Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB II

LANDASAN TEORI

2.1 Pengertian Model

Model adalah representasi dari suatu objek, benda, atau ide - ide dalam bentuk yang disederhanakan dari kondisi atau fenomena alam. Model berisi informasi-informasi tentang suatu fenomena yang dibuat dengan tujuan untuk mempelajari fenomena sistem yang sebenarnya. Model dapat merupakan tiruan dari suatu benda, sistem atau kejadian yang sesungguhnya yang hanya berisi informasi-informasi yang dianggap penting untuk ditelaah. (Mahmud Achmad, 2008).

2.2 Pengertian Prediksi

Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahannya (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat diperkecil. Prediksi tidak harus memberikan jawaban secara pasti kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi (Sri Mulyono, 2000).

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017

2.3 Pengertian Kemacetan Lalu Lintas

Kemacetan adalah kondisi dimana arus lalu lintas yang lewat pada ruas jalan yang ditinjau melebihi kapasitas rencana jalan tersebut yang mengakibatkan kecepatan bebas ruas jalan tersebut mendekati atau melebihi 0 km/jam sehingga menyebabkan terjadinya antrian. Pada saat terjadinya kemacetan, nilai derajat kejenuhan pada ruas jalan akan ditinjau dimana kemacetan akan terjadi bila nilai derajat kejenuhan mencapai lebih dari 0,5 (MKJI, 1997).

Kemacetan lalu lintas pada ruas jalan raya terjadi saat arus kendaraan lalu lintas meningkat seiring bertambahnya permintaan perjalanan pada suatu periode tertentu serta jumlah pemakai jalan melebihi dari kapasitas yang ada (Meyer, 1984).

2.4 Pengertian Warna RGB

Model warna RGB (*red, green, blue*) adalah model formal untuk menampilkan warna-warna pada monitor komputer, kartu grafis komputer, televisi dan sistem *display* lainnya (Ahmad, 2005). Sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (*Red, Green, Blue* - RGB) (Gonzales, 2004). Sorotan *electron* menghasilkan sinyal merah, hijau, biru yang dikombinasikan untuk menghasilkan berbagai warna yang dilihat pada layar (Munir, 2004).

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017

2.5 Pengertian Google Maps

Google Maps adalah sebuah layanan peta online yang dapat digunakan secara mudah dan gratis oleh pengguna. GoogleMaps juga menawarkan *API* (*Application Programming Interface*) yang dapat digunakan oleh *developer* untuk memanfaatkan layanan pada *GoogleMaps* dan menerapkannya pada aplikasi yang dibuatnya. *GoogleMaps API* merupakan suatu *library* dalam bentuk JavaScript

Pada Google Maps *API* terdapat 4 jenis pilihan model peta disediakan oleh *Google*, diantaranya:

1. *ROADMAP*, untuk menampilkan peta biasa 2 dimensi.
2. *SATELLITE*, untuk menampilkan foto satelit.
3. *TERRAIN*, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai.

HYBRID, akan menunjukkan foto satelit yang diatasnya tergambar pula apa yang tampil pada ROADMAP (jalan dan nama kota).

2.5.1 Pengertian Google Traffic

Google Traffic adalah fitur di dalam Google Maps yang menampilkan kondisi kemacetan lalu lintas yang terjadi pada jalan raya dan jalan raya utama secara *real time.* Google Traffic dapat dilihat di situs *web* Google Maps, atau dengan menggunakan aplikasi Google Maps pada perangkat *mobile*.

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017

Google Traffic bekerja dengan menganalisisa lokasi melalui data GPS yang dikirim ke *Google* oleh sejumlah besar pengguna Google Maps. Dengan menghitung kecepatan pengguna sepanjang jalan, *Google* mampu menghasilkan peta lalu lintas langsung. *Google* memproses data mentah yang masuk mengenai lokasi perangkat ponsel pengguna, dan kemudian mengolah dan memberikan informasi titik-titik terjadi kemacetan lalu lintas pada ruas jalan. Informasi kondisi kemacetan lalu lintas yang tersedia pada Google Traffic dapat digunakan oleh siapapun secara gratis.

Google Traffic memberikan warna-warna berbeda untuk setiap level kemacetan, dengan warna hijau mewakili kecepatan lalu lintas normal, warna kuning mewakili kondisi lalu lintas yang lebih lambat, merah menunjukkan kemacetan, dan merah tua (merah dan hitam) mengindikasikan hampir berhenti atau berhenti.

2.6 Pengertian WEKA

WEKA adalah aplikasi *Data Mining* open source yang berbasis java. Aplikasi ini dikembangkan pada tahun 1994 dan pertama kali oleh sebuah universitas di selandia baru yang bernama Universitas Waikato. Aplikasi WEKA ini mulai menjadi aplikasi *Data Mining* open source yang sangat terkenal pada awal perkembangannya. Hal itu dikarenakan aplikasi WEKA memiliki kelebihan yang tidak dimiliki aplikasi *Data Mining* lainnya yaitu pada aplikasi WEKA terdapat banyak algoritma yang terdapat didalam aplikasi dan disertai juga machine learning, lalu juga dalam penggunaanya tidak terlalu rumit sehingga tidak menyulitkan

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017

penggunanya, dan ditambah dengan kelebihan lainnya bahwa algoritma-algoritma yang terdapat pada aplikasi WEKA selalu baru dan ter update, sehingga dengan beberapa kelebihan. Aplikasi WEKA tersebut digunakan banyak perusahaan dalam dunia bisnis untuk membantu dalam usaha bisnisnya, akademik pun juga tak mau ketinggalan untuk menggunakan aplikasi WEKA.

Aplikasi WEKA merupakan *software* yang terdiri dari koleksi algoritma machine learning yang dapat digunakan untuk melakukan generalisasi atau formulasi dari sekumpulan data *sampling*.

2.7 Pengertian Android

Android merupakan sebuah sistem operasi yang digunakan untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi utama *mobile*. Pada awalnya sistem operasi ini dikembangkan oleh Android.Inc, yang kemudian dibeli oleh Google pada tahun 2005. Android mengembangkan usaha pada tahun 2007 dibentuklah *Open Handset Alliance* (OHA), sebuah konsorsium dari beberapa perusahaan, yaitu Texas Instrument, Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcom, Samsung Electronics, Sprint Nextel, T-*Mobile*, Packet Video, ARM Holdings, Atheros Communications, Asustek Computer INC, Garmin Ltd, Softbank, Sony EriCSSon, Toshiba Corp, dan VodaFone Group Plc dengan tujuan untuk mengembangkan standar terbuka untuk perangkat *mobile Smartphone* (Hermawan, 2010).

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017

Android memiliki empat karakteristik sebagai berikut:

1. Terbuka

   Android dibangun untuk benar-benar terbuka sehingga sebuah aplikasi dapat memanggil salah satu fungsi inti ponsel seperti membuat panggilan, mengirim pesan teks, menggunakan kamera, dan lain-lain. Android menggunakan sebuah mesin virtual yang dirancang khusus untuk mengoptimalkan sumber daya memori dan perangkat keras yang terdapat di dalam perangkat. Android merupakan *open source*, dapat secara bebas diperluas untuk memasukkan teknologi baru yang lebih maju pada saat teknologi tersebut muncul. Platform ini akan terus berkembang untuk membangun aplikasi *mobile* yang inovatif.

2. Semua aplikasi dibuat sama

   Android tidak memberikan perbedaan terhadap aplikasi utama dari telepon dan aplikasi pihak ketiga (*third-party application*). Semua aplikasi dapat dibangun untuk memiliki akses yang sama terhadap kemampuan sebuah telepon dalam menyediakan layanan dan aplikasi yang luas terhadap para pengguna.

3. Memecahkan hambatan pada aplikasi

   Android memecah hambatan untuk membangun aplikasi yang baru dan inovatif. Misalnya, pengembang dapat menggabungkan informasi yang diperoleh dari web dengan

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017

data pada ponsel seseorang seperti kontak pengguna, kalender, atau lokasi geografis.

4. Pengembangan aplikasi yang cepat dan mudah

   Android menyediakan akses yang sangat luas kepada pengguna untuk menggunakan *library* yang diperlukan dan *tools* yang dapat digunakan untuk membangun aplikasi yang semakin baik. Android memiliki sekumpulan *tools* yang dapat digunakan sehingga membantu para pengembang dalam meningkatkan produktivitas pada saat membangun aplikasi yang dibuat. (Sumber : http://www.Android.com/about/)

Google Inc. sepenuhnya membangun Android dan menjadikannya bersifat terbuka (*open source*) sehingga para pengembang dapat menggunakan Android tanpa mengeluarkan biaya untuk lisensi dari Google dan dapat membangun Android tanpa adanya batasan-batasan. Android *Software Development Kit* (SDK) menyediakan alat dan *Application Programming Interface* (API) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman *Java*.

2.8 Pengertian Wkhtmltoimage

Wkhtmltoimage merupakan *tool command line* (CLI) berbasis *open source* (LGPLv3) untuk membuat HTML menjadi PDF dan berbagai format gambar dengan memanfaatkan *Qt WebKit rendering engine*. (Sumber: https://wkhtmltopdf.org/)

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017

2.9 Pengertian Aplikasi *Hybrid*

Aplikasi *hybrid* adalah aplikasi *web* yang ditransformasikan menjadi kode native pada *platform* seperti IOS atau Android. Aplikasi *hybrid* biasanya menggunakan *browser* untuk mengijinkan aplikasi *web* mengakses berbagai fitur di *device mobile* seperti *Push Notification*, *Contacts*, atau *Offline Data Storage*. Beberapa *tools* untuk mengembangkan aplikasi *hybrid* antara lain Phonegap, Rubymotion, Intel XDK dan lain-lain.

Keuntungan membangun aplikasi *hybrid* diantaranya pemeliharaan *project* menjadi semakin mudah jika dibandingkan dengan aplikasi native. Aplikasi *hybrid* juga, bisa dibangun secara cepat untuk keperluan *cross platform* dan dana yang bisa menjadi lebih hemat jika dibandingkan dengan *native*.

2.10 Pengertian Intel XDK

Intel XDK adalah *development kit* yang dibuat oleh Intel untuk membuat aplikasi native untuk perangkat *mobile* menggunakan teknologi web seperti HTML5, CSS *dan* JavaScript. Aplikasi web dikompilasi menggunakan platform Cordova di *server* online untuk membuat aplikasi hybrid yang cross-platform. Aplikasi Intel XDK sendiri adalah aplikasi *cross-platform* yang tersedia untuk sistem operasi Windows, Linux dan OSX. Dengan teknologi seperti ini, developer dapat membuat satu aplikasi yang dapat berjalan di banyak platform *mobile* hanya dengan menggunakan satu basis kode saja. Dengan Intel XDK, *developer* dapat

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017

mengompilasi aplikasinya secara online sehingga tidak perlu menginstal berbagai *tools* seperti Cordova, SDK dan lain sebagainya di komputer sendiri. Dan yang lebih menarik lagi, Intel XDK tidak hanya menyediakan online compiling, tapi juga segala kebutuhan fase *development* seperti koding, testing, debugging, hingga publishing ke *online mobile* store seperti App Store dan Play Store. Intel XDK membenamkan *code editor Brackets* beserta keunggulan fiturnya seperti *code hinting* dan *code completion* sehingga sangat memudahkan proses penulisan kode. *Developer* juga dapat mengetes aplikasi yang sedang dibangun menggunakan emulator sehingga tidak perlu proses yang lama untuk dapat melihat progres-progres kecil.

Developer juga dapat melakukan device testing dengan bantuan aplikasi *mobile* Intel *App Preview* agar developer tidak perlu menginstal aplikasi di device untuk melakukan testing. Intel XDK juga menyediakan berbagai template aplikasi dan game dengan berbagai *framework* aplikasi seperti Ionic, AppFramework, dan *framework game* seperti Cocos2d-JS, EaselJS, Phaser, Pixi, BabylonJS dan Three.js. Dan bagi *developer* pemula yang ingin belajar membuat aplikasi atau game *mobile*, Intel XDK sudah menyediakan banyak *sample* dan demo aplikasi yang dapat langsung diamati kode program dan juga hasilnya. Hingga detik ini, Intel XDK sudah mendukung beberapa *platform mobile* populer diantaranya Android, iOS, Windows 8, Windows Phone 8.1, Windows 10, serta ekspor ke paket Chrome App.

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017

2.11 Pengertian Basis Data MySQL

Menurut Bambang Hariyanto (2004), basis data adalah kumpulan data (elementer) yang secara logik berkaitan dalam merepresentasikan fenomena/fakta secara terstruktur dalam domain tertentu untuk mendukung aplikasi pada sistem tertentu. Basisdata adalah kumpulan data yang saling berhubungan yang merefleksikan fakta-fakta yang terdapat di organisasi. Basis data mendeskripsikan *state* organisasi/perusahaan/sistem. Saat satu kejadian muncul di dunia nyata mengubah *state* organisasi/perusahaan/sistem maka satu perubahan pun harus dilakukan terhdadap data yang disimpan di basis data. Basis data merupakan komponen utama sistem informasi karena semua informasi untuk pengambilan keputusan berasal dari data di basisdata. Pengelolaaan basis data yang buruk dapat mengakibatkan ketidaktersediaan data penting yang digunakan untuk menghasilkan informasi yang diperlukan dalam pengambilan keputusan.

Menurut Connolly dan Begg (2010), *Database Management System* (DBMS) adalah sebuah sistem *software* yang memungkinkan pengguna untuk mendefinisikan, membuat, memlihara, dan mengontrol akses ke basis data.

Menurut Rudianto (2011), MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan *database* sebagai sumber dan pengolahan datanya. MySQL termasuk jenis RDBMS (*Relational Database Management System*). Itulah sebabnya, istilah seperti tabel,

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017

baris, dan kolom digunakan pada MySQL. Pada MySQL, sebuah *database* mengandung satu atau beberapa kolom. (Bambang Hariyanto, 2008)

MySQL memiliki beberapa kelebihan, yaitu:

1. MySQL memiliki performa tinggi namun merupakan *database* yang tergolong simpel sehingga mudah di-setup dan dikonfigurasi.
2. MySQL cenderung gratis untuk penggunaan tertentu.
3. MySQL mengerti bahasa SQL (*Structured Query Language*) yang merupakan pilihan sistem *database* moderen.
4. Banyak klien dapat mengakses *server* dalam satu waktu. Mereka dapat menggunakan banyak *database* secara simultan.
5. *Database* MySQL dapat diakses dari semua tempat di internet dengan hak akses tertentu.
6. MySQL dapat berjalan dalam banyak varian Unix dengan baik, sebaik seperti saat berjalan di sistem non-Unix.
7. MySQL mudah didapatkan dan memiliki *source code* yang boleh disebarluaskan sehingga bisa dikembangkan lebih lanjut.
8. Dapat dikoneksikan pada pemgoraman C, C++, Java, Perl, PHP dan Python.

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017

2.12 Pengertian *Data Mining*

Menurut Han, Jiawei (2011), *Data Mining* adalah proses menemukan pola yang menarik, dan pengetahuan dari data yang berjumlah besar. Sedangkan menurut Dunston dan Yager (2008), *Data Mining* adalah proses pencarian melalui data dengan jumlah yang besar, dalam sebuah usaha untuk menemukan pola, tren, dan hubungan. Menurut Keating dan Berry (2008), *Data Mining* adalah sebuah cara untuk mendapatkan kecerdasan pasar dari data dengan jumlah yang besar. Sedangkan menurut Liu, Sandra S. dan Chen, Jie (2009), *Data Mining* adalah proses pencarian pola tersembunyi dari berbagai *database*.

Dengan demikian berdasarkan beberapa pengertian *Data Mining* yang telah diuraikan, dapat disimpulkan bahwa *Data Mining* merupakan suatu proses analisis untuk menggali informasi yang tersembunyi dengan menggunakan statistik dan *artificial intelligence* di dalam suatu *database* dengan ukuran sangat besar, sehingga ditemukan suatu pola dari data yang sebelumnya tidak diketahui.

2.13 Pengertian Algoritma C4.5

Algoritma merupakan kumpulan perintah yang tertulis secara sistematis guna menyelesaikan permasalahan logika dari matematika. Pengertian Algoritma C4.5 merupakan algoritma yang digunakan untuk membentuk pohon keputusan. Sedang pohon keputusan dapat diartikan suatu cara untuk memprediksi atau mengklarifikasi yang sangat kuat. Pohon keputusan dapat membagi kumpulan data yang besar menjadi

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017

himpunan-himpunan record yang lebih kecil dengan menerapkan serangkaian aturan keputusan.

Secara umum algorima C4.5 untuk membangun pohon keputusan adalah sebagai berikut :

1. Pilih atribut sebagai *Node* akar.
2. Buat cabang untuk tiap-tiap nilai.
3. Bagi kasus dalam cabang.
4. Ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama (Kusrini & Luthfi, 2011)

Untuk memilih atribut sebagai *Node* akar, didasarkan pada nilai *Gain* tertinggi dari atribut-atribut yang ada. Untuk menghitung *Gain* digunakan rumus:

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(Si)$$

Keterangan :

S : Himpunan kasus

A : Atribut

n : Jumlah partisi atribut A

$|S_i|$: Jumlah kasus pada partisi ke i

$|S|$: Jumlah kasus dalam S

Setelah mendapatkan nilai *Gain*, ada satu hal lagi yang perlu kita lakukan perhitungan, yaitu mencari nilai *Entropy*. *Entropy* digunakan

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017

untuk menentukan seberapa informatif sebuah masukan atribut untuk menghasilkan keluaran atribut.

Rumus dasar dari *Entropy* tersebut adalah sebagai berikut :

$$Entropy(S) = \sum_{i=1}^{n} - pi * \log_2 pi$$

S : Himpunan Kasus

A : Fitur

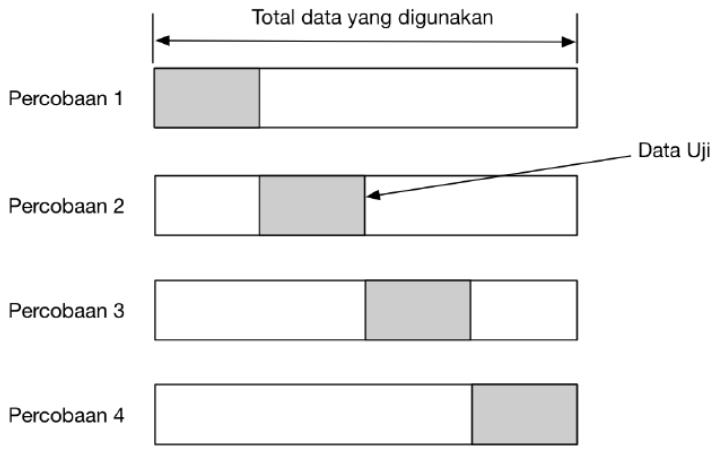
n : Jumlah partisi S

p_i : Proporsi dari Si terhadap S

2.14 Pengertian *K-Folds Cross Validation*

Dalam teknik ini data dibagi menjadi K bagian yang masing-masing memiliki jumlah yang sama. Bagian pertama digunakan sebagai data uji, bagian sisanya digunakan untuk data latih. Selanjutnya, bagian kedua digunakan sebagai data uji baru, bagian sisanya untuk data latih yang baru. Begitu seterusnya dilakukan berulang-ulang sebanyak K kali. Gambar 2.1 berikut mengilustrasikan prosedur dari *K-Fold Cross Validation*.

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017



Gambar 2.1 Metode Pembagian Data *K-Fold Cross Validation*

Keuntungan dari metode ini adalah keseluruhan data pernah digunakan sebagai data latih dan data uji, sehingga hasil *error rate* atau performa dari model dapat mewakili keseluruhan data. Tidak ada aturan khusus mengenai penentuan nilai jumlah *fold* (K) yang optimal. Nilai K yang direkomendasikan tergantung besarnya jumlah dataset. Untuk dataset yang besar, 3-*Fold Cross Validation* sudah cenderung akurat. Pilihan yang umum untuk nilai K pada *K-Fold Cross Validation* adalah 10.

2.15 Pengertian *Confusion Matrix*

Model klasifikasi yang dibangun perlu dievaluasi untuk mengetahui seberapa bagus model tersebut dalam melakukan klasifikasi yang diinginkan. Dalam mengevaluasi kinerja pengklasifikasi khususnya klasifikasi teks umumnya dilakukan dengan *accuracy* atau dengan *precision and recall* (Miner, et al, 2012). Nilai *accuracy* merepresentasikan seberapa banyak keseluruhan dokumen diklasifiksikan

dengan benar. Semakin tinggi nilai *accuracy* yang dihasilkan maka semakin bagus dan akurat model tersebut dalam melakukan klasifikasi. Persamaan untuk mendapatkan nilai *accuracy* adalah sebagai berikut:

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{Total\ kasus\ yang\ diklasifikasikan\ dengan\ benar}}{\boldsymbol{total\ kasus}}$$

Pengukuran *precision dan recall* merupakan metrik evaluasi yang paling sering digunakan pada kasus klasifikasi teks (Sokolova & Lapalme, 2009). Misalnya terdapat dua kelas A dan B, *precision* yaitu jumlah sampel berkategori A yang ditebak dengan benar sebagai A dibanding dengan jumlah total data yang ditebak sebagai A, sedangkan *recall* yaitu jumlah sampel berkategori A yang ditebak dengan benar dibandingkan dengan jumlah total sampel A. Dalam melakukan pengukuran ini biasanya dibangun *confusion matrix* yang merupakan sebuah tabel yang terdiri atas banyaknya baris data uji yang diprediksi benar dan tidak benar oleh model klasifikasi (Gambar 2.2).

Data class	Classified as *pos*	Classified as *neg*
pos	true positive (tp)	false negative (fn)
neg	false positive (fp)	true negative (tn)

Gambar 2.2 Contoh Tabel *Confusion Matrix*

Berikut merupakan rumus yang digunakan untuk menghitung *precision* dan *recall* :

$$\boldsymbol{Precision} = \frac{\boldsymbol{tp}}{\boldsymbol{tp} + \boldsymbol{fp}}$$

$$\boldsymbol{Recall} = \frac{\boldsymbol{tp}}{\boldsymbol{tp} + \boldsymbol{fn}}$$

2.16 Pengertian *Unified Modelling Language* (UML)

Penelitian ini menggunakan diagram UML (*Unified Modelling Language*) sebagai pemodelan dari aplikasi yang akan dibangun. Adapun pengertian UML menurut beberapa ahli, yaitu :

1. Menurut Widodo (2011) UML (*Unified Modelling Language*) adalah bahasa pemodelan standar yang memiliki sintak dan semantik.
2. Menurut Braun yang dikutip dalam Haviluddin (2011) *Unified Modelling Language* (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual.
3. Menurut Miftakhul Huda (2013), pemodelan (*modelling*) adalah proses merancang peranti lunak (*software*) sebelum melakukan pengkodean (*coding*).

Dalam standar pemodelan UML terdapat beberapa diagram. Diagram-diagram tersebut yaitu:

1. *Use Case Diagram*

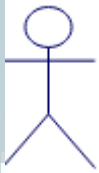
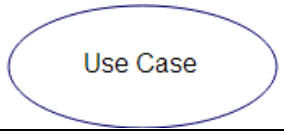
PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017

2. *Class Diagram*
3. *Statechart Diagram*
4. *Activity Diagram*
5. *Sequence Diagram*
6. *Collaboration Diagram*
7. *Component Diagram*
8. *Deployment Diagram* (Munawar, 2005).

2.16.1 Pengertian *Use Case Diagram*

Use Case Diagram adalah gambaran dari beberapa atau seluruh aktor dan *Use case* dengan tujuan mengenali interkasi mereka dalam suatu sistem. *Use Case Diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditentukan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *Use case* merepresentasikan sebuah interkasi antara aktor dengan sistem (Sugiarti, 2013).

Tabel 2.1 Simbol *Use Case Diagram*

Simbol	Nama	Keterangan
	*Actor*	Aktor hanya memberikan informasi ke sistem, aktor hanya menerima informasi dari sistem, aktor memberikan dan menerima informasi ke sistem dan dari sistem.
	*Use case*	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan

Simbol	Nama	Keterangan
		antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal di frase nama *Use case*.
	Association Relationship	Komunikasi antara aktor dan *Use case* yang berpartisipasi pada *Use case* atau *Use case* memiliki interaksi dengan aktor.

2.16.2 Pengertian *Class Diagram*

Diagram kelas atau *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Diagram kelas mendefinisikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat diantara mereka (Sugiarti, 2013).

Class diagram terdiri dari beberapa simbol. Beberapa simbol pada *class diagram* dijelaskan pada tabel 2.2.

Tabel 2.2 Simbol *Class Diagram*

No	Simbol	Keterangan
1	nama_kelas +atribut +operasi()	Simbol ini disebut *Class*. Yang mana penulisan di dalamnya tidak diperbolehkan menggunakan spasi. Simbol ini memiliki 3 susunan, yaitu nama kelas, atribut, dan *method*.
2		Simbol ini disebut *Association*. Garis ini merupakan garis yang digunakan untuk menghubungkan atau merelasikan kelas satu

No	Simbol	Keterangan
		dengan kelas yang lainnya dengan makna umum.
3		Simbol ini disebut *Generalisasi*. Yang mana digunakan untuk menghubungkan antar kelas dengan arti umum-khusus.
4		Simbol ini disebut *Aggregation*. Simbol ini digunakan untuk menghubungkan antar kelas dengan makna untuk semua bagian. Relasi ini digunakan jika kelas yang satu adalah semua bagian dari kelas yang lainnya.

2.16.3 Pengertian *Activity Diagram*

Activity diagram menggambarkan rangkaian alir aktivitas dalam sistem yang dirancang, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *Use case* atau interaksi. *Activity diagram* berupa *flowchart* yang digunakan untuk memperlihatkan alir kerja dari sistem. Diagram aktivitas juga banyak digunakan dalam mendefinisikan hal-hal berikut:

a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
b. Urutan atau pengelompokan tampilan dari sistem atau *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan pengujian yang perlu didefinisikan kasus ujinya.

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017

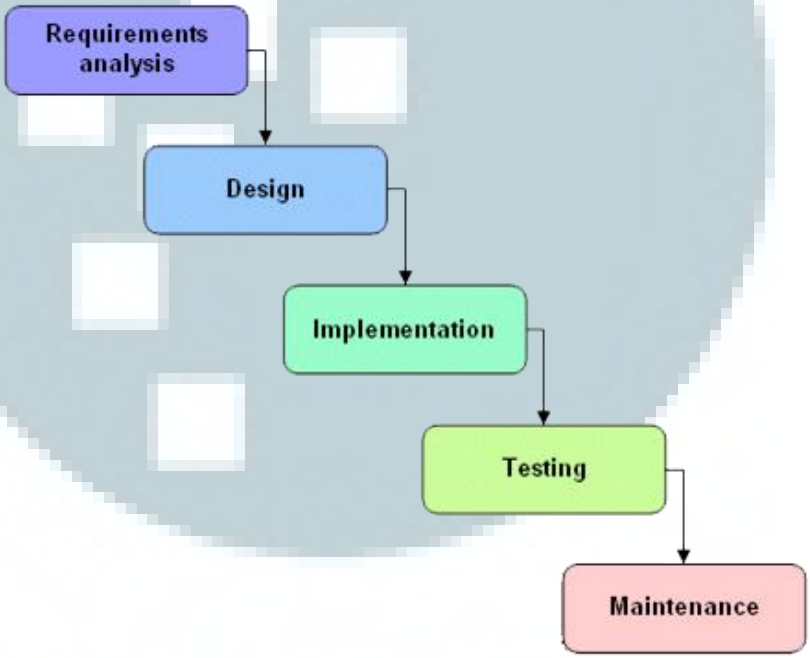
d. Rancangan menu yang ditampilkan pada perangkat lunak (Sugiarti, 2013).

Tabel 2.3 Simbol *Activity Diagram*

Simbol	Nama	Keterangan
	Status Awal	Status awal aktivitas sistem, sebuah aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan atau *decision*	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	Control Flow	Alur dari proses.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
nama swimlane	*Swimlane*	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.17 Pengertian *System Development Life Cycle* (SDLC)

Menurut Roger S. Pressman (2001), model ini biasa disebut juga dengan model *waterfall* atau disebut juga *classic life cycle*. Adapun pengertian dari SDLC ini adalah suatu pendekatan yang sistematis dan berurutan. Tahapan-tahapannya adalah *Requirements* (analisis sistem), *Analysis* (analisis kebutuhan sistem), *Design* (perancangan), *Coding* (implementasi), *Testing* (pengujian) dan *Maintenance* (perawatan).



Gambar 2.3 Tahapan *Waterfall*

Penjelasan-penjelasan SDLC Model *Waterfall*, adalah sebagai berikut:

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017

1. *Requirement Analysis*

   Pada tahap awal ini dilakukan analisa guna menggali secara mendalam terkait kebutuhan sistem. Terdapat bermacam-macam kebutuhan, seperti halnya kebutuhan *user*, kebutuhan data, dan kebutuhan teknologi.

2. *Design*

   Selanjutnya, mengacu pada hasil analisa kebutuhan sistem tersebut dilakukan penggambaran rancangan sistem dan *database*. Pada fase perancangan sistem ini digunakan *Unified Modeling Language* (UML) dan *Entity Relationship Diagram* (ERD) pada rancangan *database.*

3. *Implementation*

   Rancangan yang telah dibuat dalam tahap *design*, akan diterjemahkan ke dalam suatu bentuk atau bahasa yang dapat dibaca dan diterjemahkan oleh komputer untuk diolah. Tahap ini juga dapat disebut dengan tahap implementasi, yaitu tahap yang mengkonversi hasil perancangan sebelumnya ke dalam sebuah bahasa pemrograman yang dimengerti oleh komputer. Kemudian komputer akan menjalankan fungsi-fungsi yang telah didefinisikan sehingga mampu memberikan layanan-layanan kepada penggunanya.

4. *Testing*

   Pengujian program dilakukan untuk mengetahui kesesuaian sistem berjalan sesuai prosedur ataukah tidak. *Testing* juga

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017

dapat digunakan untuk memastikan kevalidan dalam proses *input*, sehingga dapat menghasilkan *output* yang sesuai.

5. *Maintenance*

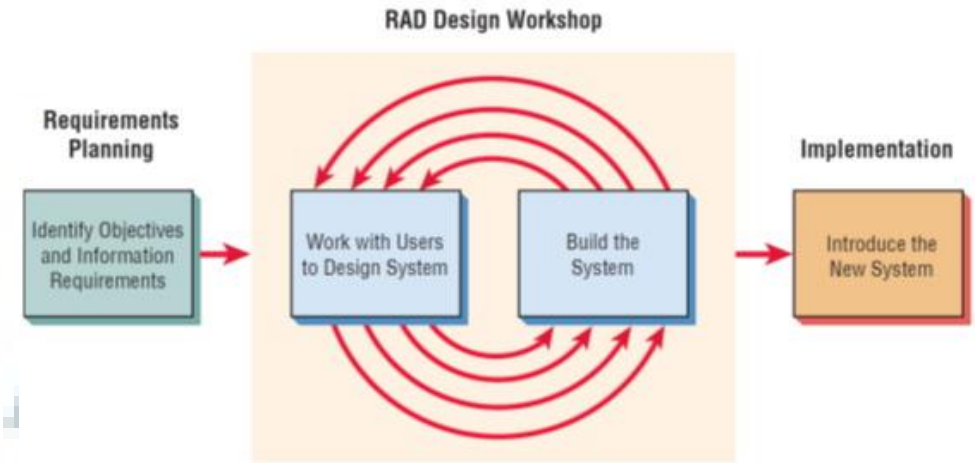
Setelah hasil pengujian sistem disetujui oleh *user* dan pihak pengembang, sistem yang dibuat akan diimplementasikan. Selanjutnya, dilakukan pemeliharaan secara berkala dan dilakukan perbaikan-perbaikan terhadap masalah yang ditemukan pada sistem.

2.18 Pengertian *Rapid Application Development* (RAD)

Menurut Pressman (2002), *Rapid Application Development* (RAD) adalah sebuah model proses pengembangan perangkat lunak sekuensial linear yang menekankan siklus pengembangan yang sangat pendek. Model RAD ini merupakan pengembangan dari metode sekuensial linier dimana perkembangan cepat dicapai dengan menggunakan model pendekatan berbasis komponen. Jika kebutuhan *user* dapat dipahami dengan baik, proses RAD memungkinkan pengembangan sistem dalam periode waktu yang lebih singkat

Sedangkan menurut Kendall (2010), RAD adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak. RAD bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara perancangan dan penerapan suatu sistem informasi.

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017



Gambar 2.4 Tahapan RAD

Pada gambar 2.4 dapat dilihat bahwa, metode RAD memiliki 3 tahapan utama dalam pengembangannya (Kendall, 2010):

1. *Requirements Planning* (Perencanaan Syarat-Syarat)

   Pada tahap awal ini dilakukan analisa guna menggali secara mendalam terkait kebutuhan sistem. Terdapat bermacam-macam kebutuhan, seperti halnya kebutuhan *user*, kebutuhan data, dan kebutuhan teknologi.

   Dalam fase ini, pengguna dan penganalisis bertemu untuk mengidentifikasikan tujuan-tujuan aplikasi atau sistem serta untuk megidentifikasikan syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. Orientasi dalam fase ini adalah menyelesaikan masalah-masalah perusahaan. Meskipun teknologi informasi dan sistem bisa mengarahkan sebagian dari sistem yang diajukan, fokusnya akan selalu tetap

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017

pada upaya pencapaian tujuan-tujuan perusahaan (Kendall, 2010).

2. RAD *Design Workshop* (*Workshop* Desain RAD)

Fase ini adalah fase untuk merancang dan memperbaiki, dimana dalam tahapan ini disebut *Workshop*. Penganalisis dan dan pemrogram dapat bekerja membangun dan menunjukkan representasi visual desain dan pola kerja kepada pengguna.

Workshop desain ini dapat dilakukan selama beberapa hari tergantung dari ukuran aplikasi yang akan dikembangkan. Selama *Workshop* desain RAD, pengguna merespon prototipe yang ada dan penganalisis memperbaiki modul-modul yang dirancang berdasarkan respon pengguna. Apabila seorang pengembangnya merupakan pengembang atau pengguna yang berpengalaman, Kendall menilai bahwa usaha kreatif ini dapat mendorong pengembangan sampai pada tingkat terakselerasi (Kendall, 2010).

3. *Implementation* (Implementasi)

Pada fase implementasi ini, penganalisis bekerja dengan para pengguna secara intens selama *Workshop* dan merancang aspek-aspek bisnis dan nonteknis perusahaan. Segera setelah aspek-aspek ini disetujui, sistem-sistem dibangun, dan disaring, sistem-sistem baru atau bagian dari sistem diujicoba dan kemudian diperkenalkan kepada organisasi (Kendall, 2010).

PEmodelan Prediksi Kemacetan..., Dimas Dwi Cahyo, FTI UMN, 2017