

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian sebelumnya [20], membahas permasalahan dimana ketika anotasi dilakukan secara manual oleh pakar (manusia) dapat menimbulkan makna bias, membutuhkan waktu yang lama, dan juga memerlukan biaya yang mahal. Berdasarkan permasalahan tersebut, sehingga dilakukan penelitian ini dengan tujuan dapat membangun pemodelan anotasi teks dengan Semi-Supervised Annotation (SSA). Pada penelitian [20], digunakan vektorisasi teks menggunakan BoW. Sedangkan pelabelan menggunakan *pseudo-labeling*, yaitu 20% data diberikan label secara manual oleh pakar kemudian data tersebut *training* menggunakan Logistic Regression dan Naïve Bayes sehingga menghasilkan sebuah model. Kemudian 80% data sisanya dianotasi menggunakan model yang dibangun berdasarkan 20% data tersebut. Model SSA terbaik dipilih dari model yang memiliki tingkat akurasi tertinggi dari beberapa kali percobaan dengan *threshold* yang berbeda. Kinerja SSA dilihat dari *F1 score* dan akurasi dari metode Confusion Matrix. *Dataset* yang digunakan adalah *dataset* IMDB yang terdiri dari 50.000 *dataset*. Penelitian ini memberikan hasil bahwa diketahui nilai akurasi SSA bergantung pada jumlah *data train* berlabel (*annotated dataset*), dan kecocokan *dataset* dengan algoritma *Machine Learning* yang digunakan. Diantara model *Machine Learning* Logistic Regression dan Naive Bayes, ketika diujikan pada *dataset* IMDB, Logistic Regression memiliki hasil akurasi dan *F1 score* lebih tinggi daripada Naive Bayes.

Dalam penelitian [21] dijelaskan bahwa NLP menggunakan Bahasa Arab masih jarang dilakukan penelitian, sehingga kamus kata atau corpus yang tersedia masih belum cukup untuk melakukan pembelajaran terawasi (*supervised learning*). Pembelajaran terawasi membutuhkan pelabelan data dalam jumlah besar, yang sering kali membutuhkan waktu lama, dan memerlukan biaya yang mahal untuk memberikan label secara manual oleh pakar (manusia). Selain itu, pelabelan manual dapat menimbulkan bias manusia pada model. Pelabelan

manual tidak disarankan ketika mengolah *dataset* yang besar. Penelitian ini bertujuan untuk mengembangkan sistem pendeteksi komentar rumor pada Twitter, dengan spesifikasi Bahasa Arab menggunakan Semi-Supervised Expectation-Maximization (E-M). Hasil penelitian yang diperoleh adalah dengan menggunakan sistem ini dapat mengadaptasi untuk bahasa lain. E-M Semi-Supervised mendapatkan akurasi yang lebih baik daripada menggunakan Gaussian Naïve Bayes classifier sebagai dasar, dengan ukuran set pelatihan hingga 60% dari set pelatihan penuh yang mencapai skor F1 sebesar 78,6%. Secara umum, ketika data cocok dengan asumsi model campuran, E-M dapat membuat perbedaan dengan sejumlah kecil data berlabel dan sejumlah besar data tidak berlabel. Namun, seiring bertambahnya ukuran data berlabel, E- Performa M menurun.

Selanjutnya pada penelitian [22] bertujuan untuk menerapkan metode untuk memberikan pelabelan sentimen berdasarkan kata kunci dengan tetap memperhatikan konteks kalimat. Penelitian tersebut dilatarbelakangi oleh penggunaan metode Lexicon dalam pelabelan *dataset* hanya memberikan label berdasarkan makna setiap kata penyusunnya. Sehingga tidak dapat mengetahui arah konteks komentar, dan kurang dapat efektif untuk mengatasi kalimat sarkas. Metode penelitian yang diusulkan oleh peneliti menggunakan metode TF-IDF untuk word embedding, dan algoritma LSI/LSA untuk pemaknaan kata terhadap konteks kalimat. Pada hasil penelitian mengkombinasikan metode menggunakan TF-IDF dan LSA mampu menemukan detail permasalahan. Contohnya kata "vaksin" di TF-IDF menempati ranking pertama positif, negatif, maupun netral dengan masing-masing memiliki bobot 0.69, 0.77, dan 0.69.

Penelitian [23] terdapat permasalahan penelitian dimana pendekatan yang ada memiliki batasan mendasar dalam ketahanannya karena *classifier* yang buruk menyebabkan pelabelan yang tidak akurat. mengusulkan penyelesaian masalah dengan pelabelan sentimen *Semi-Supervised Learning* menggunakan *lexicon*. Kemudian menghitung *confidence score* data tanpa label menggunakan *threshold* yang telah ditentukan. Apabila data memiliki *confidence score* yang tinggi maka data tersebut akan ditambahkan dalam *dataset* berlabel untuk mendapatkan label semu. Dalam penelitian ini dengan pendekatan berbasis XAI, skor diberikan pada

kata per kalimat. Artinya, dalam proses membangun leksikon, skor dihitung dan diperbarui satu per satu. Penelitian membandingkan akurasi antarmetode yang digunakan. Didapatkan hasil bahwa Gam-CAM memiliki kemampuan klasifikasi lebih baik daripada 5 algoritma klasifikasi lainnya.

Kemudian pada penelitian [24] dijelaskan bahwa penggunaan *Deep Learning* memberikan hasil yang baik ketika menggunakan data label dalam jumlah besar. Tetapi, memberikan label pada data jumlah besar menghabiskan waktu dan biaya yang mahal. Sehingga, pada penelitian ini diusulkan penyelesaian permasalahan penelitian dengan menggunakan algoritma *Deep Learning* sebagai *autoencoder* berbasis graf, yaitu Graph Regularized Variational Ladder Networks (GRVLN). Digunakan 4 *dataset* pada penelitian ini, yaitu MNIST, CIFAR-10, SVHN, dan STL-10. Hasil penelitian dijelaskan bahwa pada data MNIST dengan data latih 60000 dan data uji 10000, GRVLN memberikan hasil *error rate* paling rendah dengan 0.77 ± 0.023 . Data CIFAR-10 dengan data latih 50000 dan data uji 10000 memiliki *error rate* 16.13 ± 0.35 , Data SVHN dengan data latih 73257 dan data uji 26032 memiliki *error rate* 6.98 ± 0.76 . Data STL-10 dengan data latih 5000 dan data uji 100000 memiliki *error rate* 22.39 ± 0.074 .

Penelitian [25] menggunakan metode pelabelan manual, dengan pakar sebagai manusia yang menentukan label dalam korpus. Tetapi, untuk membuat korpus berlabel lengkap dengan kualitas tinggi memerlukan banyak usaha, waktu, dan biaya, serta dapat menjadi tugas yang berat. Tujuan dilakukan penelitian ini adalah untuk mengetahui bagaimana kombinasi antara TF-IDF dengan Random Forest untuk meningkatkan akurasi ketika digunakan pada SSL. Penelitian ini menjabarkan hasil penelitian bahwa pada Data1 dengan jumlah kelas data sebanyak 3, Random Forest memiliki *F1-score* 0.65 sedangkan Naive Bayes 0.62. Hal ini menunjukkan bahwa RF bekerja lebih baik daripada NB pada data 3 kelas. Sedangkan pada Data2 yang memiliki 2 kelas, NB memiliki *F1-score* 0.76 sedangkan RF 0.71. Hal ini menunjukkan bahwa NB bekerja lebih baik daripada RF pada data 2 kelas. Sehingga kesimpulannya, kinerja model SSL yang

digunakan bergantung pada jumlah data latih dan kesesuaian pola dalam dokumen dengan mesin.

Selanjutnya, penelitian [26] terdapat permasalahan penelitian yaitu deteksi ujaran kebencian dapat dilakukan dengan konsep analisis sentimen menggunakan anotasi teks. Untuk menghindari subjektivitas maka harus dilakukan oleh ahli bidang hate speech. Tentunya proses pelabelan akan membutuhkan waktu yang lama, dan memungkinkan terjadinya bias data. Sehingga diusulkan sebuah solusi penyelesaian dengan membuat model anotasi teks *Semi-Supervised Learning* secara otomatis menggunakan K-Nearest Neighbor (KNN). Penelitian ini dilakukan dengan alur: data yang telah ditentukan (data label dan non-label), dilakukan vektorisasi menggunakan VboW dan VTFIDF, lalu membuat pemodelan algoritma KNN. Hasil vektor dan *training* algoritma akan digunakan untuk anotasi otomatis pada data non-label. Dilakukan pembobotan dari model vector dan KNN kemudian dihitung skor polaritas. Jika skor melebihi ambang batas (*threshold*), maka akan masuk ke *data training*. Sedangkan jika kurang, akan dianotasi ulang. Hasil akurasi KNN sebesar 57.25%, dan setelah dilakukan vektorisasi TFIDF dapat meningkatkan akurasi sebesar 2,43%. Sehingga akurasi akhir yang didapatkan adalah 59.68%.

Permasalahan penelitian pada [27] adalah kumpulan data yang berlabel umumnya dibutuhkan dalam penelitian analisis sentimen. Namun, untuk memperoleh data yang keseluruhan berlabel membutuhkan usaha dan biaya yang lebih besar. Diperlukan penyelesaian permasalahan yaitu dengan diusulkannya solusi membuat sebuah pemodelan *Semi-Supervised Learning* (SSL Model) untuk analisis sentimen menggunakan pendekatan Ensemble. Metode yang digunakan adalah TFIDF sebagai vectorizer, dan algoritma SVM dan Random Forest/RF (sebagai Ensemble Classifier). Alur penelitian yang pertama *dataset* dilatih menggunakan TFIDF, yang kemudian memberikan hasil tokenisasi vektor unigram, bigram, trigram. Performa dihitung *F1 score*. Setiap model vector menghasilkan pseudo-label. Penggunaan *threshold* antara 0,6-0,9 untuk menghasilkan label yang lebih akurat. Data lebih tinggi dari *threshold* masuk ke *data training*, sedangkan yang lebih rendah akan dianotasi kembali (masuk

unlabeled data). Iterasi dilakukan sampai data *unlabeled* habis. Hasil akurasi kedua algoritma menurun ketika diterapkan dalam SSL. Rata-rata diff baseline dan SSL pada SVM -0.02, dan RF -0.03. nilai rata-rata diff F1 *score* SVM -0.01, dan RF -0.03. artinya, algoritma SVM lebih baik dalam mempertahankan akurasi SSL dibandingkan RF.

Selanjutnya pada penelitian [28] dijelaskan bahwa untuk mengurangi dampak meningkatnya spam opini, diusulkan pendekatan deteksi spam opini, yang sebagian besar mengadopsi metode klasifikasi Supervised Learning. Namun, dalam praktiknya, data yang diberikan sebagian besar tidak diberi label dan oleh karena itu diperlukan pendekatan pembelajaran *Semi-Supervised Learning*. Menggunakan 2 macam *dataset*, yaitu *dataset review* hotel dan *dataset Yelp review*. Metode klasifikasi yang diusulkan adalah Naïve bayes, kombinasi SVM dan *self-training (co-training)*, TVSM, *label propagation*. Berdasarkan metode yang diusulkan dalam penelitian, pada metode naïve bayes, dengan bigram dan 100% data label memiliki akurasi 0.93. *self-training* dengan bigram dan 20% data label memiliki akurasi 0.93. *co-training* SVM dengan TFIDF+ling.feature dan 20% data label memiliki akurasi 0.88. TSVM dengan TFIDF+bigram dan 20% data label memiliki akurasi 0.83. *Label spreading* dengan TFIDF+bigram dan 20% data label memiliki akurasi 0.65.

Menurut penelitian [29] menyebutkan bahwa metode berbasis lexicon tidak memerlukan data berlabel, sehingga menghasilkan akurasi rendah karena berbagai alasan seperti masalah cakupan leksikon, mempertimbangkan polaritas tingkat kata tanpa konteks, dan yang lainnya. Oleh karena itu, maka diusulkan penelitian dengan tujuan meningkatkan performa klasifikasi sentimen (SSentiA) dari data yang tidak memiliki label. pada 4 *dataset* biner (TripAdvisor, IMDB, Amazon, Clothing) hasil LRSentiA lebih baik daripada SentiStrength dan TextBlob. Sedangkan, pada 2 *dataset* 3-class LRSentiA menghasilkan F1score tertinggi di *dataset* TripAdvisor (0.74), dan akurasi tertinggi pada *dataset* Amazon (0.795). LRSentiA menghasilkan *pseudo-label* yang dapat digunakan untuk meningkatkan performa model *Semi-Supervised Learning*. Uji korelasi antara skor kepercayaan (mudah dibedakan antar kata) dan akurasi menggunakan Chi-Square.

Tabel 2. 1 Penelitian Sebelumnya

No	Judul Nama, Terbit	Jurnal, Tahun	Tujuan Penelitian	Metode	Hasil Penelitian	Perbedaan
1	Model Semi-Supervised Learning Menggunakan Logistic Regression Untuk Anotasi Sentimen, Siti Khomsah, Agus Sasmito Aribowo. 2022 [20]		Membangun model Semi-Supervised Learning untuk anotasi sentiment (Semi-Supervised Annotation /SSA)	<i>Dataset: dataset</i> IMDB vektorisasi: BoW klasifikasi: Logistic Regression dan Naïve Bayes. pengukuran kinerja: Confusion Matrix pada parameter F1. Pelabelan sentimen (SSL) menggunakan algoritma <i>Machine Learning</i> Logistic Regression	Penelitian ini memiliki kesimpulan bahwa akurasi SSA bergantung pada jumlah data train berlabel (<i>annotated dataset</i>), dan kecocokan <i>dataset</i> dengan algoritma <i>Machine Learning</i> yang digunakan. Diantara model <i>Machine Learning</i> Logistic Regression dan Naive Bayes, ketika diujikan pada <i>dataset</i> IMDB, Logistic Regression memiliki hasil akurasi dan F1 lebih tinggi daripada Naive Bayes.	Penelitian sebelumnya menggunakan <i>dataset</i> IMDB dan pelabelan sentimen (SSL) menggunakan algoritma <i>Machine Learning</i> Logistic Regression dan Naïve Bayes. Sedangkan penelitian yang peneliti lakukan menggunakan 3 macam <i>dataset</i> dan pelabelan sentimen (SSL) menggunakan algoritma <i>Deep Learning</i> LSTM dan GRU, dengan Word2Vec model CBoW.
2	Rumor detection in Arabic tweets using		mengembangkan sistem pendeteksi	<i>Dataset: 271000</i> tweet, terdiri dari 89	Hasil akhir: Semi-Supervised Learning mencapai hasil terbaik saat	Penelitian sebelumnya memodelkan SSL berbasis

No	Judul Nama, Terbit	Jurnal, Tahun	Tujuan Penelitian	Metode	Hasil Penelitian	Perbedaan
	Supervised Learning and Unsupervised Expectation- Maximization, Samah M. Alzanin, Aqil M. Azmi, 2019 [21]		komentar rumor pada Twitter, dengan spesifikasi Bahasa Arab menggunakan Semi-Supervised Expectation- Maximization (E- M)	event rumor, dan 88 event non-rumor. Metode: menggunakan Semi- Supervised Expectation- Maximization (E-M)	menggunakan 60% data latih yaitu dengan akurasi dan F1 <i>score</i> 78.6%. Sedangkan Supervised Learning Gaussian NB mencapai hasil terbaik saat menggunakan 100% data latih akurasi dan F1 <i>score</i> 82% dan 81,8%. Ketika ukuran data label meningkat, performa E-M menurun.	Naïve Bayes tujuan spesifik mendeteksi rumor dengan Bahasa arab, sedangkan penelitian yang peneliti lakukan memodelkan SSL dengan LSTM dan GRU, <i>dataset</i> yang digunakan sebanyak 3 <i>dataset</i> .
3	Text Mining: Analisis Teks Terkait Isu Vaksinasi COVID- 19, Novita Anggraini, Edi Surya Negara Harahap, Tri Basuki		Mengetahui metode untuk memberikan pelabelan sentimen berdasarkan kata kunci dengan tetap memperhatikan konteks kalimat.	<i>Dataset:</i> crawling komentar Instagram pada akun @kemenkes_ri. Pembobotan kata : TF-IDF. Test model: Bag of Word.	Untuk mendapatkan kata kunci, menggunakan metode trigram, kemudian kata kunci masuk Ke Bag Of Word dan akan dilakukan vektorisasi. Setelah tahap BoW, setiap kata diranking di tahap text modeling (TF-IDF). untuk mengetahui akurasi menggunakan algoritma <i>Machine</i>	Penelitian sebelumnya menggunakan metode pelabelan dengan TF-IDF dan menemukan detail permasalahan terhadap kata menggunakan LSA, sedangkan penelitian yang peneliti lakukan bertujuan memberi label data

No	Judul Nama, Terbit	Jurnal, Tahun	Tujuan Penelitian	Metode	Hasil Penelitian	Perbedaan
	Kurniawan, [22]	2021		Akurasi: Support Vector Machine, Naive Bayes, dan Random Forest Classifier. Analisis teks: algoritma LSI/LSA.	<i>Learning</i> Support Vector Machine, Naïve Bayes, dan Random Forest Classifier. Ketika kedua metode TFIDF dikombinasikan dengan algoritma LSA, maka akan diperoleh kata kunci yang merujuk pada isu melalui TF-IDF, dan detail permasalahan dari kata kunci melalui algoritma LSA.	menggunakan Semi-Supervised Learning.
4	Semi-Supervised Learning based on Auto-generated Lexicon using XAI in Sentiment Analysis. Hohyun Hwang, Younghoon Lee. 2021 [23]		Merancang model pelabelan semu dengan pendekatan <i>Lexicon</i> menggunakan XAI	<i>Dataset:</i> Airline, Amazon, Clothing, Hotel, IMDB, Steam, Yelp review Metode: <i>Explainable</i> AI (XAI). Algoritma klasifikasi: LR, SVM, LIME, SHAP,	Dalam pendekatan berbasis XAI, skor diberikan pada kata per kalimat. Artinya, dalam proses membangun leksikon, skor dihitung dan diperbarui satu per satu. Penelitian membandingkan akurasi antarmetode yang digunakan. Didapatkan hasil bahwa Gam-CAM memiliki	Penelitian sebelumnya menggunakan <i>dataset</i> berupa 7 (tujuh) set data, klasifikasi dengan 6 algoritma, konsep pelabelan menggunakan lexicon. Sedangkan penelitian yang peneliti lakukan menggunakan <i>dataset</i> 3 (tiga)

No	Judul Nama, Terbit	Jurnal, Tahun	Tujuan Penelitian	Metode	Hasil Penelitian	Perbedaan
				Grad-CAM, LPR. Metode <i>Word-embedding</i> TF-IDF dan Word2Vec	kemampuan klasifikasi lebih baik daripada 5 algoritma klasifikasi lainnya.	macam, dan konsep klasifikasi dengan algoritma LSTM dan GRU.
5	Graph Regularized Variational Ladder Networks (GRVLN) for Semi-Supervised Learning, CONG HU dan XIAO-NING SONG, 2020 [24]		Membuat pemodelan autoencoder/AE (gabungan regulasi graph Laplacian dan variational autoencoder) untuk meningkatkan kinerja Semi-Supervised Learning.	<i>Dataset:</i> MNIST <i>dataset</i> , CIFAR-10 <i>dataset</i> , SVHN <i>dataset</i> , dan STL-10. Klasifikasi: Support vector machine (SVM), Transductive Support Vector Machine (TSVM), Variational AutoEncoding(VAE), EmbedCNN, SWWAE, S3C,	Pada data MNIST dengan data latih 60000 dan data uji 10000, GRVLN memberikan hasil error rate paling rendah dengan 0.77 ± 0.023 . Data CIFAR-10 dengan data latih 50000 dan data uji 10000 memiliki <i>error rate</i> 16.13 ± 0.35 , Data SVHN dengan data latih 73257 dan data uji 26032 memiliki error rate 6.98 ± 0.76 . Data STL-10 dengan data latih 5000 dan data uji 100000 memiliki error rate 22.39 ± 0.074 . Penelitian ini menginisiasi auto-	Penelitian sebelumnya menggunakan algoritma klasifikasi Support Vector Machine (SVM), Transductive Support Vector Machine (TSVM), Variational autoEncoding (VAE), Embed CNN, SWWAE, S3C, Graph Regularized Deep Neural Network (GR-DNN), Modifikasi Ladder network (LN), sedangkan penelitian yang peneliti lakukan

No	Judul Nama, Terbit	Jurnal, Tahun	Tujuan Penelitian	Metode	Hasil Penelitian	Perbedaan
				Graph regularized Deep Neural Network (GR-DNN), Modifikasi Ladder network (LN)	encoder berbasis <i>Deep Learning</i> , Graph Regularized Variational Ladder Network (GRVLN). GRVLN memberikan hasil yang efektif dalam proses <i>training</i> dan <i>testing</i> .	menggunakan LSTM dan GRU.
6	Semi-Supervised Learning Models for Sentiment Analysis on Marketplace <i>Dataset</i> . Wisnalmawati, Agus Sasmito Aribowo, Yunie Herawati, 2022 [25]		mengetahui kinerja TFIDF dan random forest saat digunakan dalam metode Semi-Supervised Learning (SSL), dan mengetahui pengaruh banyaknya data berlabel dalam pelatihan terhadap	<i>Dataset</i> : komentar <i>review</i> pada marketplace shopee sebanyak 8523 dan 5421. Metode: <i>ensemble multi-classifier</i> yang bekerja pada unigram, bigram, dan trigram. Klasifikasi: random forest dan naïve bayes	Pada Market Data 1, <i>dataset</i> dengan tiga kelas, sebaiknya menggunakan Random Forest (F1-score RF 0,65, dan 0,62 untuk NB). Pada <i>dataset</i> Market Data 2 yang terdiri dari dua kelas, lebih baik menggunakan Naïve Bayes (skor F1 untuk RF 0,71, dan 0,76 untuk NB)	Penelitian sebelumnya menggunakan algoritma Random Forest dan Naïve Bayes, sedangkan penelitian yang peneliti lakukan menggunakan algoritma LSTM dan GRU.

No	Judul Jurnal, Nama, Tahun Terbit	Tujuan Penelitian	Metode	Hasil Penelitian	Perbedaan
		akurasi model SSL			
7	Semi-Supervised Text Annotation for Hate Speech Detection using K-Nearest Neighbors and Term Frequency-Inverse Document Frequency. Nur Heri Cahyana, Shoffan Saifullah, Yuli Fauziah, Agus Sasmito Aribowo, Rafal Drezewski. 2022 [26]	Membuat model anotasi teks Semi-Supervised Learning secara otomatis menggunakan K-Nearest Neighbor (KNN)	<i>Dataset: review 3</i> aplikasi pembayaran digital, total <i>dataset</i> sebanyak 11000. Vektorisasi: VBoW, VTFIDF, klasifikasi: algoritma K-NN	Hasil akurasi KNN sebesar 57.25%, dan setelah dilakukan vektorisasi TFIDF dapat meningkatkan akurasi sebesar 2,43%. Sehingga akurasi akhir yang didapatkan adalah 59.68%.	Penelitian sebelumnya menggunakan metode KNN dan TF-IDF untuk meningkatkan akurasi, sedangkan penelitian yang peneliti lakukan menggunakan LSTM dan GRU.

No	Judul Jurnal, Nama, Tahun Terbit	Tujuan Penelitian	Metode	Hasil Penelitian	Perbedaan
8	Semi-Supervised Learning for sentiment classification with ensemble multi-classifier approach. Agus Sasmito Aribowoa, Halizah Basiron, Noor Fazilla Abd Yusof. 2022 [27]	Membuat sebuah pemodelan Semi-Supervised Learning (SSL Model) untuk analisis sentimen menggunakan pendekatan Ensemble.	<i>Dataset:</i> IMDB dan US Airlines <i>dataset</i> . Vektorisasi : TF-IDF menghasilkan Unigram, Bigram, Trigram. Klasifikasi: algoritma SVM dan Random Forest/RF (sebagai Ensemble Classifier).	Hasil akurasi kedua algoritma menurun ketika diterapkan dalam SSL. Rata-rata diff baseline dan SSL pada SVM -0.02, dan RF -0.03. nilai rata-rata diff F1 score SVM -0.01, dan RF -0.03. artinya, algoritma SVM lebih baik dalam mempertahankan akurasi SSL dibandingkan RF.	Penelitian sebelumnya menggunakan <i>dataset</i> IMDB dan US Airlines, algoritma SVM, sedangkan penelitian yang akan peneliti lakukan menggunakan LSTM dan GRU.
9	Analyzing the effectiveness of Semi-Supervised Learning approaches for opinion spam	Bertujuan mengetahui efektivitas Semi-Supervised Learning (SSL) dalam mendeteksi	<i>Dataset:</i> review hotel dan Yelp <i>review dataset</i> Metode klasifikasi: Naïve Bayes, kombinasi SVM dan	Pada metode Naïve Bayes, dengan Bigram dan 100% data label memiliki akurasi 0.93. <i>self-training</i> dengan Bigram dan 20% data label memiliki akurasi 0.93. <i>co-training</i> SVM dengan TFIDF+ling.feature dan 20% data	Penelitian sebelumnya menggunakan algoritma Naïve Bayes, sedangkan penelitian yang peneliti lakukan menggunakan LSTM dan GRU. Vektorisasi yang

No	Judul Nama, Terbit	Jurnal, Tahun	Tujuan Penelitian	Metode	Hasil Penelitian	Perbedaan
	classification. Alexander Lightharta, Cagatay Catal, Bedir Tekinerdogan 2021. [28]		klasifikasi spam opini.	<i>Self-Training (co-training)</i> , TVSM, Label Propagation.	label memiliki akurasi 0.88. TSVM dengan TFIDF+Bigram dan 20% data label memiliki akurasi 0.83. Label spreading dengan TFIDF+Bigram dan 20% data label memiliki akurasi 0.65.	digunakan TFIDF, sedangkan penelitian ini menggunakan Word2Vec CboW.
10	SSentiA: A Self- Supervised Sentiment Analyzer for classification from <i>unlabeled</i> <i>data</i> . Salim Sazzed, Sampath Jayarathna. 2021. [29]		meningkatkan performa klasifikasi sentimen (SSentiA) dari data yang tidak memiliki label. SSentiA manfaatkan pelabelan semu melatih Supervised <i>Machine Learning</i>	<i>Dataset</i> : TripAdvisor, IMDB, Amazon, Clothing, dan Cornell movie <i>review</i> . Teknik anotasi: Kombinasi antara teknik manual dan otomatis. Teknik vektorisasi: BoW, TFIDF.	pada 4 <i>dataset</i> biner (TripAdvisor, IMDB, Amazon, Clothing) hasil LRSentiA lebih baik daripada SentiStrength dan TextBlob. Sedangkan, pada 2 <i>dataset</i> 3-class LRSentiA menghasilkan <i>F1score</i> tertinggi di <i>dataset</i> TripAdvisor (0.74), dan akurasi tertinggi pada <i>dataset</i> Amazon (0.795). LRSentiA meghasilkan pseudo-label yang dapat digunakan untuk meningkatkan	Penelitian sebelumnya menggunakan <i>dataset</i> sebanyak 5 macam, menggunakan algoritma SVM, dan BoW, TFIDF, Chi Square, sedangkan penelitian yang peneliti lakukan menggunakan algoritma LSTM dan GRU dengan 3 macam <i>dataset</i> .

No	Judul Nama, Terbit	Jurnal, Tahun	Tujuan Penelitian	Metode	Hasil Penelitian	Perbedaan
			untuk klasifikasi sentimen	Metode uji non-parametrik: Chi Square	performa model Semi-Supervised Learning. uji korelasi antara skor kepercayaan (mudah dibedakan antar kata) dan akurasi menggunakan Chi-Square.	

Pada penelitian sebelumnya, dapat ditarik kesimpulan bahwa data yang tercantum pada jurnal memberikan wawasan untuk penelitian yang akan dilakukan. Mencakup berbagai topik yang berhubungan dengan subjek penelitian yang akan dilakukan. Dari penelitian sebelumnya memberikan gambaran yang lebih luas tentang perkembangan pelabelan sentimen menggunakan metode Semi-Supervised Learning yang dikombinasikan baik dengan algoritma klasifikasi yang berbeda maupun teknik *word embedding* yang berbeda. Kolom hasil penelitian memberikan gambaran hasil yang telah dilakukan oleh penelitian sebelumnya. Sehingga dapat memberikan pengetahuan atau referensi baru untuk dilakukan penelitian yang akan datang.

2.2 Landasan Teori

2.2.1 *Natural Language Processing*

Merupakan salah satu bagian kecerdasan buatan yang berhubungan dengan pemrosesan bahasa dengan tujuan agar dapat dengan mudah mengerti oleh manusia dan mesin. Implementasi *Natural Language Processing* antara lain mesin untuk menerjemahkan berbagai bahasa, analisis sentimen, aplikasi asisten pribadi seperti Alexa, Siri [30]. Terdapat 3 (tiga) hal inti yang dapat memengaruhi hasil penelitian terkait NLP yaitu tahap *labeling*, *preprocessing*, dan *word embedding*.

1. Pelabelan

Merupakan salah satu tahap penting dalam penelitian klasifikasi data teks. Pelabelan sentimen adalah tahap memberikan satu identitas (kelas data) untuk mengetahui kelas dari pendapat/opini seseorang terhadap suatu isu [31]. Pengelompokan data menjadi beberapa kelas sentimen yang digunakan yaitu positif, netral, negatif [32] atau hanya menggunakan 2 kelas yaitu positif dan negatif [33]. Penetapan jumlah kelas data bergantung pada tujuan penelitian yang akan dilakukan.

2. *Preprocessing*

Dataset yang telah didapatkan dalam penambahan data teks disebut data mentah atau data yang belum diproses. Sehingga sebelum dilakukan pemrosesan lebih lanjut, data mentah harus dilakukan *preprocessing*/pemrosesan awal. *Data preprocessing* adalah tahap untuk merapikan data yang bertujuan untuk mempersiapkan *dataset* yang tidak terstruktur menjadi data terstruktur yang memudahkan dalam langkah selanjutnya [34]. *Data preprocessing* terdapat 6 tahap yaitu:

a. *Drop null*

Null dapat memengaruhi kinerja algoritma dalam melakukan proses selanjutnya. Sehingga adanya nilai kosong (null) pada suatu *dataset* harus dilakukan penanganan. Terdapat 2 (dua) cara yang dapat dilakukan, yaitu dengan menghapus null, dan mengisi null dengan nilai tertentu [35].

b. *Drop duplicate*

Nilai duplikat harus ditangani dengan benar karena dapat memengaruhi generalisasi model. Terdapat kemungkinan jika duplikat tidak ditangani dengan benar, maka data yang sama dapat muncul di kumpulan data pengujian yang juga ada di kumpulan data pelatihan [36].

c. *Cleaning data*

Merupakan proses untuk membersihkan data dari komponen tertentu, contoh URL, *username*, *hashtags* [37].

d. *Case folding*

Case folding adalah proses yang bertujuan untuk mengubah seluruh huruf dalam *dataset* menjadi huruf kecil atau *lowercase* [34], [38].

e. *Tokenizing*

Tokenizing atau tokenisasi adalah tahap pemisahan kalimat dalam *dataset* berdasarkan tiap kata penyusunnya. Kata yang telah dipisah dari rangkaian kalimat disebut *token* atau *term*. *Term* ini akan diberikan bobot kata pada tahap pembobotan kata [16].

f. *Slangword Removal*

Slangword removal atau penghapusan kata slang bertujuan untuk mengubah kata tidak baku menjadi kata baku [39]. Contoh kata slang antara lain “ngga”, “dapet”, “ngapain”. Dalam [40] dijelaskan bahwa bahasa gaul adalah penggunaan informal kosakata secara khas dan lebih bersifat metaforis, elips, jelas, dan singkat daripada bahasa biasa. Selain itu, bahasa gaul bukanlah bahasa resmi yang diakui dalam kamus. Hanya gaya bahasa yang digunakan dalam sekelompok orang tertentu.

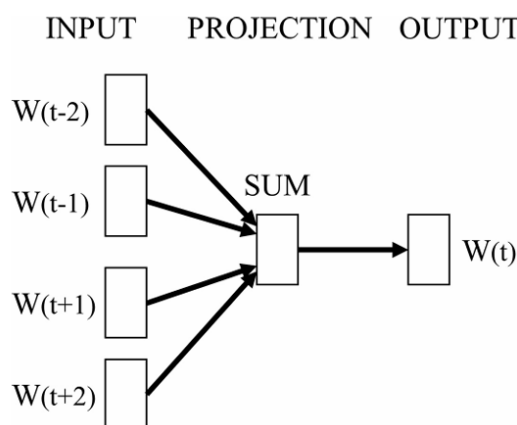
g. *Under Sampling*

Under Sampling merupakan salah satu metode yang dapat digunakan untuk mengatasi ketidakseimbangan kuantitas data antarkelas. Cara kerja *under sampling* adalah dengan membuang sampel (secara acak) dari kelas mayoritas, sehingga kuantitas dari kelas minoritas dan mayoritas akan sama [41].

3. Word2Vec

Word2Vec adalah metode yang dibuat oleh Perusahaan Google untuk merepresentasikan vektor kata. Sehingga dengan menggunakan Word2Vec dapat mengubah setiap kata ke *vector* dalam yang berisi nilai 0 dan 1 (biner), dengan nilai 1 menunjukkan letak kata dalam vektor tersebut. Metode Word2Vec sangat sistematis untuk mendeteksi hubungan antarkata [42]. Didalam Word2Vec terdapat 2 metode, yaitu CBOW (Continuous Bag Of Word) dan Skip-gram. Dalam penelitian ini akan digunakan metode CBOW (Continuous Bag Of Word). Metode Continuous Bag of Word termasuk kedalam model jaringan saraf (Neural Network) yang ditujukan untuk menemukan dan mengambil kata.

Model CBoW akan memprediksi kata yang berada di tengah dengan menggunakan kata yang berada disekitarnya. Konsep CBoW berdasarkan pemikiran bahwa kata-kata yang muncul berdekatan diharapkan memiliki arti yang mirip, dan kata-kata yang muncul berjauhan terdapat kemungkinan akan memiliki arti yang berbeda. Sehingga, diharapkan kata tengah dapat muncul dengan memperhatikan kondisi kemunculan kata-kata tetangganya. Arsitektur Continuous Bag of Word (CBOW) ditunjukkan oleh Gambar 2.1.



Gambar 2. 1 Arsitektur CBoW [43]

Pada arsitektur CBoW di atas, W merepresentasikan suatu kata yang berisi hasil nilai dari pengubahan bentuk biner (*one-hot encoding*), dan t merupakan sebelum dan sesudah kata target $W(t)$. Lapisan tersembunyi proyeksi (*projection*) diperoleh dari perkalian nilai konteks dengan matriks bobot dari

lapisan input. Hasil yang didapatkan dari perkalian tersebut dikalikan kembali dengan bobot output untuk mendapatkan vektor akhir $\mathbf{W}(t)$.

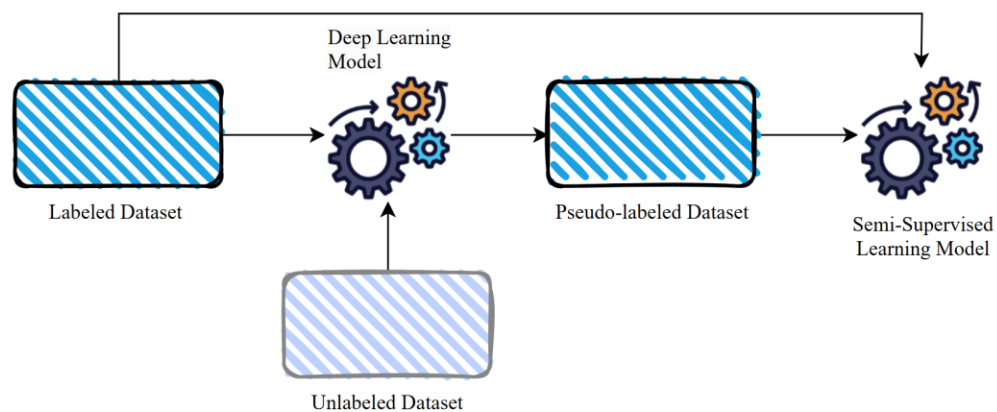
2.2.2 Analisis Sentimen

Sebuah metode yang digunakan untuk membentuk gambaran terkait opini dari proses penggalian data teks yaitu dengan menggunakan analisis sentimen. Analisis sentimen adalah suatu metode dalam memahami, mengekstraksi data sentimen yang biasanya akan dikategorikan berdasarkan polaritasnya [44]. Analisis sentimen berfokus pada analisis orientasi sentimen korpus komentar yang menunjukkan bahwa opini tersebut positif, netral, atau negatif dengan menggunakan teknik pemrosesan bahasa alami, pembelajaran mesin, temu kembali informasi, dan yang lainnya [45], [46]. Saat ini analisis sentimen memiliki dua kategori, yaitu klasifikasi berdasarkan aturan (*rule-based classification*), pembelajarannya mesin menggunakan kata-kata emosi dengan tujuan mengklasifikasikan fitur, dan pemilihan karakteristik sentimen menggunakan kamus emosi [47].

Dalam [48] dijelaskan bahwa analisis sentimen memiliki 4 (empat) tingkatan, antara lain tingkat analisis dokumen, tingkat analisis kalimat, tingkat analisis frasa, dan tingkat analisis aspek. Tingkat analisis dokumen digunakan untuk mengklasifikasikan halaman atau bab pada buku, yaitu dengan melakukan analisis dan memberikan polaritas tunggal terhadap seluruh dokumen. Pada tingkat analisis kalimat, setiap kalimat akan dianalisis dan ditentukan polaritas kalimat yang sesuai. Polaritas dalam kalimat akan ditentukan dengan metode yang sama seperti pada tingkat analisis dokumen, namun dengan data pelatihan yang lebih besar. Analisis sentimen pada tingkat frasa banyak dikembangkan dalam penelitian, salah satunya berguna dalam penilaian ulasan produk yang terdiri dari beberapa baris. Analisis sentimen tingkat aspek berfokus pada semua aspek yang digunakan dalam kalimat, dan memberikan polaritas pada semua aspek yang dihitung pada keseluruhan kalimat.

2.2.3 Semi-Supervised Learning

Ketika sebuah mesin dapat meningkatkan kinerjanya secara otomatis dengan belajar dari data yang tidak memiliki berlabel dan tanpa interaksi eksternal, disebut *Semi-Supervised Learning* [49]. Salah satu model *Semi-Supervised Learning* adalah dengan memisahkan data label dan tanpa label, kemudian menggunakan data tanpa label untuk *corpus Unsupervised Learning* dan data label untuk *corpus Supervised Learning* [50].



Gambar 2. 2 Alur *Semi-Supervised Learning* [50]

Semi-Supervised Learning (SSL) merupakan *framework* untuk memberikan label pada sejumlah besar data yang tidak berlabel. Teknik SSL dapat meningkatkan kinerja model pada tugas *Machine Learning*, misalnya klasifikasi teks, terjemahan mesin, klasifikasi gambar. Cara kerja SSL adalah dengan mengambil *dataset* berlabel yang sudah ada dan hanya menggunakan sebagian kecil data pelatihan sebagai data berlabel, sementara memperlakukan sisa data sebagai *dataset* tidak berlabel .

Seperti yang terdapat pada Gambar 2.2, model dilatih pada data berlabel dengan *supervised learning*. Untuk data yang tidak berlabel, model yang sama digunakan untuk mendapatkan prediksi untuk sekumpulan sampel yang tidak berlabel. Prediksi dengan keyakinan maksimum disebut label semu (*pseudo-label*), yang memiliki probabilitas prediksi maksimum. Ketika model mampu memprediksi label dengan tingkat kepercayaan tinggi (*high-confidence*), maka akan diketahui bahwa label semu dapat diidentifikasi sebagai *ground-truth* [51].

Tingkat kepercayaan (nilai *confidence*) model dapat diukur dengan rumus :

$$\begin{aligned} total\ weight &= \Sigma(\text{jumlah total probabilitas}) \\ Confidence(x_i) &= \frac{\max_k P(y = k | x_i)}{total\ weight} \end{aligned} \quad (2.1)$$

Dimana y_i adalah prediksi model untuk sampel x_i yang tidak berlabel, k adalah indeks kelas, dan $P(y_i | x_i)$ adalah probabilitas dari prediksi tersebut. Misalkan diketahui :

Probabilitas kelas 1 adalah 0.2

Probabilitas kelas 2 adalah 0.5

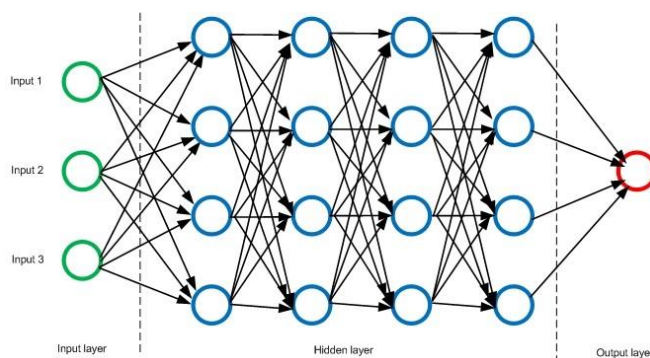
Probabilitas kelas 3 adalah 0.3.

Maka nilai *confidence* :

$$Confidence(x_i) = \frac{\max(0.2, 0.5, 0.3)}{(0.2 + 0.5 + 0.3)} = 0.5$$

2.2.4 Deep Learning

Deep Learning termasuk dalam cabang bidang pembelajaran mesin (*Machine Learning*). *Deep Learning* terdiri dari algoritma pemodelan abstraksi tingkat tinggi menggunakan fungsi transformasi non-linier yang disusun secara berlapis dan mendalam [52]. Arsitektur model *Deep Learning* dapat dilihat pada Gambar 2.3.



Gambar 2. 3 Arsitektur *Deep Learning*

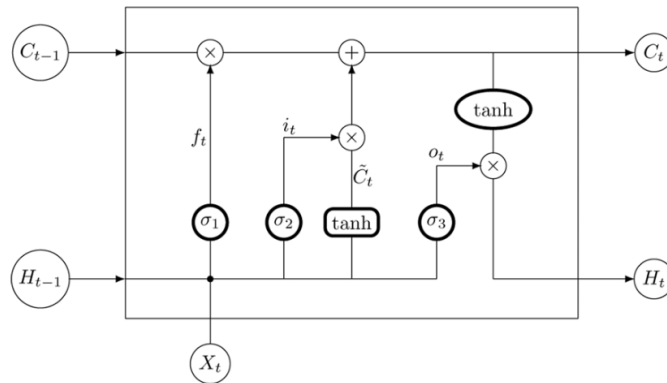
Deep Learning dapat diimplementasikan dalam Supervised Learning, Unsupervised Learning, *Semi-Supervised Learning*, Reinforcement Learning.

Tugas (*task*) dalam penelitian menggunakan *Deep Learning* antara lain klasifikasi teks (*text classification*), pengenalan citra, pengenalan suara, dan lain-lain.

2.2.5 Algoritma Long Short Term Memory (LSTM)

Algoritma LSTM sudah banyak digunakan dan berhasil dalam menangani permasalahan klasifikasi, terutama pada data yang membutuhkan pengurutan [53]. LSTM merupakan pengembangan dari algoritma Recurrent Neural Network (RNN), sehingga dalam prosesnya LSTM mampu memproses data dengan ukuran yang panjang (*long term dependency*) [54]. Cara kerja LSTM memproses data selanjutnya berdasarkan data masa lampau [55]. LSTM mampu menambah atau menghilangkan suatu informasi ke *block state* menggunakan 4 (empat) komponen yaitu *cell state*, *forget gate*, *input gate*, *output gate* [56]. Gambar 2.4 menunjukkan gambaran umum terkait alur pemrosesan data pada algoritma LSTM. algoritma LSTM membawa/meneruskan informasi dari *cell state* sebelumnya.

Pada tahap pertama, data dari *hidden state* pada *time step* sebelumnya (H_{t-1}), akan diproses pada *forget gate* (f_t). Pada *forget gate* akan ditentukan data mana yang akan dieliminasi dari *cell state*. apabila hasil perhitungan data menghasilkan nilai 0, maka data akan dihapus. Sedangkan jika hasil perhitungan data menghasilkan nilai 1, maka data akan diteruskan pada *cell state* sekarang [57]. Tahap selanjutnya adalah tahap *input gate*. Gate ini bertujuan menghitung data baru yang ditambahkan pada *cell state*. *Input gate* dibagi menjadi 2 lapisan yaitu sigmoid dan tanh yang masing-masing memiliki fungsi berbeda. Lapisan sigmoid berfungsi untuk menentukan besar nilai yang akan diperbarui dari *cell state*. Sedangkan lapisan tanh berfungsi membuat *vector* dari informasi baru. Gabungan dari lapisan sigmoid dan tanh melakukan *update* data. *Cell state* (C_t) adalah bagian LSTM yang bertugas membawa informasi/data dari sel sebelumnya ke sel selanjutnya. Informasi tiap *timestep* dalam *cell state* akan diperbarui menggunakan informasi yang telah diolah dalam *forget* dan *input gate*. Tahap terakhir adalah pada *output gate*. *Output gate* menentukan informasi berupa hasil keluaran, dan mengatur banyaknya output yang masuk pada *hidden state* selanjutnya.



Gambar 2. 4 Struktur algoritma LSTM [58]

Berikut merupakan rumus dan contoh perhitungan yang terdapat dalam LSTM pada setiap *gate* dan *state* yaitu:

$$\begin{aligned}
 & x_t = 0.5 \\
 \text{Misalkan diketahui nilai dari} & \quad h_{t-1} = 0.1 \\
 & \quad C_{t-1} = 0.2 \\
 & \quad W_f = 0.4, b_f = 0.1 \\
 \text{nilai bobot dan bias} & \quad W_i = 0.3, b_i = 0.1 \\
 & \quad W_c = 0.5, b_c = 0.1 \\
 & \quad W_o = 0.6, b_o = 0.1
 \end{aligned}$$

Forget gate :

$$\begin{aligned}
 f_t &= \sigma(W_f \cdot [h_{t-1}, X_t] + b_f) \\
 f_t &= \sigma(0.4 \cdot [0.1, 0.5] + 0.1) \\
 f_t &= \sigma(0.4 \cdot [0.6] + 0.1) \\
 f_t &= \sigma(0.34) \\
 f_t &= \frac{1}{1 + e^{-0.34}} \approx 0.584
 \end{aligned}$$

Input gate :

$$\begin{aligned}
 i_t &= \sigma(W_i \cdot [h_{t-1}, X_t] + b_i) \\
 i_t &= \sigma(0.3 \cdot [0.1, 0.5] + 0.1) \\
 i_t &= \sigma(0.3 \cdot [0.6] + 0.1) \\
 i_t &= \sigma(0.28) \\
 i_t &= \frac{1}{1 + e^{-0.28}} \approx 0.569
 \end{aligned}$$

Candidate cell state :

$$C_t = \tanh(W_C \cdot [h_{t-1}, X_t] + b_C)$$

$$C_t = \tanh(0.5 \cdot [0.1, 0.5] + 0.1)$$

$$C_t = \tanh(0.5 \cdot [0.6] + 0.1)$$

$$C_t = \tanh(0.4)$$

$$C_t \approx 0.379$$

Cell state :

$$C_t = f_t * C_{t-1} + i_t * C_t$$

$$C_t = 0.584 * 0.2 + 0.569 * 0.379$$

$$C_t \approx 0.1168 + 0.2156 \approx 0.332$$

Output gate :

$$o_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o)$$

$$o_t = \sigma(0.6 \cdot [0.1, 0.5] + 0.1)$$

$$o_t = \sigma(0.6 \cdot [0.6] + 0.1)$$

$$o_t = \sigma(0.46)$$

$$o_t = \frac{1}{1 + e^{-0.46}} \approx 0.613$$

Hidden state :

$$h_t = o_t * \tanh(C_t)$$

$$h_t = 0.613 * \tanh(0.332)$$

$$h_t \approx 0.613 * 0.32 \approx 0.196$$

Sehingga dapat diperoleh nilai *cell state* dan *hidden state* yang akan diteruskan pada *timestep* berikutnya adalah:

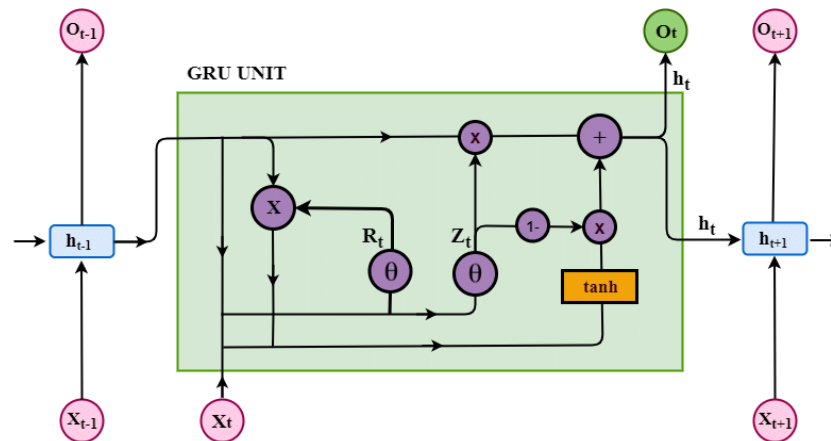
$$Cell\ State(C_t) = 0.332$$

$$Hidden\ State(h_t) = 0.196$$

2.2.6 Algoritma Gated Recurrent Unit (GRU)

Algoritma GRU memiliki struktur model yang mirip dengan Algoritma LSTM dengan versi yang lebih sederhana. Pada Gambar 2.5, dapat dijelaskan bahwa Algoritma GRU memiliki 2 *gate*, yaitu *reset gate* (R), dan *update gate* (Z). *Reset gate* (R) menambahkan masukan (*input*) yang baru pada memori sebelumnya (X). Sedangkan *update gate* (Z) menentukan seberapa banyak

informasi sebelumnya yang harus disimpan. Jika mengubah nilai semua *reset gate* menjadi 1 dan *update gate* menjadi 0 maka akan sama dengan model algoritma RNN yang standar [59].



Gambar 2. 5 Arsitektur Algoritma Gated Recurrent Unit (GRU)

Struktur internal GRU lebih sederhana dibandingkan dengan LSTM dan lebih mudah untuk dilatih karena lebih sedikit komputasi yang terlibat. Penyederhanaan ini antara lain *input gate* dan *forget gate* digabungkan menjadi satu gerbang yang disebut sebagai *update gate*, *cell state* dan *hidden state* digabungkan. *Gate* atau gerbang pada arsitektur GRU bertugas mengatur aliran informasi, sehingga GRU dapat mempelajari dan memilah informasi mana yang penting untuk disimpan atau dihapus. Informasi yang disimpan akan menjadi bahan untuk membuat keputusan/prediksi. Berikut merupakan rumus dan contoh perhitungan yang terdapat dalam GRU pada setiap *gate* dan *state* yaitu:

Misalkan diketahui nilai dari	$x_t = 0.5$
	$h_{t-1} = 0.1$
	$W_z = 0.4, b_f = 0.1$
nilai bobot dan bias	$W_i = 0.3, b_i = 0.1$
	$W = 0.5, b = 0.1$

Update gate :

$$z_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o)$$

$$z_t = \sigma(0.4 \cdot [0.1, 0.5] + 0.1)$$

$$z_t = \sigma(0.4 \cdot [0.6] + 0.1)$$

$$z_t = \sigma(0.34)$$

$$z_t = \frac{1}{1 + e^{-0.34}} \approx 0.584$$

Reset gate:

$$r_t = \sigma(W_r \cdot [h_{t-1}, X_t] + b_r)$$

$$r_t = \sigma(0.3 \cdot [0.1, 0.5] + 0.1)$$

$$r_t = \sigma(0.3 \cdot [0.6] + 0.1)$$

$$r_t = \sigma(0.28)$$

$$r_t = \frac{1}{1 + e^{-0.28}} \approx 0.569$$

Candidate cell state :

$$h_t = \tanh(W_c \cdot [r * h_{t-1}, X_t] + b)$$

$$r * h_{t-1} = 0.569 * 0.1 = 0.0569$$

$$h_t = \tanh(0.5 \cdot [0.0569, 0.5] + 0.1)$$

$$h_t = \tanh(0.5 \cdot 0.5569 + 0.1)$$

$$h_t = \tanh(0.37845)$$

$$h_t \approx 0$$

Hidden state:

$$h_t = (1 - z_t) * h_{t-1} + z_t * h_t$$

$$h_t = (1 - 0.584) * 0.1 + 0.584 * 0.361$$

$$h_t = 0.416 * 0.1 + 0.584 * 0.361$$

$$h_t \approx 0.0416 + 0.2109 \approx 0.253$$

Sehingga dapat diperoleh nilai *hidden state* yang akan diteruskan pada *timestep* berikutnya adalah:

$$Hidden State (h_t) = 0.253$$

2.2.7 VADER Lexicon

VADER (Valence Aware Dictionary for Sentiment Reasoning) memiliki cara kerja dengan menghitung polaritas sebuah kata dalam teks. Lebih lanjut dijelaskan

dalam [60], apabila kata yang dianalisis terdapat dalam kamus, maka skornya ditambahkan untuk mendapatkan skor polaritas keseluruhan. Misalnya, jika suatu teks cocok dengan kata yang ditandai positif di kamus, maka total skor polaritas teks tersebut bertambah. Jika skor polaritas keseluruhan teks tersebut positif, maka teks tersebut tergolong positif. Metode VADER sensitif terhadap kata yang memiliki kecenderungan, misalnya kata ‘hebat’, ‘sangat’, dan ‘buruk’ [61].

2.2.8 Confusion Matrix

Confusion Matrix merupakan salah satu metode yang umum digunakan dalam metode evaluasi algoritma yang melakukan komputasi data berlabel (Supervised Learning). Cara kerja Confusion Matrix adalah dengan mengolah data dengan tujuan membandingkan hasil prediksi dengan data sesungguhnya. Terdapat empat macam bagian evaluasi dari Confusion Matrix, yaitu akurasi, *recall*, *precision*, dan *F1 score*. Confusion Matrix dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Confusion Matrix

Aktual	Prediksi	
	Positif	Negatif
Positif	True Positive (TP)	False Negative (FN)
Negatif	False Positive (FP)	True Negative (TN)

1. Akurasi

Ukuran tingkat kedekatan antara nilai sebenarnya dengan nilai hasil prediksi model [62]. Hasil model dapat dikatakan semakin baik apabila menghasilkan nilai akurasi yang mendekati 100%.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \times 100\% \quad (2.2)$$

2 *Recall*

Atau disebut dengan sebut *sensitivity* merupakan metode yang digunakan untuk mengevaluasi seberapa baik model mengelompokkan data berlabel positif secara keseluruhan [62].

$$Recall = \frac{TP}{(TP + FN)} \times 100\% \quad (2.3)$$

3 *Precision*

Mengukur tingkat ketepatan suatu model yang dibangun dalam mengklasifikasikan suatu data kedalam kelas positif [63].

$$Precision = \frac{(TP)}{(TP + FP)} \times 100\% \quad (2.4)$$

4 *F1 score*

F1 score adalah perbandingan rata-rata *recall* dan presisi (*precision*), dengan rumus sebagai berikut:

$$F1\ score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \times 100\% \quad (2.5)$$