

BAB II

LANDASAN TEORI

2.1 *Mobile Legends*

Mobile Legend merupakan *game mobile* bergenre MOBA (*Multiplayer Online Battle Arena*) yang dimainkan secara tim dengan *gameplay* 5 vs 5. Dalam permainannya dilakukan 5 orang pemain dalam satu tim sehingga membuat *game* ini menjadi menarik untuk dimainkan oleh siapapun. *Mobile Legends* disebut-sebut sebagai *game mobile* paling populer sepanjang 2017 (Informasi tersebut terungkap dari laporan retrospektif App Annie untuk periode 2017), Selain itu *game* ini dimainkan di *smartphone* bukan *game* MOBA (*Massive Online Battle Arena*) yang dimainkan di *computer* sehingga orang-orang dapat dengan mudah memainkan *game* ini di manapun, dan juga ini berbasis *online* sehingga kita dapat bermain dengan siapapun. (Agung mustofa, 2019)

2.2 E-Sport

Olahraga elektronik (juga dikenal sebagai permainan kompetitif, permainan pro, *Esports*, *e-sports*, *electronic sports*, atau *pro gaming* di Korea Selatan) merupakan suatu istilah untuk kompetisi Permainan video pemain jamak, umumnya antara para pemain profesional. *Esports* secara umum merupakan singkatan dari *electronic sports*, *electronic* yang dimaksud adalah *electronic device*, merupakan sebuah alat bantu *electronic* pada setiap kegiatan *esports* yang dan Sport adalah suatu kegiatan yang tidak terbatas hanya kegiatan fisik, melainkan mental pikiran dan dilakukan antar individu maupun kelompok.

2.3 Data Mining

Data Mining adalah teknik pengolahan data yang digunakan untuk mencari hubungan antara data yang tidak diketahui oleh pengguna dengan menyajikannya secara lebih mudah dipahami sehingga dapat menjadi dasar mengambil keputusan. Menurut Hermawati (2013) *Data Mining* adalah proses yang secara otomatis menggunakan satu atau lebih teknik pembelajaran mesin (*machine learning*) untuk menganalisis dan mengekstraksi pengetahuan. Suatu proses dalam melakukan pencarian informasi secara otomatis yang dapat bermanfaat dalam menyimpan data yang memiliki ukuran besar merupakan bentuk *data mining* sehingga menemukan pola yang baru dan berguna dalam membuat keputusan (Han et al, 2011).

2.3.1 Naïve Bayes

Algoritma *Naive Bayes* berbasiskan perhitungan probabilistik dengan asumsi bahwa setiap fitur yang digunakan saling lepas. (Olson dan Delen, 2008) Menyatakan *Naive Bayes* merupakan metode klasifikasi teks yang paling populer digunakan. Algoritma ini memiliki kelebihan dari sisi kecepatan pembelajaran dan toleransinya terhadap nilai yang hilang dari fitur. Untuk menangani data numerik, algoritma ini menggunakan *probability density function*, artinya data dianggap mengikuti distribusi normal untuk kemudian dihitung nilai rata-rata dan simpangan bakunya.

Untuk merepresentasikan sebuah kelas, terdapat karakteristik petunjuk yang dibutuhkan untuk melakukan klasifikasi yang berguna untuk menjelaskan bahwa peluang masuknya sampel karakteristik tertentu kedalam kelas *posterior*. Peluang munculnya suatu kelas (sebelum masuknya sampel tersebut, seringkali disebut *prior*), dikali dengan peluang kemunculan karakteristik sampel secara global disebut juga *evidence*. Nilai *evidence* selalu tetap untuk setiap kelas pada satu sampel. Nilai *posterior* tersebut dibandingkan dengan nilai *posterior* kelas lainnya untuk menentukan ke kelas apa suatu sampel .

Klasifikasi Naive Bayes diasumsikan bahwa ada atau tidak ciri tertentu dari sebuah kelas tidak ada hubungannya dengan ciri dari kelas lainnya. Persamaan dari teorema Bayes adalah sebagai berikut:

$$P(H|X) = \dots\dots\dots (2-1)$$

Dengan :

X = Data dengan kelas yang belum diketahui;

H = Hipotesis data X merupakan suatu label kelas tertentu;

$P(H|X)$ = Probabilistik hipotesis H berdasarkan kondisi X (*posteriori probability*); $P(H)$: Probabilistik hipotesis H (*prior probability*);

$P(X|H)$ = Probabilitias X berdasarkan kondisi pada hipotesis H;

$P(X)$ = Probabilistik X;

Untuk menjelaskan teorema *naive bayes*, perlu diketahui bahwa proses klasifikasi memerlukan sejumlah petunjuk untuk menentukan kelas apa yang cocok bagi sampel yang dianalisis tersebut. Karena itu, teorema *bayes* tersebut akan disesuaikan sebagai berikut:

$$P(C|F_1 \dots F_n) = \dots\dots\dots (2-2)$$

Dengan :

C = Sebuah kelas; F₁ ... F_N = Karakteristik Petunjuk.

$$\text{Posterior} = \dots\dots\dots (2-3)$$

Penjabaran lebih lanjut sebagai Berikut :

$$\begin{aligned} P(C|F_1, \dots, F_n) &= P(C) P(F_1, \dots, F_n | C) \\ &= P(C) P(F_1|C) \\ &= P(C) P(F_1|C) \\ &= P(C)P(F_1|C)P(F_2|C,F_1)P(F_3|C,F_1,F_2),P(F_4, \dots, F_n|C,F_1,F_2) \\ &= P(C)P(F_1|C)P(F_2|C,F_1)P(F_3|C,F_1,F_2), \dots, (F_n|C,F_1,F_2, \dots, F_{n-1}) \end{aligned}$$

Adapun langkah – langkah pada *naïve bayes* yang akan dilakukan menggunakan rumus di atas sebagai berikut (Suyanto,2017) :

1. Penentuan atribut yang akan digunakan .
2. Penentuan Data Set.
3. Lakukan Perhitungan Jumlah Kasus Pada Atribut .
4. Tentukan Kasus Baru yang akan di klasifikasikan menggunakan metode naive bayes.
5. Klasifikasikan Kasus baru berdasarkan kasus yang sama dengan kasus yang lama.
6. Kalikan semua Kelas Variable untuk mendapatkan nilai dari masing masing kelas.
7. Bandingkan Hasil perkalian dari masing masing kelas maka akan di ketahui konsentrasi skripsi intelegent system atau multimedia berdasarkan hasil nilai terbesar dari masing-masing kelas.

8. Lakukan Rekomendasi Bahasa Pemrograman berdasarkan hasil perbandingan antara konsentrasi skripsi intelegent system dan multimedia.

2.4 Android

Juhara, P.Z (2016: 01) Menyatakan bahwa dalam buku yang berjudul “Panduan Lengkap Pemrograman Android” Android adalah sistem operasi berbasis Linux yang dimodifikasi untuk perangkat bergerak (*mobile devices*) yang terdiri dari sistem operasi, *middleware*, dan aplikasi – aplikasi utama. Awalnya *android* dikembangkan oleh Android Inc. Perusahaan ini kemudian dibeli oleh *Google* pada tahun 2005. Sistem operasi Android kemudian diluncurkan bersamaan dengan dibentuknya organisasi *Open Handset Al-liance* tahun 2007. Selain *Google*, beberapa nama – nama besar juga ikut seta dalam *Open Handset Alliance*, antara lain *Motorola*, *Samsung*, *LG*, *Sony Ericsson*, *T- Mobile*, *Vodafone*, *Toshiba*, dan *Intel*.

2.5 Android Studio

Android Studio yang merupakan Lingkungan Pengembangan Terpadu - *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan *IntelliJ IDEA*. Selain merupakan editor kode *IntelliJ* dan alat pengembang yang berdaya guna, *Android Studio* menawarkan fitur lebih banyak untuk meningkatkan produktivitas saat membuat aplikasi Android, misalnya sistem versi berbasis *Gradle* yang fleksibel, emulator yang cepat dan kaya fitur, lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android, *instant Run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat *Android Package Kit (APK)* baru, template kode dan integrasi *GitHub* untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh, alat pengujian dan kerangka kerja yang ekstensif, alat *Lint* untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah masalah lain, dukungan C++ dan NDK, dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian *Google Cloud Messaging* dan *App Engine*.(Akmal,2019)

2.6 Metode Prototype

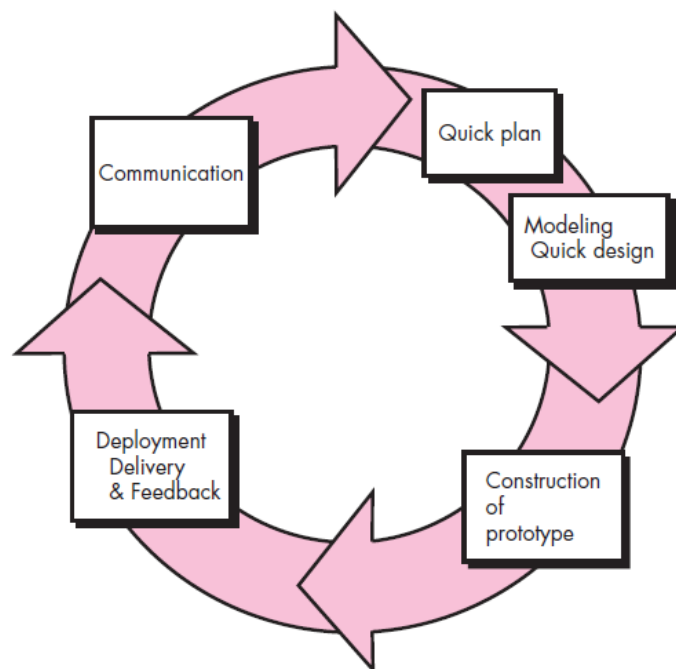
Pressman menguraikan bahwa dalam melakukan perancangan sistem yang akan dikembangkan dapat menggunakan metode *prototype*. Metode ini cocok digunakan untuk

mengembangkan sebuah perangkat lunak yang dikembangkan kembali. Metode ini dimulai dengan pengumpulan kebutuhan pengguna. Kemudian membuat sebuah rancangan kilat yang selanjutnya akan dievaluasi kembali sebelum di produksi secara benar.

Prototype bukanlah merupakan sesuatu yang lengkap, tetapi sesuatu yang harus dievaluasi dan dimodifikasi kembali. Segala perubahan dapat terjadi pada saat *prototype* dibuat untuk memenuhi kebutuhan pengguna dan saat yang sama memungkinkan pengembangan untuk lebih memahami kebutuhan pengguna secara baik.

Berikut adalah tahapan dalam metode *prototype* :

1. Komunikasi (*Communication*) dan pengumpulan data awal, yaitu komunikasi dengan klien dan user untuk menentukan kebutuhan.
2. Perencanaan cepat (*Quick Plan*), yaitu pembuatan perencanaan analisis terhadap kebutuhan pengguna.
3. Pemodelan perancangan cepat (*Modeling Quick Design*), yaitu membuat rancangan desain program.
4. Pembentukan *prototype* (*Construction of prototype*), yaitu pembuatan aplikasi berdasarkan dari pemodelan desain yang telah dibuat.
5. Penyerahan sistem dan umpan balik (*Development Delivery and Feedback*), yaitu memproduksi perangkat secara benar sehingga dapat digunakan oleh pengguna.



Gambar 2.1 Metode *Prototype* (Pressman, 2012)

2.7 Database

Database adalah sebuah struktur yang umumnya dikategorikan dalam hal Sebuah Basis *Data flat* dan sebuah Basis Data relasional. Basis data relasional lebih disukai karena lebih masuk akal dibandingkan Basis *data flat* Ada tabel-tabel yang menyimpan data. Setiap tabel terdiri dari kolom dan baris. Sebuah kolom mendefinisikan jenis informasi apa yang akan disimpan. Diperlukan kolom khusus untuk setiap jenis informasi yang ingin di simpan Abdul Kadir (Kadir, 2009).

Kalau kolom mendefinisikan jenis informasi apa yang akan disimpan, maka sebuah baris adalah data aktual yang disimpan. Setiap baris dari tabel adalah masukan dari tabel tersebut dan berisi nilai-nilai untuk setiap kolom tabel tersebut

2.7.1 SQLite

SQLite merupakan *database management system* yang populer untuk menyimpan lokal / *client* pada aplikasi perangkat lunak. *SQLite* merupakan *database engine* yang paling banyak digunakan di dunia untuk keperluan seperti *browser*, *system* operasi, dan aplikasi *mobile*. *SQLite* pada android menyimpan data internal aplikasi itu sendiri, aplikasi lain dapat menggunakannya (Safaat, 2012).

2.8 Black-Box Testing

Black Box Testing atau Pengujian Kotak Hitam atau juga disebut *Behavioral Testing*, berfokus pada persyaratan fungsional dari perangkat lunak. Artinya, teknik *Black-Box Testing* memungkinkan untuk mendapatkan set kondisi masukan yang sepenuhnya akan melaksanakan semua persyaratan fungsional untuk suatu program (Pressman, 2010).

Black-Box Testing bukan merupakan alternatif dari pengujian *White Box Testing*. Sebaliknya, *Black-Box Testing* adalah pendekatan komplementer yang mungkin untuk mengungkap kelas yang berbeda dari kesalahan daripada metode *White Box Testing*.

Black Box Testing mencoba untuk menemukan kesalahan dalam kategori berikut.

1. Fungsi tidak benar atau hilang.
2. Kesalahan interface atau antarmuka.

3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja atau perilaku.
5. Kesalahan inisialisasi dan terminasi

2.9 Unified Modeling Language (UML)


Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OOP). Definisi ini merupakan definisi yang sederhana. Pada kenyataannya, pendapat orang-orang tentang UML berbeda satu sama lain. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan persepsi tentang apa yang membuat sebuah proses rancang-bangun perangkat lunak efektif.






UML lahir dari penggabungan banyak bahasa permodelan grafis berorientasi objek yang berkembang pesat pada akhir 1980-an dan awal 1990-an. UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera *Rational Software Corp.* UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan (Nugroho, 2015).

2.9.1 Use case Diagram

Use case Diagram adalah diagram yang mendeskripsikan interaksi antara pengguna dengan aplikasi. Kesimpulannya *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem (Nugroho, 2015). Simbol dan keterangan *use case* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol dan keterangan Use case diagram


SIMBOL	KETERANGAN
Aktor 	Mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan <i>use case</i>




<i>Use case</i> 	Abstraksi dan interaksi antara sistem dan aktor
<i>Association</i> 	Abstraksi dari penghubung antara aktor dan <i>use case</i>
 Generalisasi	Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
 <<include>>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
<<extend>> 	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

2.9.2 Activity Diagram

Activity Diagram atau Diagram Aktivitas menggambarkan alur aktivitas dalam aplikasi, menjelaskan proses masing-masing alur berawal dan proses aplikasi berakhir. Diagram aktivitas juga menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi (Nugroho, 2015). Simbol dan keterangan *activity diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol dan keterangan activity diagram

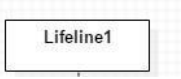



SIMBOL	KETERANGAN
Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki status awal.

<p>Aktivitas</p> 	<p>Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja</p>
<p>Percabangan/decision</p> 	<p>Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.</p>
<p>Status akhir</p> 	<p>Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir</p>

2.9.3 Sequence Diagram

Sequence Diagram biasanya digunakan untuk tujuan analisa dan desain, memfokuskan pada identifikasi metode di dalam sebuah sistem (Nugroho, 2015). Simbol dan keterangan *sequence diagram* dapat dilihat pada Tabel 2.3.





Tabel 2.3 Simbol dan keterangan *sequence diagram*

TABEL	KETERANGAN
<p>Objek</p> 	<p>Berpartisipasi secara berurutan dengan mengirimkan atau menerima pesan dan ditempatkan di bagian atas diagram</p>
<p>Garis Hidup Objek</p> 	<p>Menandakan kehidupan objek selama urutan dan diakhiri tanda X pada titik dimana kelas tidak lagi berinteraksi</p>
<p>Objek sedang aktif berinteraksi</p> 	<p>Persegi panjang yang sempit panjang ditempatkan diatas sebuah garis hidup dan menandakan ketika suatu objek mengirim atau menerima pesan</p>
<p>Message</p> 	<p>Perilaku sistem yang menandai adanya suatu alur informasi atau transisi kendali antar elemen</p>

2.9.4 Class Diagram

Class Diagram atau Diagram Kelas merupakan diagram yang memodelkan sekumpulan kelas, *interface*, kolaborasi, dan relasinya. Diagram kelas digambarkan dengan bentuk kotak (Nugroho, 2015). Simbol dan keterangan *class diagram* dapat dilihat pada Tabel 2.4.

Tabel 2.4 Simbol dan keterangan *class diagram*

SIMBOL	KETERANGAN
<i>CLASS</i> 	Himpunan dari objek-objek yang berbagai atribut serta operasi yang sama.
<i>Nary Association</i> 	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
<i>Generalization</i> 	Hubungan dimana objek anak (<i>descendent</i>) berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>oncestor</i>).
<i>Realization</i> 	Operasi yang benar-benar dilakukan suatu objek.

2.10 Penelitian Terdahulu

No	Nama Peneliti	Judul Penelitian	Hasil Penelitian	Perbedaan Penelitian
1	Alexander, Dharmawan	OPTIMALISASI SUSUNAN PEMAIN DAN PREDIKSI KEMENANGAN GAME MENGUNAKAN ALGORITMA NAÏVE BAYES (2022)	Berdasarkan perhitungan menggunakan algoritma naïve bayes maka dapat ditarik kesimpulan sebagai berikut : 1. Dengan parameter specialty, role, dan counter didapatkan lebih stabil dan akurat karena tidak akan berubah sepanjang pertandingan dibandingkan dengan parameter yang ditentukan oleh penelitian sebelumnya yaitu tingkat kesulitan, daya tahan hero, efek skill, dan output damage. 2. Hasil perhitungan aplikasi mencapai akurasi 80% dari data yang diperoleh yaitu 12 kemenangan dari total 15 pertandingan.	1. Penelitian masih Berbasis Web 2. Berfokus pengoptimalian dalam susunan Pemain / tim
2	Eduardus Hardika Sandy , Atmaja.	Prediksi Kemenangan eSport DOTA 2 Berdasarkan Data Pertandingan (2020)	Berdasarkan serangkaian kegiatan yang telah dilakukan maka dapat disimpulkan bahwa algoritma Naive Bayes dapat digunakan untuk melakukan prediksi dengan baik. Akurasi yang didapatkan sebesar 98,804 % dengan 3-fold cross validation dengan jumlah data 50000. Data yang digunakan dalam penelitian ini adalah data pertandingan saja, penelitian selanjutnya dapat mencoba mengkombinasikan antara data pertandingan dengan data lain	Objek Penelitian yang digunakan berbeda dengan penelitian ini.

			seperti player rating, item purchased, ability used dan lainnya sehingga dihasilkan informasi yang lebih detail.	
3	Listijo, Sri Murjani, Tri Purwani, and Sinta Tridian Galih	PREDIKSI KEMENANGAN DAN SUSUNAN TIM PADA GAME MOBILE LEGENDS BANG BANG MENGGUNAKAN ALGORITMA NAÏVE BAYES (2022)	Aplikasi yang dirancang berdasarkan kebutuhan para pemain Mobile Legends yang ingin menuju ke tingkat professional untuk memudahkan dalam pemilihan hero saat mode Draft Pick dan memastikan pada sesi pemilihan hero, hero yang dipilih memiliki nilai persentase kemenangan pada early game yang cukup untuk mengalahkan tim lawan dan mencapai kemenangan	Penelitian tersebut memfokuskan pada prediksi kemenangan susunan tim pada objek penelitian.
4	Khadijah, Khadijah, Nur Sabilly, and Fajar Agung Nugroho	ANALISIS SENTIMEN ULASAN LEAGUE OF LEGENDS: WILD RIFT DI GOOGLE PLAY MENGGUNAKAN NAÏVE BAYES CLASSIFIER(2023)	Pada penelitian ini, model terbaik dipilih untuk analisis sentimen terhadap League of Legends: Wild Rift di Google Play dengan menggunakan algoritma Naïve Bayes Classifier (NBC). Beberapa kesimpulan diperoleh dari penelitian tersebut. Pertama, proses transformasi kata tidak baku menjadi kata baku meningkatkan performa model algoritma NBC pada kedua ekstraksi fitur, yaitu BOW dan TF-IDF. Kedua, proses stemming pada dataset preprocessing secara umum menghasilkan kinerja klasifikasi algoritma NBC yang lebih rendah pada sebagian besar percobaan ketika menggunakan	Objek Penelitian yang digunakan berbeda dengan penelitian ini.

			fitur TF-IDF dan BOW. Dan ketiga, hasil pengujian menunjukkan bahwa ekstraksi fitur BOW menghasilkan performa model algoritma NBC yang lebih baik daripada ekstraksi fitur TF-IDF.	
5	MUSTHOFA, Agung; ROSYAD, Udung Noor	Efektivitas Event Premium Starlight Member Game Mobile Legend untuk Menarik Minat Beli Voucher Game (2019)	Bahwa dengan adanya event starlight member ini merangsang para responden atau masyarakat untuk memperhatikan dan melihat ada suatu hal yang menarik yang disajikan di dalam game mobile legend yaitu hadiah-hadiah yang menarik yang tidak bisa didapatkan oleh semua orang dan hanya bisa didapatkan oleh orang-orang yang menjadi starlight member.	Objek Penelitian yang digunakan berbeda dengan penelitian ini.
6	PRASETYA, M. Octa; EPENDI, Usman	Sentimen Analisis Hero MobileLegends Dengan Algoritma Naive Bayes(2022)	Berdasarkan hasil dari penelitian yang dilakukan oleh penulis, scrapped data berhasil mengumpulkan data yang berasal dari para pengguna game mobile legends di playstore yang berjumlah 56911 data dan diambil sebanyak 100 data yang akan dijadikan data sampel pengguna. Metode yang digunakan ini adalah Naive Bayes Classifier (NBC). Banyak sekali data komentar atau pendapat yang berhasil dikumpulkan menggunakan algoritma Naive Bayes Classifier ini. Naive Bayes Classifier	Objek Penelitian yang digunakan menganalisis data teks dan mengidentifikasi maksud teks tersebut. Sedangkan penelitian yang dilakukan melakukan prediksi dengan menggunakan metode naive bayes.

			<p>merupakan metode klasifikasi processing nya memiliki akurasi yang cukup bagus dan akurat. Pada penelitian selanjutnya penulis mengharapkan untuk menggunakan dua ataupun lebih metode yang di gunakan sebagai refrensi yang berguna untuk pembandingan metode manakah yang cocok untuk di gunakan serta lebih akurat dan lebih baik untuk melakukan prosesing data dari sentimen analisis selanjutnya.</p>	
--	--	--	---	--