

BAB II

LANDASAN TEORI

2.1 Kecelakaan Lalu Lintas

“Kecelakaan lalu lintas merupakan suatu kejadian atau peristiwa dimana beberapa kendaraan saling bersinggungan sehingga menyebabkan kerugian baik dalam bentuk materi maupun kesehatan. Terdapat lima jenis dan bentuk kecelakaan lalu lintas, yaitu: berdasarkan korban kecelakaan, berdasarkan lokasi kejadian, berdasarkan waktu terjadinya, berdasarkan posisi kecelakaan, dan berdasarkan jumlah kendaraan yang terlibat” (Depkimpraswil, 2004).

Kecelakaan lalu lintas juga terjadi disebabkan oleh “daerah rawan atau suatu lokasi dimana angka kecelakaan cukup tinggi dengan kejadian kecelakaan berulang-ulang dalam suatu daerah dan rentang waktu yang relatif sama yang diakibatkan oleh suatu penyebab tertentu” (Depkimpraswil, 2004).

2.1.1 Faktor-Faktor Kecelakaan Lalu Lintas

Kecelakaan lalu lintas merupakan suatu masalah yang perlu mendapatkan perhatian yang lebih dari pihak pemerintah maupun masyarakat. Ada beberapa faktor yang menyebabkan terjadinya kecelakaan lalu lintas diantara sebagai berikut:

1. Faktor manusia

Faktor manusia merupakan faktor utama yang sering menyebabkan kecelakaan lalu lintas karena sengaja melanggar rambu-rambu lalu lintas, ketidaktahuan terhadap aturan lalu lintas atau pura-pura tidak tahu. Selain itu manusia juga sebagai pengguna jalan raya sering sekali lalai dan juga dalam berkendara melebihi batas kecepatan yang telah ditentukan.

2. Faktor kendaraan

Faktor kendaraan yang paling sering terjadi karena kelalaian perawatan terhadap kendaraan seperti rem blong, setir macet, dan lain-lain.

3. Faktor jalan dan lainnya

Faktor jalan terkait dengan kecepatan, rencana jalan, geometrik jalan, pagar pengaman, median jalan, jarak pandang dan kondisi permukaan jalan. Kondisi jalan yang rusak atau berlobang dapat membahayakan pemakai jalan terutama bagi pemakai sepeda motor dan mobil.

4. Faktor Cuaca

Ketika cuaca hujan dapat memengaruhi kinerja atau kondisi kendaraan yang sedang berjalan seperti jalan menjadi lebih licin, jarak pandang juga terpengaruh karena penghapus kaca tidak bisa bekerja secara sempurna atau lebatnya hujan mengakibatkan jarak pandang menjadi lebih pendek. Asap dan kabut juga mempengaruhi jarak pandang terutama di daerah pegunungan.

2.1.2 Karakteristik Jenis Korban

Kecelakaan lalu lintas merupakan “suatu peristiwa yang tidak disangka-sangka dan tidak disengaja melibatkan kendaraan dengan atau tanpa pemakai jalan lainnya, yang mengakibatkan korban manusia (mengalami luka ringan, luka berat, dan meninggal) dan kerugian harta benda” (UU No. 14 tahun 1992). PP RI No. 43 tahun 1993 tentang prasarana dan lalu lintas mendefinisikan kriteria korban kecelakaan lalu lintas sebagai berikut:

1. Luka ringan adalah korban yang mengalami luka-luka yang tidak harus dirawat lama seperti luka goresan.
2. Luka berat adalah korban yang karena luka-lukanya menderita cacat tetap atau harus dirawat dalam jangka waktu lebih dari 30 hari sejak terjadinya kecelakaan. Arti cacat tetap: bila sesuatu anggota badan hilang atau tidak digunakan sama sekali dan tidak dapat sembuh/ pulih untuk selama-lamanya.
3. Meninggal adalah korban yang dipastikan mati sebagai akibat kecelakaan lalu lintas dalam waktu paling lama 30 hari setelah kejadian tersebut.

2.1.3 Pengertian Umur

Umur adalah lamanya waktu hidup yaitu terhitung sejak lahir sampai dengan sekarang. Penentuan umur dilakukan dengan menggunakan hitungan tahun. Menurut (Elizabeth, 2005) usia adalah “umur individu yang terhitung mulai saat dilahirkan sampai berulang tahun”.

Pembagian umur berdasarkan psikologi perkembangan (Santrock, 2008) terbagi atas tingkatan umur manusia pada tabel 2.1 dibawah ini.

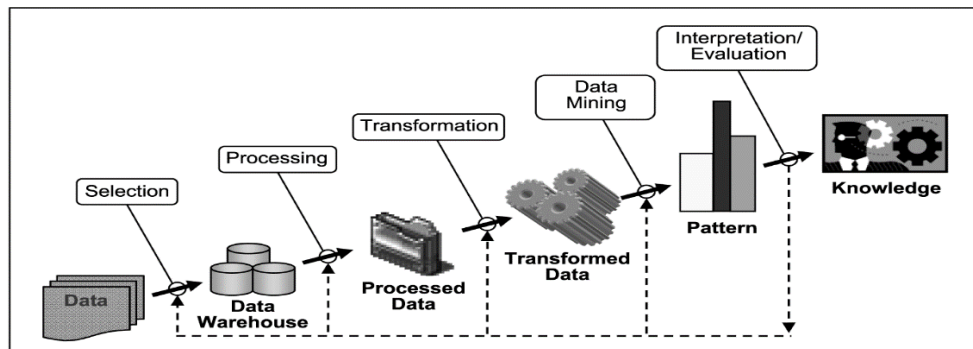
Tabel 2.1 Pembagian Umur

| No | Masa | Umur |
|----|---------------|------------------------------------|
| 1 | Bayi | Dari lahir hingga 18 atau 24 bulan |
| 2 | Anak awal | 24 bulan - 5 atau 6 tahun |
| 3 | Anak tengah | 6 - 11 tahun |
| 4 | Remaja awal | 12 - 15 tahun |
| 5 | Remaja akhir | 15 - 19 tahun |
| 6 | Dewasa awal | 20 – 30 tahun |
| 7 | Dewasa tengah | 31 – 59 tahun |
| 8 | Dewasa akhir | 60 - kematian |

2.2 *Knowledge Discovering In Database (KDD)*

Knowledge Discovering In Database (KDD) merupakan “proses yang menggunakan metode data mining dalam keseluruhan proses non-trivial untuk mencari data dan mengidentifikasi pola tertentu dalam data, dimana pola yang ditemukan bersifat sah, dengan demikian pola yang ditemukan dapat bermanfaat dan dapat dimengerti” (Fayyad, 1996) yang dikutip oleh (Kusrini dkk, 2009).

KDD bertugas untuk mengekstrak pola atau model dari data dengan menggunakan suatu algoritma yang spesifik. Berikut adalah tahapan-tahapan dalam KDD yaitu:



Gambar 2.1 Tahapan-tahapan KDD

Berikut adalah keterangan dari gambar 2.1:

1. *Data Selection* adalah pemilihan data dari sekumpulan data operasional yang dilakukan sebelum tahap penggalian informasi dalam KDD.
2. *Preprocessing* adalah tahapan sebelum proses data mining dilaksanakan, perlu dilakukan proses *cleaning* dengan tujuan untuk membuang duplikasi data atau data ganda, memeriksa data yang inkonsisten, memperbaiki kesalahan pada data dan menghapus data yang tidak diperlukan.
3. *Transformation* adalah proses perubahan data yang telah dipilih sehingga data tersebut sesuai untuk proses data mining.
4. *Data mining* adalah proses mencari pola atau informasi yang menarik dan dimengerti oleh *user* dengan data terpilih melalui teknik dan metode tertentu.
5. *Interpretation/ Evaluation* adalah pola informasi yang dihasilkan dari proses data mining perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pengguna (*user*). Pada tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan atau sesuai dengan fakta yang ada pada sebelumnya atau tidak.
6. *Knowledge presentation* merupakan suatu gambaran dari teknik visualisasi dan berupa pengetahuan yang digunakan untuk memberikan pengetahuan kepada *user*.

2.3 Data Mining

Menurut (Han, 2006), “data mining adalah sebuah ilmu yang mempelajari mengenai metode yang digunakan untuk mengekstrak pengetahuan atau menemukan pola dari suatu data”. Data mining adalah proses memperkerjakan satu atau lebih teknik pembelajaran (*machine learning*) untuk menganalisa dan mngekstraksi pengetahuan (*knowledge*) secara otomatis. Berikut merupakan beberapa pengelompokan data mining:

1. Klasifikasi dalam kelompok klasifikasi terdapat target kelas/ atau variabel kategori. Algoritma untuk melakukan klasifikasi antara lain *Modified Klasifikasi Nearest Neighbor, Naive Bayes Classifier, Neural Network*, pohon keputusan dan *Support Vector Machines*.
2. Klastering adalah mencari kelompok atau pengelompokan atribut tanpa mengetahui hasil berapa kelompok. Algoritma yang digunakan pada klastering antara lain, *Fuzzy c-means, K-means* dan sebaliknya.
3. Deskripsi digunakan untuk menggambarkan suatu permasalahan pola atau kecenderungan dalam suatu data.
4. Estimasi dalam kelompok estimasi hampir sama dengan klasifikasi bedanya estimasi mengarah ke numerik dari pada pengkategorian (kelas).
5. Prediksi dalam kelompok prediksi memiliki kemiripan dengan estimasi dan klasifikasi, tapi dalam kelompok prediksi mengetahui hasil dan nilai pada masa akan datang dan kelompok klasifikasi dan estimasi dapat diterapkan pada kelompok prediksi.
6. Asosiasi digunkana untuk mencari hubungan atau keterkaitan agar dapat menemukan atribut yang muncul dan assosiasi digunakan untuk dapat memenuhi suatu kebutuhan dalam menganalisa.

2.4 Association Rules

Association Rule merupakan suatu prosedur untuk mencari hubungan tiap *item* dalam suatu *dataset* yang digunakan. Pada *assotiation rule* memberikan informasi dalam bentuk “*if - then*” atau “jika - maka”. Istilah jika diwakili dengan *antecedent* sedangkan maka diwakili dengan istilah *consequent*.

Pada *association rule* terdapat istilah *support*. *Support* merupakan suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu *itemset* dari keseluruhan transaksi. Untuk mencari nilai *support 1-itemset* menggunakan persamaan 2.1 sedangkan untuk mencari nilai *2-itemset* menggunakan persamaan 2.2 di bawah ini.

$$Support(A) = \frac{JumlahTransaksiyangMengandungA}{TotalTransaksi} \times 100\% \dots\dots\dots(2.1)$$

$$Support(A, B) = \frac{JumlahTransaksiyangMengandungAdanB}{TotalTransaksi} \times 100\% \dots\dots\dots (2.2)$$

Untuk mencari hubungan antara dua atau lebih *item* secara *conditional* disebut dengan istilah *confidence*. Misalnya “jika A maka B” bentuk demikian disebut sebagai pola hubungan. Jika pola hubungan nilai *confidence* dibawah *threshold* maka bisa diabaikan. Persamaan rumus *confidence* dijelaskan pada persamaan 2.3 di bawah ini:

$$Confidence(A \rightarrow B) = \frac{Jumlah\ Transaksi\ yang\ Mengandung\ A\ dan\ B}{Jumlah\ Transaksi\ yang\ Mengandung\ A} \times 100\% \dots (2.3)$$

Salah satu algoritma yang merupakan teknik asosiasi pada data *mining* adalah algoritma *Frequent Pattern Growth (FP-growth)*.

2.5 Algoritma *Frequent Pattern-Growth (FP-Growth)*

Frequent Pattern-Growth (FP-Growth) merupakan salah satu *alternative* algoritma yang bisa digunakan dalam penentuan himpunan data yang paling sering muncul (*frequent itemset*) pada sebuah kumpulan data (Riarianti, 2014).

Untuk mencari *frequent itemset* dengan menggunakan *FP-growth* adalah dengan cara menggunakan *FP-tree* yang merupakan perluasan dari pohon *prefix*. Langkah pertama yang dilakukan yaitu pembentukan *FP-tree*. Setelah pembentukan *FP-tree* dapat diterapkan algoritma *FP-growth* untuk langsung mengekstrak *frequent itemset* dari *FP-tree* yang telah ada dengan menggunakan prinsip *divide and conquer*.

2.5.1 Pembentukan *Frequent Pattern Tree (FP-Tree)*

Untuk mencari *frequent itemset* dengan algoritma *FP-Growth* dapat menggunakan struktur data perluasan dari penggunaan sebuah pohon *prefix* atau disebut *FP-Tree*. Setelah menggunakan *FP-Tree* maka algoritma *FP-Growth* bisa mengekstrak *frequent itemset* dari *FP-Tree* yang telah dibentuk dengan menggunakan prinsip *divide and conquer* (Ririanti, 2014).

FP-Tree dibangun dengan cara memetakan setiap data transaksi ke dalam lintasan tertentu. Jika terdapat transaksi yang memiliki *item* yang sama maka lintasan yang terbentuk akan terjadinya saling menimpa. Proses pemampatan dengan struktur *FP-Tree* akan efektif jika banyak terdapat data transaksi yang memiliki *item* yang sama (Ririanti, 2014).

Pembangunan *FP-Tree* dimulai dari pembentukan sebuah akar yang diberi *label null*, selanjutnya terdiri dari sekumpulan pohon yang beranggotakan *item-item* tertentu. Pada setiap simpul *FP-Tree* memiliki beberapa informasi, yaitu *Label item* menjelaskan jenis *item* yang direpresentasikan oleh simpul tersebut, *Support count* merepresentasikan total lintasan transaksi yang melalui simpul tersebut dan *Pointer* merupakan penghubung yang menghubungkan simpul-simpul dengan label *item* sama pada tiap lintasan yang ditandai dengan garis panah putus-putus.

2.5.2 Penerapan *Frequent Pattern Growth (FP-growth)*

Setelah tahap pembangunan *FP-tree* dari sekumpulan data selesai maka langkah selanjutnya yaitu penerapan algoritma *FP-growth* yang bertujuan untuk mencari *frequent itemset*. Algoritma *FP-Growth* terdiri dari 3 langkah utama (Ririanti, 2014), yaitu :

1. Pembangkitan *Conditional Pattern Base*

Pada tahapan ini dilakukan pembangkitan *conditional pattern base*. *Conditional Pattern Base* adalah subdata yang memiliki lintasan *prefix* (*prefix path*) dan pola akhiran (*suffix pattern*). Pembangkitan ini diperoleh dari hasil pembangunan *FP-Tree*.

2. Pembangkitan *Conditional FP-Tree*

Pada tahapan ini dilakukan penjumlahan terhadap *support count* dari setiap *item* pada setiap *conditional pattern base*. Setiap *item* yang

jumlah *support count* lebih besar atau sam dengan *minimum support count* akan dibangkitkan dengan *conditional FP-Tree*.

3. Pencarian *Frequent item set*

Pada tahapan ini penentuan *conditional FP-Tree*. Jika *conditional FP-Tree* yang terbentuk adalah lintasan tunggal (*single path*) maka dapat melakukan kombinasi *item* untuk setiap *conditional FP-Tree* dengan tujuan untuk memperoleh *frequent item set*.

2.5.3 Lift rasio

Untuk mengevaluasi kuat tidaknya sebuah aturan asosiasi maka bisa menggunakan *lift rasio*. *Lift rasio* merupakan perbandingan antara *confidence* sebuah aturan dengan dengan nilai *benchmark confidence*. Sedangkan *Benchmark Confidence* merupakan perbandingan antara jumlah semua *item* yang menjadi *consequent* terhadap jumlah total transaksi.

Rumus untuk mencari *Benchmark Confidence* adalah

$$Benchmark\ Confidence = \frac{Nc}{N} \dots\dots\dots (2.4)$$

Keterangan :

Nc = Jumlah transaksi dengan *item* yang menjadi *consequent*.

N = Jumlah transaksi basis data

Sedangkan rumus untuk mencari *lift rasio* yaitu :

$$LR = \frac{Confidence\ (A,B)}{Benchamark\ Confidence\ (A,B)} \dots\dots\dots (2.5)$$

Jika nilai *lift rasio* lebih besar dari 1 maka menunjukkan *rule* atau aturan tersebut ada manfaat dan bisa digunakan. Lebih tinggi nilai *lift rasio* maka lebih besar kekuatan asosiasinya.

2.6 User Acceptance Test (UAT)

Pengujian yang dilakukan dengan *User Acceptance Test (UAT)* yaitu membagi kuesioner kepada *user* dan kemudian dihitung hasil pengujiannya. Pengujian ini dilakukan untuk melihat apakah sistem yang dibuat sesuai dengan yang diharapkan atau belum.

Menurut (Betha, 2006) yang dikutip melalui (Mutiara dkk, 2014), *User Acceptance Test* adalah proses pengujian oleh *user* dan menghasilkan dokumen untuk dijadikan bukti bahwa aplikasi yang dikembangkan dapat diterima user dan hasil pengujiannya dianggap memenuhi kebutuhan pengguna. *User Acceptance Test* menggunakan angket atau kuisioner yang berisi pertanyaan seputar sistem yang telah dibangun yang disebarakan kepada responden untuk menemukan *feedback* dari *user*. Pertanyaan dalam angket berbentuk objektif, dimana para responden dapat memilih salah satu dari beberapa alternatif jawaban yang telah diberi bobot atau skor. Persamaan penilaian dengan skala *likert* sebagai berikut:

1. Menentukan jumlah kategori
2. Penentuan Total Skor

Untuk mendapatkan rangkuman hasil penilaian dapat dilakukan dengan rumus sebagai berikut :

$$T \times P_n \dots \dots \dots (2.6)$$

Keterangan

T = Frekuensi jawaban yang dipilih

Pn = Bobot skor likert

3. Interval mendapatkan interval dan interpretasi persen dari kategori digunakan rumus sebagai berikut:

$$Interval (I) = \frac{100\%}{Jumlah\ Kategori} \dots \dots \dots (2.7)$$

4. Interpretasi Skor Perhitungan

Untuk mendapatkan skor perhitungan interpretasi harus diketahui skor tertinggi dan skor terendah dengan rumus sebagai berikut :

X = Skor Terendah Likert x Jumlah Pertanyaan

Y = Skor Tertinggi Likert x Jumlah Pertanyaan

Selanjutnya ditentukan indeks persen untuk mengetahui hasil akhir dengan rumus sebagai berikut :

$$Indeks \% \frac{Total\ Skor}{Y} \times 100 \dots \dots \dots (2.8)$$

2.7 Penelitian Terkait

Berikut ini adalah beberapa hasil yang telah diperoleh dari penelitian terkait, mengenai hasil dari penggunaan metode agar memberikan hasil yang maksimal dalam penelitian selanjutnya sebagai berikut:

Tabel 2.2 Penelitian Terkait

| NO | PENULIS | JUDUL | TAHUN | KESIMPULAN |
|----|---------------------|---|-------|--|
| 1 | M. Ridwan Effendi | Prediksi Pola Kecelakaan Lalu Lintas Menggunakan Metode Analisa <i>Asosiatif</i> | 2017 | Hasil dari penelitian ini mengetahui jenis merek kendaraan yang sering mengalami kecelakaan yaitu suzuki di susul yamaha dan kawasaki |
| 2 | Rizky Mei Anggreani | Perbandingan Algoritma <i>Apriori</i> Dan Algoritma <i>FP-Growth</i> Untuk Rekomendasi Pada Transaksi Peminjaman Buku Di Perpustakaan Universitas Dian Nuswantoro | 2016 | hasil kinerja kedua algoritma semakin besar minimum support nya maka itemsets dan <i>Association rule</i> yang terbentuk tidak ada, maka dari itu dicari <i>minimum support</i> yang paling mendekati agar <i>frequent itemsets</i> dan <i>Association rule</i> yang terbentuk sempurna atau mendekati sempurna. Maka dapat disimpulkan bahwa kedua algoritma tersebut layak diimplementasikan |
| 3 | Wardiman alimuddin | Analisa Tingkat Kecelakaan Lalu Lintas Dengan Metode <i>Association Rule</i> Menggunakan Algoritma <i>Apriori</i> | 2016 | Tingkat kecelakaan lalu lintas terbesar yaitu terjadi pada tingkat ringan sebesar 638 kali atau sebesar 68% dari total kecelakaan lalu lintas |
| 4 | Yuyun Dwi Lestari | Penerapan Data Mining Menggunakan Algoritma <i>FP-Tree</i> Dan <i>FP-Growth</i> | 2015 | Hasil pengolahan data penjualan obat paling banyak terjual adalah <i>fungoral, alopurinol,</i> |

| NO | PENULIS | JUDUL | TAHUN | KESIMPULAN |
|-----------|----------------|---|--------------|--|
| | | Pada Data Transaksi Penjualan Obat | | <i>diatizem HCL, batugin dan ketoconazole</i> |
| 5 | Ririanti | Implementasi Algoritma <i>FP- Growth</i> Pada Aplikasi Prediksi Persediaan Sepeda Motor | 2014 | Hasil akhir penelitian ini diperoleh jenis sepeda motor mana yang lebih banyak terjual. |

