BAB 2
TINJAUAN PUSTAKA

2.1. Landasan Teori

2.1.1. Algoritma

Menurut Skiena (2008), Algoritma adalah sebuah prosedur untuk menyelesaikan suatu pekerjaan atau masalah. Untuk menjadi sesuatu yang menarik, algoritma harus dapat menyelesaikan suatu masalah yang umum dan di spesifikasi dengan baik.

Algoritma berisi langkah-langkah penyelesaian suatu masalah. Langkah-langkah tersebut dapat berupa runtunan aksi, pemilihan aksi, dan pengulangan aksi. Ketiga jenis langkah tersebut membentuk konstruksi suatu algoritma. Jadi, sebuah algoritma dapat dibangun dari tiga buah struktur dasar, yaitu :

1. Runtunan (*Sequence*)

   Sebuah runtunan terdiri dari satu atau lebih instruksi. Tiap instruksi dikerjakan secara berurutan sesuai dengan urutan penulisannya, yakni sebuah instruksi dilaksanakan *set*elah instruksi sebelumnya selesai dikerjakan.

2. Pemilihan (*Selection*)

   Adakalanya sebuah instruksi dikerjakan jika kondisi tertentu dipenuhi. Tiap–tiap instruksi akan diseleksi oleh kondisi, apabila instruksi memenuhi kondisi yang diminta, maka instruksi akan dijalankan.

3. Pengulangan (*Repetition*)

   Salah satu kelebihan komputer adalah kemampuannya untuk mengerjakan pekerjaan yang sama berulang kali tanpa mengenal lelah. Kita tidak perlu menulis instruksi yang sama berulang kali, tetapi cukup melakukan pengulangan dengan instruksi yang tersedia.

2.1.2. Artificial Intelligence

Menurut Nilsson (2009), *artificial intelligence* adalah semua yang ditujukan untuk membuat sebuah mesin yang cerdas dimana kecerdasan yang dimaksud adalah yang memungkinkan suatu entitas dapat berfungsi secara tepat dengan pandangan terdepan di dalam lingkungannya.

Perbandingan Kecerdasan Buatan dan Kecerdasan Manusia menurut Kaplan (Turban, McLean, & Wetherbe, 1999), AI mempunyai beberapa kelebihan dibandingkan dengan kecerdasan alami/manusia. Adapun kelebihan AI sebagai berikut:

- AI bersifat Permanen, sepanjang sistem dan program masih terpelihara, sedangkan kecerdasan alami seseorang tidak dapat disimpan.
- AI menawarkan kemudahan untuk digandakan dan disebarkan.
- Pengetahuan dalam AI dapat lebih murah dibanding kecerdasan alami/manusia, biaya membeli jasa dengan komputer lebih murah dibanding membiayai manusia dengan tugas yang sama.
- AI bersifat konsisten dan teliti, sedangkan manusia tidak konsisten.
- AI dapat didokumentasikan.

2.1.3. *Natural language processing* (NLP)

Natural Language adalah bahasa-bahasa yang biasa digunakan oleh manusia. Untuk dapat memproses bahasa-bahasa tersebut ke dalam komputer maka terbentuklah sebuah ilmu yang dinamakan *Natural language processing*. Menurut Chopra et al. (2013), *Natural language processing* adalah bagian dari ilmu *artificial intelligence* dan linguistik, dibentuk untuk membuat komputer dapat mengerti pernyataan atau kata-kata yang ditulis dalam bahasa manusia.

NLP terdiri dari beberapa bagian utama yaitu:

- *Parser*, suatu sistem untuk mengambil kalimat input, dan menguraikannya kata per kata serta untuk menentukan jenis kata apa saja yang dapat mengikuti kata tersebut.

(Menguraikan kalimat ke dalam beberapa bagian gramatikal seperti subjek, objek, kata kerja, kata benda, kata sifat dan sebagainya).

- Sistem Representasi Pengetahuan, suatu sistem untuk menganalisa hasil dari *parser* dan menentukan maknanya.
- *Output Translator*, suatu hasil terjemahan yang merepresentasikan sistem pengetahuan serta melakukan langkah-langkah yang berupa jawaban atas bahasa alami. *Output translator* merupakan *output* khusus yang sesuai dengan program komputer lainnya.

Pustejovsky dan Stubbs (2012) menjelaskan bahwa ada beberapa area utama penelitian pada field NLP, diantaranya:

1. *Question Answering Systems* (QAS)

   Kemampuan komputer untuk menjawab pertanyaan yang diberikan oleh *user*. Daripada memasukkan *keyword* ke dalam *browser* pencarian, dengan QAS, *user* bisa langsung bertanya dalam bahasa natural yang digunakannya, baik itu Inggris, Mandarin, ataupun Indonesia.

2. *Summarization*

   Pembuatan ringkasan dari sekumpulan konten dokumen atau email. Dengan menggunakan aplikasi ini, *user* bisa dibantu untuk mengkonversikan dokumen teks yang besar ke dalam bentuk slide presentasi

3. *Machine Translation*

   Produk yang dihasilkan adalah aplikasi yang dapat memahami bahasa manusia dan menerjemahkannya ke dalam bahasa lain. Salah satu contoh yang dapat kita lihat adalah Google Translate yang telah menjadi alat penerjemah di dalam kehidupan sehari-hari jutaan orang di berbagai belahan dunia.

4. *Speech Recognition*

   Field ini merupakan cabang ilmu NLP yang cukup sulit. Namun saat ini telah banyak dikembangkan *speech*

recognition untuk digunakan di *smartphone* atau komputer dalam mengenali bahasa yang diucapkan, yang biasanya berupa pernyataan dan perintah.

5. *Document Classification*

   Field ini dapat dikatakan sebagai area penelitian NLP yang paling sukses dan telah memberi kemudahan bagi kita untuk mengklasifikasikan dokumen berdasarkan kategori yang ada. *Document classification* sangat berguna digunakan pada aplikasi seperti email untuk menyaring spam, klasifikasi kategori artikel atau berita, dan ulasan film.

2.1.4. Machine Learning

2.1.4.1. Pengertian *Machine Learning*

Machine learning adalah cabang dari ilmu kecerdasan buatan (*artificial intelligence*) yang berfokus pada pembangunan dan studi sebuah sistem agar mampu belajar dari data-data yang diperolehnya. Menurut Arthur Samuel, *Machine learning* adalah bidang studi yang memberikan kemampuan program komputer untuk belajar tanpa secara eksplisit di program.

Machine learning sebagai sebuah solusi dapat digunakan jika sebuah kasus masalah memenuhi tiga hal sebagai berikut

1. A *Pattern* exist (terdapat sebuah pola). Jika data yang ada bersifat acak. Maka *machine learning* tidak akan dapat menyelesaikan permasalahan tersebut.
2. Persoalan tersebut tidak dapat diselesaikan secara matematis. *Machine learning* menggunakan teori ilmu statistic (probabilitas dan pendekatan). Sehingga ketika sebuah peristiwa tidak dapat diselesaikan dengan pendekatan perhitungan matematik biasa, maka *Machine learning*-lah solusinya.
3. Terdapat data. Sebagaimana namanya *machine learning* adalah sebuah proses dimana sebuah aplikasi sanggup belajar dari data yang diberikan. Jika tidak terdapat data, maka *machine learning* tidak akan bisa menyelesaikan masalah tersebut.

Teknik dari *machine learning* dibagi menjadi tiga kategori yaitu *supervised learning*, *unsupervised learning* dan *reinforcement learning* (Russell & Norvig, 2010).

Supervised learning merupakan suatu teknik yang diggunakan untuk mendapatkan beberapa metode yang akan digunakan sebagai *classifier* untuk memprediksi input yang diberikan. Beberapa contoh dari algoritma *supervised learning* adalah Naïve Bayes, SVM (*Support Vector Machine*), dan *Neural Network*s.

Unsupervised learning merupakan teknik yang sama dengan *supervised learning*. Sama halnya dengan *supervised*, *unsupervised learning* juga berfungsi sebagai metode *classifier* untuk memprediksi input yang diberikan. Tetapi perbedaannya terletak pada metode evaluasi dimana pada *unsupervised learning* tidak dilakukan perbandingan dengan data manual, melainkan model yang merupakan hasil prediksi dengan masing-masing data.

Reinforcement learning merupakan metode pembelajaran terhadap apa yang akan dilakukan atau bagaimana memetakan situasi terhadap aksi yang akan dilakukan oleh sistem untuk mendapatkan *reward* yang maksimal. Metode ini berbeda dengan 2 metode lainnya dimana metode ini tidak membutuhkan skenario dari aksi yang akan dilakukan, tetapi lebih pada menentukan aksi yang dapat memberikan *reward* yang maksimal.

2.1.4.2. Algoritma *Machine Learning*

2.1.4.2.1. *Support Vector Machine*

Support Vector Machine pertama kali diperkenalkan oleh Cortes & Vapnik (1995), kemudian dikembangkan oleh banyak peneliti hingga saat ini. Support Vector Machine digunakan untuk mengklasifikasikan data yang berada dalam suatu dataset struktural dan untuk mendiskriminasikan antara kelas berbeda dari sampel yang ada (Ben-Hur et al., 2013). Support Vector Machine (SVM) dikenal sebagai teknik pembelajaran mesin (machine learning) paling mutakhir

setelah pembelajaran mesin sebelumnya yang dikenal sebagai Neural Network (NN). Baik SVM maupun NN tersebut telah berhasil digunakan dalam pengenalan pola. Pembelajaran dilakukan dengan menggunakan pasangan data input dan data output berupa sasaran yang diinginkan. Pembelajaran dengan cara ini disebut dengan pembelajaran terarah (supervised learning). Dengan pembelajaran terarah ini akan diperoleh fungsi yang menggambarkan bentuk ketergantungan input dan outputnya. Selanjutnya, diharapkan fungsi yang diperoleh mempunyai kemampuan generalisasi yang baik, dalam arti bahwa fungsi tersebut dapat digunakan untuk data input di luar data pembelajaran (Kerami & Murfi, 2004).

Kelebihan dalam memilih solusi untuk menyelesaikan suatu masalah, kelebihan dan kelemahan masing-masing metode harus diperhatikan. Selanjutnya metode yang tepat dipilih dengan memperhatikan karakteristik data yang diolah. Dalam hal SVM, walaupun berbagai studi telah menunjukkan kelebihan metode SVM dibandingkan metode konvensional lain, SVM juga memiliki berbagai kelemahan. Kelebihan SVM antara lain sebagai berikut:

1. Generalisasi

Generalisasi didefinisikan sebagai kemampuan suatu metode (SVM, *neural network*, dsb.) untuk mengklasifikasikan suatu *pattern*, yang tidak termasuk data yang dipakai dalam fase pembelajaran metode itu. Vapnik menjelaskan bahwa generalization *error* dipengaruhi oleh dua faktor: *error* terhadap *training set*, dan satu faktor lagi yang dipengaruhi oleh dimensi VC (Vapnik-Chervonenkis). Strategi pembelajaran pada *neural network* dan umumnya metode *learning machine* difokuskan pada usaha untuk meminimimalkan *error*

pada *training-set*. Strategi ini disebut *Empirical risk minimization* (ERM).

Adapun SVM selain meminimalkan *error* pada *training-set*, juga meminimalkan faktor kedua. Strategi ini disebut *Structural Risk minimization* (SRM), dan dalam SVM diwujudkan dengan memilih *hyperplane* dengan margin terbesar. Berbagai studi empiris menunjukkan bahwa pendekatan SRM pada SVM memberikan *error* generalisasi yang lebih kecil daripada yang diperoleh dari strategi ERM pada *neural network* maupun metode yang lain.

2. *Curse of dimensionality*

Curse of dimensionality didefinisikan sebagai masalah yang dihadapi suatu metode *pattern recognition* dalam mengestimasikan parameter (misalnya jumlah *hidden* neuron pada *neural network*, *stopping criteria* dalam proses pembelajaran dsb.) dikarenakan jumlah sampel data yang relatif sedikit dibandingkan dimensional ruang vektor data tersebut. Semakin tinggi dimensi dari ruang vektor informasi yang diolah, membawa konsekuensi dibutuhkannya jumlah data dalam proses pembelajaran. Pada kenyataannya seringkali terjadi, data yang diolah berjumlah terbatas, dan untuk mengumpulkan data yang lebih banyak tidak mungkin dilakukan karena kendala biaya dan kesulitan teknis. Dalam kondisi tersebut, jika metode itu "terpaksa" harus bekerja pada data yang berjumlah relatif sedikit dibandingkan dimensinya, akan membuat proses estimasi parameter metode menjadi sangat sulit.

3. *Feasibility*

SVM dapat diimplementasikan relatif mudah, karena proses penentuan *support vector* dapat dirumuskan dalam QP *problem*. Dengan demikian jika kita memiliki

library untuk menyelesaikan QP *problem*, dengan sendirinya SVM dapat diimplementasikan dengan mudah. Selain itu dapat diselesaikan dengan metode sekuensial sebagaimana penjelasan sebelumnya.

Dari banyaknya kelebihan diatas SVM juga mempunyai banyak kekurangan antara lain

1. Sulit dipakai dalam *problem* berskala besar. Skala besar dalam hal ini dimaksudkan dengan jumlah sample yang diolah.
2. SVM secara teoritik dikembangkan untuk *problem* klasifikasi dengan dua class. Dewasa ini SVM telah dimodifikasi agar dapat menyelesaikan masalah dengan class lebih dari dua, antara lain strategi *One versus rest* dan strategi *Tree Structure*.

2.1.4.2.2. Naïve Bayes

Naïve Bayes *Classifier* merupakan sebuah metoda klasifikasi yang berakar pada teorema Bayes. Metode pengklasifikasian dengan menggunakan metoda probabilitas dan statistic yang dikemukakan oleh ilmuan Inggris Thomas Bayes, yaitu memprediksi peluang masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema Bayes. Ciri utama dari Naïve Bayes *classifier* ini adalah asumsi yang sangat kuat (naïf) akan indepensi dari masing-masing kondisi/kejadian.

Menurut Olson & Delen (2008) menjelaskan Naïve Bayes untuk setiap kelas keputusan, menghitung probabilitas dengan syarat bahwa kelas keputusan adalah benar, mengingat *vector* informasi obyek. Algoritma ini mengasumsikan bahwa atribut obyek adalah independen. Probabilitas yang terlihat dalam memproduksi perkiraan akhir dihitung sebagai jumlah frekuensi dari "*master*" tabel keputusan.

Naïve Bayes *Classifier* bekerja sangat baik dibandingkan dengan model *classifier* lainnya. Hal ini dibuktikan oleh Xhemali, Hinde Stone dalam jurnalnya "Naïve Bayes vs *Decision Trees* vs *Neural Network*s in the Classification *of Training Web* Pages" (Xhemali, 2009) yang mengatakan bahwa "Naïve Bayes *Classifier* memiliki tingkat akurasi yang lebih baik dibanding model *classifier* lainnya".

Keuntungan penggunakan metode Naïve Bayes *Classifier* adalah bahwa metoda ini hanya membutuhkan jumlah data pelatihan (*training* data) yang kecil untuk menentukan parameter yang diperlukan dalam proses pengklasifikasian. Karena yang diasumsikan sebagai variabel independen, maka hanya varians dari suatu varibel dalam sebuah kelas yang dibutuhkan untuk menentukan klasifikasi, bukan keseluruhan dari matriks kovarian.

2.1.4.2.3. Logistic Regression

Regresi logistik adalah sebuah pendekatan untuk membuat model prediksi seperti halnya regresi linear atau yang biasa disebut dengan istilah *Ordinary Least Squares (OLS) regression*. Perbedaannya adalah pada regresi logistik, peneliti memprediksi variabel terikat yang berskala dikotomi. Skala dikotomi yang dimaksud adalah skala data nominal dengan dua kategori, misalnya: Ya dan Tidak, Baik dan Buruk atau Tinggi dan Rendah.

Analisis regresi logistik digunakan untuk menjelaskan hubungan antara variabel respon yang berupa data dikotomik/biner dengan variabel bebas yang berupa data berskala interval dan atau kategorik (Hosmer, Jovanovic, dan Lemeshow, 1989). Variabel yang dikotomik/biner adalah variabel yang hanya mempunyai dua kategori saja, yaitu kategori yang menyatakan kejadian sukses (Y=1) dan kategori yang

menyatakan kejadian gagal (Y=0). pada model model linear umum komponen acak tidak harus mengikuti sebaran normal, tapi harus masuk dalam sebaran keluarga eksponensial. Sebaran bernoulli termasuk dalam salah satu dari sebaran keluarga eksponensial. Variabel respon Y ini, diasumsikan mengikuti distribusi Bernoulli.

Metode untuk mengestimasi parameter-parameter yang tidak diketahui dalam model regresi logistik ada 3 yaitu:

1. Metode kemungkinan maksimum (*Maximum Likelihood Method*).
2. Metode kuadrat terkecil tertimbang noniterasi (*Noniterative Weight Least Square Method*).
3. Analisis fungsi diskriminan (*Discriminant Fuction Analysis*).

Pada dasarnya metode maksimum Likelihood merupakan metode kuadrat terkecil tertimbang dengan beberapa proses iterasi, sedangkan metode *noniterative weight least square method* hanya menggunakan satu kali iterasi. kedua metode ini *asymptoticaly equivalent*, artinya jika ukuran sampel besar keduanya akan menghasilkan estimator yang identik. Penggunaan fungsi diskriminan mensyaratkan variabel penjelas yang kuantitatif berdistribusi normal. Oleh karena itu, penduga dari fungsi diskriminan akan *overestimate* bila variabel penjelas tidak berdistribusi normal.

Dari Ketiga metode di atas, metode yang banyak digunakan adalah metode *maximum likelihood* dengan alasan lebih praktis (Nachrowi dan Usman, 2002). Metode *maximum likelihoood* ini menduga parameter dengan nilai yang memaksimalkan fungsi likelihood (*likelihood function*).

2.1.4.2.4. Gradient Boosting

Gradient Boosting merupakan sebuah teknik *machine learning* untuk regresi dan klasifikasi permasalahan, sehingga menghasilkan sebuah prediksi model dalam bentuk *ensemble* dari model prediksi yang lemah. *Gradient Boosting* menggunakan *tree* sebagai dasar dari *learning algorithm*. *Gradient Boosting* diterapkan pada beberapa *loss function* yang populer seperti *least-squares* (LS), *least absolute deviation* (LAD), Huber, dan *Logistic binomial log-likelihood*. *Gradient Boosting* dari *regression trees* menghasilkan regresi dan klasifikasi yang kompetitif dan kuat (Friedman, 2001). Hal tersebut juga dibuktikan oleh Taieb & Hyndman (2013) bahwa dalam penelitiannya membuktikan bahwa *Gradient Boosting* dapat menjadi algoritma prediksi yang efektif untuk permasalahan klasifikasi dan regresi.

2.1.4.2.5. LDA (*Linear Discriminant Analysis*)

LDA terkenal dengan skema untuk *feature extraction* dan *dimension reduction* (Duda, Hart, & Stork, 2000). LDA telah banyak digunakan diberbagai aplikasi seperti *face recognition* (Belhumeur, Hespanha, & Kriegman, 1997), *image retrieval* (Swets & Weng, 1996), *microarray data classification* (Dudoit, Fridlyand, & Speed, 2002). LDA klasik mentransformasi data ke dalam *lower-dimensional vector space* sehingga rasio jarak *between-class* ke jarak *within-class* dimaksimalkan, sehingga mencapai *discriminant* maksimal (Ye, Janardan, & Li, 2005). Dalam beberapa tahun terakhir, banyak peneliti yang telah melakukan kemajuan terhadap LDA seperti *probabilistic* LDA (Shafey et al., 2013), *sparse* LDA (Zhang et al., 2014), dan Stepwise LDA (Siddiqi et al., 2015).

2.1.4.3. Library Machine Learning

2.1.4.3.1. Scikit-learn

Scikit-learn merupakan sebuah modul Python yang mengintegrasi berbagai algoritma *machine learning* dalam skala menengah baik untuk *supervised learning* maupun *unsupervised learning.* (Pedregosa et al., 2011). *Scikit-learn* juga merupakan *project open-source* yang mempunyai ambisi untuk menyediakan *machine learning tools* yang efisiensi dan stabil kepada *non-machine learning experts* dan dapat digunakan dalam berbagai bidang ilmiah lainnya. (Buitinck et al., 2013).

Scikit-learn menyediakan seluruh tahapan yang dibutuhkan di dalam *machine learning* seperti *preprocessing, feature selection, learning algorithms,* dan *model evaluation.*

2.1.4.3.2. NumPy

NumPy adalah sebuah package untuk komputasi ilmiah yang ditulis dalam bahasa Python. Selain kegunaan ilmiah, NumPy juga dapat digunakan sebagai wadah *multi-dimensional data* yang efisien. NumPy dilisensi dengan lisensi BSD, memungkinkan penggunaan kembali dengan beberapa batasan (NumPy, n.d.).

2.1.4.3.3. Pandas

Pandas adalah *library open-source* Python yang menyediakan kinerja tinggi dan mudah untuk digunakan untuk keperluan struktur data dan data analisis untuk bahasa pemrograman Python dan dilisensi dengan lisensi BSD (Pandas, n.d.).

2.1.4.3.4. Imbalanced-learn

Imbalanced-learn adalah sebuah *library open-source* yang ditulis dalam bahasa pemrograman Python

yang menyediakan berbagai macam metode untuk menyelesaikan *imbalanced dataset* yang sering dijumpai pada *machine learning* dan *pattern recognition* (Lemaitre, Nogueira, & Aridas, 2017).

2.1.4.3.5. NLTK (*Natural Language Toolkit*)

NLTK adalah kumpulan dari serangkaian *program modules, datasets,* dan *tutorials* yang mendukung penelitian dan pengajaran dalam komputasi linguistik dan *natural language processing*. NLTK ditulis dalam bahasa pemrograman Python dan dilisensi dengan lisensi *open-source* GPL (Bird, 2006).

2.1.5. Deep learning

2.1.5.1. Pengertian *Deep learning*

Deep learning adalah sebuah kelas dari teknik machine learning yang menggunakan banyak lapisan proses informasi non-linear untuk supervised atau unsupervised feature extraction dan transformation serta pattern analysis dan classification (Deng L, Yu D, 2014). Teknologi machine learning yang ada saat ini telah digunakan untuk banyak aspek dalam kehidupan modern di masyarakat saat ini, mulai dari pencarian web hingga penyaringan konten di jejaring sosial, bahkan juga digunakan pada rekomendasi di situs-situs e-commerce. Teteapi teknik *machine learning* konvensional ini terbatas pada kemampuan dalam mengolah data dalam bentuk mentahnya. Selama beberapa dekade, membangun sebuah pattern-*recognition* atau sistem machine learning membutuhkan teknik yang teliti dan kemampuan ahli yang tinggi untuk merancang fitur ekstraktor yang mengubah data mentah menjadi sebuah representasi internal atau fitur vektor yang sesuai. Representation learning adalah sebuah metode yang memungkinkan mesin untuk mendapatkan data mentah dan secara otomatis menemukan representasi yang dibutuhkan untuk pendeteksian atau klasifikasi (LeCun, Bengio & Hinton, 2015).

Metode *deep learning* ini adalah *representation learning* dengan berbagai tingkat representasi, diperoleh dengan menyusun modul sederhana namun dalam bentuk non-linear yang masing-masingnya dapat mengubah representasi dari satu tingkat menjadi tingkat yang lebih tinggi dan lebih abstrak.

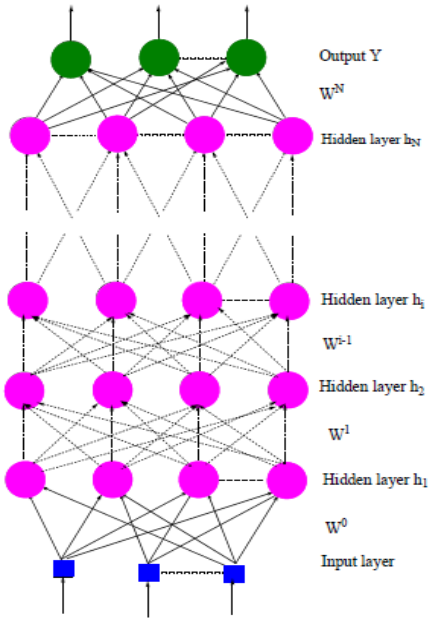
Deep learning membuat kemajuan besar dalam memecahkan masalah yang telah menghalangi usaha terbaik dari komunitas AI selama bertahun-tahun. *Deep learning* akan memiliki banyak kesuksesan di masa depan karena hanya membutuhkan sedikit teknik, sehingga dapat dengan mudah meningkatkan jumlah komputasi dan data yang dapat dilakukan (LeCun, Bengio & Hinton, 2015). Algoritma baru dan arsitektur *deep learning* yang baru hanya akan mempercepat perkembangan *deep neural network* ini.

2.1.5.2. Arsitektur *Deep learning*

2.1.5.2.1. MLP (*MultiLayer Perceptron*)

MLP adalah sebuah model dari cabang *artificial neural network* yang memetakan kumpulan data-data *input* menjadi sebuah *output*. *Multilayer Perceptron* adalah model yang paling banyak digunakan dalam aplikasi *neural network* yang menggunakan algoritma *training back-propagation*. Definisi arsitektur dalam jaringan MLP adalah sebuah titik yang sangat relevan, karena hanya dengan kurangnya koneksi dapat membuat sebuah model jaringan tidak mampu menyelesaikan sebuah masalah akibat parameter yang tidak mencukupi, di sisi lain jaringan yang terlalu banyak dapat menyebabkan terjadinya *over-fitting* pada data *training* (Ramchoun et al., 2016). Terutama ketika memakai lapisan dan neuron dalam jumlah yang besar. Mengoptimalkan jumlah koneksi dan *hidden layer* untuk membuat sebuah model MLP dapat memecahkan masalah masih menjadi sebuah tantangan

yang harus dilakukan. Jumlah lapisan-lapisan dari model bergantung terhadap masalah (Egrioglu et al., 2008).



Gambar 2.1 Struktur jaringan MLP atau *Feedforward network* (Sumber: Ramchoun et al., 2016, p.27).

MLP biasa disebut juga dengan jaringan *Feedforward* karena sifatnya yang membawa informasi dari lapis masukan (*input layer*) untuk dibawa dan ditransformasi ke depan hingga lapis luaran (*output layer*). MLP adalah bentuk dari model *perceptron* asli yang diciptakan oleh Rosenblatt di sekitar tahun 1950 (Rosenblatt, 1958). Model ini memiliki satu atau lebih *hidden layer* di antara lapisan input dan outputnya, neuron diorganisir di lapisan tersebut, jaringan selalu dimulai dari lapisan bawah ke lapisan atas, neuron pada lapisan yang sama tidak saling berhubungan seperti terlihat pada Gambar 2.1. Jumlah neuron pada lapisan input sama dengan jumlah pengukuran untuk pola masalah dan jumlah neuron pada lapisan output sama dengan jumlah class. Untuk jumlah lapisan pilihan dan neuron di masing-masing lapisan dan koneksi disebut dengan masalah arsitektur. Hal yang harus dilakukan adalah mengoptimalkan model tersebut dengan jaringan yang tepat dengan parameter secukupnya dan

generalisasi yang baik untuk klasifikasi atau regresi (Ramchoun et al., 2016).

2.1.5.2.2. CNN (*Convolutional Neural Network*)

CNN pertama kali dikembangkan oleh Kunihiko Fukushima dengan nama NeoCognitron, seorang peneliti dari NHK *Broadcasting Science Research Laboratories.* (Fukushima, 1980). Pada awalnya CNN dirancang untuk *Computer Vision*, ternyata model CNN terbukti efektif untuk *Natural language processing* (NLP) dan telah mencapai hasil yang sangat baik dalam *semantic parsing* (Grefenstette et al., 2014), *search query retrieval* (Shen et al., 2014), *sentence modeling* (Kalchbrenner et al., 2014) dan kasus NLP lainnya (Collobert et al., 2011).

Sebuah CNN terdiri dari beberapa *layer.* Berdasarkan LeNet5 (Scarlet.Stanford, 2013) CNN terdiri dari 3 *layer* antara lain:

1. *Convolution Layer*

   Convolution Layer mengoperasi konvolusi pada output dari *layer* sebelumnya. *Layer* tersebut adalah proses utama yang mendasari sebuah CNN. Konvolusi adalah suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada output fungsi lain secara berulang. Dalam pengolahan data, konvolusi berarti mengaplikasikan sebuah *kernel* pada data disemua offset yang memungkinkan. Tujuan dilakukan konvolusi pada data adalah untuk mengekstraksi fitur dari *input.* Konvolusi akan menghasilkan transformasi *linear* dari data *input* sesuai informasi spasial pada data. Bobot pada *layer* tersebut menspesifikasikan *kernel* konvolusi yang digunakan, sehingga *kernel* konvolusi dapat dilatihan berdasarkan *input* pada CNN.

2. *Subsampling Layer*

Subsampling adalah proses mereduksi ukuran sebuah data. Dlaam pengolahan data, *subsampling* juga bertujuan untuk meningkatkan invariansi posisi dari fitur. Dalam sebagian besar CNN, metode *subsampling* yang digunakan adalah *max pooling*. *Max pooling* membagi output dari *convoluion layer* menjadi beberapa *grid* kecil lalu mengambil nilai maksimal dari setiap *grid* untuk menyusun matriks yang telah direduksi.

3. *Fully Connected Layer*

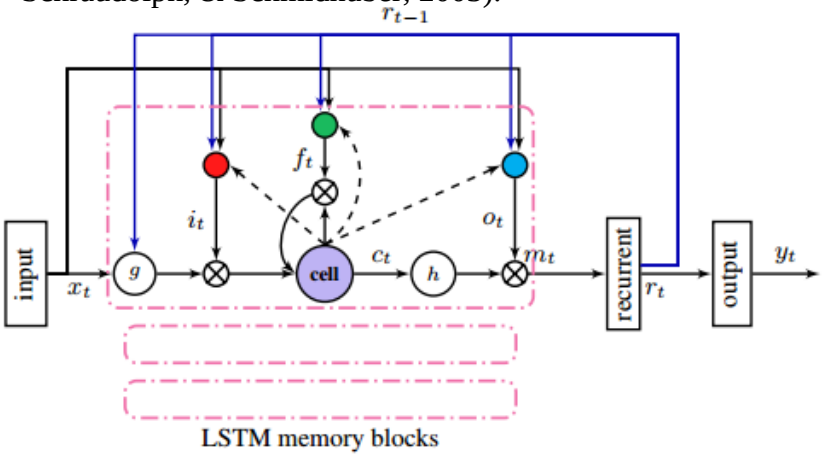
Layer tersebut adalah *layer* yang biasanya digunakan dalam penerapan MLP dan bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara *linear*. Setiap *neuron* pada *convolution layer* perlu ditransformasi menjadi data satu dimensi terlebih dahulu sebelum dapat dimasukkan ke dalam sebuah *fully connected layer*. Karena hal tersebut menyebabkan data kehilangan informasi spasialnya dan tidak reversibel, *fully connected layer* hanya dapat diimplementasikan di akhir jaringan.

2.1.5.2.3. LSTM (*Long Short-Term Memory)*

LSTM adalah arsitektur *recurrent neural network* (RNN) yang dirancang khusus untuk *model temporal sequences* dan lebih akurat dibanding dengan RNN biasa (Sak, Senior, & Beaufays, 2014). LSTM mencapai hasil yang paling dikenal dalam *natural language text compression* (Mahoney, 2017) dan *unsegmented connected handwriting recognition* (Graves et al., 2009), dan *automatic speech recognition* (Graves, Mohammed, & Hinton, 2013).

LSTM berisi unit khusus yang disebut blok memori yang berada pada *recurrent hidden layer*. Blok memori berisi sel memori yang menyimpan jaringan sementara selain unit perkalian khusus yang disebut *gates* untuk

mengontrol arus informasi. Setiap blok memori di jaringan original terdapat sebuah *input gate* dan *output gate*. *Input gate* mengendalikan aliran input aktivasi ke dalam sel memori. *Output gate* mengendalikan aliran output aktivasi ke seluruh jaringan. Setelah itu, *forget gate* akan ditambahkan ke blok memori. Hal tersebut untuk mengatasi kelemahan dari model LSTM mencegah pengolahan terus menerus *input streams* yang tidak tersegmentasi ke dalam *subsequences* (Gers, Schmidhuber, & Cummins, 2000). *Forget gate* menskalakan keadaan internal sel sebelum menambahkannya sebagai masukan ke sel melalui koneksi *self-recurrent* sel. Karena itu secara adaptif melupakan atau mengatur ulang ingatan sel. Selain itu, arsitektur LSTM modern mengandung *peephole connections* dari sel internalnya ke *gate* di sel yang sama untuk mempelajari waktu yang tepat untuk *output* (Gers, Schraudolph, & Schmidhuber, 2003).



Gambar 2.2 Arsitektur LSTMP RNN. Satu blok memori tunggal ditampilkan untuk kejelasan (Sumber: Sak, Senior, & Beaufays, 2014, p.339).

2.1.5.2.4. GRU (*Gated Recurrent Unit)*

GRU diusulkan oleh Cho et al. pada tahun 2014 untuk membuat setiap *recurrent unit* adaptif menangkap dependensi dari skala waktu yang berbeda. Sama halnya dengan LSTM, GRU memiliki *gating units* yang

memodulasi arus informasi di dalam unit, tanpa harus memiliki sel memori yang terpisah. GRU juga dirancang untuk secara adaptif dapat melakukan *reset* atau *update* isi memorinya. Setiap GRU memiliki sebuah *reset gate* dan *update gate* yang mengingatkan pada *forget gate* dan *input gate* dari LSTM. Namun, beda halnya dengan LSTM, GRU sepenuhnya memperlihatkan isi memori masing-masing *timestep* dan nilai antara isi memori sebelumnya dengan isi memori baru, meskipun secara adaptif dikontrol oleh *update gate*.

2.1.5.3. Library Deep learning

2.1.5.3.1. Keras

Keras adalah sebuah *High-Level Application Program Interface* (API) untuk *neural networks* yang ditulis dalam bahasa Python dan dapat dijalankan diatas TensorFlow atau Theano. Keras dikembangkan dengan tujuan untuk dapat menerapkan eksperimen yang cepat untuk para peneliti (Keras, n.d.).

Keras merupakan bersifat *open-source* dan dapat membantu para peneliti untuk melakukan pekerjaan *neural-networks* dari awal hingga selesai. Keras menyediakan *datasets* dan beberapa fitur berupa *preprocessing, model, layer,* dan *evaluate model*.

2.1.5.3.2. Theano

Theano adalah sebuah *library* Python yang dapat digunakan untuk *define, optimize,* dan *evaluate mathematical expression* yang melibatkan *multi-dimensional arrays* secara efisien (Theano, 2017).

2.1.6 My*Personality*

My*Personality* merupakan *project* yang diciptakan oleh Michal Kosinski (Kosinski et al., 2015). Aplikasi ini digunakan oleh

pengguna Facebook untuk mengisi kuesioner-kuesioner tentang diri mereka, kemudian memberikan hasil dari kuesioner tersebut yang berupa sifat mereka berdasarkan model kepribadian *BIG 5 Personality Traits.*

User-user yang telah menggunakan program ini kemudian di data profilnya dan dikumpulkan menjadi *dataset* yang telah digunakan oleh banyak peneliti di dunia.

2.1.7 *Resampling*

Datasets yang memiliki dua kelas dikatakan tidak seimbang (*imbalanced*) ketika salah satu kelas memiliki jumlah yang jauh lebih sedikit (minoritas) daripada kelas lainnya (mayoritas) (Japkowicz & Stephen, 2002).

Terdapat beberapa teknik untuk melakukan penyeimbangan (*balanced)* kelas antara lain:

1. *Over-sampling*

   Over-sampling dilakukan dengan duplikasi kelas minoritas secara acak agar kelas minoritas berjumlah sama dengan kelas mayoritas. Melakukan duplikasi pada kelas minoritas dapat menyebabkan pengambilan keputusan oleh sistem semakin kecil dan lebih spesifik yang menyebabkan sistem menjadi *over-fitting* (Han, Wang, & Mao, 2005).

2. *Under-sampling*

   Under-sampling dilakukan dengan penghapusan kelas mayoritas secara acak agar kelas mayoritas berjumlah sama dengan kelas minoritas. Kelemehan dalam melakukan *under-sampling* secara acak adalah besar kemungkinan suatu datasets akan kehilangan data yang berisi informasi berguna (Han, Wang, & Mao, 2005).

2.1.8 Python

Python dikembangkan oleh Guido van Rossum pada tahun 1990 di CWI, Amsterdam sebagai kelanjutan dari bahasa pemrograman ABC.

Saat ini pengembangan Python terus dilakukan oleh sekumpulan pemrogram yang dikoordinir Guido dan Python *Software Foundation*. Python *Software Foundation* adalah sebuah organisasi non-profit yang dibentuk sebagai pemegang hak cipta intelektual Python sejak versi 2.1 dan dengan demikian mencegah Python dimiliki oleh perusahaan komersial. Saat ini distribusi Python sudah mencapai versi 2.6.1 dan versi 3.6.

Nama Python dipilih oleh Guido sebagai nama bahasa ciptaannya karena kecintaan Guido pada acara televisi *Monty Python s Flying Circus*. Oleh karena itu seringkali ungkapan-ungkapan khas dari acara tersebut seringkali muncul dalam korespondensi antar pengguna Python.

Sisi utama yang membedakan Python dengan bahasa lain adalah dalam hal aturan penulisan kode program. Bagi para programmer di luar python siap-siap dibingungkan dengan aturan indentasi, tipe data, *tuple*, dan *dictionary*. Python memiliki kelebihan tersendiri dibandingkan dengan bahasa lain terutama dalam hal penanganan modul, ini yang membuat beberapa programmer menyukai python. Selain itu python merupakan salah satu produk yang *open source*, *free*, dan *multiplatform*.

Beberapa fitur yang dimiliki Python adalah:

- Memiliki *library* yang luas; dalam distribusi Python telah disediakan modul-modul siap pakai untuk berbagai keperluan.
- Memiliki tata bahasa yang jernih dan mudah dipelajari.
- Memiliki aturan layout kode sumber yang memudahkan pengecekan, pembacaan kembali dan penulisan ulang kode sumber.
- Berorientasi objek.
- Memiliki sistem pengelolaan memori otomatis (*garbage collection*, seperti java)
- Modular, mudah dikembangkan dengan menciptakan modul-modul baru; modul-modul tersebut

- Dapat dibangun dengan bahasa Python maupun C/C++.
- Memiliki fasilitas pengumpulan sampah otomatis, seperti halnya pada bahasa pemrograman Java,
- Python memiliki fasilitas pengaturan penggunaan ingatan komputer sehingga para pemrogram tidak perlu melakukan pengaturan ingatan komputer secara langsung.

2.1.9 Eclipse IDE

Menurut Vogel & Arthorne (2015), Eclipse dikenal sebagai *Integrated Development Environment* (IDE) untuk Java. Eclipse pun menguasai pasar global sebesar 65% sebagai IDE untuk *Java programming*.

IDE Eclipse dapat diintegrasikan dengan komponen *software* tambahan, yang biasa disebut dengan *plug-in* dan beberapa *plug-in* dapat dikelompokkan menjadi *features*. Beberapa proyek dan perusahaan telah meningkatkan pemakaian IDE Eclipse untuk digunakan sebagai dasar untuk membuat sebuah aplikasi. Aplikasi ini dikenal sebagai Eclipse *Rich Client Platform* (Eclipse RCP). IDE Eclipse yang digunakan pada penelitian ini ialah Eclipse Neon (versi 4.6).

2.1.10 Notepad++

Notepad++ adalah sebuah *source code* editor gratis dan dapat menggantikan posisi Notepad serta mendukung beberapa bahasa pemrograman. Notepad++ dijalankan di atas Microsoft Windows dan diatur oleh GPL *License* (Ho, 2016).

2.1.11 Microsof Excel 2010

Microsoft Excel 2010 adalah sebuah *spreadsheet software* yang dikembangkan oleh Microsoft untuk Windows, macOS, Android, dan iOS. Microsoft Excel 2010 telah banyak digunakan oleh ilmuwan untuk pengumpulan data, perhitungan, dan analisis (Zhang et al., 2010).

2.1.12 XAMPP

XAMPP adalah sebuah *package* distribusi Apache yang berukuran kecil dan ringan memiliki teknologi *web development* yang paling umum. XAMPP merupakan *tool* yang baik untuk siswa dalam melakukan *developing* dan *testing* aplikasi berbasis PHP dan MySQL (Dvorski, 2007)

.

2.1.13 Apache HTTP Server

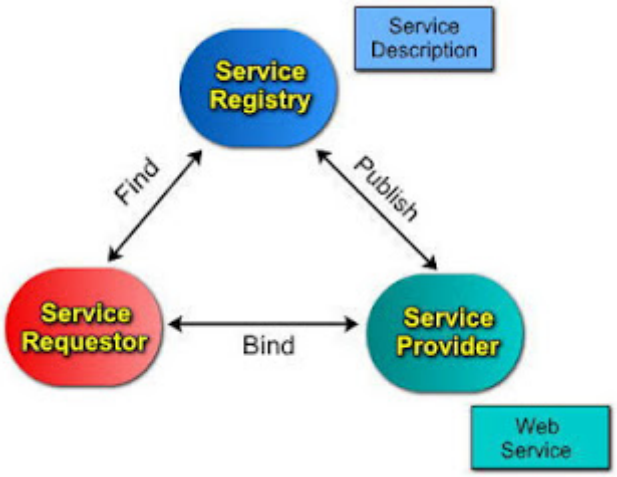
Apache HTTP Server adalah sebuah *open-source* HTTP *server* untuk sistem operasi modern seperti UNIX dan Windows. Tujuan dari Apache HTTP Server adalah untuk menyediakan layanan HTTP yang sinkron dengan standar HTTP saat ini (Apache, n.d.).

2.1.14 *Web Service*

Web service adalah aplikasi sekumpulan data (*database*), perangkat lunak (*software*) atau bagian dari perangkat lunak yang dapat diakses secara remote oleh berbagai piranti dengan sebuah perantara tertentu. Secara umum,*web service* dapat diidentifikasikan dengan menggunakan URL seperti hanya *web* pada umumnya. Namun yang membedakan *web service* dengan *web* pada umumnya adalah interaksi yang diberikan oleh *web service* . Berbeda dengan URL *web* pada umumnya, URL *web service* hanya menggandung kumpulan informasi, perintah, konfigurasi atau sintaks yang berguna membangun sebuah fungsi-fungsi tertentu dari aplikasi.

Web service dapat diartikan juga sebuah metode pertukaran data, tanpa memperhatikan dimana sebuahdatabase ditanamkan, dibuat dalam bahasa apa sebuah aplikasi yang mengkonsumsi data, dan di *platform* apa sebuah data itu dikonsumsi. *Web service* mampu menunjang interoperabilitas. Sehingga *web service* mampu menjadi sebuah jembatan penghubung antara berbagai sistem yang ada.

Menurut W3C *Web service Architecture Working Group* pengertian *Web service* adalah sebuah sistem *software* yang di desain untuk mendukung interoperabilitas interaksi mesin ke mesin melalui sebuah jaringan. *Interface web service* dideskripsikan dengan menggunakan format yang mampu diproses oleh mesin (khususnya WSDL). Sistem lain yang akan berinteraksi dengan *web service* hanya memerlukan SOAP, yang biasanya disampaikan dengan HTTP dan XML sehingga mempunyai korelasi dengan standar *Web* (Booth, 2004).



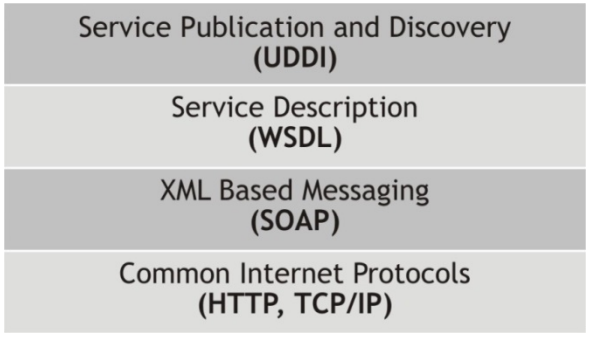
Gambar 2.3 Bentuk Arsitektur *Web service*
(Sumber: http://www.ibm.com/developerworks/*library*/ws-wsilover/*WebServices*Architecture.jpg)

Bagian-bagian dari entitas arsitektur *web service* dapat dilihat pada Gambar 2.3 dan terdiri dari bagian-bagian sebagai berikut:

- *Service* Provider: Berfungsi untuk menyediakan layanan/*service* dan mengolah sebuah registry agar layanan-layanan tersebut dapat tersedia.
- *Service* Registry: Berfungsi sebagai lokasi central yang mendeskripsikan semua layanan/*service* yang telah di-register.
- *Service Request*or: Peminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.

Web pada umumnya digunakan untuk melakukan respon dan *request* yang dilakukan antara *client* dan *server*. Sebagai contoh, seorang pengguna layanan *web* tertentu mengetikan alamat url *web* untuk membentuk sebuah*request*. *Request* akan sampai pada *server*, diolah dan kemudian disajikan dalam bentuk sebuah respon. Dengan singkat kata terjadilah hubungan *client-server* secara sederhana.

Sedangkan pada *web service* hubungan antara *client* dan *server* tidak terjadi secara langsung. Hubungan antara*client* dan *server* dijembatani oleh *file web service* dalam format tertentu. Sehingga akses terhadap databaseakan ditanggani tidak secara langsung oleh *server*, melainkan melalui perantara yang disebut sebagai *web service* . Peran dari *web service* ini akan mempermudah distribusi sekaligus integrasi database yang tersebar di beberapa *server* sekaligus.



Gambar 2.4 *Layer* dari komponen-komponen *Web service*
(Sumber : http://*file*s.ekowins.*web*node.com/200000001-856f5864af/komponen.jpg)

Komponen-komponen *Web service* :

1. *Layer* 1: Protokol internet standar seperti HTTP dan TCP/IP.
2. *Layer* 2: *Simple Object Access Protocol* (SOAP), adalah sebuah *XML-based mark-up language* untuk pergantian pesan diantara aplikasi-aplikasi. SOAP berguna seperti sebuah amplop yang digunakan untuk pertukaran data objek didalam *network*. SOAP mendefinisikan empat aspek didalam

komunikasi: *Message envelope*, *Encoding*, RPC *call convention*, dan bagaimana menyatukan sebuah *message* didalam protokol *transport*.

3. *Layer* 3: *Web Service Definition Language* (WSDL), adalah sebuah *XML-based language* untuk mendeskripsikan XML. Ia menyediakan *service* yang mendeskripsikan *service request* dengan menggunakan protokol-protokol yang berbeda dan juga *encoding*. Ia akan memfasilitasi komunikasi antar aplikasi. WSDL akan mendeskripsikan apa yang akan dilakukan oleh *web service*, bagaimana menemukannya dan bagaimana untuk mengoperasikannya.
4. *Layer* 4: *Universal Description Discovery* and Integration (UDDI), adalah sebuah *service* registry bagi pengalokasian *web service* . UDDI mengkombinasikan SOAP dan WSDL untuk pembentukan sebuah registry API bagi pendaftaran dan pengenalan *service*. Ia menyediakan sebuah area umum dimana sebuah organisasi dapat mengiklankan keberadaan mereka dan *service* yang mereka berikan (*web service*). UDDI adalah sebuah framework yang mendefinisikan sebuah *XML-based registry* dimana sebuah organisasi dapat meng-upload informasi mengenai *service* yang mereka berikan. *XML-based registry* berisi nama-nama dari organisasi tsb, beserta *service* dan deskripsi dari *service* yang mereka berikan.

2.1.15 *The Big Five Traits*

Kepribadian adalah pemikiran, emosi, dan perilaku tertentu yang menjadi ciri dari seseorang dalam menghadapi dunianya. Menurut McAdams & Olson (2010) kepribadian adalah seperangkat perbedaan individu yang dipengaruhi oleh perkembangan individu antara lain nilai, sikap, kenangan pribadi, hubungan sosial, kebiasaan, dan keterampilan.

Big five merupakan pendekatan yang digunakan untuk melihat kepribadian manusia melalui *trait* yang tersusun dalam lima buah

domain kepribadian yang telah dibentuk dengan menggunakan analisis faktor.

Dimensi *Big five* kebanyakan berasal dari pendekatan leksikal (bahasa) terhadap *trait*. Dengan kata lain, orang mendeskripsikan, menguji, dan mengkategorisasikan orang lain, dan peringkat yang dihasilkan disederhanakan ke dalam lima dimensi. *Trait-trait* dari *Big five* adalah sebagai berikut :

1. *Extraversion* (E)

   Extraversion adalah jenis kepribadian yang mengidentifikasikan intensitas dari interaksi interpersonal seseorang, tingkat aktivitas, kebutuhan akan stimulasi, dan kapasitas untuk berbahagia (Cervone & Pervin, 2015). Orang yang tinggi pada dimensi ini cenderung penuh semangat, antusias, dominan, ramah dan komunikatif. Orang yang sebaliknya akan cenderung pemalu, tidak percaya diri, submisif, dan pendiam.

2. *Agreeableness* (A)

   Agreeableness adalah jenis kepribadian yang mengukur orientasi interpersonal seseorang dalam hal pikiran, perasaan, dan tindakan (Cervone & Pervin, 2015). Orang yang tinggi pada dimensi *agreeableness* cenderung ramah, kooperatif, mudah percaya dan hangat. Orang yang rendah dalam dimensi ini cenderung dingin, konfrontatif dan kejam.

3. *Neuroticism* (N)

   Neuroticism adalah jenis kepribadian yang mengidentifikasi sebaik apa seseorang dapat mengatur emosinya (Cervone & Pervin, 2015). Orang yang tinggi dalam dimensi *Neuroticism* biasanya gugup, sensitif, tegang dan mudah cemas. Orang yang rendah dalam dimensi ini cenderung tenang dan santai.

4. *Conscientiousness*

   Conscientiousness adalah jenis kepribadian yang mengevaluasi bagaimana seseorang terorganisir, gigih, termotivasi, dan memiliki tujuan untuk dicapai (Cervone & Pervin, 2015). Orang yang tinggi dalam dimensi *Conscientiousness* umumnya berhati-hati, dapat diandalkan, teratur, dan bertanggung jawab. Orang yang rendah dalam dimensi ini cenderung ceroboh, berantakan dan biasanya tidak dapat diandalkan.

5. *Openness* (O)

   Openness adalah jenis kepribadian yang mengukur bagaimana seseorang mengapresiasi dan memiliki toleransi terhadap suatu hal atau pengalaman baru (Cervone & Pervine, 2015). Orang yang tinggi dalam dimensi *Openness* umumnya terlihat imajinatif, ceria, menyenangkan, kreatif dan artistik. Sebaliknya, orang yang rendah dalam dimensi ini umumnya dangkal, membosankan dan sederhana.

2.1.16 LIWC

Linguistic Inquiry and Word Count (LIWC) adalah program penghitung jumlah kata yang merujuk kepada kamus gramatikal, psikologikal dan kategori content kata (Pennebaker et al, 2007). LIWC telah digunakan secara efisien untuk mengklasifikasikan kata beserta dimensi psikologikalnya untuk memprediksi hasil kepribadian dari seseorang. Membuat LIWC ini menjadi alat analisis teks yang digunakan secara luas dalam ilmu *social sciences*. LIWC menghasilkan 81 *features* yang berbeda ke dalam 5 kategori.

Features tersebut terdiri dari *Standard Count* (jumlah kata, jumlah kata yang lebih dari enam huruf, banyaknya preposisi, dan sebagainya), *Psychological Processes* (emosi, kecerdasan seseorang, dan sebagainya), *Relativity* (kata yang menunjukkan waktu, mengungkapkan tentang masa lalu atau masa depan, dan sebagainya), *Personal Concern* (pekerjaan, isu keuangan, kesehatan), dan beberapa dimensi lainnya (jumlah tanda baca yang digunakan dan kata-kata umpatan atau hinaan). *Feature-feature* tersebut juga mengandung *syntactic* (contohnya rasio dari *pronouns*) dan *semantic information* (contohnya kata-kata emosi yang bersifat positif) yang telah divalidasi oleh para ahli.

LIWC	Gender		Age		Extraversion		Agreeableness		Conscientious.		Neuroticism		Openness	
Category	[34] *d*	our β	[30] β	our β	[27] ρ	our β	[27] ρ	our β	[27] ρ	our β	[27] ρ	our β	[27] ρ	our β
Total function words	-	-0.04	-	0.16	-	-0.04	-	0.02	-	0.02	-	0.03	-	0.09
Total pronouns	0.36	0.07	-	-0.02	*ns*	*ns*	0.11	*ns*	*ns*	-0.03	*ns*	0.04	-0.21	0.07
Personal pronouns	-	0.14	-	-0.08	-	*ns*	-	*ns*	-	-0.04	-	0.04	-	0.05
1st pers singular	0.17	0.13	-0.14	-0.22	*ns*	*ns*	*ns*	-0.03	*ns*	-0.06	0.12	0.05	-0.16	0.05
1st pers plural	*ns*	*ns*	-0.13	0.21	0.11	0.03	0.18	0.05	*ns*	0.05	*ns*	-0.04	-0.1	*ns*
2nd person	-0.06	0.05	-	0.04	0.16	*ns*	*ns*	0.02	*ns*	*ns*	-0.15	*ns*	-0.12	0.02
3rd pers singular	-	0.09	-	0.15	-	*ns*	-	*ns*	-	*ns*	-	0.02	-	*ns*
3rd pers plural	-	-0.05	-	0.26	-	-0.06	-	-0.04	-	*ns*	-	0.02	-	0.03
3rd pers overall	0.2	-	-	-	*ns*	-	*ns*	-	*ns*	-	*ns*	-	*ns*	-
Impersonal pronouns	-	-0.09	-	0.11	-	-0.05	-	*ns*	-	*ns*	-	0.02	-	0.08
Articles	-0.24	-0.24	-	0.28	*ns*	-0.05	*ns*	*ns*	0.09	0.02	-0.11	-0.02	0.2	0.13
Common verbs	-	0.04	-	0.02	-	-0.03	-	*ns*	-	*ns*	-	0.04	-	0.03
Auxiliary verbs	-	0.02	-	0.08	-	-0.06	-	*ns*	-	*ns*	-	0.05	-	0.07
Past tense	0.12	-0.03	-0.16	*ns*	*ns*	-0.04	0.1	0.02	*ns*	-0.02	*ns*	*ns*	-0.16	*ns*
Present tense	0.18	0.08	0.04	*ns*	*ns*	*ns*	*ns*	*ns*	*ns*	*ns*	*ns*	0.04	-0.16	0.03
Future tense	*ns*	-0.07	0.14	0.09	*ns*	-0.05	*ns*	*ns*	*ns*	*ns*	*ns*	0.03	*ns*	0.05
Adverbs	-	0.05	-	-0.07	-	-0.04	-	*ns*	-	*ns*	-	0.05	-	0.04
Prepositions	-0.17	-0.13	-	0.27	*ns*	-0.04	*ns*	0.03	*ns*	0.06	*ns*	*ns*	0.17	0.06
Conjunctions	-	0.03	-	0.12	-	-0.02	-	0.02	-	0.02	-	0.02	-	0.06
Negations	0.11	*ns*	-	-0.12	*ns*	-0.06	*ns*	-0.05	-0.17	-0.03	0.11	0.07	-0.13	0.02
Quantifiers	-	-0.09	-	0.24	-	-0.02	-	0.03	-	0.05	-	*ns*	-	0.05
Numbers	-0.15	-0.13	-	0.05	-0.12	-0.06	0.11	0.02	*ns*	0.02	*ns*	*ns*	-0.08	0.06
Swear words	-0.22	-0.21	-	-0.17	*ns*	*ns*	-0.21	-0.15	-0.14	-0.09	0.11	0.06	*ns*	*ns*
Social processes	-	0.08	-0.13	0.21	0.15	0.04	0.13	0.02	*ns*	*ns*	*ns*	*ns*	-0.14	*ns*
Family	0.12	0.22	-	0.28	0.09	0.03	0.19	0.03	*ns*	0.03	*ns*	*ns*	-0.17	-0.12
Friends	0.09	0.08	-	0.26	0.15	0.05	0.11	0.04	*ns*	0.02	-0.08	*ns*	*ns*	-0.04
Humans	*ns*	0.04	-	0.06	0.13	0.06	*ns*	-0.05	-0.12	*ns*	*ns*	*ns*	-0.09	*ns*
Affective processes	0.11	0.11	-	-0.05	0.09	0.07	*ns*	0.02	*ns*	*ns*	*ns*	*ns*	-0.12	-0.04
Positive emotion	*ns*	0.21	0.12	0.14	0.1	0.13	0.18	0.13	*ns*	0.1	*ns*	-0.08	-0.15	-0.07
Negative emotion	0.1	-0.12	-0.05	-0.31	*ns*	-0.07	-0.15	-0.17	-0.18	-0.13	0.16	0.15	*ns*	0.03
Anxiety	0.16	0.08	-	-0.13	*ns*	-0.04	*ns*	-0.02	*ns*	-0.02	0.17	0.06	*ns*	0.07
Anger	*ns*	-0.22	-	-0.25	*ns*	-0.05	-0.23	-0.19	-0.19	-0.12	0.13	0.11	*ns*	0.02
Sadness	0.1	0.08	-	-0.15	*ns*	-0.04	*ns*	-0.02	-0.11	-0.04	0.1	0.09	*ns*	*ns*
Cognitive processes	0.07	-0.03	0.07	0.1	*ns*	-0.05	*ns*	0.02	-0.11	*ns*	0.13	0.04	-0.09	0.1
Insight	0.09	-0.05	0.11	0.04	*ns*	-0.09	*ns*	*ns*	*ns*	-0.02	*ns*	0.05	*ns*	0.13
Causation	*ns*	-0.05	*ns*	-0.01	-0.09	-0.06	-0.11	-0.02	-0.12	*ns*	0.11	0.02	*ns*	0.08
Discrepancy	0.07	*ns*	-	0.02	*ns*	-0.05	*ns*	-0.02	-0.13	-0.03	0.13	0.07	-0.12	0.02
Tentative	*ns*	-0.12	-	0.07	-0.11	-0.08	*ns*	*ns*	-0.1	-0.03	0.12	0.06	*ns*	0.07
Certainty	0.14	*ns*	-	0.09	0.1	*ns*	*ns*	0.03	-0.1	0.04	0.13	*ns*	*ns*	0.06
Inhibition	-	0.03	-	0.09	-0.13	*ns*	*ns*	*ns*	*ns*	0.04	0.09	*ns*	*ns*	*ns*
Inclusive	*ns*	0.04	-	0.23	0.09	0.04	0.18	0.05	*ns*	0.05	*ns*	-0.02	0.11	0.06
Exclusive	*ns*	-0.05	*ns*	*ns*	*ns*	-0.07	*ns*	*ns*	-0.16	-0.03	0.1	0.05	*ns*	0.05
Perceptual Processes	0.12	*ns*	-	-0.06	0.09	-0.04	*ns*	*ns*	-0.1	-0.07	*ns*	0.03	-0.11	0.1
See	*ns*	*ns*	-	*ns*	*ns*	-0.02	0.09	*ns*	*ns*	-0.04	*ns*	*ns*	*ns*	0.04
Hear	0.1	-0.07	-	-0.1	0.12	-0.04	*ns*	*ns*	-0.12	-0.06	*ns*	0.02	-0.08	0.08
Feel	0.17	0.04	-	-0.07	*ns*	-0.02	0.1	*ns*	*ns*	-0.04	0.1	0.03	*ns*	0.05
Biological processes	*ns*	0.05	-	-0.06	0.14	0.04	0.09	-0.06	*ns*	-0.06	*ns*	0.05	-0.09	0.02
Body	-	-0.02	-	-0.14	0.1	*ns*	0.09	-0.09	*ns*	-0.09	*ns*	0.06	-0.04	0.04
Health	-	0.05	-	0.07	-	*ns*	-	*ns*	-	*ns*	-	0.06	-	*ns*
Sexual	*ns*	0.05	-	-0.14	0.17	0.1	0.08	-0.04	*ns*	-0.04	*ns*	*ns*	*ns*	*ns*
Ingestion	-	0.02	-	0.12	-	*ns*	-	-0.03	-	-0.03	-	*ns*	-	0.03
Relativity	-	-0.06	-	0.16	-	*ns*	-	0.05	-	0.08	-	-0.03	-	-0.03
Motion	0.07	*ns*	-	0.12	-	0.02	-	0.05	-	0.07	-	-0.04	-	-0.04
Space	*ns*	-0.18	-	0.21	*ns*	*ns*	0.16	*ns*	*ns*	0.02	-0.09	*ns*	-0.11	0.07
Time	*ns*	0.02	-0.19	0.08	*ns*	*ns*	0.12	0.06	0.09	0.09	*ns*	-0.03	-0.22	-0.07
Work	-0.12	-0.08	-	-0.02	-0.08	-0.05	*ns*	0.03	*ns*	0.1	*ns*	-0.03	*ns*	-0.02
Achievement	-	-0.17	-	0.16	-0.09	*ns*	*ns*	0.05	0.14	0.11	*ns*	-0.06	*ns*	-0.02
Leisure	*ns*	-0.08	-	0.03	0.08	0.06	0.15	0.04	*ns*	0.03	*ns*	-0.07	-0.17	*ns*
Home	0.15	0.19	-	0.18	*ns*	*ns*	0.19	0.03	*ns*	0.04	*ns*	-0.02	-0.2	-0.06
Money	-0.1	-0.12	-	0.24	*ns*	*ns*	-0.11	-0.04	*ns*	0.03	*ns*	*ns*	*ns*	0.03
Religion	-	-0.03	-	0.21	0.11	*ns*	*ns*	0.06	*ns*	0.04	*ns*	-0.04	*ns*	*ns*
Death	-	-0.18	-	-0.1	*ns*	-0.08	-0.13	-0.09	-0.12	-0.08	*ns*	0.08	0.15	0.09
Assent	-	0.07	-	-0.22	*ns*	0.05	*ns*	0.04	-0.09	*ns*	*ns*	-0.04	-0.11	-0.05
Nonfluencies	-	-0.03	-	0.02	-	*ns*	-	*ns*	-	*ns*	-	0.03	-	*ns*
Fillers	-	-0.02	-	-0.24	-	*ns*	-	-0.04	-	-0.08	-	0.03	-	0.04
participants (N)	*9,130*	*74,859*	*3,087*	*74,859*	*576*	*72,709*	*576*	*72,772*	*576*	*72,781*	*576*	*71,968*	*576*	*72,809*

Tabel 2.1 Korelasi Kategori LIWC dengan jenis kelamin, umur dan *Big five Model Personality* (Schwartz, et al., 2013)

2.1.17 SPLICE

Structured Programming for Linguistic Cue Extraction (SPLICE) adalah alat analisis linguistic yang dibangun oleh Kevin Moffit dan Justin Scott Giboney dari Universitas Arizona, Amerika Serikat (Moffit & Giboney, 2012). SPLICE merupakan tool yang cukup baru dan telah digunakan dalam berbagai penelitian yang berhubungan dengan NLP atau linguistik. SPLICE juga dapat diakses melalui *web* untuk membantu pengembangan aplikasi yang dapat terbantu dengan kekuatan teknik pemrosesan bahasa natural yang mumpuni. Berikut adalah fitur-fitur yang menjadi kategori kamus SPLICE:

1. *Quantity* (5 *features*)
   Fitur Berdasarkan frekuensi atau jumlah.
2. *Part of speech* (8 *features*)
   Semua fitur linguistik dalam kategori *Part of Speech* (POS) yang dikalkulasikan dengan POS *tagger* berdasarkan *Brown corpus.*
3. *Immediacy* (2 *features*)
   Fitur yang mengindikasikan kesiapan.
4. *Pronouns* (10 *features*)
   Fitur berdasarkan jumlah kata ganti.
5. *Positive Self Evaluation* (3 *features*)
   Fitur yang berhubungan dengan evaluasi positif pembicara.
6. *Negative Self Evaluation* (3 *features*)
   Fitur yang berhubungan dengan evaluasi negatif pembicara.
7. *Influence* (8 *features*)
   Fitur yang mengindikasikan pembicara mempengaruhi seseorang.
8. *Deference* (4 *features*)
   Fitur yang mengindikasikan keterlibatan pembicara di luar mempengaruhi
9. *Whissel* (3 *features*)
   Fitur yang menghitung hasil *Whissel Dictionary of Affect* dalam bahasa.

10. *Complexity* (12 *features*)
    Fitur kompleksitas teks.
11. *Spoken Word* (18 *features*)
    Fitur yang berhubungan dengan cara menulis atau berbicara.
12. *Tense* (4 *features*)
    Jumlah kata kerja dalam kalimat tertentu.
13. *Sentiwordnet* (3 *features*)
    Kalkulasi nilai positif, negatif, and objektivitas berdasarkan Senti*word*net.
14. *Readability* (11 *features*)
    Fitur yang memberikan hasil *readability*.

2.2 Penelitian Terkait

Berbagai penelitian mengenai prediksi atau hubungan antara kepribadian berbasis model "Big 5 *Traits*" telah banyak dilakukan sebelumnya. Berbagai penelitian tersebut juga menggunakan data dari berbagai sosial media, salah satunya Facebook yang merupakan target dari penelitian kita. Dari penelitian-penelitian tersebut, terdapat perkembangan dan kemajuan dalam akurasi menentukan sifat dari pengguna berdasarkan berbagai faktor dan fitur. Tidak hanya dari segi linguistik, beberapa peneliti juga mulai menggunakan fitur lainnya seperti umur, gender, jumlah foto, jumlah likes dan sebagainya. Berikut adalah beberapa penelitian prediksi kepribadia yang telah dilakukan para peneliti sebelumnya:

2.2.1 Personality and Patterns of Facebook Usage

Kesuksesan seorang individu bergantung terhadap apa yang ditampilkan oleh dirinya kepada orang lain. Kesuksesan di lingkungan kerja, percintaan dan mendapat dukungan positif dari seseorang sangatlah bergantung terhadap apa yang orang lain pikirkan tentang diri kita. Interaksi manusia, sosialisasi dan komunikasi di dalam dunia online yang semakin berkembang berarti menunjukkan pandangan diri kita terhadap orang lain juga semakin penting.

Salah satu sosial media yang paling berkembang, Facebook, telah mempengaruhi sebagian besar populasi manusia di dunia. Sekitar 800 juta pengguna menggunakan Facebook 40 menit setiap hari (Bachrach et al., 2012). Profil Facebook menjadi sumber informasi penting bagi kita untuk menciptakan pandangan kita terhadap orang lain. Dari beberapa penelitian sebelumnya, menunjukkan bahwa kepribadian seseorang dapat diprediksi oleh orang lain berdasarkan profil dari Facebook mereka.

Penelitian ini berfokus pada bagaimana prediksi kepribadian dapat diambil melalui fitur-fitur berbeda dari profil Facebook. Peneliti mengekstraksi *high-level features* dari profil Facebook dan menunjukkan korelasinya dengan kepribadian pengguna, sesuai dengan kuesioner model kepribadian *Five Factor Model*. Profil fitur yang digunakan dalam penelitian ini dibagi ke dalam dua aspek yaitu,

1. Aspek profil yang bergantung sepenuhnya terhadap kegiatan dari pengguna itu sendiri yang meliputi: Jumlah foto yang di upload, jumlah event yang diikuti dan jumlah grup yang dibentuk, serta jumlah objek atau foto yang di like oleh *user.*
2. Aspek pro*file* yang bergantung kepada kegiatan pengguna dan teman dari si pengguna yang meliputi: Jumlah berapa kali pengguna di tag dalam foto dan ukuran pertemanan si pengguna (number of friends or *network* size).

Feature	Details
Friends	number of Facebook friends
Groups	number of associations with groups
Likes	number of Facebook “likes”
Photos	number of photos uploaded by *user*
Statuses	number of status updates by *user*
Tags	number of times others “tagged” *user* in photos

Tabel 2.2 Fitur Profil Facebook yang digunakan dalam penelitian (Sumber: Bachrach et al., 2012)

Penelitian ini menggunakan Five Factor Model yang saat ini paling banyak digunakan dan diterima, dimana model untuk menentukan kepribadian manusia ini telah sangat baik dipelajari. Beberapa penelitian sebelumnya menunjukkan bahwa kepribadian berkorelasi dengan banyak aspek dalam kehidupan, termasuk kesuksesan dalam dunia kerja, daya tarik, keharmonisan pernikahan dan kebahagiaan. Penelitian sebelumnya telah menunjukkan kepribadian berkorelasi dengan total penggunaan internet, sosial media dan situs jaringan sosial lainnya. Tetapi, penelitian ini lebih berfokus pada jumlah berapa kali seorang pengguna menggunakan alat-alat atau fitur tersebut daripada bagaimana pengguna menggunakan alat atau fitur tersebut.

Beberapa hipotesis telah didapatkan dari hubungan antara kepribadian dan fitur profil Facebook. Ross et al. (2009), mempelajari mengenai hubungan antara kepribadian dan penggunaan jaringan sosial. Mereka menghasilkan beberapa hipotesis antara lain:

1. Relasi positif antara *Extraversion* dengan penggunaan Facebook, jumlah teman di Facebook dan jumlah grup si pengguna.
2. Relasi Positif antara *Neuroticism* dengan informasi privasi yang dimunculkan di Facebook.
3. Relasi Positif antara *Agreeableness* dengan jumlah teman di Facebook.
4. Relasi positif antara *Openness* dan banyaknya jumlah fitur Facebook yang dipakai.
5. Relasi negative antara *Conscientiousness* and penggunaan Facebook secara keseluruhan.

Sayangnya penelitian di atas menggunakan *dataset* yang relative cukup kecil (n = 97) dan sampel homogen (sebagian besar murid perempuan yang mengambil mata kuliah sama di universitas yang sama) yang membuat kurangnya kekuatan analisis dan sulitnya diterapkan kepada populasi pada umumnya.

Beberapa penelitian sebelumnya menggunakan *dataset* atau sampel yang terbatas dan homogen sehingga menghasilkan sesuatu yang kontradiksi. Tujuan utama penelitian ini adalah menggunakan sampel pengguna yang besar dan representatif untuk menjawab pertanyaan bagaimana profil Facebook dapat menentukan kepribadian. Peneliti mengemukakan hipotesis sebagai berikut:

1. *Openness* dan *Neuroticism* memiliki korelasi positif dengan jumlah update status, foto, grup dan jumlah likes pengguna.
2. *Conscientiousness* memiliki korelasi negatif dengan semua aspek dari penggunaan Facebook.
3. *Extraversion* memiliki korelasi positif dengan semua aspek dari penggunaan Facebook.
4. *Agreeableness* memiliki korelasi positif dengan jumlah teman, grup dan jumlah likes pengguna.

Peneliti menggunakan *dataset* dari 180.000 pengguna yang didapatkan dari my*Personality*, sebuah aplikasi Facebook yang dibuat pada tahun 2007. Aplikasi ini memungkinkan pengguna Facebook untuk mengisi kuesioner standar model kepribadian *Big five Personality* and untuk mendapatkan hasil dari kepribadian mereka. *Dataset* dalam penelitian ini memiliki rata-rata umur 24.15 (SD=6.55) dan dominan gender perempuan (58%).

Hasil dari penelitian ini menunjukkan bahwa:

- *Openness* berkorelasi positif dengan jumlah likes, grup dan update status dari pengguna.
- *Conscientiousness* berkorelasi negatif dengan jumlah likes dan jumlah grup. Tetapi memiliki korelasi positif dengan jumlah foto yang diupload.
- *Extraversion* memiliki korelasi positif dengan semua aspek.
- *Agreeableness* memiliki korelasi positif dengan jumlah *user* di tag di dalam foto. Tetapi memiliki korelasi negatif dengan jumlah likes dari *user*

- *Neuroticism* memiliki korelasi positif dengan jumlah likes dan grup dari *user*. Tetapi memiliki korelasi negatif dengan jumlah teman dari *user*.

Personality Trait	Pro*file* Feature	Pearson Correlation
Openness	Likes	0.102
	Statuses	0.062
	Groups	0.077
Conscientiousness	Likes	-0.088
	Groups	0.06977
	Photos	0.0330
Extraversion	Statuses	0.117
	Likes	0.034
	Groups	0.069
	Friends	0.177
Aggreableness	Likes	-0.036
Neuroticism	Likes	0.075
	Friends	-0.059

Tabel 2.3 Korelasi antara fitur profil Facebook dengan jenis dari 5 kepribadian (Sumber: Bachrach et al., 2012)

2.2.2 Recognising Personality Traits Using Facebook Status Updates

User Generated Content (UGC) di situs *online* jaringan sosial berpotensial menyediakan sumber informasi yang kaya untuk aplikasi bisnis yang dapat digunakan untuk berbagai hal contohnya *online marketing*. Dalam penelitian ini, peneliti berkontribusi dalam bidang ini dengan mengeksplorasi penggunaan teknik *Machine learning* (ML) untuk mengetahui kepribadian pengguna secara otomatis melalui *update* status di Facebook.

Jenis kepribadian pada umumnya di deskripsikan dengan menggunakan kepribadian lima dimensi yang diketahui dengan nama *Big five*. *Big five* tersebut adalah *Extraversion*, *Neuroticism*, *Agreeableness*, *Openness* dan *Conscientiousness*. Karena ada kemungkinan lebih dari satu jenis kepribadian dapat muncul dalam

seorang *user* yang sama, maka untuk setiap kepribadian peneliti melakukan *training classifier* binary untuk memisahkan *user* yang memiliki kepribadian tersebut dan yang tidak memiliki kepribadian tersebut.

Peneliti menggunakan berbagai fitur sebagai input untuk *classifier* yaitu:

1. LIWC, sebagai fitur linguistic yang terdiri dari 81 fitur yang merujuk pada penelitian Tausczik & Pennebaker (2010).
2. Social *network features*, yang terdiri dari
    - *Network* size
    - Betweenness
    - nBetweenness
    - Density
    - Brokerage
    - nBrokerage
    - Transitivity
3. Time-related *Features*, yang terdiri dari
    - Frekuensi status update dalam sehari
    - Jumlah status update dari jam 06-11 pagi
    - Jumlah status update dari jam 11-16
    - Jumlah status update dari jam 16-21
    - Jumlah status update dari jam 21-00
    - Jumlah status update dari jam 00-06 pagi
4. Other *Features*
    - Jumlah status per *user*
    - Jumlah kata kapital
    - Jumlah huruf kapital
    - Jumlah kata yang digunakan lebih dari satu kali
    - Jumlah url
    - Jumlah penggunaan PROPNAME – alias yang digunakan untuk penyamaran nama

Value	cEXT	cNEU	cAGR	cCON	cOPN
Yes	96	99	134	130	176
No	154	151	116	120	74

Tabel 2.4 Distribusi Kepribadian Berdasarkan Data Pengguna Facebook (Sumber: Farnadi et al., 2013)

Penelitian ini membandingkan penggunaan tiga algoritma *Machine learning* yaitu *Support Vector Machine* (SVM), Nearest Neighbor with k=1 (kNN) dan Naïve Bayes (NB). Semua hasil akan diperoleh menggunakan WEKA (Witten & Frank, 2015) dan dibandingkan dengan algoritma kelas mayoritas (Base). Untuk menginvestigasi bagaimana setiap grup dari fitur berkontribusi terhadap hasil, peneliti melakukan *training classifier* biner menggunakan ketiga algoritma tersebut. Semua hasil kemudian di rata-rata menggunakan *10-fold cross-validation* dan tes two-tailed paired t-tes dilakukan untuk mengevaluasi perbedaan signifikan dengan baseline $p < .05$.

Features	*Algorithm*	cEXT	cNEU	cAGR	cCON	cOPN
-	Base	0.38	0.36	0.29	0.27	0.50
LIWC	SVM	0.58•	0.48•	0.47•	0.55•	0.60•
	kNN	0.58•	0.54•	0.50•	0.54•	0.54
	NB	0.58•	0.52•	0.52•	0.48 •	0.60 •
Social	SVM	**0.71**•	0.36	**0.60**•	0.45 •	0.50
	kNN	0.62•	0.53•	0.52•	0.47•	0.60•
	NB	0.67•	**0.62**•	0.52•	0.55•	**0.63**•
Time	SVM	0.38	0.36	0.33	**0.59**•	0.50
	kNN	0.63•	0.54•	0.53•	0.50•	0.55•
	NB	0.51•	0.44	0.26	0.26*	0.60
Other	SVM	0.40	0.40	0.35	0.52•	0.50
	kNN	0.45•	0.57•	0.50•	0.51•	0.57•
	NB	0.54•	0.51•	0.46•	0.57•	0.59•

Tabel 2.5 Hasil Klasifikasi berdasarkan Precision (Sumber: Farnadi et al., 2013)

Features	*Algorith*	cEXT	cNEU	cAGR	cCON	cOPN
-	Base	0.62	0.55	0.54	0.52	0.70
LIWC	SVM	0.61	0.57	0.50	0.54	0.70
	kNN	0.57	0.53	0.50	0.54	0.54*
	NB	0.53*	0.55	0.53	0.47	0.62*
Social	SVM	**0.68**•	0.6•	**0.57**•	0.52	0.70
	kNN	0.62	0.53	0.51	0.47	*0.60
	NB	0.59	0.56	0.50	0.54	0.46*
Time	SVM	0.62	0.60•	0.54	**0.55**	0.70
	kNN	0.62	0.54*	0.53	0.51	0.56*
	NB	0.61	0.43*	0.46*	0.48*	0.38*
Other	SVM	0.62	**0.61**•	0.52	0.53	0.70
	kNN	0.47*	0.56	0.50	0.51	0.57*
	NB	0.60	0.49*	0.49	0.55	0.67

Tabel 2.6 Hasil Klasifikasi berdasarkan Recall

(Sumber: Farnadi et al., 2013)

Features	*Algorithm*	cEXT	cNEU	cAGR	cCON	cOPN
-	Base	0.47	0.45	0.37	0.36	0.58
LIWC	SVM	0.56•	0.49	0.45•	**0.54**•	**0.61**•
	kNN	0.57•	0.52•	0.50•	0.53•	0.54
	NB	0.53	0.51	0.48•	0.44•	0.60
Social	SVM	**0.62**•	0.45	0.50•	0.41•	0.58
	kNN	**0.62**•	0.53•	0.51•	0.46•	0.59
	NB	0.58•	0.54•	0.47•	0.48 •	0.46*
Time	SVM	0.47	0.45	0.39	0.47•	0.58
	kNN	**0.62**•	0.53•	**0.53**•	•0.5	0.55
	NB	0.50	0.30*	0.31*	0.33*	0.32*
Other	SVM	0.47	0.46	0.38	0.43•	0.58
	kNN	0.46	**0.56** •	0.49•	0.51•	0.57
	NB	0.52	0.45	0.44•	0.49•	0.59

Tabel 2.7 Hasil Klasifikasi berdasarkan F-measure

(Sumber: Farnadi et al., 2013)

Class	*Features*	*Algorithm*	Precision	Recall	F-measure
cEXT	Social	SVM	0.71	0.68	0.62
	Social+Time	NB	0.69	0.68	0.65

cNEU	Other	SVM	0.40	0.61	0.46
	Other+Social	NB	0.63	0.55	0.53
cAGR	Social	SVM	0.60	0.57	0.50
	Social+Time	SVM	0.60	0.57	0.49
cCON	LIWC	SVM	0.55	0.54	0.54
	LIWC+Social	kNN	0.55	0.54	0.54
	LIWC+Social+Time	kNN	0.56	0.55	0.55
	LIWC+Social+Time+Other	kNN	0.54	0.54	0.53
cOPN	LIWC	SVM	0.60	0.70	0.61
	LIWC+Social	SVM	0.62	0.71	0.62
	LIWC+Social+Time	SVM	0.61	0.70	0.62

Tabel 2.8 Hasil Klasifikasi berdasarkan Gabungan dari Fitur-Fitur

(Sumber: Farnadi et al., 2013)

Features	cEXT	cNEU	cAGR	cCON	cOPN
Network size	0.31●	-0.18●	0.07	0.14●	0.02
Betweeness	0.25●	-0.13●	0.05	0.11	0.04
nBetweenness	0.22●	-0.03	0.11	0.12	-0.06
Density	-0.24●	0.10	-0.08	-0.14●	0.05
Brokerage	0.25●	-0.13●	0.05	0.11	0.04
nBrokerage	0.23●	-0.08	0.09	0.08	-0.01
Transitivity	-0.27●	0.14●	-0.15●	-0.02	-0.06

Tabel 2.9 Hasil Korelasi antara Fitur Sosial dan Jenis Kepribadian (nilai dengan simbol (●) adalah nilai yang secara signifikan ($p < .05$) berkorelasi dengan kepribadian)

(Sumber: Farnadi et al., 2013)

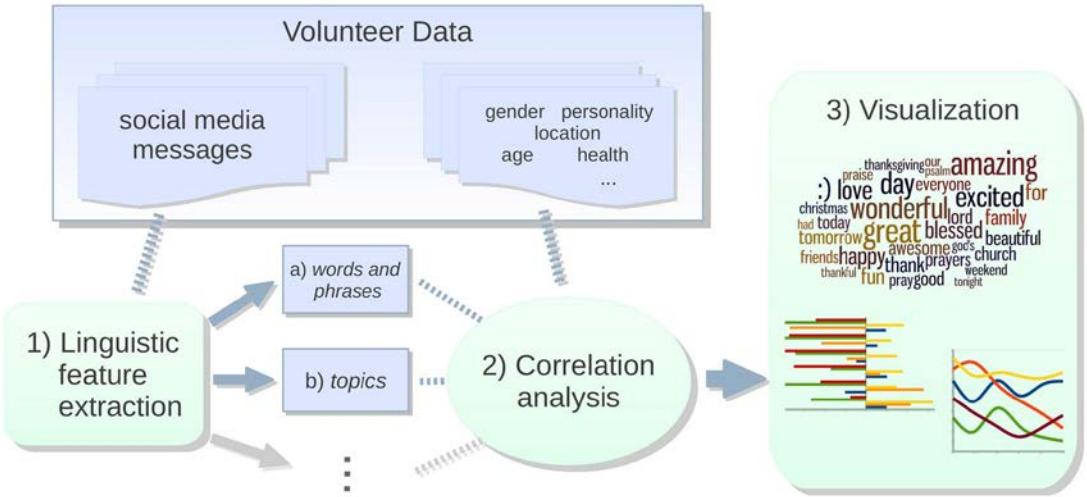
2.2.3 Personality, Gender, and Age in the Language of Social Media: The Open-Vocabulary Approach

Penelitian ini dilakukan oleh Schwartz et al. pada tahun 2013. Peneliti berpendapat bahwa ilmu sosial telah memasuki era *data science*, memanfaatkan teknologi dan sumber dari sosial media yang belum ada sebelumnya. Melalui media seperti Facebook dan Twitter, yang digunakan terus menerus oleh 1/7 dari populasi dunia

(Facebook, 2012), penggunaannya untuk memprediksi pasar saham dan penghitungan estimasi kebahagiaan sepanjang waktu. Untuk menggunakan data-data tersebut dalam jumlah yang besar, peneliti menggabungkan komputasi bahasa dan ilmu sosial. Teknik yang peneliti gunakan memanfaatkan apa yang orang-orang bagi atau tulis di sosial media dengan mencari kata-kata, frase dan topik dari atribut seseorang seperti jenis kelamin, umur dan lokasi mereka.

Peneliti menggunakan *differential language analysis* (DLA), metode khusus peneliti untuk analisis *open-vocabulary*, untuk mencari fitur bahasa dari jutaan pesan Facebook yang membedakan demografis dan atribut psikologis. Dari dataset yang dikumpulkan lebih dari 15,4 juta pesan Facebook dikumpulkan dari 75 ribu sukarelawan (Kosinski et al, 2015), peneliti mengekstrak 700 juta contoh kata, frasa, dan secara otomatis menghasilkan topic dan korelasi mereka dengan jenis kelamin, umur dan kepribadian. Peneliti mereplikasi analisis bahasa tradisional dengan menerapkan *Linguistic Inquiry and Word Count* (LIWC) (Pennebaker et al., 2007), sebuah tool yang popular di bidang psikologi, terhadap dataset dalam penelitian.

Kepribadian yang dihasilkan akan diklasifikasikan berdasarkan *Five Factor Model* atau *Big 5*, yang klasifikasinya terbagi dalam 5 dimensi yaitu: *extraversion*, *agreeableness*, *conscientiousness*, *neuroticism*, dan *openness.* Dengan memeriksa kata-kata yang digunakan oleh seseorang, para peneliti telah lama menemukan pemahaman mengenai psikologi manusia. Seperti dalam penelitian Tauszczik & Pennebaker (2010) mengemukakan bahwa “Bahasa adalah cara yang paling umum dan dapat diandalkan orang untuk menerjemahkan pikiran dan emosi internal mereka ke dalam bentuk yang bisa dimengerti oleh orang lain. Kata-kata dan bahasa adalah hal yang sangat penting dalam psikologi dan komunikasi.”



Gambar 2.5 Infrastruktur dari *differential language analysis* (DLA)

(Sumber: Schwartz et al., 2013)

Metodologi yang digunakan dalam penelitian ini seperti terlihat pada Gambar 2.5 yaitu:

1. Feature Extraction
   - *Words and phrases*: urutan dari 1 sampai 3 kata yang ditemukan dengan menggunakan *emoticon-aware tokenizer* dan *filter collocation* (24.530 fitur).
   - Topics: secara otomatis melakukan pengelompokan kata menjadi satu topik dengan menggunakan teknik Latent *Dirichlet Allocation* (500 fitur).
2. Analisis Korelasi

   Peneliti menemukan korelasi (β dari *least square linear regression*) antara setiap fitur bahasa dan masing-masing demografis atau hasil psikometik. Semua hubungan yang direpresentasikan dalam penelitian ini berada di signifikan terkecil dari *Bonferroni-corrected* $p < 0.001$.
3. Visualisasi

   Representasi grafis dari hasil analisis korelasi.

Open Vocabulary: Differential Language Analysis adalah teknik yang digunakan dalam penelitian ini dan terbagi menjadi tiga kunci karakteristik yaitu:

1. *Open-vocabulary* – tidak terbatas pada daftar kata yang telah ditentukan. Tetapi, fitur linguistik termasuk kata-kata, frasa, dan topik (kumpulan kata-kata yang berhubungan secara semantis) ditentukan secara otomatis dari teks (*data-driven*), artinya DLA diklasifikasikan dengan pendekatan *open-vocabulary*.

2. *Discriminating* – mencari fitur linguistik yang membedakan atribut psikologi dan demografis, menggunakan tes yang signifikan.

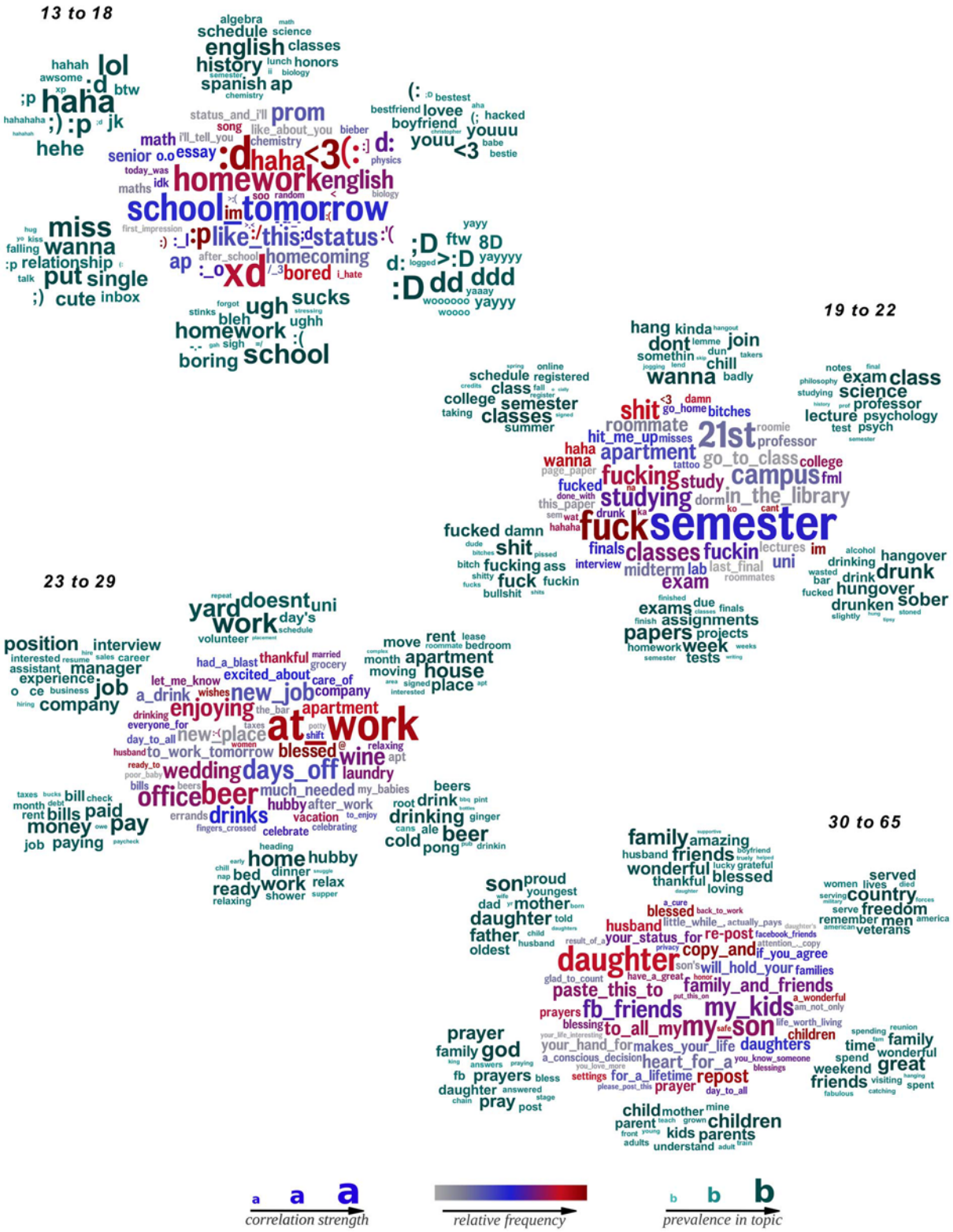
3. *Simple* – menggunakan teknik statistik yang sederhana, cepat dan mudah diterima.

Tabel 2.10 Statistik jenis kelamin, umur dan lima faktor model kepribadian (Sumber: Schwartz et al., 2013)

	N	*Mean*	*Standard deviation*	*skewness*
Gender	74859	0.62	0.49	-0.50
Age	74859	23.43	8.96	1.77
Extraversion	72709	-0.07	1.01	-0.34
Agreeableness	72772	0.03	1.00	-0.40
Conscientiousness	72781	-0.04	1.01	-0.09
Neuroticism	71968	0.14	1.04	-0.21
Openness	72809	0.12	0.97	-0.48

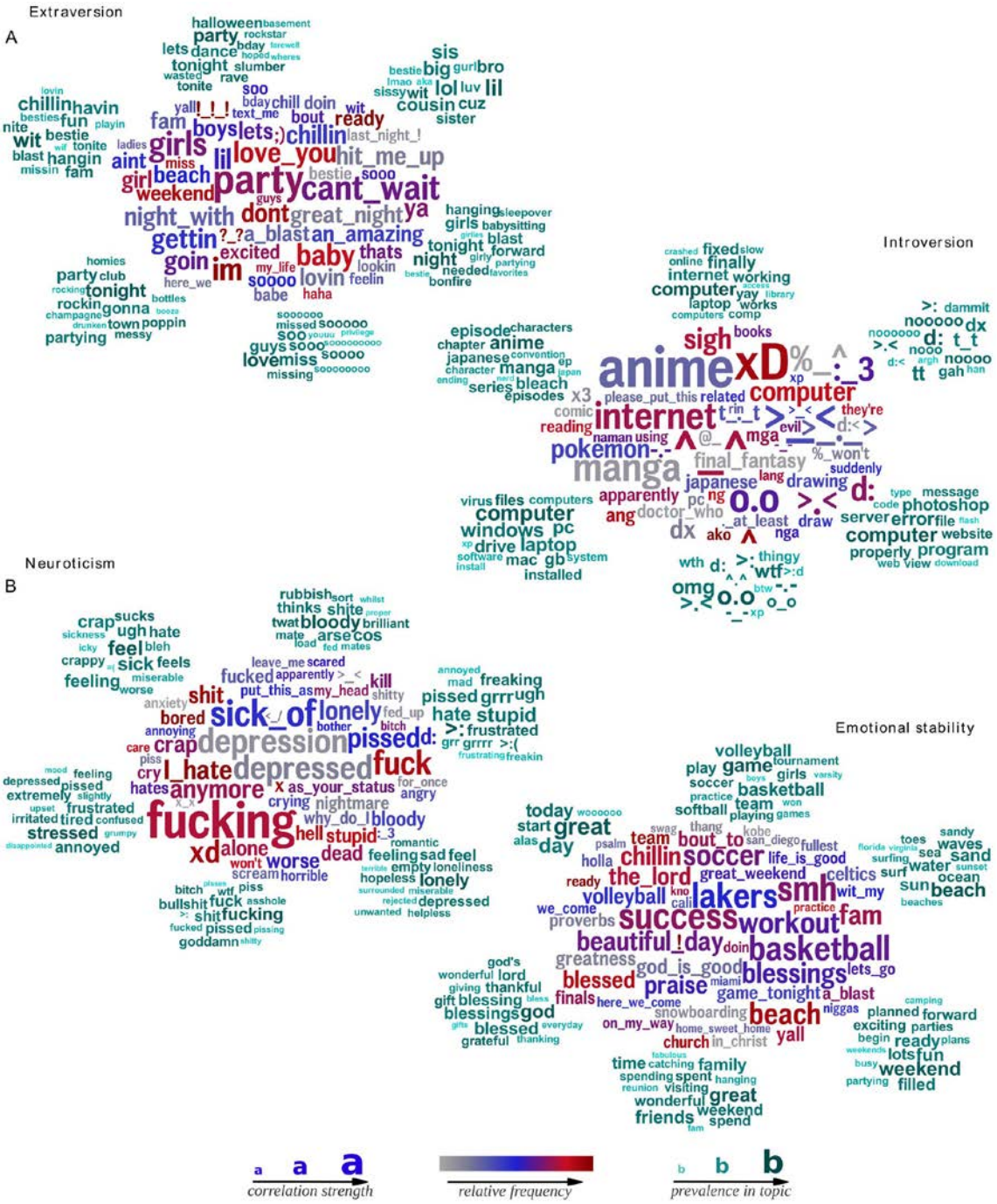
Tabel 2.10 diatas memperlihatkan statistik dari tujuh variabel dependen yang ada dalam penelitian kali ini. Jenis kelamin dihitung dengan nilai 0 bagi pria dan 1 bagi wanita. Umur antara 13 hingga 65 tahun. Untuk visualisasi lebih jelas, peneliti membuat *words clouds* untuk memperlihatkan penyebaran kata berdasarkan variabel-variabel tersebut di atas. *Words clouds* yang dibuat oleh peneliti dapat dilihat pada Gambar 2.6 yang merupakan *words clouds* dengan korelasi terhadap umur dan Gambar 2.7 yang merupakan *words clouds* dengan korelasi terhadap karakter *extraversion* dan *introversion* dari stabilitas emosional.

Gambar 2.6 Kata-kata, frasa dan topik yang membedakan subjek umur 13-18, 19-22, 23-29, dan 30-65. (Sumber: Schwartz et al., 2013)



Gambar 2.7 Kata-kata, frasa dan topik yang membedakan karakter *extraversion* dari *introversion* dari stabilitas emosional

(Sumber: Schwartz et al., 2013)



Tabel 2.11 Perbandingan LIWC dan fitur *open-vocabulary*
(Sumber: Schwartz et al., 2013)

	Gender	**Age**	***Extraversion***	***Agreeableness***	***Conscientiousness***	***Neuroticism***	***Openness***
Features	***accuracy***	**R**	**R**	**R**	**R**	**R**	**R**
LIWC	78.4%	.65	.27	.25	.29	.21	.29
Topics	87.5%	.80	.32	.29	.33	.28	.28
Wordphrases	91.4%	.83	.37	.29	.34	.29	.41
Wordphrases + Topics	91.9%	.84	.38	.31	.35	.31	.42
Topics + LIWC	89.2%	.80	.33	.29	.33	.28	.38
Wordphrases + LIWC	91.6%	.83	.38	.30	.34	.30	.41
Wordphrases + Topics + LIWC	91.9%	.84	.38	.31	.35	.31	.42

Dari tabel 2.11 di atas dapat dilihat bahwa model yang menggunakan metode *open vocabulary* memiliki akurasi yang lebih tinggi dibandingkan dengan model yang dibangun dengan fitur LIWC. Kemudian juga dapat dilihat bahwa tambahan fitur LIWC terhadap metode *open vocabulary* tidak dapat meningkatkan akurasi yang menyimpulkan bahwa *open vocabulary* dapat digunakan secara independen dalam sistem ini tanpa fitur lainnya seperti LIWC

2.2.4 Sistem Prediksi Kepribadian "The *Big Five Traits*" Dari Data Twitter

Penelitian ini dilakukan oleh Wijaya et al. (2016). Penelitian ini mencoba untuk mebuat sebuah sistem prediksi kepribadian berdasarkan informasi *user* Twitter dalam bahasa Inggris. Model kepribadian yang digunakan adalah *Five Factor* model.

Dataset yang digunakan dalam penelitian ini sebanyak 4429 *user* yang aktif melakukan *tweet* bahasa Inggris di sosial media Twitter. Tahapan yang dilakukan dalam penelitian ini terbagi menjadi 4 yaitu: *Twitter data extraction, feature extraction, WEKA processing,* dan *system prediction*.

1. *Twitter data extraction*

Cara ekstraksi data dari Twitter dilakukan melalui Twitter API. Ekstraksi yang dilakukan adalah mengambil *username*, *tweet* dari *user*, jumlah *follower*, jumlah *following*, *tweet* favorit, jumlah *mentions* dan jumlah *hashtags*.

Penelitian ini juga melakukan *preprocessing* data sebelum dilanjutnya ke tahap selanjutnya. Tahap *preprocessing* yang dilakukan adalah:

- *Removal of retweets*
- *Removal of emojis*
- *Removal of hyperlinks*

2. *Feature Extraction*

Fitur yang digunakan dalam penelitian ini adalah LIWC dan MRC. Total sebanyak 27 fitur digunakan dalam penelitian dan fitur tersebut dibagi dalam tiga bagian yaitu: *word* level, message level dan Boolean.

3. WEKA *Processing*

Proses ini menggunakan WEKA untuk menentukan kalkulasi seperti *standard deviation* (sd), *average* (*mean*), *maximum value* dan *minimum value*.

4. *System Prediction*

Tahap akhir adalah menampilkan hasil prediksi ke dalam aplikasi yang dibangun dengan Android.

Classifier dibangun untuk mengidentifikasi kombinasi model kepribadian *Big five* sebanyak 32 *classes.* Sistem prediksi dilakukan dengan menggunakan algoritma *Support Vector Machine (*SVM*),* ZeroR, dan Naïve Bayes. Evaluasi dilakukan dengan 5, 10, 15, dan 20-*fold cross validation*.

Tabel 2.12 Akurasi sistem prediksi kepribadian dengan menggunakan K-*Fold cross validation*
(Sumber: Wijaya et al., 2016)

K-*fold cross validation*	**Akurasi**
5 folds	23.9557%
10 folds	24.4751%
15 folds	24.0912%
20 folds	24.0912%

Perbandingan akurasi algoritma *Support Vector Machine*, *Naïve Bayes*, and *ZeroR* terdapat pada Tabel 2.13 di bawah ini. Akurasi paling tinggi didapatkan dengan menggunakan algoritma *Support Vector Machine*.

Tabel 2.13 Perbandingan akurasi algoritma *Support Vector Machine*, Naïve Bayes, and ZeroR. (Sumber: Wijaya et al., 2016)

Classifier	**Akurasi**
Support Vector Machine	24.4751%
Naïve Bayes	9.0765%
ZeroR	21.0883%

Evaluasi juga dilakukan dengan mengevaluasi lima kepribadian *Big five* yaitu: *Extraversion, Emotional Stability, Agreeableness, Conscientiousness,* dan *Openness.*

Tabel 2.14 Akurasi prediksi kepribadian
(Sumber: Wijaya et al., 2016)

Personality Trait	**Akurasi**
Extraversion	61.5263%
Emotional *Stability*	80.876%
Agreeableness	64.1906%
Conscientiousness	77.8597%
Openness	70.49%

Peneliti membandingan hasil akurasi penelitian ini dengan penelitian sebelumnya dan menyimpulkan *contextual features* dapat meningkatkan akurasi sedikit lebih tinggi.

2.2.5 Personality Prediction Based on Twitter Information in Bahasa

Penelitian tentang prediksi kepribadian telah banyak dilakukan dengan menggunakan berbagai sosial media. Penelitian yang dilakukan oleh Ong et al. (2017) ini menggunakan sosial media Twitter dan berfokus pada bahasa Indonesia. Beberapa statistik mengemukakan bahwa Indonesia merupakan salah satu pengguna sosial media Twitter tertinggi, sekitar 2.4% dari *tweets* di dunia berasal dari *user* yang berlokasi di Jakarta, Indonesia (Carley et al., 2015).

Sistem kepribadian dalam penelitian ini dibangun berdasarkan model kepribadian *Big five* oleh McCrae dan Costa yang membagi kepribadian seseorang menjadi 5 jenis yaitu, *Agreeableness, Conscientiousness, Emotional Stability, Extraversion,* dan *Openness*. Tujuan penelitian ini dibuat adalah untuk membangun sistem prediksi kepribadian dengan bahasa Indonesia, melakukan beberapa skenario percobaan untuk meningkatkan akurasi, dan perbandingan 2 algoritma *machine learning* yang diimplementasikan ke model prediksi.

Dataset yang yang digunakan pada penelitian ini sebanyak 359 dimana 329 data sebagai data *training* dan sisa 30 data sebagai data *testing*. Dataset *user* yang dipilih berdasarkan kriteria:

1. *User* post di Twitter minimal sekali setiap bulan
2. *User* menggunakan bahasa Indonesia

Informasi *user*s diekstraksi menjadi 12 fitur yaitu:

1. Jumlah *tweets*
2. Jumlah *followers*
3. Jumlah *following*
4. Jumlah favorit
5. Jumlah *retweets*

6. Jumlah *tweet* yang di retweet
7. Jumlah *tweet* dengan kutip
8. Jumlah *mentions*
9. Jumlah *replies*
10. Jumlah *hashtags*
11. Jumlah URLs
12. Rata-rata perbedaan waktu setiap *tweet*

Pelabelan dataset itu dilakukan dengan bantuan ahli psikologi dengan tabel distribusi seperti di bawah ini.

	Agreeableness	***Conscientiousness***	**Emotional *Stability***	***Extraversion***	***Openness***
High	134	92	150	202	163
Low	195	237	179	127	166

Tabel 2.15 Tabel distribusi dataset *training* (Sumber: Ong et al., 2017)

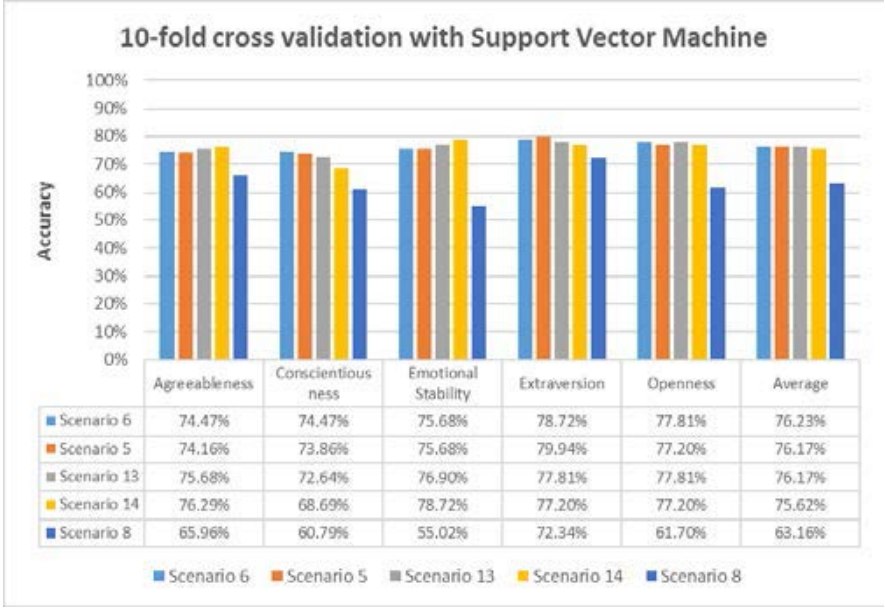
	Agreeableness	***Conscientiousness***	**Emotional *Stability***	***Extraversion***	***Openness***
High	19	16	21	23	16
Low	11	14	9	6	14

Tabel 2.16 Tabel distribusi dataset *testing* (Sumber: Ong et al., 2017)

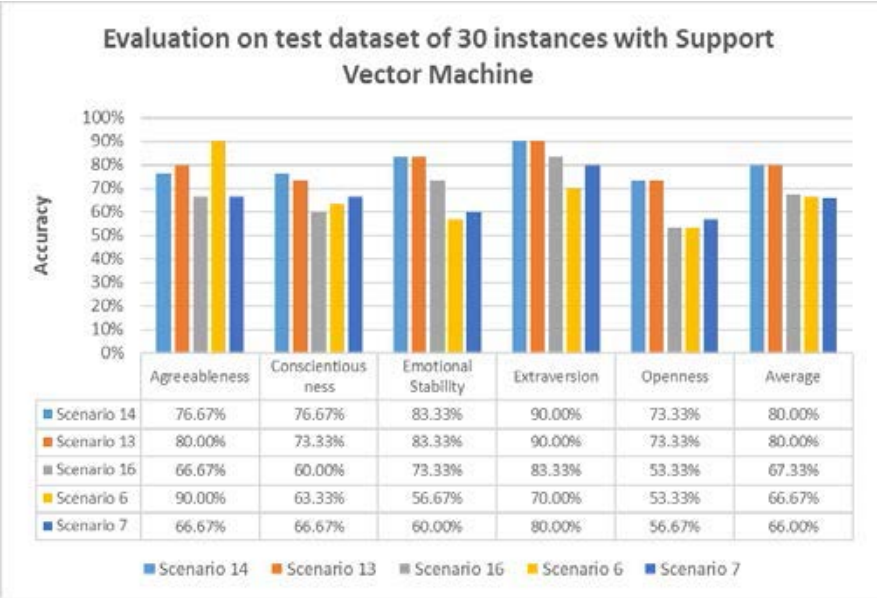
Selanjutnya, tahap yang dilakukan adalah *preprocessing* data untuk menghilangkan beberapa elemen secara otomatis dan manual seperti, penghilangan *emoticon*, penghilangan URLs/*hyperlinks*, penghilangan st*op words* dan elemen lainnya. Setelah melalui tahap *preprocessing*, sistem prediksi dibangun dengan menggunakan Support Vector Machine dan XGBoost, dimana SVM dijalankan dengan WEKA dan XGBoost dengan R. Setelah proses *training*, sistem di evaluasi dengan *10-fold cross validation* dan memasukkan 30 data *testing* ke dalam sistem. Sistem prediksi ini dibangun dengan beberapa skenario seperti terlihat pada Tabel 2.17.

Scenario	Minimum occurrence of n-gram		n-gram weighting scheme		LDA topic *features*		Stop *words* omission	
	1	2	Boolean	TF	Use LDA	Don't use LDA	Omit	Don't omit
1	✓		✓		✓		✓	
2	✓		✓		✓			✓
3	✓		✓			✓	✓	
4	✓		✓			✓		✓
5	✓			✓	✓		✓	
6	✓			✓	✓			✓
7	✓			✓		✓	✓	
8	✓			✓		✓		✓
9		✓	✓		✓		✓	
10		✓	✓		✓			✓
11		✓	✓			✓	✓	
12		✓	✓			✓		✓
13		✓		✓	✓		✓	
14		✓		✓	✓			✓
15		✓		✓		✓	✓	
16		✓		✓		✓		✓

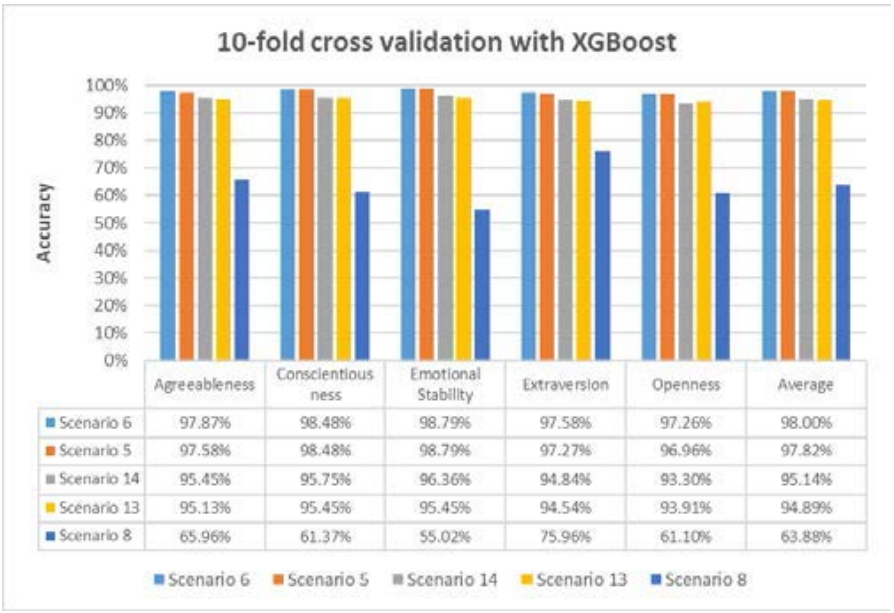
Tabel 2.17 Tabel skenario untuk tahap evaluasi (Sumber: Ong et al., 2017)



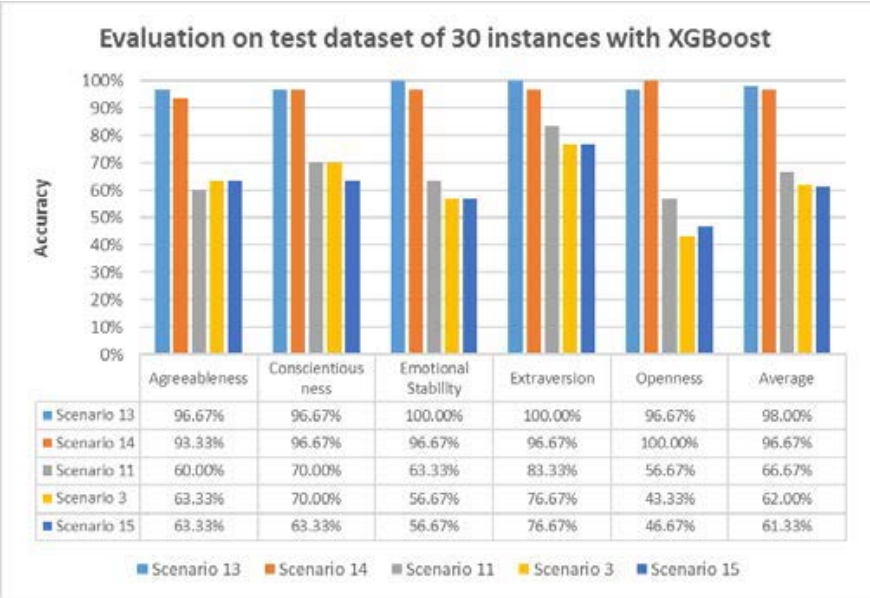
Gambar 2.8 Hasil akurasi SVM dengan *10-fold cross validation* (Sumber: Ong et al., 2017)



Gambar 2.9 Hasil akurasi SVM dengan menggunakan dataset *testing* (Sumber: Ong et al., 2017)



Gambar 2.10 Hasil akurasi XGBoost dengan menggunakan *10-fold cross validation* (Sumber: Ong et al., 2017)



	Agreeableness	Conscientiousness	Emotional Stability	Extraversion	Openness	Average
Scenario 13	96.67%	96.67%	100.00%	100.00%	96.67%	98.00%
Scenario 14	93.33%	96.67%	96.67%	96.67%	100.00%	96.67%
Scenario 11	60.00%	70.00%	63.33%	83.33%	56.67%	66.67%
Scenario 3	63.33%	70.00%	56.67%	76.67%	43.33%	62.00%
Scenario 15	63.33%	63.33%	56.67%	76.67%	46.67%	61.33%

Gambar 2.11 Hasil akurasi XGBoost dengan menggunakan dataset *testing* (Sumber: Ong et al., 2017)

Dari penelitian dan hasil akurasi pada gambar diatas menunjukkan bahwa evaluasi dengan menggunakan *10-fold cross validation* pada SVM menghasilkan akurasi tertinggi dengan 76.2310%, sedangkan XGBoost mendapatkan 97.9962%. Hasil evaluasi juga menunjukkan bahwa penggunaan fitur topik LDA dan frekuensi TF dapat meningkatkan akurasi sitem prediksi.

Hasil akhir dari penelitian menunjukkan penggunaan XGBoost pada sistem jauh lebih baik daripada menggunakan *Support Vector Machine*.

2.2.6 Personality Traits Recognition on Social Network – Facebook

Penelitian ini dilakukan oleh Alam, Stepanov, dan Riccardi pada tahun 2013. Untuk melakukan suatu interaksi sosial, diperlukan untuk mengetahui perilaku atau kebiasaan seseorang. Kepribadian adalah salah satu aspek fundamental, dimana dengan kepribadian kita dapat memahami perilaku orang tersebut. Telah terbukti bahwa ada korelasi yang kuat antara kepribadian seorang pengguna dengan cara pengguna tersebut berperilaku di sosial media seperti Facebook.

Kepribadian adalah hal yang paling kompleks dari semua atribut seorang manusia dan juga merupakan suatu keunikan tersendiri setiap manusia. Hal ini sudah menjadi tujuan jangka panjang dari para psikolog sejak lama untuk memahami kepribadian manusia dan dampaknya terhadap tingkah laku manusia tersebut. Banyak teori mengenai jenis kepribadian, namun model yang paling banyak digunakan saat ini adalah *Big five Model Personality* yang terbagi menjadi:

- O (*Openness*): Artistik, rasa ingin tahu, imajinatif, dsb.
- C (*Conscientiousness*): Efisien, terorganisir, dsb.
- E (*Extraversion*): Enerjik, aktif, asertif, dsb.
- A (*Agreeableness*): Belas kasih, kooperatif, dsb.
- N (*Neuroticism*): Kecemasan, tegang, dsb.

Dataset yang digunakan pada penelitian ini adalah 250 dataset yang disediakan oleh my*Personality* (Celli et al., 2013) pada *workshop* "*Workshop on Computational Personality Recognition* (*Shared Task*)". Pada workshop ini disarankan agar peneliti memisahkan data *training* sebesar 66% dan data *testing* 34%. Peneliti mengikuti saran tersebut dan memisahkan dataset berdasarkan pembagian diatas. Distribusi dataset berdasarkan kepribadian *Big five* dapat dilihat pada Tabel 2.18.

Fitur yang digunakan oleh peneliti adalah *open-vocabulary* dengan menggunakan '*string to word vector*' weka untuk mengubah data teks menjadi fitur vektor menggunakan TF-IDF (Manning,

2008) sebagai nilai fitur. Sedangkan untuk *classifier* yang digunakan adalah SMO (*Sequential Minimal Optimization for Support Vector Machine*), *Bayesian Logistic Regression* (BLR) dan *Multinomial Naïve Bayes* (MNB).

Cl	**Train-set**		**Test-set**	
	Y (%)	**N (%)**	**Y (%)**	**N (%)**
O	4863(74.3)	1682(25.7)	2507(74.3)	865(25.7)
C	3032(46.3)	3513 (53.7)	1524(45.2)	1848(54.8)
E	2784(42.5)	3761 (57.5)	1426(42.3)	1946(57.7)
A	3506(53.6)	3039 (46.4)	1762(52.3)	1610(47.7)
N	2449(37.4)	4096 (62.6)	1268(37.6)	2104(62.4)

Tabel 2.18 Tabel Distribusi dataset my*Personality*
(Sumber: Alam, Stepanov & Riccardi, 2013)

Class	**Pre-Avg**	**Re-Avg**	**F1-Avg**	**Acc**	**Chance (%)**
O	57.46	58.28	57.68	65.84	61.78
C	58.02	58.09	57.99	58.16	50.36
E	57.47	57.57	57.49	58.21	51.05
A	58.40	58.41	58.40	58.45	50.10
N	56.89	56.99	56.92	59.25	52.94
Mean	**57.65**	**57.87**	**57.70**	**59.98**	**53.25**

Tabel 2.19 Tabel Hasil SMO (Sumber: Alam, Stepanov & Riccardi, 2013)

Pada Tabel 2.19, *chance* (%) adalah akurasi yang dihitung secara dengan melabelkan secara acak dataset yang telah didistribusikan sebelumnya. Penghitungan dilakukan 100 kali dengan *seed* (1-100). Hasil penggunaan BLR dan MNB dapat dilihat pada Tabel 2.20 dan Tabel 2.21 secara berurutan. Sebagai tambahan untuk evaluasi, peneliti melakukan *10-fold cross validation* terhadap algoritma *Multinomial Naïve Bayes* (MNB) yang dapat dilihat pada Tabel 2.22.

Dari penelitian ini, didapatkan bahwa *classifier* MNB bekerja lebih baik dari *classifier* SMO dan *classifier* BLR.

Class	Pre-Avg	Re-Avg	F1-Avg	Acc
O	55.03	55.86	55.02	62.57
C	56.99	57.06	56.90	57.00
E	56.06	56.17	56.02	56.58
A	57.79	57.71	57.68	57.95
N	55.38	55.52	55.41	57.59
Mean	**56.25**	**56.46**	**56.21**	**58.34**

Tabel 2.20 Tabel Hasil BLR

(Sumber: Alam, Stepanov & Riccardi, 2013)

Class	Pre-Avg	Re-Avg	F1-Avg	Acc
O	59.83	59.71	59.77	69.48
C	59.06	59.11	59.07	59.34
E	57.99	58.13	57.98	58.57
A	59.09	58.71	58.49	59.16
N	58.84	57.90	57.95	62.40
Mean	**58.96**	**58.71**	**58.65**	**61.79**

Tabel 2.21 Tabel Hasil *Multinomial Naïve Bayes*

(Sumber: Alam, Stepanov & Riccardi, 2013)

Class	Pre-Avg	Re-Avg	F1-Avg	Acc
O	58.6±1.6	58.4±1.4	58.4±1.5	68.5±1.7
C	59.2±1.4	59.2±1.3	59.2±1.3	59.4±1.4
E	58.2±1.6	58.3±1.6	58.1±1.6	58.6±1.5
A	57.2±1.6	56.9±1.5	56.7±1.5	57.6±1.5
N	59.6±2.1	58.5±1.7	58.6±1.7	63.0±1.9
Overall	**58.5±0.9**	**58.3±0.8**	**58.2±0.9**	**61.4±4.5**

Tabel 2.22 Tabel Hasil *Multinomial Naïve Bayes* dengan *10-fold cross validation* (Sumber: Alam, Stepanov & Riccardi, 2013)

2.2.7 Deep Learning-Based Document Modeling for Personality Detection from Text

Penelitian ini dilakukan oleh Majumder, Poria, Gelbukh, & Cambria pada tahun 2017. Kepribadian adalah sebuah kombinasi dari

tingkah laku, emosi, motivasi, dan pola pikir individu. Kepribadian kita memiliki dampak yang besar di dalam kehidupan kita, mempengaruhi pilihan hidup, kesejahteraan, kesehatan, dan banyak pilihan lainnya. Prediksi otomatis ciri kepribadian seseorang memiliki banyak penerapan yang penting. Dalam konteks sentimen analisis (Cambria, 2016), misalnya, produk dan layanan yang direkomendasikan seseorang seharusnya dievaluasi secara positif oleh pengguna lain dengan tipe kepribadian yang serupa. Deteksi kepribadian juga bisa dimanfaatkan untuk polaritas kata disambiguasi dalam leksikon sentimen (Cambria, 2016). Karena konsep yang sama dapat menyampaikan polaritas yang berbeda terhadap tipe orang yang berbeda. Dalam diagnosis kesehatan mental, diagnosis tertentu berkorelasi dengan ciri kepribadian tertentu. Dalam forensik, mengetahui ciri kepribadian membantu mengurangi siklus prasangka. Dalam manajemen sumber daya manusia, ciri kepribadian mempengaruhi kesesuaian seseorang terhadap pekerjaan tertentu.

Kepribadian biasanya digambarkan secara formal dalam kaitannya dengan *Big five Personality Traits* (Digman, 1990), yang merupakan nilai biner (ya/tidak):

- *Extraversion* (EXT). Apakah orang itu ramah, banyak bicara, dan energik atau pendiam dan suka menyendiri?
- *Neuroticism* (NEU). Apakah orang itu sensitif dan gugup atau kukuh dan percaya diri?
- *Agreeableness* (AGR). Apakah orang itu dapat dipercaya, terus terang, dermawan, dan sopan atau tidak bisa dipercaya, rumit, dan sombong?
- *Conscientiousness* (CON). Apakah orang itu efisien dan terorganisir atau ceroboh?
- *Openness* (OPN). Apakah orang itu pandai menciptakan sesuatu hal dan memiliki rasa ingin tahu atau terima apa adanya dan waspada?

Metode yang peneliti gunakan meliputi *input data preprocessing* dan *filtering*, *feature extraction*, dan *classification*. Peneliti menggunakan dua jenis *feature*: jumlah *document-level stylistic features* yang sama,

semantic features per-kata yang digabungkan menjadi representasi panjang variabel dari *text input.* Representasi panjang variabel ini dimasukkan ke dalam CNN, di mana diproses secara hirarkis dengan menggabungkan kata-kata menjadi *n-grams, n-grams* menjadi kalimat, dan kalimat menjadi keseluruhan dokumen. Nilai yang diperoleh kemudian digabungkan dengan *document-level stylistic features* untuk membentuk representasi dokumen yang akan digunakan untuk klasifikasi terakhir.

Untuk metode yang lebih rinci akan dijelaskan di bawah:

- *Preprocessing*. Meliputi pemisahan kalimat serta pembersihan data seperti merubah seluruh kalimat menjadi huruf kecil.
- *Document-level feature extraction*. Peneliti menggunakan *Mairesse baseline feature set,* yang mencakup *global features* seperti *word count* dan panjang rata-rata kalimat.
- *Filtering*. Beberapa kalimat dalam sebuah esai mungkin tidak membawa petunjuk kepribadian apapun. Kalimat tersebut dapat diabaikan dalam *semantic feature extraction* untuk dua alasan: pertama, kalimat tersebut dapat menjadi *noise* yang dapat menggangu kinerja klasifikasi, dan kedua, penghapusan kalimat tersebut dapat mengurangi ukuran *input* dan juga waktu *training*, tanpa mempengaruhi hasil secara negatif.
- *Word-level feature extraction*. Peneliti menggunakan *word embedding*, secara khusus menggunakan *word*2vec *embeddings* (5).
- *Classification*. Untuk klasifikasi, peneliti menggunakan *deep* CNN yang terdiri dari 7 *layers*: *input* (*word vectorization*), *convolution* (*sentence vectorization*), *max pooling* (*sentence vectorization*), *1-max pooling* (*document vectorization*), *concatenation* (*document vectorization*), *linear with Sigmoid activation* (*classification*), dan *two-neuron softmax output* (*classification*).

Peneliti menggunakan *stream-of-consciousness essay dataset* yang disediakan oleh James Pennebaker dan Laura King. *Dataset* tersebut terdiri dari 2.468 esai tanpa nama yang telah diberi label kepribadian dari

penulis: EXT, NEU, AGR, CON, dan OPN. Peneliti menghilangkan satu esai dari *dataset* yang hanya memiliki kalimat “Err:508,”.

Peneliti menggunakan 10-*fold cross-validation* untuk melakukan evaluasi terhadap hasil *training*. Pada Tabel 2.23 dapat dilihat akurasi tertinggi pada *traits Extraversion* adalah 58.09%, untuk *traits Neuroticism* 59.38%, *traits Agreeableness* 56.71%, *traits Conscientiousness* 57.30%, dan untuk *traits Openness* 62.68%. Semua *traits* menggunakan arsitektur CNN+Mairesse.

Document vector *d*	Filter	Classifier	Convolution filter	*Personality traits*				
				EXT	NEU	AGR	CON	OPN
N/A	N/A	Majority	N/A	51.72	50.02	53.10	50.79	51.52
Word n-grams	Not used	SVM	N/A	51.72	50.26	53.10	50.79	51.52
Mairesse[12]	N/A	SVM	N/A	55.13	58.09	55.35	55.28	59.57
Mairesse (our experiments)	N/A	SVM	N/A	55.82	58.74	55.70	55.25	60.40
Published state of the art per *trait*[12]	N/A	N/A	N/A	56.45	58.33	56.03	**56.73**	60.68
CNN	N/A	MLP	1, 2, 3	55.43	55.08	54.51	54.28	61.03
CNN	N/A	MLP	2, 3, 4	55.73	55.80	55.36	55.69	61.73
CNN	N/A	SVM	2, 3, 4	54.42	55.47	55.13	54.60	59.15
CNN + Mairesse	N/A	MLP	1, 2, 3	54.15	57.58	54.64	55.73	61.79
CNN + Mairesse	N/A	SVM	1, 2, 3	55.06	56.74	53.56	56.05	59.51
CNN + Mairesse	N/A	sMLP/FC	1, 2, 3	54.61	57.81	55.84	**57.30**	62.13
CNN + Mairesse	Used	sMLP/MP	1, 2, 3	**58.09**	57.33	**56.71**	56.71	61.13
CNN + Mairesse	Used	MLP	1, 2, 3	55.54	58.42	55.40	56.30	**62.68**
CNN + Mairesse	Used	SVM	1, 2, 3	55.65	55.57	52.40	55.05	58.92
CNN + Mairesse	Used	MLP	2, 3, 4	55.07	**59.38**	55.08	55.14	60.51
CNN + Mairesse	Used	SVM	2, 3, 4	56.41	55.61	54.79	55.69	61.52
CNN + Mairesse	Used	MLP	3, 4, 5	55.38	58.04	55.39	56.49	61.14
CNN + Mairesse	Used	SVM	3, 4, 5	56.06	55.96	54.16	55.47	60.67

*Bold indicates the best result for each *trait*.

Tabel 2.23 Hasil Akurasi setiap *Traits* dari berbagai konfigurasi (Sumber: Majumder et al., 2017)