# BAB 3
# ANALISIS DAN PERANCANGAN

## 3.1 Analisis Sistem

Analisis sistem berjutuan untuk mengidentifikasi masalah yang muncul pada pembangunan sistem, hal ini bertujuan untuk membantu ketika proses perancangan berlangsung. Dalam analisa sistem ini, meliputi beberapa bagian, yaitu:

1. Analisis Masalah
2. Analisis Karakteristik Sumber Data Informasi Kemacetan
3. Analisis Sistem
4. Analisis Kebutuhan Non Fungsional
5. Analisis Kebutuhan Fungsional

### 3.1.1 Analisis Masalah

Analisis Masalah dari pengekstraksian informasi yang akan di bangun adalah:

1 Bagaimana cara mengekstrak informasi dari media sosial Twitter ke dalam kelas-kelas yang telah ditentukan sebelumnya.

2 Bagaimana caranya agar hasil pengekstraksian informasi dapat di visualisasikan pada peta *smartphone* berbasis Android.

### 3.1.2 Analisis Karakteristik Sumber Data Informasi Kemacetan

Langkah pertama yang harus dilakukan agar dapat menghasilkan pengkestraksian informasi kemacetan yang baik adalah dengan mempelajari bagaimana sumber data dalam menyampaikan informasi. Dalam menyampaikan informasi sumber data tersebut memiliki beberapa aturan dan objek.

### 3.1.2.1 Media Sosial Twitter

Dalam menyampaikan informasi media Sosial Twitter mempunyai beberapa objek. Berikut adalah penjelasan mengenai objek-objek yang ada pada Twitter dengan contoh informasi kemacetan di kota Bandung Jalan Supratman.

a. Tweet

Tweet merupakan pesan yang dapat menyampaikan informasi setiap penggunanya dengan maksimal karakter yaitu, 140 karakter. Setiap ada tweet yang baru, maka tweet tersebut akan berada pada *history* tweet paling atas.




**Gambar 3-1 Kumpulan tweet yang menyampaikan informasi kemacetan dan tweet dari akun Irfa Yuliandhari yang berada pada history tweet paling atas**

b. Hashtag (#)

Hashtag merupakan simbol untuk menandai kata kunci atau topik informasi dari sebuah tweet. Untuk daerah Bandung biasanya menandai kondisi kemacetan di kota Bandung dengan menggunakan hastag #LalinBdg atau #lalinBdg.




**Gambar 3-2 Hashtag (#) dengan kata kunci #LalinBdg**

c. Re-Tweet

Re-Tweet adalah cara untuk mengulang tweet pada sebuah topik yang sedang dibicarakan, selain itu re-tweet juga berfungsi untuk membalas tweet akun lain dengan mengetikan kata "RT" didepan nama akun yang di-re-tweet.




**Gambar 3-3 Tweet dari akun All About Bandung yang di-re-tweet oleh akun Irfa Yuliandhari**

d. At (@)

Merupakan tanda yang manandakan nama akun dari pengguna Twitter, atau bisa juga sebagai tanda untuk menandai seseorang pada sebuah tweet (*mention*).




**Gambar 3-4 @infobdg merupakan nama akun Twitter dari All About Bandung**




**Gambar 3-5 Mention yang dilakukan oleh akun Dewi U Prasetiowati (@dewiutarip) kepada akun All About Bandung (@infobdg)**

Selain dari kelima objek di atas, adapula beberapa objek umum lainnya seperti twittersearch dan waktu tweet berlangsung. Twittersearch akan mencari kata sesuai inputan dengan mengabaikan besar kecilnya huruf.




**Gambar 3-6 Tweetsearch**




**Gambar 3-7 Tweet dari akun All About Bandung (@infobdg) yang menandakan tweet-nya terjadi pada tanggal 17 Januari**

Berdasarkan hasil analisis dengan contoh informasi kemacetan kota Bandung di atas, didapatkan rangkuman hasil dari analisis karakteristik media sosial Twitter, yaitu sebagai berikut:

**Tabel 3-1 Tabel rangkuman hasil analisis karakteristik media sosial Twitter dengan contoh lokasi di kota Bandung Jalan Supratman**

| No | Nama Objek | Hasil |
|---|---|---|
| 1 | Pencarian (#lalinBdg Supratman) | #LalinBdg |
| 2 | Tweet (kondisi kemacetan) | Macet |
| 3 | Tweet (tempat kemacetan) | Jl A.Yani arah cicadas/cicaheum, jl. supratman |

### 3.1.2.2 *Website* Lewat Mana

Dalam penyampaian informasi kemacetan, *website* Lewat Mana mempunyai kemiripan dengan sumber data yang telah dijelaskan sebelumnya, yaitu media sosial Twitter. Kemiripan tersebut adalah pada bentuk penyampaiannya yang berupa *timeline*. Berikut adalah *screenshot website* Lewat Mana:




**Gambar 3-8 Gambar *screenshot website* Lewat Mana**

Berikut adalah *screenshoot* penyampaian informasi kemacetan pada *website* Lewat Mana yang berupa *timeline* dengan contoh informasi kemacetan di kota Bandung Jalan Suprtaman:




**Gambar 3-9 Gambar *screenshot website* Lewat Mana**

Berdasarkan hasil analisis dengan contoh informasi kemacetan kota Bandung Jalan Jakarta di atas, didapatkan rangkuman hasil dari analisis karakteristik *webiste* Lewat Mana, yaitu sebagai berikut:

**Tabel 3-2 Tabel rangkuman hasil analisis karakteristik *website* Lewat Mana dengan contoh lokasi di kota Bandung**

| No | Nama Objek | Hasil |
|---|---|---|
| 1 | Pencarian<br>(Bandung Jalan Supratman) | Bandung, bandung |
| 2 | kondisi kemacetan | padat, macet |
| 3 | tempat kemacetan | Supratman, supratman |

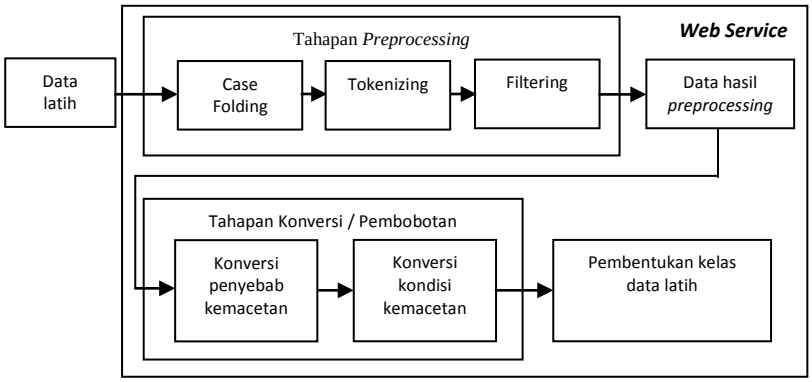
Sesuai dengan kedua hasil analisis diatas, maka dapat disimpulkan bahwa format kondisi kemacetan yang akan diambil sebagai informasi kemacetan adalah kata-kata baku saja, hal tersebut dilakukan agar sumber data baru yang masuk ke dalam sistem dapat langsung di ekstrak informasinya.

### 3.1.3 Analisis Sistem

Sistem yang akan dibangun meliputi cakupan dua bagian, yaitu dari *web service* dan klien (Android). Proses ekstraksi, dilakukan pada sisi *web service* sedangkan pada sisi klien (Android), dilakukan proses pengambilan data hasil ekstraksi yang dikombinasikan dengan data spasial atau peta. Berikut adalah diagram alur dari sistem yang akan di bangun:

1. Data Latih

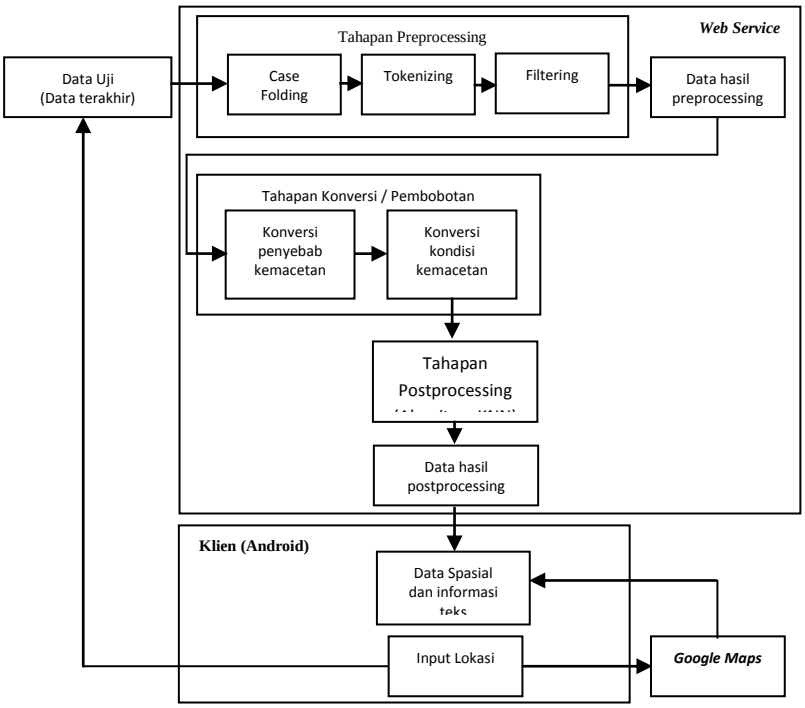
   Pada tahap ini, data latih akan melalui tahapan *preprocessing* untuk tahapan *cleaning* datanya, setelah mendapatkan hasil dari tahap *preprocessing*, lalu dilakukan pembobotan serta pembentukan kelas data latih.




**Gambar 3-10 Gambar diagram alur data latih pada *web service***

2. Data Uji

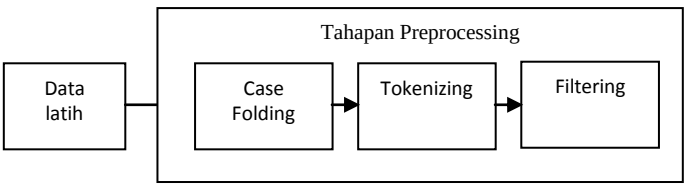
   Pada tahap ini, data uji atau dalam hal ini adalah data terakhir berdasarkan inputan lokasi dari pengguna, melalui tahap *preprocessing* sebagai *cleaning* datanya serta melalui tahap pembobotan dan tahap *postprocessing* dengan menggunakan algoritma KNN untuk mendapatkan kelas dari data uji tersebut.




**Gambar 3-11 Gambar diagram alur data uji pada *web service* dan pada klien (Android)**

### 3.1.3.1 Analisis *Preprocessing* Ekstraksi Informasi Dari Sumber Data Informasi Kemacectan

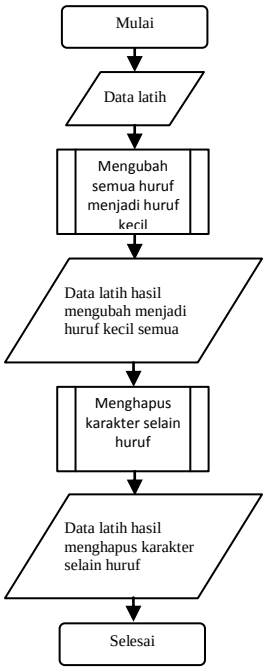
Pada tahap preprosesing terdapat beberapa tahapan yang harus dilalui sebelum inputan (data latih) masuk ke dalam tahapan algoritma KNN. Tahapan tersebut antara lain, *Case Folding*, *Tokenizing* dan *Filtering*.




**Gambar 3-12 Alur kerja tahapan preprocessing**

#### 3.1.3.1.1 *Case Folding*

*Case folding* merupakan tahapan mengubah semua inputan huruf menjadi huruf kecil semua (*lower case*). Pada tahap ini juga karakter selain huruf akan dihilangkan. Hal ini dapat memudahkan ketika tahapan *tokenizing* berlangsung. Berikut adalah alur proses tahapan dari *case folding*:




**Gambar 3-13 Flowchart tahapan case folding**

Pada tahapan ini, ada beberapa aturan proses agar hasil *case folding* dapat sesuai dengan yang diinginkan. Adapun aturan-aturan tersebut adalah:

**Tabel 3-3 Tabel aturan pada tahapan *case folding***

| Kondisi | Aksi |
|---|---|
| Inputan memiliki huruf kapital | Mengubah semuan inputan menjadi huruf kecil |
| Inputan memiliki karakter khusus | Menghapus karakter khusus tersebut dari inputan |
| Inputan memiliki spasi (*white space*) | Tidak ada aksi |
| Inputan memiliki huruf kecil | Tidak ada aksi |

Adapun karakter-karakter khusus tersebut, yaitu:

**Tabel 3-4 Tabel karakter-karakter khusus**

| Karakter khusus | | |
|---|---|---|
| ^ | , | : |
| @ | ( | ; |
| % | ) | \| |
| $ | - | ] |
| # | < | } |
| ‘ | > | { |
| ~ | + | \ |
| & | = | / |
| ! | * | “ |
| ` | _ | 1 |
| 2 | 3 | 4 |
| 5 | 6 | 7 |
| 8 | 9 | 0 |

Berikut adalah contoh penerapan tahap *case folding* pada penelitian ini:

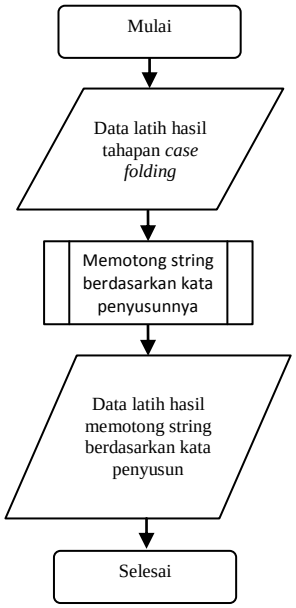
**Tabel 3-5 Tabel penerapan Case Folding**

| Nama | Contoh data | |
|---|---|---|
| | **Twitter** | **Lewat Mana** |
| Input | A. Yani arah cicadas,, macet dr mulai perempatan jl. Supratman | macet parah dr mulai jembatan layang kircon sampe jl supratman, karna ad bbrapa ruas jalan yg trgnang banjir |
| Output | a yani arah cicadas macet dr mulai perempatan jl supratman | macet parah dr mulai jembatan layang kircon sampe jl supratman karna ad bbrapa ruas jalan yg trgnang banjir |

#### 3.1.3.1.2 Tokenizing

Pada tahap *tokenizing* dilakukan pemotongan string berdasarkan setiap kata penyusunnya. Pemotongan tersebut dilakukan ketika inputan memiliki spasi (*white space*).

Berikut adalah alur proses tahapan *tokenizing*:




**Gambar 3-14 Flowchart tahapan tokenizing**

Pada tahapan ini, ada beberapa aturan proses agar hasil *tokenizing* dapat sesuai dengan yang diinginkan. Adapun aturan-aturan tersebut adalah:

**Tabel 3-6 Tabel aturan pada tahapan *tokenizing***

| Kondisi | Aksi |
|---|---|
| Inputan memiliki spasi (*white space*) | Memotong menjadi per tiap bagian kata atau string |
| Inputan memiliki kata pada d*atabase wordlist* | Tidak ada aksi |

Contoh tahapan *tokenizing* pada penilitian ini yaitu:

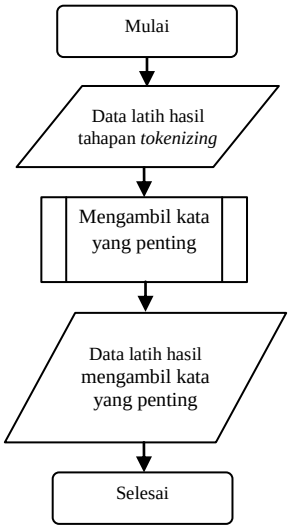
**Tabel 3-7 Tabel penerapan tahap *tokenzing***

| Nama | Contoh data | |
|---|---|---|
| | Twitter | Lewat Mana |
| Input | a yani arah cicadas macet dr mulai perempatan jl supratman | macet parah dr mulai jembatan layang kircon sampe jl supratman karna ad bbrapa ruas jalan yg trgnang banjir |
| Output | String 1: a<br>String 2: yani | String 1: macet<br>String 2: parah |

| Nama | Contoh data | |
|---|---|---|
| | **Twitter** | **Lewat Mana** |
| | String 3: arah<br>String 4: cicadas<br>String 5: macet<br>String 6: dr<br>String 7: mulai<br>String 8: perempatan<br>String 9: jl<br>String 10: supratman | String 3: dr<br>String 4: mulai<br>String 5: jembatan<br>String 6: layang<br>String 7: kircon<br>String 8: sampe<br>String 9: jl<br>String 10: supratman<br>String 11: karna<br>String 12: ad<br>String 13: bbrapa<br>String 14: ruas<br>String 15: jalan<br>String 16: yg<br>String 17: trgnang<br>String 18: banjir |

#### 3.1.3.1.3 *Filtering (Word List)*

*Filtering* pada penelitian ini merupakan tahapan mengambil kata yang penting dari tahap *tokenizing*. Pada penelitian ini menggunakan algoritma *worldlist* untuk menyimpan kata penting tersebut. Berikut adalah alur proses tahapan *filtering*:




**Gambar 3-15 Flowchart tahapan *filtering***

Pada tahapan ini, ada beberapa aturan proses agar hasil *filtering* dapat sesuai dengan yang diinginkan. Adapun aturan-aturan tersebut adalah:

**Tabel 3-8 Tabel aturan pada tahapan *filtering***

| Aturan | Hasil |
|---|---|
| Inputan cocok dengan isi dari *database wordlist* | Menyimpan kata atau *string* tersebut pada *database* |
| Inputan cocok dengan inputan nama jalan dari pengguna | Tidak menghapus nama jalan tersebut |
| Inputan tidak cocok dengan isi dari *database wordlist* | Tidak menyimpan kata atau *string* tersebut pada *database* |

Isi pada *database wordlist* merupakan kumpulan kata-kata penting yang relevan dengan penelitian. Kata-kata tersebut diambil dari hasil kuesioner kepada para pengguna jalan. Adapun isi dari *database wordlist* tersebut, yaitu:

**Tabel 3-9 Kata pada database *wordlist***

| **Kata pada *database wordlist*** | | | | |
|---|---|---|---|---|
| **Kondisi** | | | **Kombinasi** | |
| Macet Sekali | | | Macet + sekali | |
| Macet Parah | | | Macet + parah | |
| Macet | | | Macet | |
| Padat Merayap | | | Padat + Merayap | |
| Padat | | | Padat | |
| **Penyebab** | | | **Kombinasi** | |
| Banjir | | | Banjir | |
| Perbaikan Jalan | | | Perbaikan + Jalan | |
| Kecelakaan | | | Kecelakaan | |
| **Nama Jalan** | | | | |
| Jl. A.B.C. | Jl. A.B.C. | Jl. Neglasari | Jl. Ayudia | Jl. Aksan |
| Jl. Alani | Alun-alun Timur | Jl. Jakin | Jl. Ambon | JI. Karangarum |
| Jl. Nanas | Jl. Angandara | Jl. Sirnagalih | Jl. Tasikmalaya | Jl. Arjuna |
| Jl. Astanaanyar | Jl. Astina | Jl. Bengkel | Jl. Aceh | Jl. Babakan |

| **Kata pada *database wordlist*** | | | | |
|---|---|---|---|---|
| | | | | Ciamis |
| JI. Babatan | Jl. Bali | Jl. Balonggede | J1. Banda | Jl. Mohamad Toha |
| JI. Bangka | Jl. Banceuy | Jl. Barata | Jl. Cendana | Jl. Baros |
| Jl. Bawean | Jl. Badaksinga | JI. Dipati Ukur | Jl. Bengawan | Jl. Ciumbulleuit |
| Jl. Ciungwanara | JI. Hsssanudin | J1. Ranggamalela | Jl. Belitung | Jl. Blanak |
| Jl. Blimbing | J1. Buahbatu | Jl. Bukit Jarian | J1. Bukit Tunggul | JI. Bungur |
| J1. Bungsu | J1. Burangrang | Jl. Aceh | JI. Dokter Sutomo | J1. Borneo |
| J1. Suria Kancana | Jl. Panumbangjaya | Jl. Ranggagading | Jl.Soka | J1. Tarate |
| J1. Kidang Pananjung | Jl. Geusanulun | J1. Braga | Jl. Brantas | Jl. Jatayu |
| Jl. Sukajadi | Jl. Marconi | J1. Hegarmanah Wetan | Jl. Hegarmanah Kulon | Jl. Pajajaran |
| Jl. Purnawarman | Jl. Dokter Wahidin | Jl. Kembang Tanjung | Jl. Culan | JI. Karangtineung |
| Jl.Kembang Dayang | JI. Sukawangi | Jl. Sulawesi | JI. Seram | Jl. Pecinan Lama |
| Jl. Malabar | Jl. Keristelek | Jl. Kolektor | Jl. Erguloh | Jl. Kejaksaan |
| Jl. Curie | Jl. Sawunggaling | Jl. Jakarta | JI. Dago/JI.Ir. Haji Juanda | Jl. Dahlia |
| JI. Dalem Kaum | Jl. Pabrik Daging | Jl. Sukajadi | JI. Cimandiri | Jl. Dasarata |
| JI. Dederuk | Jl. Karangsari | Jl. Hegarmanah | Jl. Van Deventer | Jl. Raden Dewi |
| JI. Kembang Sepatu | Jl. Jambal | JI. Japati | Jl. Jeruk | Jl. Johar |
| JI. Dukuh | JI. Dursasana | Jl. Dorna | JI. Ehrlich | Jl. Wastukancana |
| Jl. Cikutra | JI. Prof. Eyckman | Jl. Industri | JI. Haji Wasyid | Jl. Pajajaran |
| Jl. Sirnasari | Jl. Flores | Jl. Garuda | Jl. Sultan Tirtayasa | Jl. Gabus |

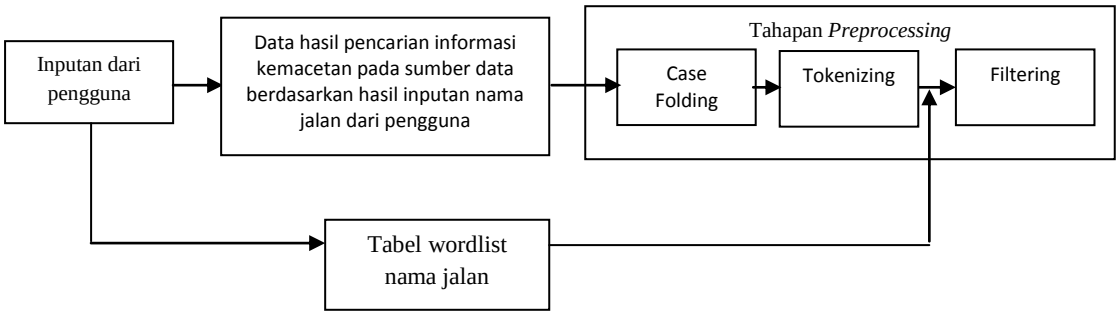
| **Kata pada *database wordlist*** | | | | |
|---|---|---|---|---|
| Jl. Lamping | Jl. Galunggung | Jl. Sirnamanah | Jl. Gardujati | Jl. Bahureksa |
| Jl. Taman Sari | J1. Gunung Agung | Jl. Gunung Kareumbi | Jl. Gunung Putri | Jl. Guntur |
| Jl. Lebak | Jl. Karanglayung | Jl.Gajahiumantung | Jl. Siliwangi | Jl. Dokter Sukimin |
| Jl. Linggawastu | Jl. Pa Jiping | Jl. Halimun | Jl. Halmahera | Jl. Kiayi Gede Utama |
| Jl. Tubagus Ismail | J1.Hata | J1. Jamuju | J1. Sultan Agung | Jl. Lamping |
| Jl. Dokter Rajiman | Jl. Menjangan | Jl. Serang | Jl. Tengku Angkasa | Jl. Somawinata |
| Jl. Abdurachman Saleh | J1. Bagusrangin | Jl. Wira Angun-angun | Jl. Hegarmanah | Jl. Dokter Susilo |
| J1. Ganeca | J1. Pangeran Kornel | Jl. Hotel Homan | Jl. Taman Sari | Lapangan Ciujung |
| Jl. Clujung | Jl. Suren | Jl. Sumur Bandung | Jl. Taman Sari | Jl. Sukabumi |
| Taman Nusantara/Tm.Lalu Lintas | Jl. Surapati | Lapangan Jap Lun | J1. Jap Lun | Jl. Jawa |
| J1. Jun Liong | Jl. Aryajipang | Jl. Jo Sun Bie | Taman Sari | Jl. Kiputih |
| Jl. Surapati | J1. Kalih Apo | JI. Kamuning | Jl. Cisanggarung | Jl. Cihapit |
| JI. Aceh | JI. Kenari | Jl. Kapatihan | Jl. Karapitan | JI. Kartini |
| Jl. Kebonjati | Jl. Kebon Jukut | Jl. Setasion Timur | Jl. Kebon Kawung | JI. Mohamad Toha |
| Jl. Kebon Sirih | Jl. Aceh | Kebon Waru Tengah | Kebon Waru Utara | Lapangan Kebon Waru |
| Kebon Waru Selatan | Jl. Kejaksaan | Jl. Kencana | Jl. Gereja | JI. Kelenteng |
| JI. Karangtingal | Jl.. Cisangkuy | Jl. Cilaki | JI. Kiaracondong | JI. Kihiur |
| Jl. Pajajaran . | Jl. Tengku Angkasa | J1. Mundinglaya | Jl. Haji Akbar | J1.Kopo |
| Jl. Korawa | Jl.. Kosambi | Jl. Rakata | J1. Kresna | Jl. Dayang |

| **Kata pada *database wordlist*** | | | | |
|---|---|---|---|---|
| | | | | Sumbi |
| Jl. Begawan Sempani | Jl. Layang | Jl.. Laksana | Jl. Sangkuriang | Jl. Babakan Ciamis |
| J1. Gereja/J1. Perintis Kemerdekaan | Jl. Nijiand | Jl.. Cihampelas | Jl. Lempuk | J1. Lengkong Besar |
| Jl. Lengkong Kecil | Jl. Lesmana | J1.:Rangga Gempol | Lapangan Bengawan | Jl.. Marjuk |
| Jl. Haji Moh. Iskat | Jl. Wastukancana | J1.Lombok | Jl. Setasion Selatan | Jl. Gambir |
| Jl. Gelapnyawang | Jl. Madura | Jl. Adipati Kertabumi | Jl. Gudang Utara | Jl. Gudang Selatan |
| JL Sukaresmi | JL Malabar | Jl. Malabar | JI. Malabar | JL Sangkuning |
| JL Mangga | JL Manglit | Lapangan Cirebon | JI. Cirebon | J1. Mataramatrako |
| J1. Patrakomala | J1. Situ | J1. Anggrek | J1.Aceh | JI. Merak |
| JI. Pajajaran | J1. Merdeka | Jl. Tamblong Dalam | J1. Moch. Yunus | JI. Taman Maluku |
| JI. Garut | J1. Raden Patah | J1. Bogor | JI. Sulanjana | JI. Nakula |
| J1. Nangka | JI. Naripan | JI. Naripan Dalam | J1. Gandapura | JI. Cipunagara |
| J1.Nias | JI. Kaca-kaca Wetan | J1. Ksatriyan | J1. Ciwulan | J1. Cipaganti |
| J1. Tongkeng | J1. Panaitan | J1. Sunda | Lapangan Dipati Ukur | Taman Anggrek |
| Jl. Anggrek | J1. Kiayi Luhur | Jl. Lembong - JI. Bungsu | JI. Pagarsih | J1. Sukaasih |
| J1. Pandu | Jl. Otto Iskandardinata | J1.Tegallega | J1. Gatot Subroto | Jl. Taman Merdeka |
| Jl. Taman Sari | JI. Pasang | Jl. Ps.Baru | JI. Belakang Pasar | JI. Pasar Utara |
| JI. Pasar Barat | JI. Pasar Selatan | JI. Pasirkaliki | Jl. Pasirkoja | Jl. Pasundan |
| JI. Pasundan | Jl. Pasteur | JI. Patuha | Jl.Peda | JI. Pejagalan |
| Jl. Imam Bonjol | Taman Pendawa | JI. Pendawa | Jl. Pepetek | JI. Lengkong Dalam |
| Jl. Pujuh | Jl. Pulolaut | Jl. Pungkur | JI. Purabaya | JI. Puter |

| **Kata pada *database wordlist*** | | | | |
|---|---|---|---|---|
| Jl. Putri | JI. Ahmad Yani | Jl. Jenderal Sudirman | JI. Dokter Rum | JI. Singaperbangsa |
| JI. Cianjur | JI. Progo | Lapangan Pacuan Kuda | Jl. Hadi Sucipto | Jl. Kapten Sumantri |
| JI. Sumedang | J1.Rama | JI. Rambutan | JI. Rancabentang | JI. Rasamala |
| JI. Jurang | JI. Samoja | Jl. Kabupaten | JI. Kabupaten | JI. Kabupaten |
| JI. Diponegoro | Jl. Pasar Baru | Jl. Pacar | Jl. Dipati Ukur | Jl. Riau |
| Jl. Kaca Piring | Jl. Banda | Jl. Dokter Cipto | Jl. Rontgen | Jl.Riau |
| Lapangan Dokter Otten | Jl. Haji Moh. Mesri | JI. Prabu Dimuntur | Jl. Bosscha | Jl. Salam |
| Lapangan Sabang | J1. Sabang | Jl. Sadewa | Jl. Salak | Ji. Dokter Saleh |
| Jl. Samiaji | Jl. Dokter Samyudo | Jl. Saninten | Jl. Saparua | Jl. Satrugna |
| J1.Sawo | Jl. Merdeka | Jl. Sindangsirna | Jl. Semar | Jl. Sepat |
| J1. Pudak | JI. Riau | Jl. Simpang | JL Sinta | Jl. Arjuna |
| Jl. Dokter Slamet | Jl. Patrakomala | Jl. Sukapakir | Jl. Sumbadra | Jl. Sumbawa |
| Jl. Sunda | Jl. Suniaraja | Jl. Suwatama | Lapangan Panglayungan | Jl. Morse |
| Jl. Centeh | Jl. Setasion Timur | Jl. Setasion Barat | J1. Srikandi | JI. Kembang Sepatu |
| JI. Ermawar | JI. Sumarsana | JI. Astanaanyar | JI. R. Pagergunung | JI. Sumatra |
| Taman Merdeka | JI. Tamblong | J1. Tampomas | Jl. Tanjan | JI. Tanjung |
| J1. Tanjunganom | Jl. Cilaki | JI. Ciateul | Jl. Moh. Toha | Jl. Otto Iskandardinata |
| Jl. Tegallega | Jl. Tilpon | Tegallega | J1. Telukbuyung | J1. Taman Khewan |
| Jl..Teri | JI. Ternate | Jl. Terrasana | JI. Dokter Rubini | JI. Tikukur |
| JI. Banda | Jl. Dokter Abdul Rivai | Jl. Titiran | Jl. Cakranegara | Taman Campaka |
| JI. Cemara | Jl. Ciateul | Jl. Cibadak | Taman | Taman |

| Kata pada *database wordlist* | | | | |
|---|---|---|---|---|
| | | | Cibeunying Utara | Cibeunying Selatan |
| JI. Cibeunying | JI. Cibuni | Jl. Cibunut Dalam | Taman Cibunut | J1.Buton |
| JI. Cihapit | JI. Cikapayang | JI. Cikapundung Timur | JI. Cikapundung Barat | Jl. Cikuray |
| Jl. Imam Bonjol | Jl. Cilaki | Jl. Cilentah | J1. Ciliwung | JI. Cimanuk |
| Jl. Cipaera | JI. Sastra | JI. Ciremay | JI. Cisangkuy | Taman Citarum |
| JI. Citarum | JI. Cicendo | Jl. Citrayuda | Jl. Cumi-cumi | JI. Bapa Husen |
| Jl. Lengkong Tengah | Jl.Rais | JI. Fanatayuda | Jl. Lengkong Dalam | JI. Haji Hassan |
| JI. Multatuli | JI. Hariangbanga | JI. Dokter Gunawan | JI. Kolam | Jl. Wayang |
| JI. Aruna | JI. Waringin | Lapangan Waringin | JI. Trunojoyo | J1. Westhoff |
| JI. Suriani | JI. Wibisana | JI. Cilamaya | JI. Diponegoro | Lapangan Diponegoro |
| Jl. Windu | J1. Srigading | Jl. Banten | JI. Krawang | J1.Tera |
| Taman Ganeca | Jl. Pelana | Jl. Maulana Yusuf | Jl. Rumah Sakit | Jl. Tengku Umar |

Selain penyebab dan kondisi kemacetan, salah satu parameter yang akan menjadi acuan *filtering* adalah nama jalan. Hasil pencarian nama jalan dari kedua sumber data akan di cocokan dengan nama jalan yang ada pada tabel *wordlist*.




**Gambar 3-16 Alur kerja nama jalan sebagai paramater filtering**

Contoh tahapan Filtering pada penelitian ini, yaitu:

**Tabel 3-10 Tabel penerapan tahap *filtering***

| Nama | Contoh data | |
|---|---|---|
| | **Twitter** | **Lewat Mana** |
| Input | String 1: a<br>String 2: yani<br>String 3: arah<br>String 4: cicadas<br>String 5: macet<br>String 6: dr<br>String 7: mulai<br>String 8: perempatan<br>String 9: jl<br>String 10: supratman | String 1: macet<br>String 2: parah<br>String 3: dr<br>String 4: mulai<br>String 5: jembatan<br>String 6: layang<br>String 7: kircon<br>String 8: sampe<br>String 9: jl<br>String 10: supratman<br>String 11: karna<br>String 12: ad<br>String 13: bbrapa<br>String 14: ruas<br>String 15: jalan<br>String 16: yg<br>String 17: trgnang<br>String 18: banjir |
| Output | String 1: macet<br>String 2: supratman | String 1: macet<br>String 2: parah<br>String 3: supratman<br>String 4: banjir |

Setelah data dari Twitter dan Lewat Mana melalui tahap *preprocessing*, kedua data dari dua sumber data tersebut dimasukkan dalam satu tabel dengan data dari Twitter terlebih dahulu dan data dari Lewat Mana setelahnya. Maka didapatkan tabel sebagai berikut:

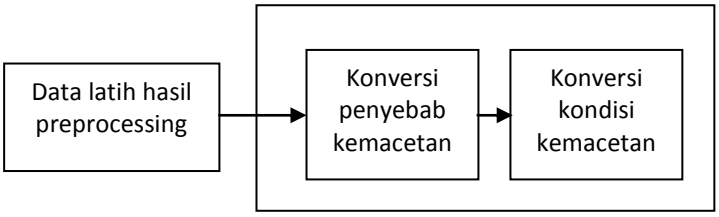
**Tabel 3-11 Tabel hasil tahap *preprocessing***

| Data-ke | Jalan | Penyebab | Kondisi |
|---|---|---|---|
| 1 | supratman | Tidak ada | Macet |
| 2 | supratman | Banjir | Macet Parah |

### 3.1.3.2 Analisis Algoritma KNN

Algoritma KNN adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Data pembelajaran diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi data pembelajaran [4].

Dalam penelitian ini, ada beberapa Proses yang harus dilakukan sebelum masuk ke dalam inti proses algoritma KNN berlangsung. Proses-proses tersebut dilakukan sebagai pembentukan data *sample* dan merupakan acuan untuk data uji yang akan dimasukan, dimana dalam hal ini data uji tersebut adalah permintaan lokasi berdasarkan inputan dari pengguna. Berikut adalah alur kerja proses tersebut:




**Gambar 3-17 Alur kerja proses konversi**

**Contoh Kasus:** Terdapat permintaan lokasi kemacetan oleh pengguna dengan Jalan Supratman di Kota Bandung. Setelah melewati tahap *preprocessing*, terdapat fakta kondisi kemacetan dari hasil permintaan tersebut:

**Tabel 3-12 Tabel fakta berdasarkan lokasi permintaan dari pengguna**

| Data-ke | Jam | Penyebab | Kondisi |
|---|---|---|---|
| 1 | 19:34 | Tidak ada | Macet |
| 2 | 19:23 | Tidak ada | Macet |
| 3 | 19:22 | Tidak ada | padat merayap |
| 4 | 19:20 | Tidak ada | macet total |
| 5 | 18:55 | Tidak ada | macet total |
| 6 | 18:54 | Tidak ada | padat merayap |
| 7 | 18:43 | Tidak ada | Macet |
| 8 | 18:27 | Tidak ada | macet total |

| Data-ke | Jam | Penyebab | Kondisi |
|---|---|---|---|
| 9 | 17:53 | Tidak ada | macet total |
| 10 | 17:43 | Tidak ada | macet total |
| 11 | 17:43 | Tidak ada | macet total |
| 12 | 17:43 | Tidak ada | macet total |
| 13 | 17:43 | Tidak ada | macet total |
| 14 | 16:38 | Tidak ada | Macet |
| 15 | 16:12 | Tidak ada | Macet |
| 16 | 16:11 | Tidak ada | padat merayap |
| 17 | 12:55 | Tidak ada | macet parah |
| 18 | 12:49 | Tidak ada | macet parah |
| 19 | 12:44 | Tidak ada | macet parah |
| 20 | 11:46 | Tidak ada | macet parah |
| 21 | 11:46 | Tidak ada | macet parah |
| 22 | 11:43 | Tidak ada | macet parah |
| 23 | 11:39 | Tidak ada | macet parah |
| 24 | 11:38 | Tidak ada | macet parah |
| 25 | 11:27 | Tidak ada | macet parah |
| 26 | 11:25 | Tidak ada | macet parah |
| 27 | 11:24 | Tidak ada | macet parah |
| 28 | 11:24 | Tidak ada | macet parah |
| 29 | 11:23 | Tidak ada | macet parah |
| 30 | 11:23 | Tidak ada | macet parah |
| 31 | 11:23 | Tidak ada | macet parah |
| 32 | 11:22 | Tidak ada | macet parah |
| 33 | 11:21 | Tidak ada | macet parah |
| 34 | 11:18 | Tidak ada | macet parah |
| 35 | 11:18 | Tidak ada | macet parah |
| 36 | 11:18 | Tidak ada | macet parah |
| 37 | 11:17 | Tidak ada | macet parah |
| 38 | 11:17 | Tidak ada | macet parah |
| 39 | 11:17 | Tidak ada | macet parah |
| 40 | 11:17 | Tidak ada | macet parah |
| 41 | 11:17 | Tidak ada | macet parah |

| Data-ke | Jam | Penyebab | Kondisi |
|---|---|---|---|
| 42 | 11:17 | Tidak ada | macet parah |
| 43 | 11:16 | Tidak ada | macet parah |
| 44 | 11:16 | Tidak ada | macet parah |
| 45 | 11:16 | Tidak ada | macet parah |
| 46 | 11:16 | Tidak ada | macet parah |
| 47 | 11:16 | Tidak ada | macet parah |
| 48 | 11:15 | Tidak ada | macet parah |
| 49 | 11:15 | Tidak ada | macet parah |
| 50 | 10:07 | Tidak ada | padat merayap |
| 51 | 9:32 | Tidak ada | padat merayap |
| 52 | 9:19 | Tidak ada | Macet |
| 53 | 9:15 | Tidak ada | Macet |
| 54 | 9:11 | Tidak ada | Macet |
| 55 | 8:58 | Tidak ada | Padat |

Pertama, konversikan penyebab kemacetan hasil *preprocessing* menjadi dalam bentuk nilai atau angka. Berdasarkan fakta pada tabel *wordlist*, terdapat tiga penyebab masalah kemacetan, yaitu banjir, perbaikan jalan dan kecelakaan. Ketiga permasalahan tersebut diberikan bobot dengan nilai masing-masing 1 (satu) untuk kecelakaan, 2 (dua) untuk perbaikan jalan, 3 (tiga) untuk banjir dan 0 (nol) jika tidak ada keterangan penyebab kemacetan. Pembobotan nilai atau angka tersebut didapatkan berdasarkan hasil kuesioner.

**Tabel 3-13 Tabel bobot nilai dari penyebab kemacetan**

| Penyebab | Bobot |
|---|---|
| Banjir | 3 |
| Perbaikan Jalan | 2 |
| Kecelakaan | 1 |

Setelah dilakukan konversi penyebab kemacetan tersebut menjadi nilai atau angka, maka terdapat tabel sebagai berikut:

**Tabel 3-14 Tabel penyebab kemacetan yang telah dikonversi**

| Data-ke | Jam | Penyebab | Kondisi |
|---|---|---|---|
| 1 | 19:34 | 0 | Macet |
| 2 | 19:23 | 0 | Macet |
| 3 | 19:22 | 0 | padat merayap |
| 4 | 19:20 | 0 | macet total |
| 5 | 18:55 | 0 | macet total |
| 6 | 18:54 | 0 | padat merayap |
| 7 | 18:43 | 0 | Macet |
| 8 | 18:27 | 0 | macet total |
| 9 | 17:53 | 0 | macet total |
| 10 | 17:43 | 0 | macet total |
| 11 | 17:43 | 0 | macet total |
| 12 | 17:43 | 0 | macet total |
| 13 | 17:43 | 0 | macet total |
| 14 | 16:38 | 0 | Macet |
| 15 | 16:12 | 0 | Macet |
| 16 | 16:11 | 0 | padat merayap |
| 17 | 12:55 | 0 | macet parah |
| 18 | 12:49 | 0 | macet parah |
| 19 | 12:44 | 0 | macet parah |
| 20 | 11:46 | 0 | macet parah |
| 21 | 11:46 | 0 | macet parah |
| 22 | 11:43 | 0 | macet parah |
| 23 | 11:39 | 0 | macet parah |
| 24 | 11:38 | 0 | macet parah |
| 25 | 11:27 | 0 | macet parah |
| 26 | 11:25 | 0 | macet parah |
| 27 | 11:24 | 0 | macet parah |
| 28 | 11:24 | 0 | macet parah |
| 29 | 11:23 | 0 | macet parah |
| 30 | 11:23 | 0 | macet parah |
| 31 | 11:23 | 0 | macet parah |
| 32 | 11:22 | 0 | macet parah |

| Data-ke | Jam | Penyebab | Kondisi |
|---|---|---|---|
| 33 | 11:21 | 0 | macet parah |
| 34 | 11:18 | 0 | macet parah |
| 35 | 11:18 | 0 | macet parah |
| 36 | 11:18 | 0 | macet parah |
| 37 | 11:17 | 0 | macet parah |
| 38 | 11:17 | 0 | macet parah |
| 39 | 11:17 | 0 | macet parah |
| 40 | 11:17 | 0 | macet parah |
| 41 | 11:17 | 0 | macet parah |
| 42 | 11:17 | 0 | macet parah |
| 43 | 11:16 | 0 | macet parah |
| 44 | 11:16 | 0 | macet parah |
| 45 | 11:16 | 0 | macet parah |
| 46 | 11:16 | 0 | macet parah |
| 47 | 11:16 | 0 | macet parah |
| 48 | 11:15 | 0 | macet parah |
| 49 | 11:15 | 0 | macet parah |
| 50 | 10:07 | 0 | padat merayap |
| 51 | 9:32 | 0 | padat merayap |
| 52 | 9:19 | 0 | Macet |
| 53 | 9:15 | 0 | Macet |
| 54 | 9:11 | 0 | Macet |
| 55 | 8:58 | 0 | Padat |

Kedua, konversikan kondisi kemacetan hasil preprosesing ke dalam bentuk nilai atau angka. Sesuai dengan data fakta pada tabel *wordlist*, terdapat lima kondisi kemacetan, yaitu padat, padat merayap, macet, macet parah dan macet total. Lima kondisi kemacetan tersebut diberikan bobot dengan nilai masing-masing 1 (satu) untuk padat dan padat merayap, 2 (dua) untuk macet, 3 (tiga) untuk macet parah dan macet total. Pembobotan nilai dan angka tersebut didapatkan berdasarkan hasil kuesioner.

**Tabel 3-15 Tabel bobot nilai dari kondisi kemacetan**

| Kondisi | Bobot |
|---|---|
| Macet Sekali / Macet Parah | 3 |
| Macet | 2 |
| Padat Merayap / Padat | 1 |

Dengan melakukan konversi tersebut, maka didapatkan tabel sebagai berikut:

**Tabel 3-16 Tabel kondisi kemacetan yang telah dikonversi**

| Data-ke | Jam | Penyebab | Kondisi |
|---|---|---|---|
| 1 | 19:34 | 0 | 2 |
| 2 | 19:23 | 0 | 2 |
| 3 | 19:22 | 0 | 1 |
| 4 | 19:20 | 0 | 2 |
| 5 | 18:55 | 0 | 2 |
| 6 | 18:54 | 0 | 1 |
| 7 | 18:43 | 0 | 2 |
| 8 | 18:27 | 0 | 2 |
| 9 | 17:53 | 0 | 2 |
| 10 | 17:43 | 0 | 2 |
| 11 | 17:43 | 0 | 2 |
| 12 | 17:43 | 0 | 2 |
| 13 | 17:43 | 0 | 2 |
| 14 | 16:38 | 0 | 2 |
| 15 | 16:12 | 0 | 2 |
| 16 | 16:11 | 0 | 1 |
| 17 | 12:55 | 0 | 3 |
| 18 | 12:49 | 0 | 3 |
| 19 | 12:44 | 0 | 3 |
| 20 | 11:46 | 0 | 3 |
| 21 | 11:46 | 0 | 3 |
| 22 | 11:43 | 0 | 3 |
| 23 | 11:39 | 0 | 3 |
| 24 | 11:38 | 0 | 3 |

| Data-ke | Jam | Penyebab | Kondisi |
|---|---|---|---|
| 25 | 11:27 | 0 | 3 |
| 26 | 11:25 | 0 | 3 |
| 27 | 11:24 | 0 | 3 |
| 28 | 11:24 | 0 | 3 |
| 29 | 11:23 | 0 | 3 |
| 30 | 11:23 | 0 | 3 |
| 31 | 11:23 | 0 | 3 |
| 32 | 11:22 | 0 | 3 |
| 33 | 11:21 | 0 | 3 |
| 34 | 11:18 | 0 | 3 |
| 35 | 11:18 | 0 | 3 |
| 36 | 11:18 | 0 | 3 |
| 37 | 11:17 | 0 | 3 |
| 38 | 11:17 | 0 | 3 |
| 39 | 11:17 | 0 | 3 |
| 40 | 11:17 | 0 | 3 |
| 41 | 11:17 | 0 | 3 |
| 42 | 11:17 | 0 | 3 |
| 43 | 11:16 | 0 | 3 |
| 44 | 11:16 | 0 | 3 |
| 45 | 11:16 | 0 | 3 |
| 46 | 11:16 | 0 | 3 |
| 47 | 11:16 | 0 | 3 |
| 48 | 11:15 | 0 | 3 |
| 49 | 11:15 | 0 | 3 |
| 50 | 10:07 | 0 | 1 |
| 51 | 9:32 | 0 | 1 |
| 52 | 9:19 | 0 | 2 |
| 53 | 9:15 | 0 | 2 |
| 54 | 9:11 | 0 | 2 |
| 55 | 8:58 | 0 | 1 |

Ketiga, setelah mengkonversi penyebab kemacetan dan kondisi menjadi nilai atau angka, tentukan setiap penyebab dan kondisi tersebut menjadi kelas-kelas yang telah ditentukan dengan cara menjumlahkan nilai atau angka dari penyebab dengan nilai atau angka dari kondisi. Jika hasilnya adalah kurang sama dengan 2 (dua), maka klasifikasikan menjadi kelas macet dan jika hasilnya lebih dari 2 (dua), klasifikasikan menjadi kelas Macet Sekali. Setelah dilakukan perhitungan dan pengklasifikasian, maka didapatkan data sebagai berikut:

**Tabel 3-17 *Sample* data yang telah dklasifikasikan berdasarkan kelas yang telah ditentukan**

| Data-ke | Jam | Penyebab | Kondisi | Kelas |
|---|---|---|---|---|
| 1 | 19:34 | 0 | 2 | Macet |
| 2 | 19:23 | 0 | 2 | Macet |
| 3 | 19:22 | 0 | 1 | Macet |
| 4 | 19:20 | 0 | 2 | Macet |
| 5 | 18:55 | 0 | 2 | Macet |
| 6 | 18:54 | 0 | 1 | Macet |
| 7 | 18:43 | 0 | 2 | Macet |
| 8 | 18:27 | 0 | 2 | Macet |
| 9 | 17:53 | 0 | 2 | Macet |
| 10 | 17:43 | 0 | 2 | Macet |
| 11 | 17:43 | 0 | 2 | Macet |
| 12 | 17:43 | 0 | 2 | Macet |
| 13 | 17:43 | 0 | 2 | Macet |
| 14 | 16:38 | 0 | 2 | Macet |
| 15 | 16:12 | 0 | 2 | Macet |
| 16 | 16:11 | 0 | 1 | Macet |
| 17 | 12:55 | 0 | 3 | Macet Sekali |
| 18 | 12:49 | 0 | 3 | Macet Sekali |
| 19 | 12:44 | 0 | 3 | Macet Sekali |
| 20 | 11:46 | 0 | 3 | Macet Sekali |
| 21 | 11:46 | 0 | 3 | Macet Sekali |
| 22 | 11:43 | 0 | 3 | Macet Sekali |

| Data-ke | Jam | Penyebab | Kondisi | Kelas |
|---|---|---|---|---|
| 23 | 11:39 | 0 | 3 | Macet Sekali |
| 24 | 11:38 | 0 | 3 | Macet Sekali |
| 25 | 11:27 | 0 | 3 | Macet Sekali |
| 26 | 11:25 | 0 | 3 | Macet Sekali |
| 27 | 11:24 | 0 | 3 | Macet Sekali |
| 28 | 11:24 | 0 | 3 | Macet Sekali |
| 29 | 11:23 | 0 | 3 | Macet Sekali |
| 30 | 11:23 | 0 | 3 | Macet Sekali |
| 31 | 11:23 | 0 | 3 | Macet Sekali |
| 32 | 11:22 | 0 | 3 | Macet Sekali |
| 33 | 11:21 | 0 | 3 | Macet Sekali |
| 34 | 11:18 | 0 | 3 | Macet Sekali |
| 35 | 11:18 | 0 | 3 | Macet Sekali |
| 36 | 11:18 | 0 | 3 | Macet Sekali |
| 37 | 11:17 | 0 | 3 | Macet Sekali |
| 38 | 11:17 | 0 | 3 | Macet Sekali |
| 39 | 11:17 | 0 | 3 | Macet Sekali |
| 40 | 11:17 | 0 | 3 | Macet Sekali |
| 41 | 11:17 | 0 | 3 | Macet Sekali |
| 42 | 11:17 | 0 | 3 | Macet Sekali |
| 43 | 11:16 | 0 | 3 | Macet Sekali |
| 44 | 11:16 | 0 | 3 | Macet Sekali |
| 45 | 11:16 | 0 | 3 | Macet Sekali |
| 46 | 11:16 | 0 | 3 | Macet Sekali |
| 47 | 11:16 | 0 | 3 | Macet Sekali |
| 48 | 11:15 | 0 | 3 | Macet Sekali |
| 49 | 11:15 | 0 | 3 | Macet Sekali |
| 50 | 10:07 | 0 | 1 | Macet |
| 51 | 9:32 | 0 | 1 | Macet |
| 52 | 9:19 | 0 | 2 | Macet |
| 53 | 9:15 | 0 | 2 | Macet |
| 54 | 9:11 | 0 | 2 | Macet |
| 55 | 8:58 | 0 | 1 | Macet |

Selanjutnya, setelah didapatkan data *sample*, maka dilanjutkan ke dalam inti proses ekstraksi informasi ini, yaitu proses penentuan kelas bagi data uji dengan menggunakan *k-nearest neighbor*. Langkah-langkah dalam menghitung metode *k-nearest neighbor* adalah sebagai berikut [7]:

1 Menentukan parameter K (jumlah tetangga paling dekat);.
2 Menghitung kuadrat jarak query euclid masing-masing objek terhadap data *sample* yang diberikan;
3 Urutkan objek-objek tersebut berdasarkan urutan jarak euclid terkecil;
4 Hitung jumlah mayoritas kelas berdasarkan K (jumlah terdekat ).

Berikut adalah *pseudocode* dari algoritma KNN dengan *Euclidean Distance* sebagai fungsi untuk menghitung nilai kuadrat jaraknya:

```
Procedure Knn(input: k:integer, querySet:larikPoint,
trainingSet:larikPoint)

{I.S: Menentukan kelas pada kueri terbaru berdasarkan
mayoritas kelas}
{F.S: Mendapatkan kelas pada kueri terbaru berdasarkan
mayoritas kelas}

Kamus:
   d: double
   map: SortedMap
   query: pointer

Algoritma:
   foreach (query in querySet) do
       foreach (training in trainingSet) do
           map ← SortedMap(k);
               if (query != 0) do
                   d ← EuclideanDistance(query, training);
                   map(d, training);
                   else query ← 0;
```

```
                    d ← EuclideanDistance(query, training);
                    map(d, training);
                endif
            endfor
            query↑label ← majorityVote(map);
        endfor
```

**Data Uji**: Terdapat tweet baru jam 21:00 yang menyatakan bahwa kondisi jalan macet parah (3) dan penyebabnya adalah kecelakaan (1).

**Tabel 3-18 Data uji yang dilakukan terhadap data sample**

| Data-ke | Jam | Penyebab | Kondisi | Kelas |
|---|---|---|---|---|
| Baru | 21:00 | 1 | 3 | ? |
| 1 | 19:34 | 0 | 2 | Macet |
| 2 | 19:23 | 0 | 2 | Macet |
| 3 | 19:22 | 0 | 1 | Macet |
| 4 | 19:20 | 0 | 2 | Macet |
| 5 | 18:55 | 0 | 2 | Macet |
| 6 | 18:54 | 0 | 1 | Macet |
| 7 | 18:43 | 0 | 2 | Macet |
| 8 | 18:27 | 0 | 2 | Macet |
| 9 | 17:53 | 0 | 2 | Macet |
| 10 | 17:43 | 0 | 2 | Macet |
| 11 | 17:43 | 0 | 2 | Macet |
| 12 | 17:43 | 0 | 2 | Macet |
| 13 | 17:43 | 0 | 2 | Macet |
| 14 | 16:38 | 0 | 2 | Macet |
| 15 | 16:12 | 0 | 2 | Macet |
| 16 | 16:11 | 0 | 1 | Macet |
| 17 | 12:55 | 0 | 3 | Macet Sekali |
| 18 | 12:49 | 0 | 3 | Macet Sekali |
| 19 | 12:44 | 0 | 3 | Macet Sekali |
| 20 | 11:46 | 0 | 3 | Macet Sekali |
| 21 | 11:46 | 0 | 3 | Macet Sekali |

| Data-ke | Jam | Penyebab | Kondisi | Kelas |
|---|---|---|---|---|
| 22 | 11:43 | 0 | 3 | Macet Sekali |
| 23 | 11:39 | 0 | 3 | Macet Sekali |
| 24 | 11:38 | 0 | 3 | Macet Sekali |
| 25 | 11:27 | 0 | 3 | Macet Sekali |
| 26 | 11:25 | 0 | 3 | Macet Sekali |
| 27 | 11:24 | 0 | 3 | Macet Sekali |
| 28 | 11:24 | 0 | 3 | Macet Sekali |
| 29 | 11:23 | 0 | 3 | Macet Sekali |
| 30 | 11:23 | 0 | 3 | Macet Sekali |
| 31 | 11:23 | 0 | 3 | Macet Sekali |
| 32 | 11:22 | 0 | 3 | Macet Sekali |
| 33 | 11:21 | 0 | 3 | Macet Sekali |
| 34 | 11:18 | 0 | 3 | Macet Sekali |
| 35 | 11:18 | 0 | 3 | Macet Sekali |
| 36 | 11:18 | 0 | 3 | Macet Sekali |
| 37 | 11:17 | 0 | 3 | Macet Sekali |
| 38 | 11:17 | 0 | 3 | Macet Sekali |
| 39 | 11:17 | 0 | 3 | Macet Sekali |
| 40 | 11:17 | 0 | 3 | Macet Sekali |
| 41 | 11:17 | 0 | 3 | Macet Sekali |
| 42 | 11:17 | 0 | 3 | Macet Sekali |
| 43 | 11:16 | 0 | 3 | Macet Sekali |
| 44 | 11:16 | 0 | 3 | Macet Sekali |
| 45 | 11:16 | 0 | 3 | Macet Sekali |
| 46 | 11:16 | 0 | 3 | Macet Sekali |
| 47 | 11:16 | 0 | 3 | Macet Sekali |
| 48 | 11:15 | 0 | 3 | Macet Sekali |
| 49 | 11:15 | 0 | 3 | Macet Sekali |
| 50 | 10:07 | 0 | 1 | Macet |
| 51 | 9:32 | 0 | 1 | Macet |
| 52 | 9:19 | 0 | 2 | Macet |
| 53 | 9:15 | 0 | 2 | Macet |
| 54 | 9:11 | 0 | 2 | Macet |

| Data-ke | Jam | Penyebab | Kondisi | Kelas |
|---|---|---|---|---|
| 55 | 8:58 | 0 | 1 | Macet |

Hal pertama yang harus dilakukan adalah, tentukan K tetangga terdekat. Nilai k yang terbaik untuk algoritma ini tergantung pada data. Secara umum, nilai K yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur [7]. Untuk itu, kita mengambil nilai K adalah ¼ dari data, dalam contoh kasus ini adalah 13,75 ≈ 14. Setelah itu, hitung kuadrat jarak query euclid pada masing-masing objek dengan menggunakan rumus Euclidean Distance, yaitu:

$$D(x_1, x_2) = \|x_2 - x_1\| = \sqrt{\sum_{j=1}^{p}(x_{2j} - x_{1j})^2}$$

Dimana $x_1$ adalah nilai dari penyebab dan kondisi dari data uji dan $x_2$ adalah nilai dari masing-masing penyebab dan kondisi dari data uji. Maka didapatkan hasil sebagai berikut:

**Tabel 3-19 Perhitungan dengan menggunakan Euclidean Distance**

| Data-ke | Jam | Penyebab | Kondisi | Kelas | Euclidean Distance |
|---|---|---|---|---|---|
| 1 | 19:34 | 0 | 2 | Macet | $(0-1)^2 + (2-3)^2 = 2$ |
| 2 | 19:23 | 0 | 2 | Macet | $(0-1)^2 + (2-3)^2 = 2$ |
| 3 | 19:22 | 0 | 1 | Macet | $(0-1)^2 + (1-3)^2 = 5$ |
| 4 | 19:20 | 0 | 2 | Macet | $(0-1)^2 + (2-3)^2 = 2$ |
| 5 | 18:55 | 0 | 2 | Macet | $(0-1)^2 + (2-3)^2 = 2$ |
| 6 | 18:54 | 0 | 1 | Macet | $(0-1)^2 + (1-3)^2 = 5$ |
| 7 | 18:43 | 0 | 2 | Macet | $(0-1)^2 + (2-3)^2 = 2$ |
| 8 | 18:27 | 0 | 2 | Macet | $(0-1)^2 + (2-3)^2 = 2$ |
| 9 | 17:53 | 0 | 2 | Macet | $(0-1)^2 + (2-3)^2 = 2$ |
| 10 | 17:43 | 0 | 2 | Macet | $(0-1)^2 + (2-3)^2 = 2$ |
| 11 | 17:43 | 0 | 2 | Macet | $(0-1)^2 + (2-3)^2 = 2$ |
| 12 | 17:43 | 0 | 2 | Macet | $(0-1)^2 + (2-3)^2 = 2$ |
| 13 | 17:43 | 0 | 2 | Macet | $(0-1)^2 + (2-3)^2 = 2$ |

| Data-ke | Jam | Penyebab | Kondisi | Kelas | Euclidean Distance |
|---|---|---|---|---|---|
| 14 | 16:38 | 0 | 2 | Macet | $(0-1)^2+(2-3)^2=2$ |
| 15 | 16:12 | 0 | 2 | Macet | $(0-1)^2+(2-3)^2=2$ |
| 16 | 16:11 | 0 | 1 | Macet | $(0-1)^2+(1-3)^2=5$ |
| 17 | 12:55 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 18 | 12:49 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 19 | 12:44 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 20 | 11:46 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 21 | 11:46 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 22 | 11:43 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 23 | 11:39 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 24 | 11:38 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 25 | 11:27 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 26 | 11:25 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 27 | 11:24 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 28 | 11:24 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 29 | 11:23 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 30 | 11:23 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 31 | 11:23 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 32 | 11:22 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 33 | 11:21 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 34 | 11:18 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 35 | 11:18 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 36 | 11:18 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 37 | 11:17 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 38 | 11:17 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 39 | 11:17 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 40 | 11:17 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 41 | 11:17 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 42 | 11:17 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 43 | 11:16 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 44 | 11:16 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |

| Data-ke | Jam | Penyebab | Kondisi | Kelas | Euclidean Distance |
|---|---|---|---|---|---|
| 45 | 11:16 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 46 | 11:16 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 47 | 11:16 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 48 | 11:15 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 49 | 11:15 | 0 | 3 | Macet Sekali | $(0-1)^2+(3-3)^2=1$ |
| 50 | 10:07 | 0 | 1 | Macet | $(0-1)^2+(2-3)^2=5$ |
| 51 | 9:32 | 0 | 1 | Macet | $(0-1)^2+(2-3)^2=5$ |
| 52 | 9:19 | 0 | 2 | Macet | $(0-1)^2+(2-3)^2=2$ |
| 53 | 9:15 | 0 | 2 | Macet | $(0-1)^2+(2-3)^2=2$ |
| 54 | 9:11 | 0 | 2 | Macet | $(0-1)^2+(2-3)^2=2$ |
| 55 | 8:58 | 0 | 1 | Macet | $(0-1)^2+(1-3)^2=5$ |

Urutkan berdasarkan objek-objek terkecil, yaitu:

**Tabel 3-20 Hasil pengurutan berdasarkan objek-objek terkecil**

| Data-ke | Jam | Penyebab | Kondisi | Kelas | Euclidean Distance |
|---|---|---|---|---|---|
| 17 | 12:55 | 0 | 3 | Macet Sekali | 1 |
| 18 | 12:49 | 0 | 3 | Macet Sekali | 1 |
| 19 | 12:44 | 0 | 3 | Macet Sekali | 1 |
| 20 | 11:46 | 0 | 3 | Macet Sekali | 1 |
| 21 | 11:46 | 0 | 3 | Macet Sekali | 1 |
| 22 | 11:43 | 0 | 3 | Macet Sekali | 1 |
| 23 | 11:39 | 0 | 3 | Macet Sekali | 1 |
| 24 | 11:38 | 0 | 3 | Macet Sekali | 1 |
| 25 | 11:27 | 0 | 3 | Macet Sekali | 1 |
| 26 | 11:25 | 0 | 3 | Macet Sekali | 1 |
| 27 | 11:24 | 0 | 3 | Macet Sekali | 1 |
| 28 | 11:24 | 0 | 3 | Macet Sekali | 1 |
| 29 | 11:23 | 0 | 3 | Macet Sekali | 1 |
| 30 | 11:23 | 0 | 3 | Macet Sekali | 1 |
| 31 | 11:23 | 0 | 3 | Macet Sekali | 1 |
| 32 | 11:22 | 0 | 3 | Macet Sekali | 1 |

| Data-ke | Jam | Penyebab | Kondisi | Kelas | Euclidean Distance |
|---|---|---|---|---|---|
| 33 | 11:21 | 0 | 3 | Macet Sekali | 1 |
| 34 | 11:18 | 0 | 3 | Macet Sekali | 1 |
| 35 | 11:18 | 0 | 3 | Macet Sekali | 1 |
| 36 | 11:18 | 0 | 3 | Macet Sekali | 1 |
| 37 | 11:17 | 0 | 3 | Macet Sekali | 1 |
| 38 | 11:17 | 0 | 3 | Macet Sekali | 1 |
| 39 | 11:17 | 0 | 3 | Macet Sekali | 1 |
| 40 | 11:17 | 0 | 3 | Macet Sekali | 1 |
| 41 | 11:17 | 0 | 3 | Macet Sekali | 1 |
| 42 | 11:17 | 0 | 3 | Macet Sekali | 1 |
| 43 | 11:16 | 0 | 3 | Macet Sekali | 1 |
| 44 | 11:16 | 0 | 3 | Macet Sekali | 1 |
| 45 | 11:16 | 0 | 3 | Macet Sekali | 1 |
| 46 | 11:16 | 0 | 3 | Macet Sekali | 1 |
| 47 | 11:16 | 0 | 3 | Macet Sekali | 1 |
| 48 | 11:15 | 0 | 3 | Macet Sekali | 1 |
| 49 | 11:15 | 0 | 3 | Macet Sekali | 1 |
| 1 | 19:34 | 0 | 2 | Macet | 2 |
| 2 | 19:23 | 0 | 2 | Macet | 2 |
| 4 | 19:20 | 0 | 2 | Macet | 2 |
| 5 | 18:55 | 0 | 2 | Macet | 2 |
| 7 | 18:43 | 0 | 2 | Macet | 2 |
| 8 | 18:27 | 0 | 2 | Macet | 2 |
| 9 | 17:53 | 0 | 2 | Macet | 2 |
| 10 | 17:43 | 0 | 2 | Macet | 2 |
| 11 | 17:43 | 0 | 2 | Macet | 2 |
| 12 | 17:43 | 0 | 2 | Macet | 2 |
| 13 | 17:43 | 0 | 2 | Macet | 2 |
| 14 | 16:38 | 0 | 2 | Macet | 2 |
| 15 | 16:12 | 0 | 2 | Macet | 2 |
| 52 | 9:19 | 0 | 2 | Macet | 2 |
| 53 | 9:15 | 0 | 2 | Macet | 2 |
| 54 | 9:11 | 0 | 2 | Macet | 2 |

| Data-ke | Jam | Penyebab | Kondisi | Kelas | Euclidean Distance |
|---|---|---|---|---|---|
| 3 | 19:22 | 0 | 1 | Macet | 5 |
| 6 | 18:54 | 0 | 1 | Macet | 5 |
| 16 | 16:11 | 0 | 1 | Macet | 5 |
| 50 | 10:07 | 0 | 1 | Macet | 5 |
| 51 | 9:32 | 0 | 1 | Macet | 5 |
| 55 | 8:58 | 0 | 1 | Macet | 5 |

Dengan K=14 yang telah ditentukan sebelumnya, maka kelas-kelas terdekatnya adalah:

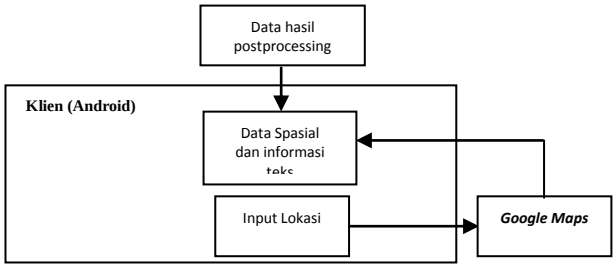
**Tabel 3-21 Kelas-kelas terdekat**

| Data-ke | Jam | Penyebab | Kondisi | Kelas | Euclidean Distance |
|---|---|---|---|---|---|
| 17 | 12:55 | 0 | 3 | Macet Sekali | 1 |
| 18 | 12:49 | 0 | 3 | Macet Sekali | 1 |
| 19 | 12:44 | 0 | 3 | Macet Sekali | 1 |
| 20 | 11:46 | 0 | 3 | Macet Sekali | 1 |
| 21 | 11:46 | 0 | 3 | Macet Sekali | 1 |
| 22 | 11:43 | 0 | 3 | Macet Sekali | 1 |
| 23 | 11:39 | 0 | 3 | Macet Sekali | 1 |
| 24 | 11:38 | 0 | 3 | Macet Sekali | 1 |
| 25 | 11:27 | 0 | 3 | Macet Sekali | 1 |
| 26 | 11:25 | 0 | 3 | Macet Sekali | 1 |
| 27 | 11:24 | 0 | 3 | Macet Sekali | 1 |
| 28 | 11:24 | 0 | 3 | Macet Sekali | 1 |
| 29 | 11:23 | 0 | 3 | Macet Sekali | 1 |

Karena mayoritas kelasnya adalah kelas Macet Sekali, sehingga data baru pada jam 21:00 yang menyatakan bahwa kondisi jalan macet parah (3) dan penyebabnya adalah kecelakaan (1), masuk ke dalam kelas **Macet Sekali**.

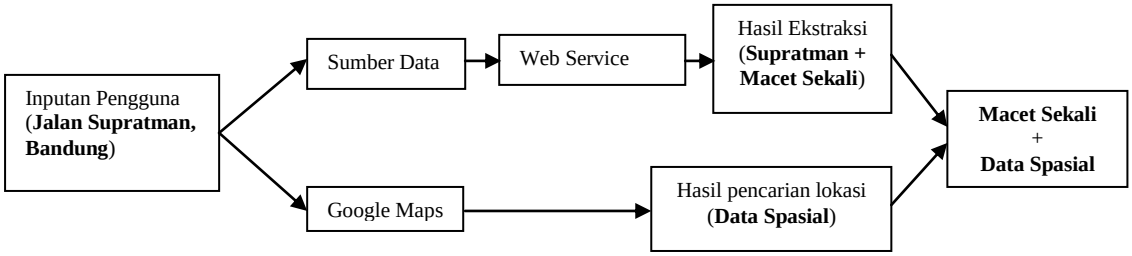
### 3.1.3.3 Menampilkan Hasil Ekstraksi

Pada tahap ini, hasil ekstraksi akan ditampilkan pada peta *smartphone* Android (klien). Hasil ekstraksi tersebut merupakan hasil dari pengolahan data baru yang telah dilakukan pada tahap *postprocessing* di *webservice*. Disisi lain, pada sisi klien, akan mencari lokasi berdasarkan inputan lokasi dari klien, setelah didapatkan hasil lokasinya, hasil tersebut akan digabungkan dengan hasil pengolahan data pada tahap *postprocessing*. Berikut adalah alur kerja pada tahapan menampilkan hasil ekstraksi:




**Gambar 3-18 Alur kerja proses menampilkan hasil ekstraksi dengan bentuk peta**

Inputan lokasi dengan jalan Supratman kota Bandung pada contoh kasus di atas, akan menjadi parameter acuan pada dua proses, yaitu proses pencarian lokasi pada Google Maps dan proses pengambilan data pada sumber data Twitter dan Lewat Mana.




**Gambar 3-19 Proses hasil ekstraksi yang digabungkan dengan hasil data spasial**

## 3.1.4 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non fungsional berfungsi untuk menghasilkan spesifikasi yang terperinici mengenai hal-hal yang di lakukan dalam proses berlangsungnya sistem. Analisis kebutuhan non fungsional pada sistem yang akan

dibangun ini terdiri dari analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras dan analisis kebutuhan perangkat pikir.

### 3.1.4.1 Analisis Kebutuhan Perangkat Lunak

Spesifikasi dari perangkat lunak yang akan digunakan dalam pembangunan sistem adalah sebagai berikut:

a. Sistem operasi Windows 7 Ultimate
b. Borland Delphi 7 32bit
c. Eclipse Indigo
d. Wamp Server 2.1e 32bit

Sedangkan spesifikasi perangkat lunak minimum untuk menjalankan aplikasi yang akan dibangun adalah sistem operasi Android 4.0

### 3.1.4.2 Analisis Kebutuhan Perangkat keras

Spesifikasi perangkat keras yang digunakan agar dapat mendukung dalam proses menjalankan perangkat lunak adalah sebagai berikut:

a. *Processor* Core i3 2.30 GHz
b. RAM 2 Gb
c. HDD 500 Gb
d. Monitor LED/LCD
e. Mouse
f. Keyboard

Sedangkan spesifikasi minimum untuk menjalankan perangkat lunak yang akan dibangun adalah sebagai berikut:

a. *Processor* 600 Mhz
b. RAM 128 Mb

### 3.1.4.3 Analisis Kebutuhan Perangkat Pikir

Analisis kebutuhan perangkat pikir digunakan untuk menganalisis karakteristik pengguna perangkat lunak yang akan dibangun. Adapun karakteristiknya dijelaskan dalam table berikut:

**Tabel 3-22 Tabel Analisis Kebutuhan Perangkat Pikir Pengguna Perangkat Lunak**

| Nama | Spesifikasi |
|---|---|
| Pengguna | Setiap pengguna jalan yang ingin mengetahui informasi kemacetan |
| Tingkat Pendidikan | Minimal SMA |
| Tingkat Keterampilan | Mengetahui cara menggunakan *smartphone* Android |
| Pengalaman | Pernah menggunakan *smartphone* Android |

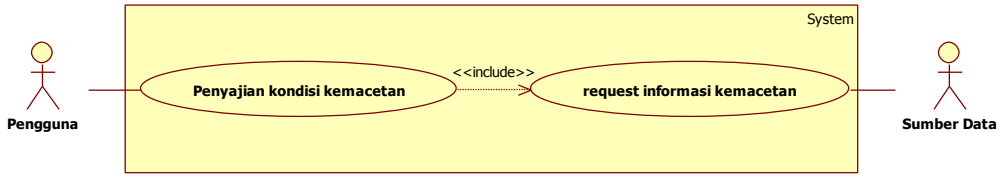
## 3.1.5 Analisis Kebutuhan Fungsional

### 3.1.5.1 Deskripsi global perangkat lunak

Perangkat lunak yang akan dibangun merupakan perangkat lunak yang dapat menampilkan hasil ekstraksi informasi mengenai kemacetan berdasarkan inputan nama jalan dari pengguna. Ekstraksi tersebut meliputi dua kelas, yaitu kelas macet dan sangat macet.

### 3.1.5.2 *Use Case* Diagram

*Use Case* Diagram menjelaskan tentang fungsionalitas pada sistem yang akan dibangun. *Use Case* Diagram pada sistem yang akan dibangun terdiri dari dua *use case* dan dua aktor. Berikut adalah *Use Case* Diagram dari sistem yang akan dibangun:




**Gambar 3-20 *Use Case* Diagram**

#### 3.1.5.2.1 Definisi Aktor

Adapun tabel dari definisi aktor yang ada pada digram *use case*, yaitu:

**Tabel 3-23 Tabel Definisi Aktor**

| No | Actor | Deskripsi |
|---|---|---|
| 1 | Pengguna | Setiap pengguna yang ingin mengetahui informasi kemacetan |
| 2 | Sumber Data | Media *digital* yang menginformasikan kondisi kemacetan |

#### 3.1.5.2.2 Definisi Use Case

Berikut adalah definisi per tiap *use case* yang ada pada diagram *use case*:

**Tabel 3-24 Tabel Definisi *Use Case***

| No | Use Case | Deskripsi |
|---|---|---|
| 1 | Penyajian kondisi kemacetan | Proses hasil dari ekstraksi informasi sesuai dengan *request* nama jalan oleh pengguna |
| 2 | *Request* data | Proses melakukan *request* data berdasarkan dengan inputan nama jalan dari pengguna |

#### 3.1.5.2.3 Skenario Use Case

Nama *Use Case*: Penyajian Kondisi Kemacetan

Skenario:

| Aksi Actor | Reaksi Sistem |
|---|---|
| Skenario Normal | |
| 1. Input lokasi | |
| | 2. Mengecek ketersediaan internet<br>3. Mengambil data dari *web service*<br>4. Menampilkan lokasi kemacetan |
| 5. Melihat data kemacetan yang telah diolah dalam bentuk map | |
| Skenario Alternatif Pertama | |
| | 1. Mengecek ketersediaan internet<br>2. Menampilkan pesan tidak ada koneksi |

| Aksi Actor | Reaksi Sistem |
|---|---|
| Skenario Alternatif Kedua | |
| 1. Input lokasi | |
| | 2. Mengecek ketersediaan internet<br>3. Mengambil data dari *web service*<br>4. Menampilkan pesan lokasi tidak ditemukan |

Nama *Use Case*: *Request* Informasi Kemacetan

Skenario:

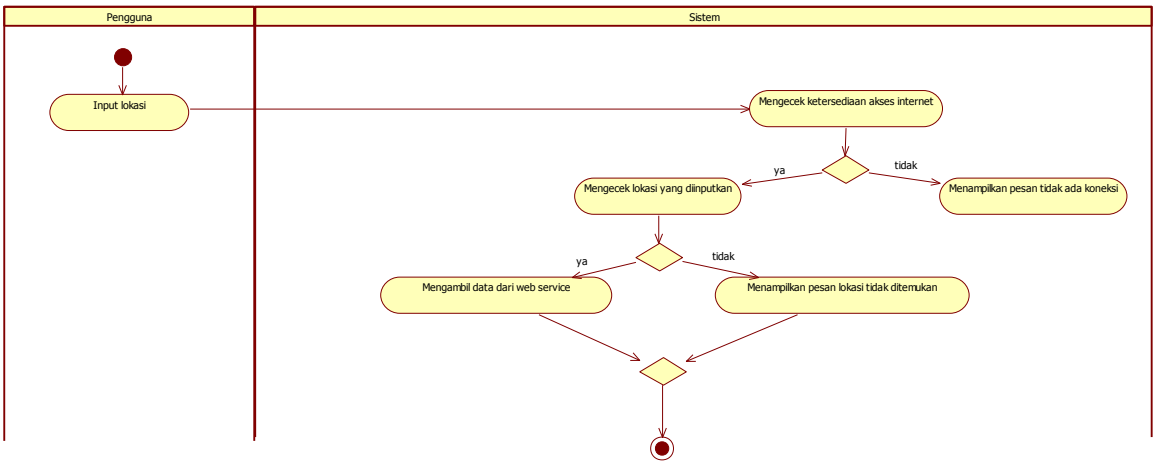
| Aksi Actor | Reaksi Sistem |
|---|---|
| Skenario Normal | |
| | 1. Mengecek lokasi yang diinputkan |
| 2. Menampilkan informasi kemacetan | |
| | 3. Mengambil data kemacetan dari sumber data |
| Skenario Alternatif | |
| | 1. Mengecek lokasi yang diinputkan |
| 2. Tidak Menampilkan informasi kemacetan | |

### 3.1.5.3 Activity Diagram

*Activity* diagram pada penelitian ini menjelaskan tentang alur kerja tahapan-tahapan aktivitas dari *use case* yang akan di bangun. *Activity* utama pada penelitian ini meliputi dua *activity*, yaitu Penyajian Kondisi Kemacetan dan Request Informasi Kemacetan. Sub *activity* dari *activity* Penyajian Kondisi Kemacetan, diambil dari aturan-aturan yang ada pada tahapan *preprocessing*.

#### 3.1.5.3.1 Activity Diagram Penyajian Kondisi Kemacetan

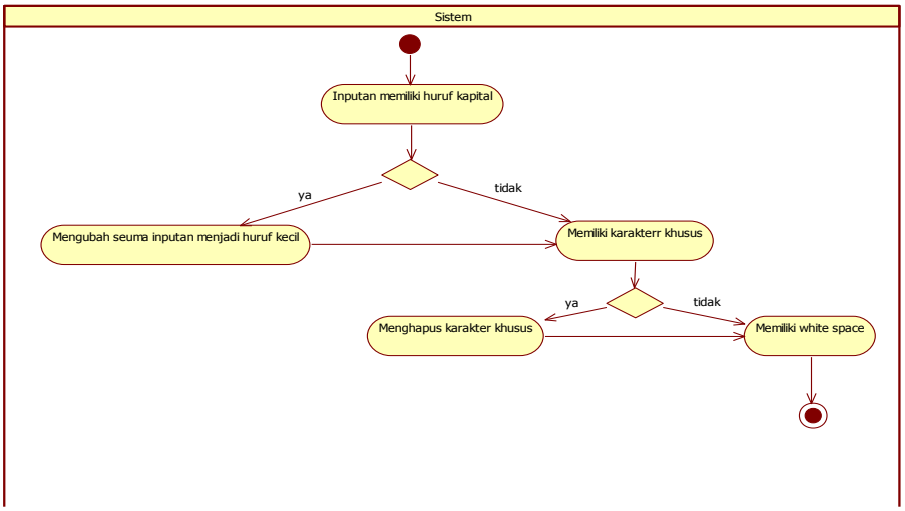
Activity ini menjelaskan tentang proses penyajian kondisi kemacetan pada sistem yang akan dibangun.




**Gambar 3-21 Activity Diagram Penyajian Kondisi Kemacetan**

### 3.1.6.3.3 Sub Activity *Case Folding*

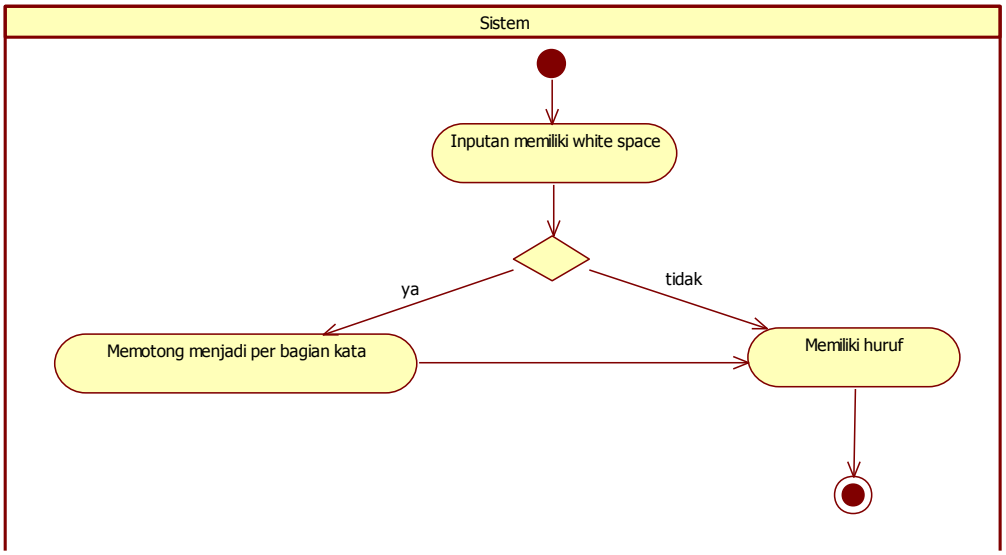
Sub Activity ini menjelasakan tentang aturan-aturan yang ada pada tahapan *preprocessing case folding.*




**Gambar 3-22 Sub Activity Diagram *Case Folding***

### 3.1.6.3.4 Sub Activity *Tokenizing*

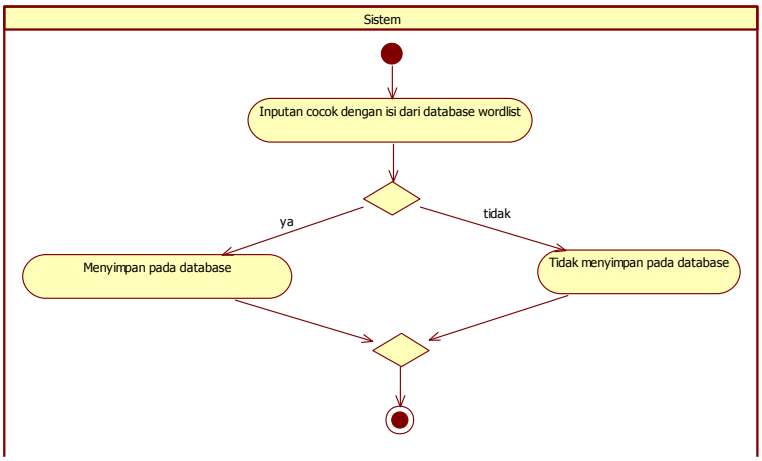
Sub Activity ini menjelasakan tentang aturan-aturan yang ada pada tahapan *preprocessing tokenizing.*




**Gambar 3-23 Sub Activity Diagram *Tokenizing***

### 3.1.6.3.5 Sub Activity *Filtering*

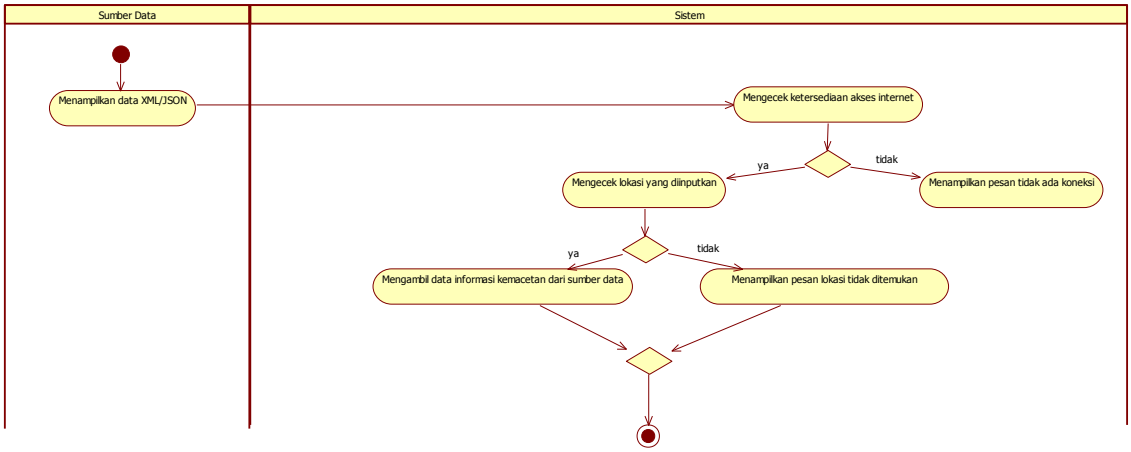
Sub Activity ini menjelasakan tentang aturan-aturan yang ada pada tahapan *preprocessing case folding.*




**Gambar 3-24 Sub Activity Diagram *Filtering***

### 3.1.5.3.2 Activity Request Informasi Kemacetan

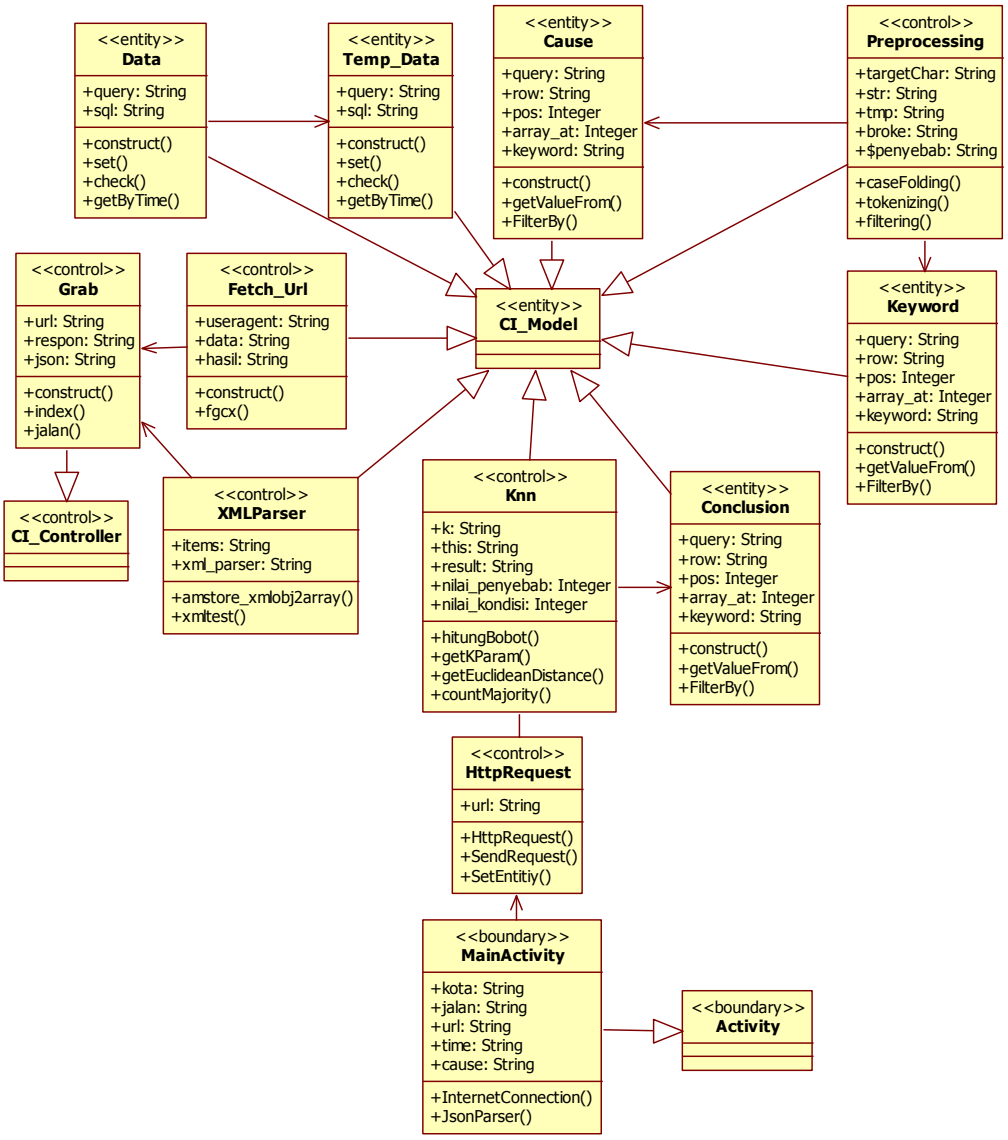
Activity ini menjelaskan tentang proses *request* inormasi kemacetan pada sumber data informasi kemacetan.




**Gambar 3-25 Activity Diagram *Request* Informasi Kemacetan**

### 3.1.5.4 Class Diagram

Berikut adalah Class Diagram pada sistem yang akan dibangun:

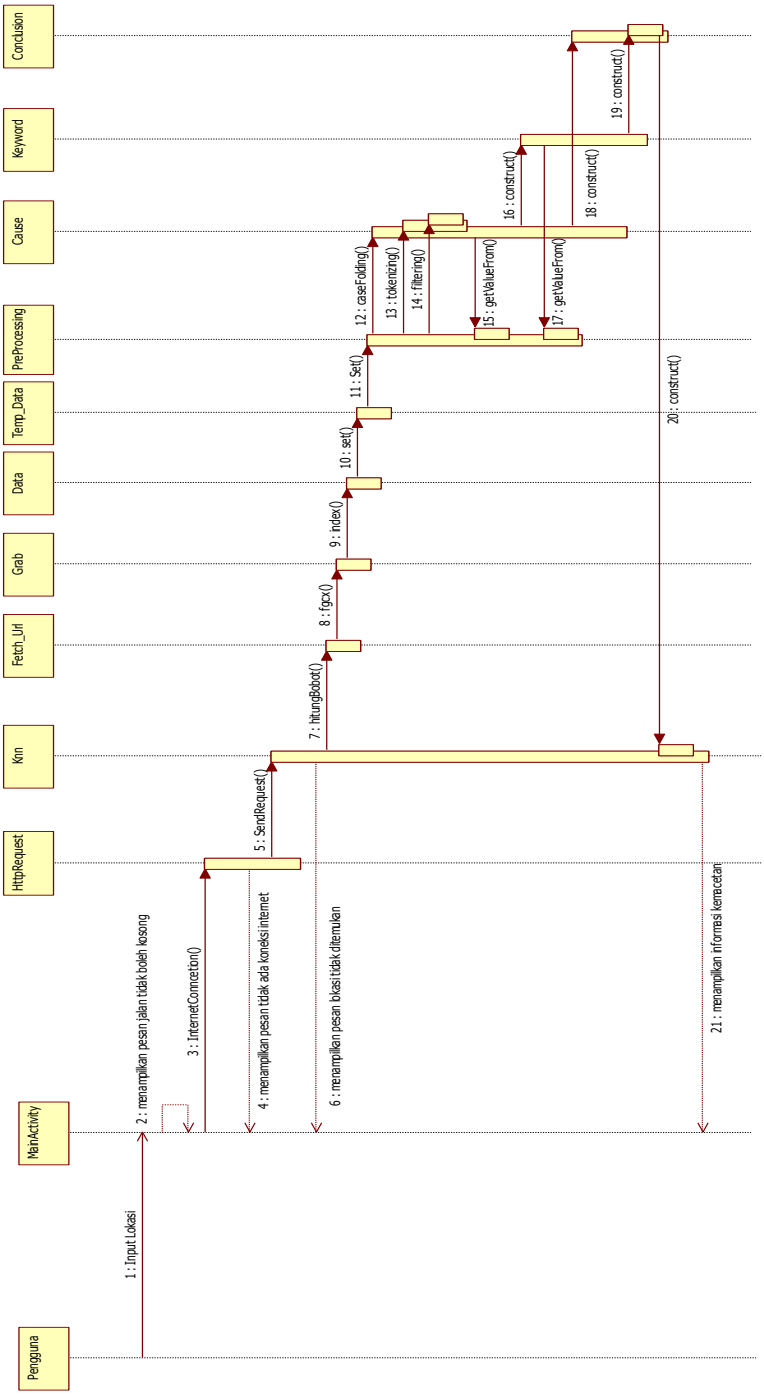



**Gambar 3-26 Class Diagram dari sistem yang akan dibangun**

### 3.1.5.5 Sequance Diagram

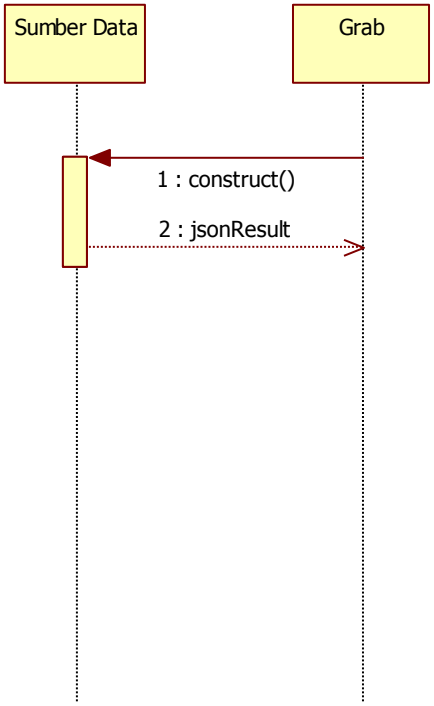
Berikut ini adalah bentiuk sequance diagram pada:

1. *Use Case*: Penyajian Kondisi Kemacetan




**Gambar 3-27 Sequance Diagram Penyajian Lokasi Kemacetan**

2. *Use Case*: *Request* Informasi Kemacetan



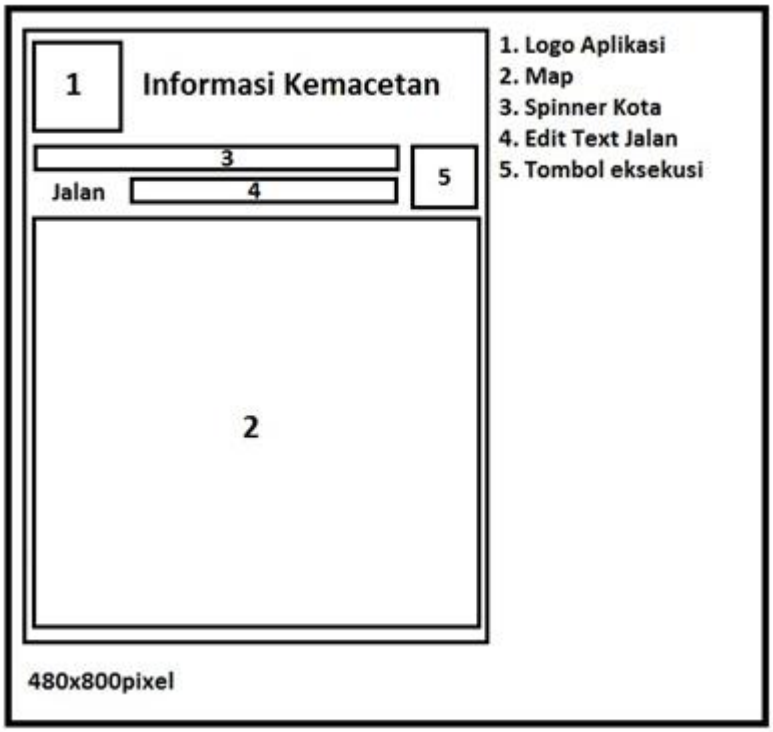

**Gambar 3-27 Sequance Diagram *Request* Informasi Kemacetan**

## 3.2 Perancangan Sistem

Tahap perancangan merupakan tahapan selanjutnya setelah tahapan analisis. Perancangan dalam membangun sistem ini meliputi perancangan antarmuka, perancangan pesan dan perancangan prosedural.

### 3.2.1 Perancangan Antarmuka

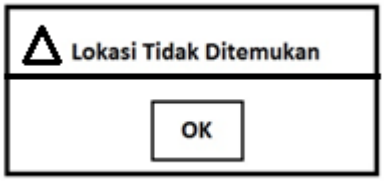
Perancangan antarmuka merupakan salah satu sub tahapan penting dalam pembangunan suatu sistem perangkat lunak. Arti perancangan antarmuka sendiri adalah proses pembuatan perancangan form-form (bentuk-bentuk) yang akan disajikan pada layar. Hasil akhir dari proses perancangan antarmuka ini disebut tampilan layar. Berikut merupakan gambar hasil akhir proses perancangan antarmuka, yaitu tampilan layar pada sistem perangkat lunak yang penulis bangun:



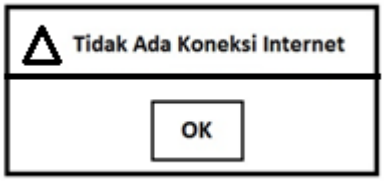

**Gambar 3-28 Perancangn Menu**

### 3.2.2 Perancangan Pesan

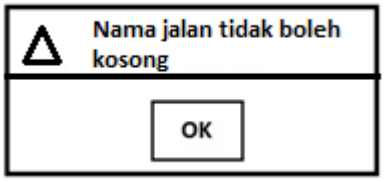
Perancangan pesan merupakan proses pembuatan perancangan teks atau rangkaian kata-kata yang akan digunakan dalam penyampaian pesan-pesan baik berupa pesan perintah, pesan pertanyaan, dan atau pesan informasi pada tampilan layar dalam perangkat lunak yang akan dibangun. Berikut ini tampilan beberapa pesan informasi pada perangkat lunak yang penulis bangun:




**Gambar 3-29 Perancangan Pesan Lokasi Tidak Ditemukan**




**Gambar 3-30 Perancangan Pesan Tidak Ada Koneksi Internet**

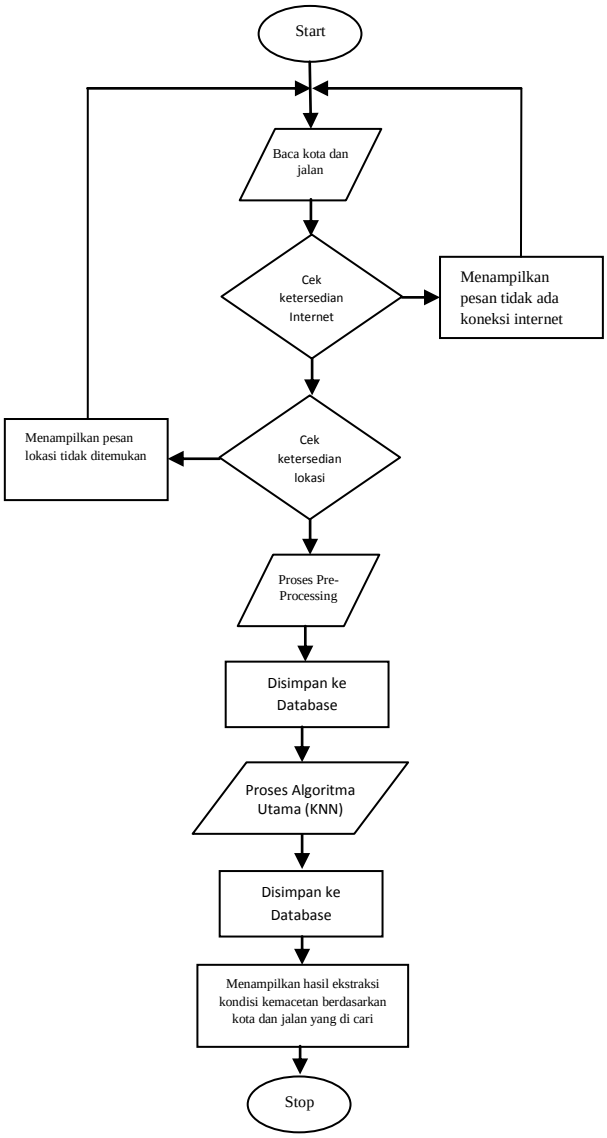



**Gambar 3-31 Perancangan Pesan Nama Jalan Tidak Boleh Kosong**

### 3.2.3 Perancangan Prosedural

Perancangan prosedural pada aplikasi yang dibangun, meliputi penyajian informasi kemacetan berdasarkan lokasi (kota dan jalan) yang telah diinputkan oleh *user* tersedia atau tidak. Perancangan prosedural tersebut adalah sebagai berikut:

1. Start
2. Read Kota dan Jalan
3. Cek koneksi internet tersedia atau tidak
4. Jika tidak tersedia menampilkan pesan tidak ada koneksi internet dan kembali ke point 2
5. Jika tersedia, cek Kota dan Jalan
6. Jika ada, menampilkan kota dan jalan yang dicari
7. Jika tidak ada, menampilkan pesan lokasi yang dicari tidak ditemukan dan kembali ke poin 2
8. Proses tahap *preprocessing*
9. Simpan di *database*
10. Proses tahap algoritma KNN
11. Simpan di *database*
12. Menampilkan hasil ekstraksi informasi di peta *smartphone* Android
13. Stop




**Gambar 3-32 Flowchart Prosedural**