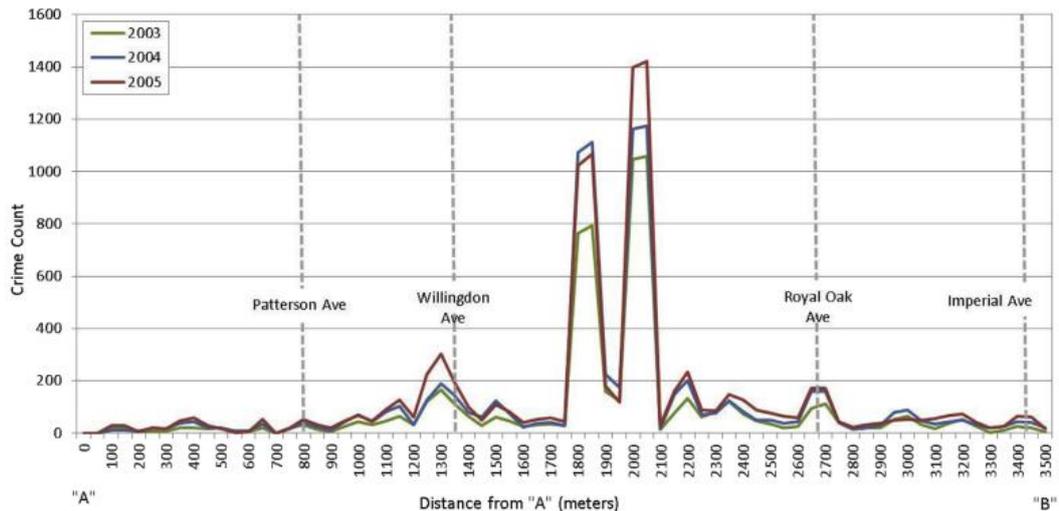


BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

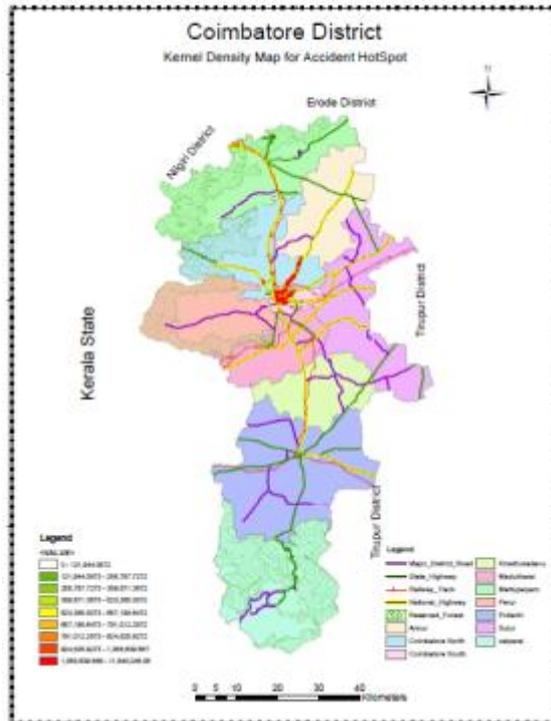
Penelitian sebelumnya pernah dilakukan oleh Spicer, et al (2015) dari *Institute of Canadian Research Studies*. Pada penelitian ini proses analisis dan visualisasi area kriminal dengan membandingkan lima teknik atau metode, yaitu *kernel density* digunakan untuk memvisualisasikan kepadatan, *aggregate count to address* digunakan untuk mengumpulkan data kriminal berdasarkan alamat tertentu, *aggregate count to street segment* digunakan untuk mengelompokkan data kriminal kemudian akan dianalisis untuk mengetahui tingkat kriminal yang tinggi, *street profile analysis* digunakan untuk menampilkan data kriminal pada lokasi tertentu menggunakan grafik, dan *line transect* untuk menganalisis pola kriminal pada jalan. Hasil penelitian ini adalah peneliti berhasil mengeksplorasi teknik baru untuk menganalisis perkembangan kejadian kriminal yang berfokus pada jalan di dalam wilayah perkotaan, yaitu menggunakan grafik *Street Profile Analysis*.



Gambar 2.1 Hasil Penelitian Spicer, et al

Sumber: Spicer et, al (2015)

Penelitian lainnya pernah dilakukan oleh Selvasofia & Arulraj (2016) dari *Karunya University India*. Pada penelitian tersebut membahas mengenai bagaimana menganalisis kecelakaan lalu-lintas menggunakan *GIS*. Objek yang diteliti adalah jarak antara titik kecelakaan dengan pusat kesehatan. Data yang digunakan pada penelitian ini dalam kisaran waktu 4 tahun dari 2011 sampai 2014. Ada beberapa metode dan teknik yang digunakan dalam penelitian ini, antara lain *Buffer Analysis*, *Kernel Density Analysis*, dan *Overlay Analysis*. Pada penelitian ini berhasil diidentifikasi lokasi kecelakaan yang paling rentan di Kabupaten Coimbatore. Lokasi yang merupakan paling rentan terjadi kecelakaan adalah di Periyanyakampalayam, Sarcarsamakulam, Thondamuthur, Madukkari dan Sular.



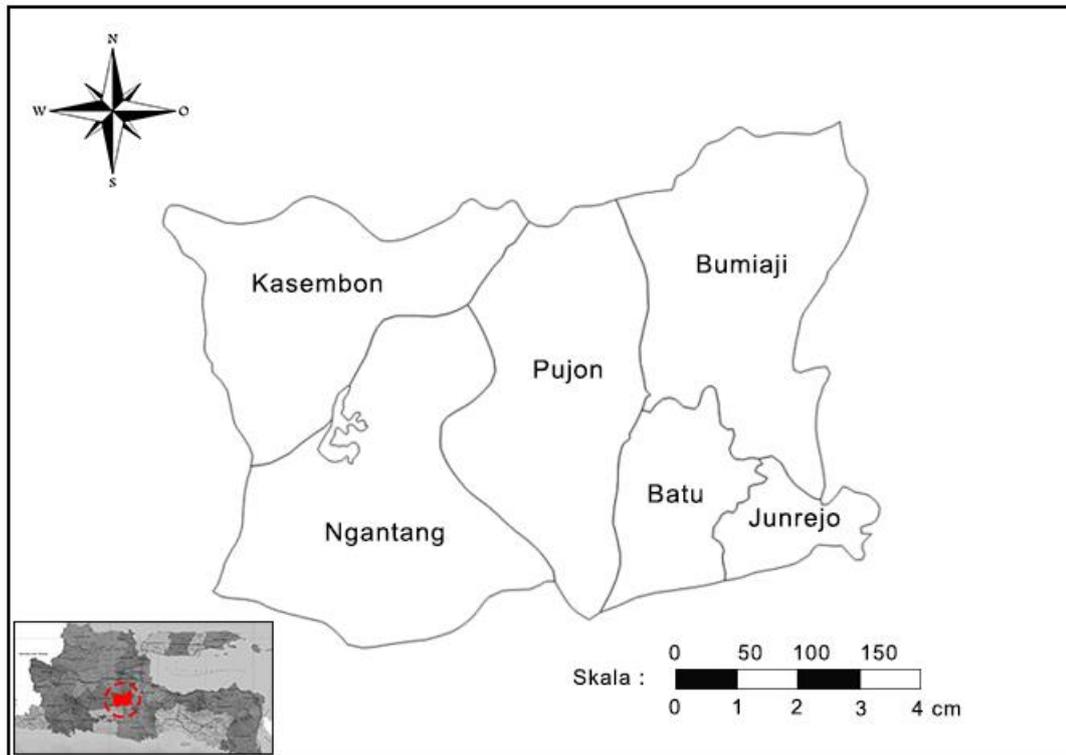
Gambar 2.2 Hasil Penelitian Selvasofia & Arulraj

Sumber: Selvasofia & Arulraj (2016)

Perbedaan penelitian ini dengan penelitian yang telah dilakukan sebelumnya adalah pada penelitian ini akan menghasilkan informasi pemetaan lokasi kecelakaan di daerah operasional Polres Kota Batu. Metode yang digunakan adalah *street profile* seperti pada penelitian Spicer, et al (2015) namun dengan fokus yang berbeda yaitu kecelakaan lalu-lintas. Hasil dari penelitian ini tidak hanya menampilkan grafik perkembangan kecelakaan, namun dilengkapi dengan sistem informasi pemetaan. Sistem ini akan melakukan beberapa analisis dan menampilkan informasi kecelakaan di daerah operasional Polres Kota Batu. Informasi ini diperuntukkan untuk pihak kepolisian mengidentifikasi lokasi rawan kriminalitas dan kecelakaan. Sedangkan untuk pihak masyarakat dapat digunakan untuk menambah kesadaran dan kewaspadaan masyarakat untuk lebih berhati-hati dalam berkendara.

2.2 Daerah Operasional Polres Kota Batu

Daerah operasional Polres Kota Batu adalah daerah yang menjadi tanggung jawab dari Polres Kota Batu. Polres Kota Batu menangani total 6 kecamatan yang terbagi dalam 1 kota dan 1 kabupaten. Kota pertama yaitu Kota Batu dengan 3 kecamatan, yaitu Kecamatan Junrejo, Kecamatan Batu, dan Kecamatan Bumiaji. Kemudian Kabupaten Malang dengan 3 kecamatan juga, yaitu Kecamatan Pujon, Kecamatan Ngantang, dan Kecamatan Kasembon.



Gambar 2.3 Peta Daerah Operasional Polres Kota Batu

2.3 Sistem Informasi

2.3.1 Sistem

Pemahaman terkait definisi sistem senantiasa berkembang sesuai dengan latar belakang pengertian sistem tersebut diterapkan. Menurut Kamus Besar Bahasa Indonesia (KBBI), sistem merupakan gabungan dari beberapa unsur perangkat yang saling berkaitan untuk mencapai sebuah tujuan. Sedangkan menurut Harary & Batell (1981), sistem biasanya melibatkan unit yang keterkaitannya disusun berdasarkan hirarki yang telah ditentukan sebelumnya.

2.3.2 Data

Data merupakan fakta atau angka yang dikumpulkan untuk diperiksa dan dipertimbangkan dan digunakan untuk membantu pengambilan keputusan. Namun data ini masih belum memiliki makna yang jelas. Data harus diolah atau dilengkapi dengan konteks sehingga dapat memiliki makna yang jelas (Cambridge, 2017). Beberapa contoh data antara lain adalah data yang menyatakan tanggal atau jam, atau menyatakan nilai mata uang.

2.3.3 Informasi

Informasi adalah hasil dari pengolahan data, biasanya oleh komputer. Hal ini menghasilkan fakta, yang memungkinkan data yang diolah digunakan dalam konteks dan sudah memiliki makna (Cambridge, 2017). Berbeda dengan data

biasa, informasi adalah data yang memiliki makna. Biasanya sebuah informasi sudah dapat dijadikan dasar untuk membuat keputusan.

2.3.4 Sistem Informasi

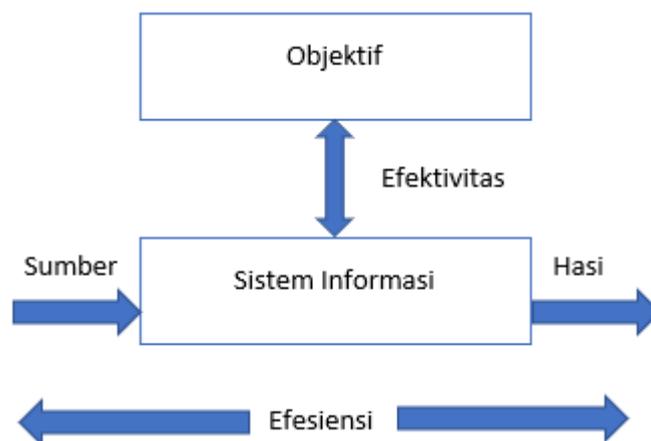
Sistem informasi merupakan sistem didalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategis dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Jogiyanto, 2005).

Sedangkan menurut Alter (2013) Sistem informasi (SI) melibatkan berbagai teknologi informasi (TI) seperti komputer, perangkat lunak, database, sistem komunikasi, internet, perangkat mobile dan banyak lagi, untuk melakukan tugas tertentu, berinteraksi dengan dan menginformasikan berbagai aktor dalam organisasi atau sosial yang berbeda. konteks. Kepentingan umum untuk bidang SI adalah semua aspek pengembangan, penerapan, penggunaan dan dampak SI dalam organisasi dan masyarakat.

Dari pengertian tersebut dapat tarik kesimpulan bahwa sistem informasi merupaan suatu alat yang didalamnya berisi kerangka kerja yang digunakan untuk memproses data berupa inputan data yang outputnya menjadi sebuah informasi yang dapat digunakan untuk tujuan tertentu.

Sistem Informasi memiliki dua fitur utama, yaitu (Tofan, 2013):

1. *Efficacy*: kemampuan untuk memberikan hasil sesuai dengan tujuan perusahaan.
2. *Efficiency*: kemampuan untuk mengoptimalkan penggunaan sumber daya, memungkinkan informasi memuaskan konsumen.



Gambar 2.4 Karakteristik Sistem Informasi

Sumber: Tofan (2013)

2.4 Sistem Informasi Geografis (SIG)

Sistem informasi geografis (SIG) memungkinkan kita memvisualisasikan, mempertanyakan, menganalisis, dan menginterpretasikan data untuk memahami

hubungan, pola, dan tren berdasarkan data spasial (Esri, 2012). SIG menguntungkan organisasi dari semua ukuran dan hampir di setiap industri. Ada ketertarikan dan kesadaran akan nilai ekonomi dan strategis SIG. SIG dapat memanfaatkan dukungan dari sistem untuk memasukkan data geografis, untuk menganalisisnya dengan berbagai cara, dan menghasilkan bentuk presentasi data (Huisman & de By, 2009). SIG mempunyai kemampuan untuk menghubungkan berbagai data pada suatu titik tertentu di bumi, menggabungkannya, menganalisis, dan akhirnya memetakan hasilnya. Data yang diolah pada SIG adalah data spasial yaitu sebuah data yang berorientasi geografis dan merupakan lokasi yang memiliki sistem koordinat tertentu. Sehingga aplikasi SIG dapat menjawab beberapa pertanyaan seperti lokasi, kondisi, tren, pola, dan pemodelan. Kemampuan inilah yang membedakan SIG dengan sistem informasi biasa.

2.4.1 Jenis Data Sistem Informasi Geografis

Umumnya jenis data yang digunakan untuk membangun SIG dibagi menjadi dua, yaitu data atribut dan data spasial (Buckley, 1997).

- Data atribut/ non-spasial, data yang merepresentasikan aspek-aspek deskriptif dari fenomena yang dimodelkannya. Misalnya data sensus penduduk, catatan survei, data statistik lainnya.
- Data grafis/ spasial, merupakan data yang merupakan representasi permukaan bumi yang memiliki referensi (koordinat) berupa peta, foto udara, citra satelit dan sebagainya atau hasil dari interpretasi data-data tersebut.

2.4.2 Komponen Sistem Informasi Geografis

Untuk membuat dan menggunakan SIG, diperlukan beberapa komponen. Komponen-komponen tersebut antara lain (Buckley, 1997):

- Orang yang mengembangkan, menjalankan, dan bertanggungjawab penuh atas SIG.
- Prosedur merupakan proses atau alur yang digunakan untuk mengolah data menjadi informasi yang berguna.
- Data merupakan bahan mentah yang dikumpulkan saat proses analisis kebutuhan sebelum mengembangkan SIG.
- *Software* merupakan perangkat lunak yang digunakan untuk mengolah, menyimpan, menganalisis, dan memvisualisasikan data spasial.
- *Hardware* merupakan perangkat keras yang dibutuhkan untuk menjalankan sistem. Misalnya, personal komputer, *scanner*, *digitizer*, dan lain sebagainya.

2.5 Quantum GIS (QGIS)

Quantum GIS (QGIS) adalah perangkat lunak untuk membangun Sistem Informasi Geografis (SIG). Perangkat lunak ini merupakan *open source* dengan

lisensi dibawah *GNU General Public License*. *QGIS* merupakan proyek tidak resmi dari *Open Source Geospatial Foundation (OSGeo)*. *QGIS* dapat dijalankan pada *Linux, Windows, Mac OSX, Windows*, serta mendukung banyak format dan fungsionalitas data vektor, raster, dan *database* (QGIS, 2015).

2.5.1 Open Street Map

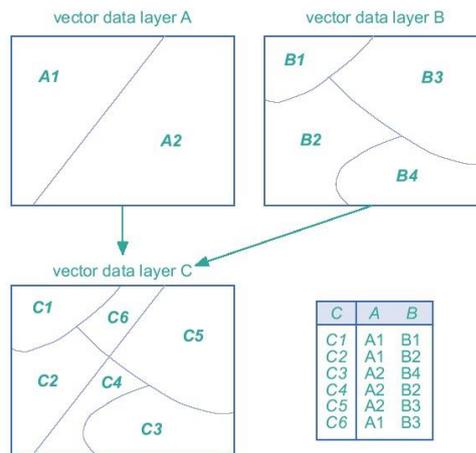
OpenStreetMap (OSM) adalah sebuah proyek berbasis web untuk membuat peta seluruh dunia yang gratis dan terbuka, dibangun sepenuhnya oleh sukarelawan dengan melakukan survey menggunakan *GPS*, mendigitasi citra satelit, dan mengumpulkan serta membebaskan data geografis yang tersedia di publik. Melalui *Open Data Commons Open Database License 1.0*, kontributor *OSM* dapat memiliki, memodifikasi, dan membagikan data peta secara luas (Openstreetmap, 2017). Terdapat beragam jenis peta digital yang tersedia di internet, namun sebagian besar memiliki keterbatasan secara legal maupun teknis. Hal ini membuat masyarakat, pemerintah, peneliti dan akademisi, inovator, dan banyak pihak lainnya tidak dapat menggunakan data yang tersedia di dalam peta tersebut secara bebas. Di sisi lain, baik peta dasar *OSM* maupun data yang tersedia di dalamnya dapat diunduh secara gratis dan terbuka, untuk kemudian digunakan dan didistribusikan kembali.

2.5.2 Map Layers

Dalam pengembangan Sistem Informasi Geografis, hal yang paling penting adalah konsep *layers* (Schenkelaars & Oosterom, 1995). Setiap layer dapat menggambarkan aspek tertentu dari keadaan dunia nyata yang dimodelkan. Dengan menggunakan konsep layer ini merupakan teknik alami untuk mengatur data dari sumber yang mungkin berbeda. Konsep layer ini juga dapat digunakan sebagai dasar operasi query yang memerlukan gabungan dari beberapa layer.

2.5.3 Overlay

Overlay merupakan teknik yang mengombinasikan dua atau lebih untuk menghasilkan data baru (Huisman & de By, 2009). Normalnya overlay dilakukan menggunakan dua *layer* data, dan mengasumsikan kedua data tersebut pada sistem dan area yang sama. Prinsip dari *overlay* adalah membandingkan karakteristik dari lokasi yang sama di kedua *layers* data, dan hasil akhirnya ditentukan oleh pengguna. Dalam domain vektor, *overlay* secara komputasi lebih mendesak daripada di domain raster. Prinsip *overlay* dapat dilihat pada Gambar 2.5. Fungsi *overlay* ini juga dapat berlaku untuk model data raster.



Gambar 2.5 Penerapan *Overlay*

Sumber: Huisman & de By (2009)

2.6 WebGIS

WebGIS adalah sebuah proses dalam merancang, mengimplementasikan dan mengirimkan data layanan geospasial menggunakan teknologi *World Wide Web* atau yang umum disebut sebagai *website*. Selama dua dekade terakhir, *WebGIS* mengalami perkembangan yang cukup pesat dimana presentasi sederhana sebuah peta statis diubah menjadi peta dinamis untuk mempermudah analisis data. Teknologi *WebGIS* telah dikembangkan oleh instansi pemerintahan, perusahaan, lembaga penelitian, dan masyarakat umum untuk digunakan sebagai pendukung keputusan, akses data spasial, eksplorasi dan visualisasi data spasial, ruang pengolahan analisis data dan pemodelan serta digunakan untuk mengintegrasikan layanan berbasis geo spasial dengan layanan proses komputasi dan lingkungan (Li, 2011).

2.6.1 Peta

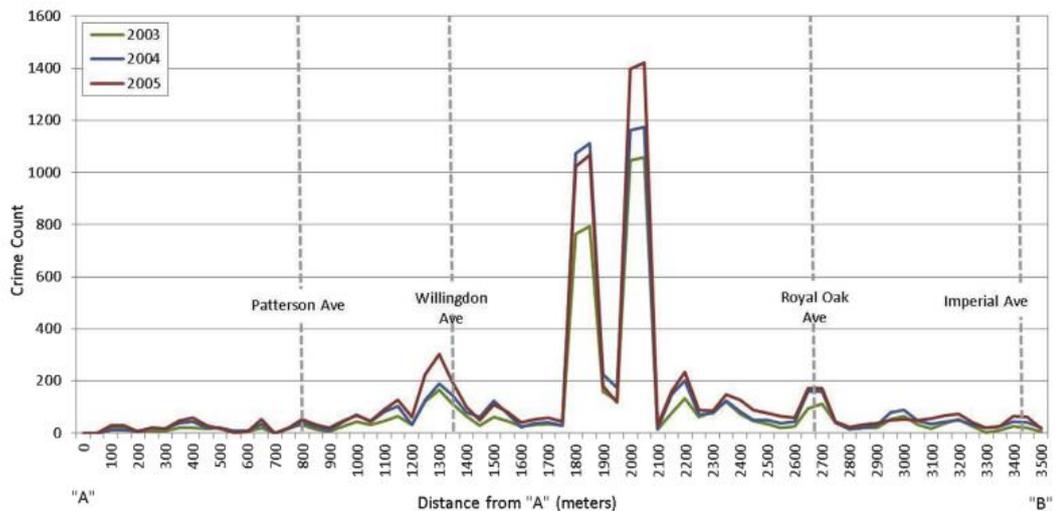
Peta merupakan salah satu cara terbaik mengetahui model dari dunia nyata. Peta sudah digunakan selama bertahun-tahun untuk merepresentasikan informasi tentang dunia nyata. Konsep dan desain peta telah berkembang menjadi sains dengan tingkat kecanggihan yang tinggi (Huisman & de By, 2009). Peta selalu merupakan representasi grafis pada tingkat detail tertentu, yang ditentukan oleh skala. Lembar peta memiliki batas fisik, dan fitur yang mencakup dua lembar peta harus dipotong-potong. Ada hubungan yang sangat kuat antara peta dengan SIG. Secara spesifik, peta dapat digunakan sebagai masukan untuk SIG.

2.6.2 Data *Vector*

Model data vektor menampilkan, menempatkan dan menyimpan data spasial dengan menggunakan titik, garis atau poligon beserta atribut-atributnya. Bentuk-bentuk dasar representasi data spasial ini, di dalam sistem model data vektor, didefinisikan oleh sistem koordinat kartesian dua dimensi (x, y) (Budiman, 2017).

2.8 Street Profile Analysis

Street Profile merupakan metode terbaru yang saat ini digunakan dalam proses pemetaan yang menggunakan objek berupa jalan (Spicer, et al, 2016). Sebagai contoh teknik pemetaan ini didasarkan pada konsep-konsep dari infrastruktur perkotaan dan dirancang untuk menunjukkan bagaimana kejahatan terjadi di tempat-tempat kecil dan dapat meningkat karena dinamika yang ada di lingkungan yang berubah-ubah. Metode ini dapat menemukan lokasi-lokasi tertentu, kemudian membantu dalam menganalisis dinamika temporal suatu kejadian. Dalam konteks ini, metode Street Profile dapat menampilkan varian kepadatan kejahatan dengan cara yang jelas. Dalam penerapannya, metode ini menampilkan perkembangan kriminal pada segmen jalan dalam bentuk grafik. Grafik ini dibuat menggunakan bantuan *Excel*. Spicer, et al (2016) pada penelitiannya menjelaskan bahwa metode ini bisa dikembangkan lagi dengan objek yang berbeda namun masih menggunakan objek jalan, misalnya analisis pada kecelakaan lalu-lintas.



Gambar 2.7 Contoh Grafik *Street Profile*

Sumber: Spicer, et al(2016)

2.9 Equivalent Accident Number (EAN)

Equivalent Accident Number atau *EAN* merupakan salah satu perhitungan yang dapat digunakan untuk menentukan tingkat kecelakaan. Pengertian *EAN* itu sendiri adalah angka yang digunakan untuk menilai kejadian kecelakaan lalu lintas. *EAN* adalah skala numerik untuk menimbang tingkat kecelakaan. Hal ini dihitung dengan membandingkan perkiraan kerugian yang disebabkan oleh berbagai tingkat kecelakaan, yaitu korban jiwa atau kematian (*FAT*), luka berat atau luka parah (*SVI*), luka ringan atau ringan (*MNI*), dan properti yang hanya rusak (*PDO*) (Sugiyanto, et al, 2017).

Teknik identifikasi perangkingan lokasi kecelakaan dilakukan dengan cara menentukan *Weighted Accident Number (WAN)*. Ada beberapa jenis atau tingkat

kecelakaan berdasarkan tingkat keparahan korban sehingga angka kecelakaan perlu diberi bobot dengan nomor kecelakaan setara menjadi *WAN*. Pada Tabel 2.1 ditunjukkan bobot *EAN* yang didapatkan dari beberapa instansi antara lain Puslitbang Jalan, Direktorat Jenderal Perhubungan Darat (Ditjen Hubdat), Polisi Republik Indonesia (Polri), dan *Accident Blackspot Investigation/ Unit Penelitian Kecelakaan Lalu Lintas (ABIU/ UPK)*

Table 2.1 Bobot Equivalent Accident Number

Tingkat Kecelakaan	Puslitbang Jalan	Ditjen Hubdat	Polri	ABIU/ UPK	Nilai Rata-rata
Meninggal (<i>FAT</i>)	12	12	10	6	10
Luka Berat (<i>SVI</i>)	3	6	5	3	4.25
Luka Ringan (<i>MNI</i>)	3	3	1	0.8	2.33
Properti Yang Hanya Rusak (<i>PDO</i>)	1	1	1	0.2	1

Sumber: Sugiyanto, et al (2017)

Untuk menghitung *WAN* dapat menggunakan rumus yang ditunjukkan pada rumus 2.1.

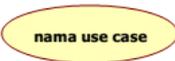
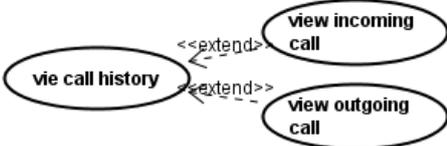
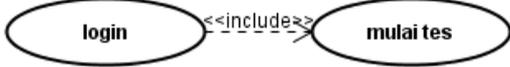
$$WAN = (10 \times FAT) + (4,25 \times SVI) + (2.33 \times MNI) + (1 \times PDO) \quad (2.1)$$

2.10 Unified Modelling Language (UML)

2.10.1 Use Case Diagram

Use Case mendeskripsikan sebuah interaksi antar satu atau lebih aktor dengan sistem informasi yang akan dibuat. Pada umumnya *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi yang akan dibangun. Dalam penerapannya, *Use Case Diagram* digunakan untuk mengidentifikasi hubungan antar fungsi dan memberi nama *use case* sesuai dengan interaksinya (Sommerville, 2011). Hal ini kemudian ditambah dengan informasi tambahan yang menggambarkan interaksi dengan sistem. Pada Tabel 2.2 merupakan simbol-simbol yang digunakan dalam membuat *use case*.

Table 2.2 Tabel Simbol Use Case Diagram

Simbol	Nama	Deskripsi
	Aktor	Aktor merepresentasikan peran-peran yang dapat diambil oleh user untuk dapat berinteraksi dengan sistem. Aktor tidak selalu berwujud manusia, aktor juga dapat berbentuk sistem lain.
	Use Case	Use Case merupakan gambaran dari aksi-aksi yang dapat dilakukan oleh aktor
	Relationship	Relationship menggambarkan hubungan khusus antar elemen yang memiliki makna tertentu
	Extend Relationship	<p>Extend relationship merupakan relasi yang menggambarkan hubungan sebuah Use Case dengan Use Case tambahan lain yang bersifat optional.</p> <p>Contoh: Use Case “view incoming calls” dan “view outgoing calls” adalah optional Use Case dari “view call history”.</p> 
	Include Relationship	<p>Include relationship merupakan relasi yang mengharuskan sebuah Use Case dijalankan terlebih dahulu agar dapat menjalankan Use Case lain.</p> <p>Contoh: Use Case “mulai tes” hanya dapat dijalankan setelah Use Case “Login” telah dijalankan.</p> 

Sumber: Rosa & Shalahuddin (2015)

2.10.2 Activity Diagram

Activity Diagram menggambarkan workflow atau aktivitas dari sebuah sistem atau menu yang ada pada perangkat lunak. Yang penting pada diagram ini bahwa diagram aktivitas yang dibuat dapat menggambarkan aktivitas sistem, bukan

aktivitas aktor (Sommerville, 2011). Pada Tabel 2.3 merupakan simbol-simbol yang digunakan dalam membuat *activity diagram*.

Table 2.3 Tabel Simbol Activity Diagram

Simbol	Deskripsi
Status awal 	Simbol yang menunjukkan status awal dari suatu sistem
Aktivitas 	Simbol yang menunjukkan suatu tugas atau pekerjaan yang dilakukan sistem, dan biasanya diawali dengan kata kerja.
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem.

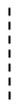
Sumber : Rosa & Shalahuddin (2015)

2.10.3 Sequence Diagram

Sequence Diagram merupakan diagram urutan pada *UML*. Diagram ini digunakan untuk memodelkan interaksi antar aktor dengan objek dalam sistem maupun interaksi antar objek itu sendiri (Sommerville, 2011). Pada *Sequence Diagram* terdapat beberapa sintaks yang digunakan untuk menggambarkan interaksi antar aktor dengan objek maupun interaksi antar objek itu sendiri. Banyaknya *sequence* yang harus dibuat biasanya sebanyak fungsional yang ada pada *Use Case*. Sehingga semakin banyak fungsi pada *Use Case* maka semakin banyak pula *sequence* yang dibuat. Pada Tabel 2.4 merupakan simbol-simbol yang digunakan dalam membuat *sequence diagram*.

Table 2.4 Tabel Simbol Sequence Diagram

Simbol	Nama	Deskripsi
	Aktor	Himpunan dari objek yang saling berbagi atribut dan metode yang sama.

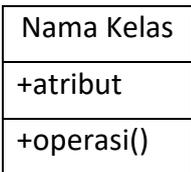
	<i>Lifeline</i>	Menyatakan rentan waktu dimana objek dapat hidup di dalamnya.
	Waktu aktif	Menyatakan jika sebuah objek dalam kondisi aktif dan tengah melakukan transaksi.
1 : nama_metode() 	<i>Call</i>	Menggambarkan jika sebuah objek memanggil operasi atau metode tertentu, baik dari objek lain atau objek itu sendiri.
1 : keluaran 	<i>return</i>	Menggambarkan sebuah <i>return</i> dari suatu objek setelah menjalankan operasi tertentu.

Sumber : Rosa & Shalahuddin (2015)

2.10.4 Class Diagram

Class Diagram digunakan oleh pengembang sistem untuk mengembangkan model sistem berorientasi objek. Dalam *Class Diagram* dapat ditunjukkan kelas dalam sebuah sistem dan asosiasi di antara kelas-kelas ini (Sommerville, 2011). Sebuah asosiasi adalah penghubung antara kelas yang menunjukkan bahwa ada hubungan antara kelas-kelas ini. Akibatnya, setiap kelas mungkin memiliki pengetahuan tentang kelas terkait. Pada Tabel 2.2 merupakan simbol-simbol yang digunakan dalam membuat *class diagram*.

Table 2.5 Tabel Simbol *Class Diagram*

Simbol	Nama	Deskripsi
	Kelas	Himpunan dari objek yang saling berbagi atribut dan metode yang sama.
 nama_interface	<i>Interface</i>	Memiliki konsep yang sama dengan <i>interface</i> dalam pemrograman berorientasi objek.
	<i>Association</i>	Menggambarkan relasi antar kelas dengan makna yang umum.

→	<i>Directed Association</i>	Menggambarkan relasi antar kelas dengan makna “kelas yang satu digunakan oleh kelas yang lain”.
→▷	<i>Generalization</i>	Menggambarkan relasi antar kelas dengan makna “generalisasi-spesialisasi (umum-khusus)”.
- - - →	<i>Dependency</i>	Menggambarkan relasi antar kelas dengan makna “kebergantungan sebuah kelas dengan kelas yang lain”.

Sumber : Rosa & Shalahuddin (2015)

2.11 Pengujian Perangkat Lunak

Pengujian perangkat lunak membutuhkan perancangan kasus uji dengan tujuan untuk mendapatkan kesalahan dengan waktu yang relatif singkat. Ada banyak metode-metode perancangan kasus uji yang telah berkembang guna menyediakan pendekatan pengujian secara sistematis bagi para pengembang. Metode perancangan kasus uji ini juga menyediakan mekanisme yang dapat membantu kelengkapan dari pengujian dan memberikan kemungkinan tertinggi untuk dapat menemukan berbagai kesalahan dalam sebuah sistem atau perangkat lunak (Pressman, 2001).

2.11.1 Black-Box Testing

Black box testing adalah metode pengujian perangkat lunak yang dilakukan pada fungsional sistem untuk mengetahui *output* sistem dari masukan yang sudah ditentukan (Khan, 2011). Pengetahuan khusus dari kode aplikasi/ struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Tes ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional. Perancang uji memilih *input* yang valid dan tidak valid dan menentukan *output* yang benar.

Black Box Testing bukanlah solusi dari masalah alternatif pada *White Box Testing*, tetapi merupakan pelengkap untuk menguji hal yang tidak bisa dilakukan oleh *White Box Testing* (Nidhra & Dondeti, 2012). Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: unit, integrasi, fungsional, sistem dan penerimaan. Metode *black box* memfokuskan pada keperluan fungsional dari *software*. Karna itu ujicoba *black box* memungkinkan pengembang *software* untuk membuat beberapa kondisi *input* yang akan digunakan untuk menguji fungsional suatu program (Kompasiana, 2010).

2.11.2 Compatibility Testing

Compatibility Testing merupakan salah satu pengujian yang dilakukan untuk menguji dan mengetahui apakah terdapat masalah pada sebuah sistem *web* saat dijalankan pada beberapa perangkat *browser* (Pressman, 2001). Pengujian ini sangat diperlukan karena setiap komputer memiliki sistem operasi, *browser*, dan kecepatan jaringan yang berbeda. Setiap perbedaan ini pasti mengakibatkan perbedaan kecepatan dan hasil dari pemrosesan sebuah sistem. Terkadang beberapa browser tidak mendukung *plugin* yang mungkin diperlukan untuk melakukan proses pada sistem *web*.

2.11.3 User Acceptance Testing

User Acceptance Testing atau *UAT* merupakan salah satu dari bagian pengembangan sebuah sistem yang digunakan untuk menentukan apakah sistem telah memenuhi kriteria penerimaan atau tidak (Watanabe, et al, 2012). Pengujian ini tidak digunakan untuk memastikan sistem tidak akan *error*, namun memastikan bahwa solusi dari sistem yang telah dibangun benar-benar bekerja untuk pengguna (Sualim, 2016).

Pengujian ini biasanya dilakukan pada tahap akhir pengembangan sebuah sistem. Jika kriteria penerimaan telah terpenuhi, maka sistem siap dirilis. Responden yang melakukan pengujian ini berasal dari pegawai atau karyawan yang kedepannya akan menggunakan sistem. Untuk jumlah responden biasanya bermacam-macam. Pada penelitian Mohd & Shahbodin (2015) dilakukan *UAT* terhadap 30 murid sebagai responden.

2.12 Likert Scale

Skala Likert merupakan salah satu cara yang berguna dalam studi dimana hubungan antara sikap dan atau persepsi individu diperikasa dan dianalisis (McLeod, et al, 2011). Penggunaan skala ini yaitu dengan meminta beberapa responden untuk memberikan opini maupun tanggapan mengenai pernyataan yang diajukan. 30 responden merupakan standar pengambilan nilai. Responden akan menilai sejauh mana mereka setuju ataupun tidak mengenai solusi dari masalah yang telah ditentukan. Responden akan memilih satu dari beberapa pilihan yang telah disediakan (Choizes, 2017). Dalam implementasi skala *likert* terdapat beberapa pertanyaan yang masing-masing memiliki bobot tertentu. Skala *likert* memiliki keuntungan sehingga dengan hanya mengharapkan jawaban ya atau tidak yang sederhana dari responden, dan bahkan tidak ada pendapat sama sekali (McLeod, 2017). Oleh karena itu data kuantitatif diperoleh, yang berarti bahwa data dapat dianalisis dengan relatif mudah.

2.12.1 Cara Menghitung Menggunakan Likert Scale

Proses perhitungan skala *likert* diawali dengan memberikan bobot nilai pada setiap jawaban. Pada Tabel 2.6 merupakan rincian bobot nilai jawaban. Sedangkan Tabel 2.7 merupakan persentase untuk setiap jawaban.

Table 2.6 Bobot Nilai

A = Sangat Setuju	5
B = Setuju	4
C = Netral	3
D = Tidak Setuju	2
E = Sangat Tidak Setuju	1

Sumber: Choizes (2017)

Table 2.7 Persentase Nilai

80 sampai 100%	Sangat Setuju
60 sampai 79,99%	Setuju
40 sampai 59,99%	Netral
20 sampai 39,99%	Tidak Setuju
0 sampai 19,99%	Sangat Kurang Setuju

Sumber: Choizes (2017)

Setelah bobot dan persentase nilai ditentukan, tahap selanjutnya adalah menghitung jumlah jawaban dengan dikalikan bobot nilai. Untuk perhitungannya adalah sebagai berikut:

1. Jawaban Sangat Setuju (A) = $n \times 5$
2. Jawaban Setuju (B) = $n \times 4$
3. Jawaban Netral (C) = $n \times 3$
4. Jawaban Tidak Setuju (D) = $n \times 2$
5. Jawaban Sangat Tidak Setuju (E) = $n \times 1$

$$\text{Total Nilai} = (n \times 5) + (n \times 4) + (n \times 3) + (n \times 2) + (n \times 1) \quad (2.2)$$

n merupakan jumlah responden yang menjawab. Untuk melakukan perhitungan selanjutnya, harus diketahui terlebih dahulu nilai tertinggi (X). Rumus yang digunakan dapat dilihat pada Rumus 2.3.

$$X = \text{Nilai Tertinggi Likert} \times (\text{Jumlah Responden} \times \text{Total Pertanyaan}) \quad (2.3)$$

Sedangkan untuk menghitung hasil dari UAT dengan menggunakan skala *likert* dapat dilihat pada rumus 2.5.

$$\text{Rumus Index \%} = (\text{Total Nilai} \div X) \times 100 \quad (2.4)$$