan banyak digunakan oleh para programmer dan software developer untuk mengembangkan berbagai tipe aplikasi, mulai dari aplikasi console, aplikasi desktop, game, dan applet (aplikasi yang berjalan di lingkungan web browser), sampai ke aplikasi-aplikasi yang berskala enterprise. Untuk memenuhi kebutuhan tipe aplikasi yang beragam tersebut, Java dikategorikan menjadi tiga edisi, yaitu: J2SE (Java 2 Platform Standart Edition) untuk membuat aplikasi-aplikasi desktop dan applet, J2EE (Java 2 Platform Enterprise Edition) untuk membuat aplikasi-aplikasi multitier berskala enterprise, dan J2ME (Java 2 Platform Micro Edition) untuk membuat aplikasi-aplikasi yang dapat dijalankan di lingkungan perangkat-perangkat mikro seperti handphone, PDA dan Smartphone[10].

G. Netbeans

NetBeans merupakan sebuah integrated development environment berbasis Java. IDE ini didesain untuk membatasi kesalahan coding dan memfasilitasi pengoreksi kesalahan (error) dengan tools seperti NetBeans FindBugs untuk menemukan dan memperbaiki masalah coding di Java dan Debugger untuk mengatur kode yang kompleks dengan field watches, breakpoints dan execution monitoring[11].

NetBeans IDE menyediakan penganalisaan kode dan editor yang bekerja menggunakan teknologi java yang terbaru, yaitu Java SE8, Java SE Embedded 8 dan Java ME Embedded 8. NetBeans IDE juga mempunyai tools untuk HTML5/JavaScript, Node.js, KnockoutJS dan AngularJS[12].

H. UML (Unified Modelling Language)

UML adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasi, membangun dan pendokumentasian dari sebuah sistem pengembangan software berbasis object oriented[13].

Adapun diagram-diagram yang ada dalam UML sebagai berikut:

1. Use Case Diagram

Use Case Diagram menggambarkan aktivitas yang dapat dilakukan oleh sistem dari sudut pandang user (pengguna sistem) sebagai pengguna (external observer) dan berhubungan dengan skenario-skenario yang dapat dilakukan oleh user. Use case diagram merupakan sekumpulan actor, usecase dan komunikasi antar actor dengan usecase[13].

2. Class Diagram

Class Diagram menggambarkan suatu grup yang memiliki kesamaan keadaan dan perilaku. Class merupakan blueprint dari object dalam sistem berorientasi object[13].

3. Statechart Diagram

Statechart Diagram menggambarkan perilaku sistem perangkat lunak yang dibuat dan perilaku class, subsistem dan seluruh aplikasi. State machine diagram berguna untuk menyediakan cara yang baik dalam memodelkan komunikasi yang terjadi dengan entitas luar[13].

4. Activity Diagram

Activity Diagram menggambarkan hubungan dari usecase dan responsibilities dari class. Activity diagram fokus pada behavior dari operation dan menunjukkan aliran control untuk menyelesaikan proses tertentu, seperti aktivitas untuk menerima pembayaran. Aliran control tersebut dimulai dari state awal atau initial state dan diakhiri oleh state yang lain yang disebut dengan end state[13].

5. Sequence Diagram

Sequence diagram menggambarkan interaksi antara objek di dalam dan di sekitar sistem dan digunakan untuk menggambarkan skenario, tipe diagram ini memperlihatkan tahapan-tahapan yang seharusnya terjadi untuk menghasilkan use case[13].

6. Component Diagram

Component Diagram menggambarkan struktur dan hubungan antar komponen, termasuk ketergantungan (dependency). Component diagram akan sangat membantu bila ada salah satu komponen yang rusak atau tidak sesuai dengan tujuan system[13].

7. Deployment Diagram

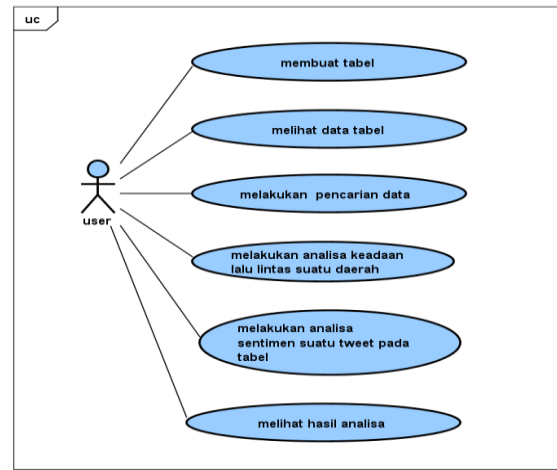
Deployment diagram digunakan untuk menggambarkan hardware dimana komponen software tersebut akan dikembangkan. Deployment diagram biasanya digambarkan setelah tahap pengembangan sistem diselesaikan. Namun deployment diagram bisa

digambarkan setelah mengidentifikasi class, interface dan relationship-nya.[13]. yang dimaksud.

III. HASIL DAN PEMBAHASAN

A. Use Case Diagram pada Aplikasi Deteksi Kemacetan

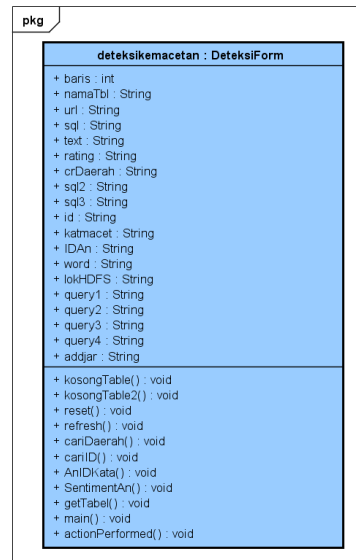
Berikut use case diagram pada Program Aplikasi Deteksi Kemacetan yang menggambarkan apa saja yang dapat dilakukan user di dalam Aplikasi Deteksi Kemacetan.



Gambar 8. Use Case Diagram pada Aplikasi Deteksi Kemacetan

B. Class Diagram pada Aplikasi Deteksi Kemacetan

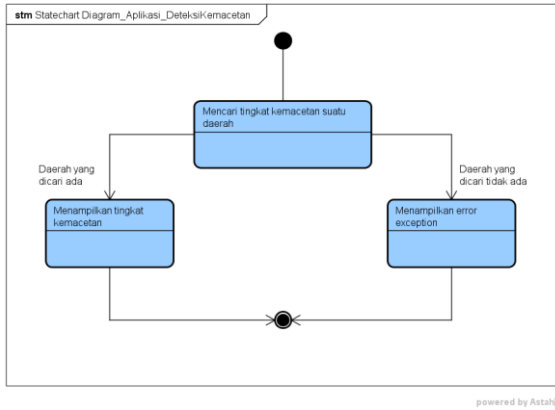
Pada Aplikasi Deteksi Kemacetan ini hanya memiliki 1 class diagram yaitu class diagram DeteksiForm. Berikut merupakan gambaran class diagram DeteksiForm:



Gambar 9. Class Diagram pada Aplikasi Deteksi Kemacetan

C. State chart Diagram pada Aplikasi Deteksi Kemacetan

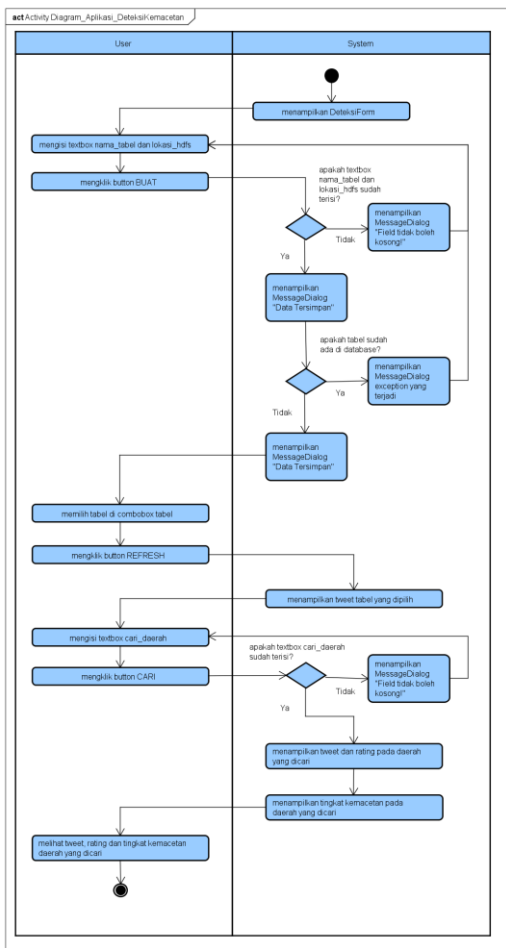
Pada Aplikasi Deteksi Kemacetan terdapat satu State Diagram yaitu pada menganalisa tingkat kemacetan suatu daerah.



Gambar 10. Statechart Diagram pada Aplikasi Deteksi Kemacetan

D. Activity Diagram pada Aplikasi Deteksi Kemacetan

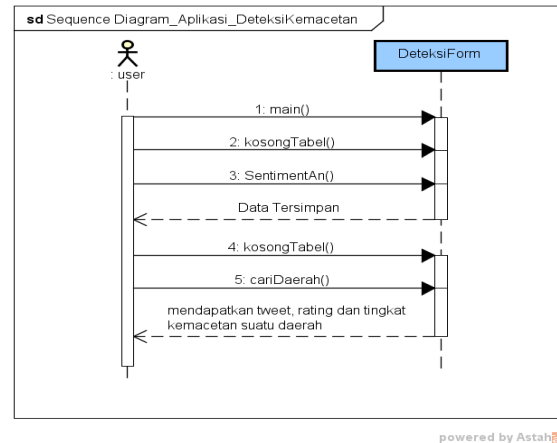
Activity Diagram pada Aplikasi Deteksi Kemacetan dijelaskan pada gambar berikut:



Gambar 11. Activity Diagram pada Aplikasi Deteksi Kemacetan

E. Sequence Diagram pada Aplikasi Deteksi Kemacetan

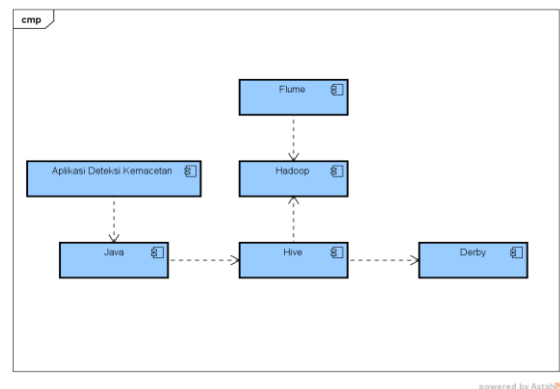
Berikut merupakan sequence diagram pada Aplikasi Deteksi Kemacetan.



Gambar 12. Sequence Diagram pada Aplikasi Deteksi Kemacetan

F. Component Diagram pada Aplikasi Deteksi Kemacetan

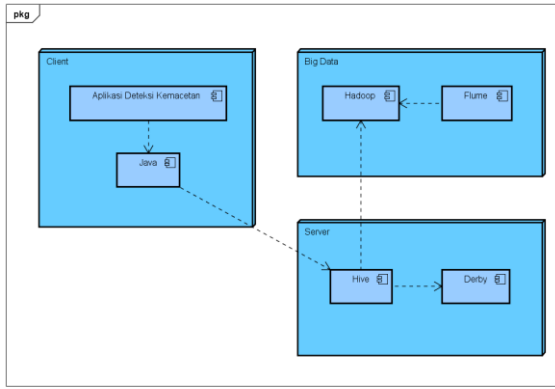
Pada Aplikasi Deteksi Kemacetan terdapat tiga komponen yang digambarkan sebagai berikut:



Gambar 13. Component Diagram pada Aplikasi Deteksi Kemacetan

G. Deployment Diagram pada Aplikasi Deteksi Kemacetan

Pada Deployment Diagram Aplikasi Deteksi Kemacetan, komponen-komponen yang digunakan terdapat pada node client, node server dan node big data yang digambarkan pada gambar berikut:



Gambar 14. Deployment Diagram pada Aplikasi Deteksi Kemacetan

H. Tahap Pengumpulan Data

Tahap pengumpulan data dijelaskan pada gambar dibawah ini:

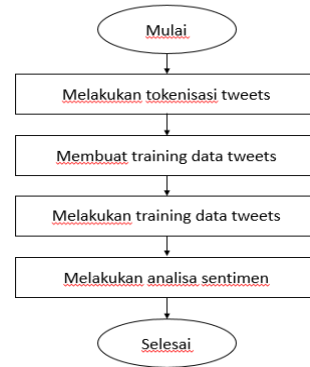


Gambar 15. Tahap Pengumpulan Data

Pada Gambar 15. merupakan alur tahapan pengumpulan data penelitian. Pertama instalasi dan konfigurasi Hadoop, kemudian melakukan instalasi dan konfigurasi Flume. Proses selanjutnya melakukan streaming tweets yakni proses pengambilan tweets menggunakan Twitter API. Tweets kemudian di download di simpan pada HDFS. Kemudian melakukan instalasi dan konfigurasi Hive, dan Derby, untuk kemudian dapat membuat tabel di Hive untuk menyimpan data tweets yang akan dianalisa.

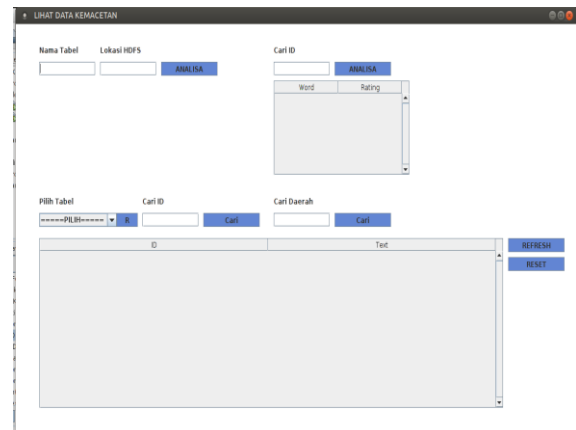
I. Tahap Analisa Data

Tahap analisa data pada penelitian ini dijelaskan pada gambar berikut:



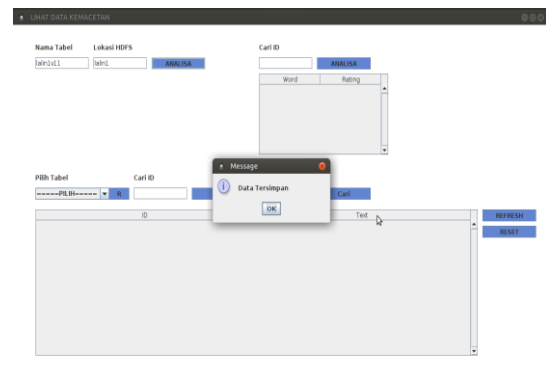
Gambar 16. Tahap Analisa Data

Pada Gambar 16 merupakan tahapan analisa data yang dimulai dari tokenisasi tweets, yaitu membuat kalimat menjadi kata-kata, seperti kemacetan parah terjadi di manggarai menjadi “kemacetan”, “parah”, “terjadi”, “di”, “manggarai”. Selanjutnya membuat training data tweets kemacetan tersebut dan melakukan pelatihan data training tweets kemacetan. Kemudian melakukan analisa sentimen yang dilakukan di Hive dan ditampilkan di aplikasi berbasis Java.



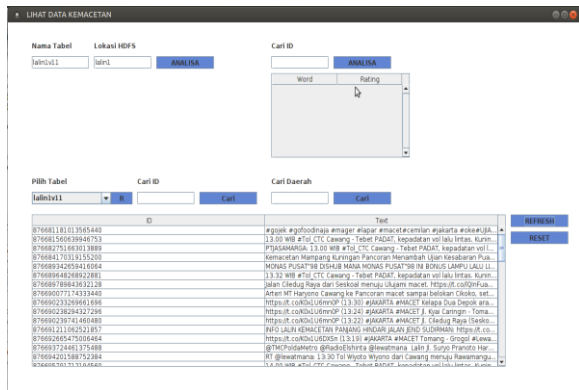
Gambar 17. Tampilan Aplikasi

Gambar 17 merupakan tampilan aplikasi untuk mendeteksi kemacetan di Jakarta. Pada aplikasi ini terdapat 7 Button, 5 TextBox, 2 Table dan 1 ComboBox.



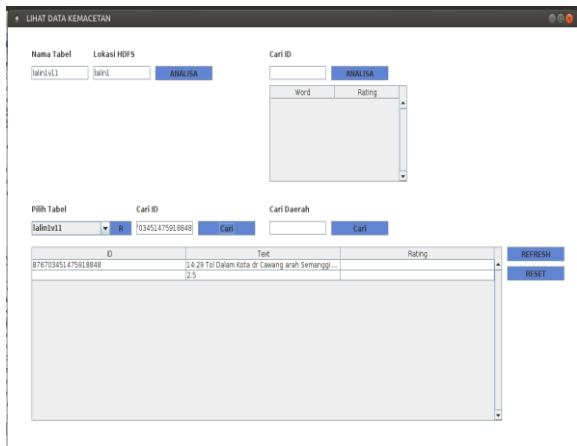
Gambar 18. Tampilan Aplikasi untuk Menentukan Tabel yang di Analisa

Pada gambar 18, aplikasi melakukan proses penyimpanan ke database dengan memasukkan Nama Tabel dan Lokasi HDFS dimana tweet tersebut disimpan setelah melakukan proses streaming.



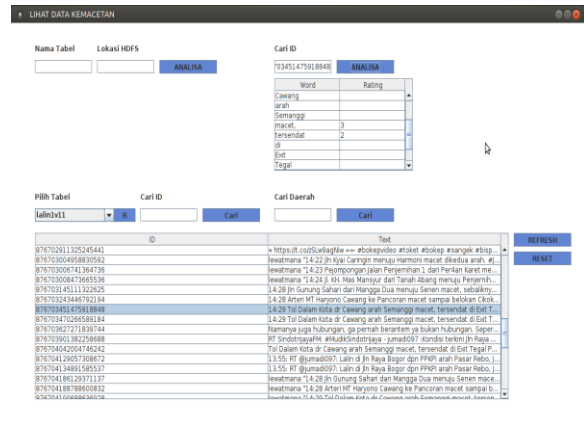
Gambar 19. Tampilan Aplikasi untuk Data pada Tabel yang di Analisa

Pada gambar 19 Aplikasi menampilkan Tabel yang dipilih melalui ComboBox dengan mengklik tombol REFRESH.



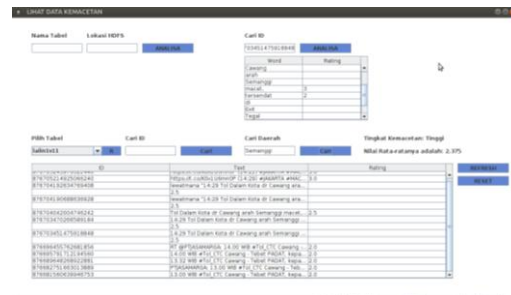
Gambar 20. Tampilan Aplikasi untuk Hasil Pencarian Data

Pada gambar 20, aplikasi menampilkan data dengan melakukan pencarian pada kolom Cari ID dan mengklik tombol CARI.



Gambar 21. Tampilan Aplikasi untuk Analisa Data

Pada gambar 21, aplikasi menampilkan di tabel bagian atas nilai-nilai tiap kata dengan melakukan pencarian pada kolom Cari ID di bagian atas dan mengklik tombol ANALISA.



Gambar 22. Tampilan Aplikasi untuk Hasil Analisa Sentimen

Pada gambar 22, aplikasi menampilkan hasil analisa sentimen pada suatu wilayah dengan mengisi kolom Cari Wilayah dan mengklik tombol CARI.

J. Analisa Hasil Penelitian

Terdapat 3 kategori kemacetan berdasarkan warna di Google Maps dan Penulis memberikan nilai pada setiap warna, dan dijelaskan pada tabel berikut.

Tabel 1. Kriteria Kemacetan berdasarkan GMaps

| Simbol | Warna | Keterangan | Nilai |
|--------|--------|------------|-------|
| | Hijau | Lancar | 1 |
| | Orange | Padat | 2 |
| | Merah | Macet | 3 |

Pada tabel 1, terdapat empat simbol yang menunjukkan warna, yaitu warna hijau untuk lancar, orange untuk padat, dan merah untuk macet.

Selanjutnya untuk menghitung akurasi Aplikasi Deteksi Kemacetan dilakukan perbandingan sample dengan Google Maps dengan menghitung nilai tiap

daerah untuk mendapatkan akurasi tiap daerah dan akurasi rata-rata. Rumus untuk menghitung jumlah akurasi adalah sebagai berikut:

$$\text{Akurasi} = (\text{Nilai terendah}) / (\text{Nilai terbesar}) \times 100$$

Dibawah ini merupakan tabel perbandingan tersebut.

Tabel 2. Sample Penilaian Daerah Kemacetan

| Daerah | Sample Tweet | Data Google Maps | Akurasi |
|-------------------|--------------|------------------|---------|
| | Nilai | Nilai | |
| Semanggi | 2.375 | 2.5 | 95% |
| Pancoran | 2.467 | 1.5 | 61% |
| Tebet | 2 | 3 | 67% |
| Pondok Indah | 3 | 3 | 100% |
| Akurasi Rata-rata | | | 81% |

Pada tabel 2, menampilkan akurasi rata-rata sebesar 81% yang didapatkan dengan membagi nilai terendah dengan nilai terbesar pada setiap daerah seperti yang dijelaskan pada rumus sebelumnya. Kemudian untuk mendapatkan akurasi rata-rata yaitu dengan menjumlahkan ke empat akurasi masing-masing daerah dan dibagi berdasarkan banyaknya daerah.

IV. PENUTUP

A. Kesimpulan

Dari teori-teori dan pembahasan sebelumnya, penulis mendapatkan kesimpulan yaitu dengan Aplikasi Deteksi Kemacetan ini dapat memberikan informasi kemacetan di daerah Jakarta serta memberikan informasi mengenai tingkat kemacetan daerah tersebut. Aplikasi ini dapat menghubungkan data twitter dengan analisa sentimen. Data Twitter digunakan pada saat pengumpulan data tweets pengguna yang memposting keadaan suatu daerah Jakarta yakni macet, padat, maupun lancar. Untuk dapat mengambil Data Twitter, Aplikasi dikoneksikan dengan Flume, Hive, dan Derby dengan membuat twitter apps untuk mendapatkan Consumer Key (API key), Consumer Secret (API secret), Access Token, dan Access Token Secret. Kemudian dimasukkan ke dalam konfigurasi twitter.conf pada Flume agar Flume dapat melakukan streaming data API Twitter.

Setelah data pada Twitter API dapat di streaming, penulis merancang Aplikasi dengan diagram UML, dan hasilnya fitur-fitur pada Aplikasi Deteksi Kemacetan dapat berfungsi sesuai dengan diagram UML yang dirancang sebelumnya. Pada Aplikasi Deteksi Kemacetan, hasil streaming data ditampilkan dan data tersebut dianalisa dengan menggunakan analisa sentimen dan hasilnya yaitu memberikan informasi tingkat kemacetan di Jakarta sesuai daerah yang dicari. Setelah mengetahui tingkat kemacetan suatu daerah, pengujian dilakukan pada Aplikasi Deteksi Kemacetan yang dibandingkan dengan Google Maps. Pengujian dilakukan dengan menggunakan 4 sample daerah di Jakarta. Sehingga penulis mendapatkan akurasi sebesar 81%. Akurasi tersebut cukup besar untuk sebuah pengujian perangkat lunak.

B. Saran

Diharapkan untuk pengembangan sistem ini selanjutnya dapat menampilkan gambar wilayah lalu lintas yang dicari keadaan lalu lintas dan penyebab lalu lintas serta meningkatkan keakuratan dalam tingkat kemacetan pada suatu daerah.

V. REFERENSI

- [1] S. Bhosale and S. Kokate, "Traffic Detection Using Tweets on Twitter Social Network," *Int. J. Sci. Res. ISSN (Online Index Copernicus Value Impact Factor)*, vol. 14611, no. 12, pp. 2319–7064, 2013.
- [2] R. P. Agus, "IPTEK: Solusi Komprehensif Atasi Kemacetan Lalu Lintas." [Online]. Available: <http://dikti.go.id/iptek-solusi-komprehensif-atasi-kemacetan-lalu-lintas/>. [Accessed: 22-May-2017].
- [3] S. Kamran, M. Shaikh, A. Naseem, and P. Kamble, "Exploiting Social Media Data for Traffic Monitoring Using the Techniques of Data Mining," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 4, no. 3, 2016.
- [4] R. S. Aisah, "Analisis Klasifikasi Sentimen Twitter terhadap Kinerja Layanan Provider Telekomunikasi menggunakan varian Naïve Bayes," Institut Pertanian Bogor, 2016.
- [5] S. Shinde and P. S. Jagtap, "Big data solution for improving traffic management System effectively for vehicular networks," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 5, no. 6, pp. 800–805, 2016.
- [6] M. Vadivukarassi, N. Puviarasan, and P. Aruna, "Sentimental Analysis of Tweets Using Naive Bayes Algorithm," *World Appl. Sci. J.*, vol. 35, no. 1, pp. 54–59, 2017.
- [7] H. Rajwani, S. Somvanshi, A. Upadhye, R. Vaidya, and T. Dange, "Dynamic Traffic Analyzer Using Twitter," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 4, no. 10, pp. 2013–2016, 2015.
- [8] M. Cindy, "What Is Twitter API," 2015. [Online]. Available: <https://www.quora.com/What-is-Twitter-API>. [Accessed: 22-May-2017].
- [9] Sangeeta, "Twitter Data Analysis Using FLUME & HIVE on Hadoop Framework," vol. 4, no. 2, pp. 119–123, 2016.
- [10] R. Wardhani and M. H. Yaqin, "Game Dasar-Dasar Hukum Islam Dalam Kitab Mabadi ' ul Fiqh Jilid I," *Teknika*, vol. 5, no. 2, pp. 473–478, 2013.
- [11] R. Margaret, "Definition NetBeans," 2014. [Online]. Available: <http://www.theserverside.com/definition/NetBeans>. [Accessed: 03-Jul-2017].
- [12] Anonim, "NetBeans IDE 8.2 Information," 2016. [Online]. Available: <https://netbeans.org/community/releases/82/>. [Accessed: 03-Jul-2017].
- [13] Y. Emma, *Sharia Banking Information System Analysis and Design*. CCIT-FTUI, 2012.