BAB II

LANDASAN TEORI

2.1 Penelitian Terkait

Berikut adalah penelitian terdahulu yang disajikan dalam mendukung dilakukannya ini, antara lain:

Tabel 2.1. Penelitian Terdahulu

No	Judul dan Peneliti Jurnal	Nama Jurnal	Metode	Hasil Penelitian
1	**Judul Jurnal:** Building a Medical Chatbot using Support Vector Machine Learning Algorithm. **Nama Peneliti:** Tamizharasi B., Jenila Livingston L.M., S. Rajkumar[19].	Journal of Physics: Conference Series, vol. 1716, no. 1, pp. 012059, 2020, doi: 10.1088/1742-6596/1716/1/012059.	SVM, Naïve Bayes, KNN.	Chatbot medis mencapai akurasi tinggi dengan Support Vector Machine (SVM) sebagai algoritma terbaik, mencapai akurasi 92,33%, lebih unggul dibandingkan KNN (87,66%) dan Naïve Bayes (81%). SVM terbukti lebih efektif dalam memprediksi penyakit dan lebih efisien dalam waktu dan ruang penyimpanan dibandingkan algoritma lain.
2	**Judul Jurnal:** Text Message Classification using Multiclass Support Vector	Telematika: Jurnal Informatika dan Teknologi	Multi-class SVM.	Penelitian ini menunjukkan bahwa algoritma Multiclass Support Vector Machine (SVM) efektif dalam mengklasifikasikan teks pesan

No	Judul dan Peneliti Jurnal	Nama Jurnal	Metode	Hasil Penelitian
	Machine on Information Service Chatbot. **Nama Peneliti:** R. P. Putra, A. H. Pratomo, dan R. I. Perwira[20].	Informasi, vol. 19, no. 3, pp. 295-310, Oct. 2022, doi: 10.31515/telematika.v19i3.7418.		dalam chatbot layanan informasi di Jurusan Informatika UPN "Veteran" Yogyakarta. Model ini menunjukkan kinerja yang baik dengan akurasi 87%, presisi 89%, dan recall 87%, menggunakan kernel RBF dan pendekatan One Versus All (OVA). Metode ini membantu chatbot dalam memberikan informasi secara otomatis terkait kategori seperti administratif, dokumen, jadwal, kegiatan, dan sapaan.
3	**Judul Jurnal:** AI Powered Anti-Cyber Bullying System using Machine Learning Algorithm of Multinomial Naïve Bayes and Optimized Linear Support Vector Machine. **Nama Peneliti:**	International Journal of Advanced Computer Science and Applications, vol. 13, no. 5, pp. 5-9, 2022, doi: 10.31515/ijacsa.v13i5.7418.	Multinomial Naïve Bayes (MNB) dan Linear Support Vector Machine (SVM).	Sistem berhasil mendeteksi dan menyaring pesan berbahaya dengan akurasi 92% dengan menggunakan SVM dibandingkan MNB dalam memblokir pesan bullying agar tidak sampai ke penerima. Dapat disimpulkan algoritma SVM lebih baik daripada MNB dalam penelitian ini.

No	Judul dan Peneliti Jurnal	Nama Jurnal	Metode	Hasil Penelitian
	T. Ige, S. Adewale[21].			
4	**Judul Jurnal:** Detection of Recovery of Covid-19 Cases using Machine Learning **Nama Peneliti:** J. V. Raj, J. V. J. Anton, J. P. Durai Raj[22].	International Journal of Current Research and Review, vol. 13, no. 6, pp. S-59–S-63, Mar. 2021, doi: 10.31782/IJCRR.2021.SP183.	SVM, ANN, DTree, KNN, Random Forest, Logistic Regression.	Dari hasil penelitian Decision Tree (DT) menunjukkan performa terbaik dengan Precision dan F1-Score tertinggi (0.9 dan 0.82), sementara SVM dan RF memiliki Accuracy tertinggi (0.8) namun kalah dalam metrik lainnya. K-Nearest Neighbors (KNN) memiliki performa terendah. Secara keseluruhan, DT adalah algoritma paling efektif di antara yang dibandingkan.
5	**Judul Jurnal:** Machine Learning Algorithms for Detection and Classifications of Emotions in Contact Center Applications **Nama Peneliti:** M. Płaza, S.	Sensors, vol. 22, no. 5311, pp. 1-18, Jul. 2022, doi: 10.3390/s22145311.	CNN, SVM, ANN, RFC, KNN.	Hasil penelitian menunjukkan bahwa SVM memberikan hasil terbaik untuk teks dengan akurasi 65.9% sedangkan CNN memberikan hasil terbaik untuk analisis suara dengan akurasi mencapai 67.5%.

No	Judul dan Peneliti Jurnal	Nama Jurnal	Metode	Hasil Penelitian
	Trusz, J. Kęczkowska, E. Boksa, S. Sadowski, Z. Koruba[23].			
6	**Judul Jurnal:** Optimasi Feature Selection Pada Komentar Media Sosial Terhadap Peralihan TV Digital Menggunakan Naïve Bayes, SVM, dan K-Nearest Neighbor **Nama Peneliti:** N. T. Romadloni and N. D. Septiyanti[24].	Decode Jurnal Pendidikan Teknologi Informasi, vol. 3, no. 2, pp. 151-160, Sep. 2023.	Naïve Bayes, SVM, KNN, Feature Selection.	SVM menunjukkan akurasi tertinggi sebesar 80,19% setelah feature selection, sedangkan Naïve Bayes dan KNN masing-masing mencapai akurasi 63,68% dan 80,02%.
7	**Judul Jurnal:** Sentiment Analysis of Public Acceptance of Covid-19 Vaccines Types	Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 7, no. 3,	Naïve Bayes, Support Vector Machine (SVM),	SVM menghasilkan akurasi tertinggi sebesar 84.89%, diikuti oleh Naïve Bayes dengan 84.65%, dan LSTM dengan 82.97%. Sinovac adalah jenis vaksin yang paling banyak mendapat

No	Judul dan Peneliti Jurnal	Nama Jurnal	Metode	Hasil Penelitian
	in Indonesia using Naïve Bayes, Support Vector Machine, and Long Short-Term Memory (LSTM) **Nama Peneliti:** D. A. Kristiyanti amd S. Hardani[25].	hal. 722-732, 2023, doi: 10.29207/resti.v7i3.4737.	Long Short-Term Memory (LSTM).	respons positif dan negatif di Twitter.
8	**Judul Jurnal:** Pembuatan Aplikasi Chatbot Berbasis Web Menggunakan Dialogflow dengan Integrasi Dialogflow Messenger pada Situs Journal of Multidisciplinary Issues **Nama Peneliti:** Ng, Ricky[26].	*Journal of Multidisciplinary Issues*, 2022.	NLP, RAD, Alpha Testing, Closed Beta Testing.	Aplikasi chatbot berbasis web untuk situs jurnal yang mampu menjawab pertanyaan pengguna secara otomatis. *Chatbot* diuji dengan *Alpha Testing* dan *Closed Beta Testing*, menunjukkan efektivitas *chatbot* dalam memberikan respon yang sesuai dan skor 85,25% dalam pengujian pengguna.

Beberapa penelitian telah menunjukkan keunggulan algoritma Support Vector Machine (SVM) dalam berbagai konteks aplikasi[19][20][21][22][23][24][25], terutama dalam hal akurasi. Dalam penelitian yang berfokus pada deteksi pesan berbahaya, SVM dibandingkan dengan Multinomial Naïve Bayes (MNB) untuk menyaring pesan-pesan berisiko, seperti pesan bullying.

Hasilnya, SVM berhasil mencapai akurasi sebesar 92%, lebih tinggi dibandingkan MNB yang memiliki akurasi lebih rendah, sehingga SVM terbukti efektif dalam memblokir pesan berbahaya agar tidak sampai ke penerima[21]. Selain itu, dalam analisis sentimen terhadap penerimaan publik terhadap vaksin Covid-19 di Indonesia, SVM menghasilkan akurasi tertinggi sebesar 84,89%, diikuti oleh Naïve Bayes dengan akurasi 84,65% dan Long Short-Term Memory (LSTM) dengan 82,97%. Ini menunjukkan keunggulan SVM dalam mengklasifikasikan sentimen pada media sosial[25].

Pada penelitian lain yang menggunakan multi-class SVM dengan kernel RBF dan pendekatan *One Versus All* (OVA) untuk klasifikasi teks dalam chatbot, algoritma ini berhasil mencapai akurasi sebesar 87% dengan presisi 89% dan recall 87%, membuktikan efektivitasnya dalam mengkategorikan teks secara otomatis dalam *chatbot* layanan informasi[20]. Dalam aplikasi sistem prediksi medis, SVM juga menunjukkan performa yang unggul dengan akurasi sebesar 92,33%, lebih tinggi daripada algoritma lain seperti K-Nearest Neighbor (KNN) yang hanya mencapai akurasi 87,66% dan Naïve Bayes dengan 81%. SVM terbukti lebih efektif dalam memprediksi penyakit dan lebih efisien dalam penggunaan waktu dan ruang penyimpanan[19]. Terakhir, dalam penelitian yang melibatkan optimasi *feature selection* pada komentar media sosial, SVM mencapai akurasi tertinggi sebesar 80,19% setelah proses optimasi, dibandingkan dengan Naïve Bayes dan KNN yang masing-masing hanya mencapai akurasi sebesar 63,68% dan 80,02%[24]. Berdasarkan berbagai penelitian sebelumnya, Support Vector Machine (SVM) konsisten memberikan akurasi yang baik. Oleh karena itu, SVM digunakan dalam model *chatbot* berbasis *retrieval-based* dengan dukungan *Natural Language Processing* (NLP) yang dioptimalkan melalui kombinasi metode TF-IDF untuk *feature extraction* guna meningkatkan akurasi *chatbot* dalam memproses dan memahami pertanyaan pengguna. Dengan menerapkan *framework* CRISP-ML, akhir dari penelitian ini mencakup tahapan *monitoring*, *maintenance*, dan *documentation* yang dievaluasi melalui *close beta testing* pada *chatbot* Digital Hub Sinar Mas Land. Pendekatan ini memberikan kontribusi signifikan dengan

menyediakan panduan komprehensif bagi penelitian lain dalam pengembangan dan pemeliharaan *chatbot* berbasis *retrieval* untuk kebutuhan bisnis perusahaan.

2.2 Tinjauan Teori

2.2.1 *Artificial Intelligence* (AI)

Artificial Intelligence (AI) didefinisikan sebagai ilmu komputer yang dirancang untuk melakukan tugas-tugas yang biasanya membutuhkan kecerdasan manusia, seperti pengenalan suara, pemrosesan bahasa alami, dan pengambilan keputusan. Berdasarkan penelitian, penerapan AI berkembang pesat dalam berbagai bidang, termasuk kesehatan, keuangan, pendidikan, dan teknologi informasi. Secara umum, AI terbagi menjadi beberapa pendekatan utama: *Machine Learning* (ML), *Natural Language Processing* (NLP), dan *Compuer Vision* yang masing-masing memiliki peranan dalam meningkatkan performa atau sistem suatu hal. Dalam beberapa dekade terakhir, AI telah berkembang pesat, terutama berkat peningkatan daya komputasi dan akses terhadap data yang melimpah. Teknologi ini telah diaplikasikan di berbagai sektor contohnya sektor properti. AI membantu dalam analisis data properti untuk memprediksi kenaikan properti, serta dalam bisnis yang menggunakan AI untuk sistem rekomendasi properti ataupun *chatbot*[27].

2.2.2 *Machine Learning* (ML)

Machine Learning (ML) adalah cabang studi di bidang pengembangan teknologi, yang artinya mesin memiliki kemampuan untuk belajar tanpa intervensi manusia secara langsung [28]. *Machine learning* bekerja untuk mencari sebuah pola kompleks pada data yang nantinya akan ditarik kesimpulan sehingga dapat menjadi penentu keputusan bagi sebuah perusahaan atau organisasi di masa mendatang. Penerapan *Machine learning* umumnya dilakukan pada pengolahan *Big Data*, memungkinkan otomatisasi pembuatan model analitik dan transformasi data besar yang awalnya sulit diinterpretasikan

menjadi lebih mudah digunakan dan diolah sebagai sumber informasi atau pengetahuan yang bermanfaat untuk meningkatkan potensi operasional dan strategi dalam kinerja perusahaan atau organisasi [29].

Dalam konteks *machine learning* (ML), algoritma dirancang untuk mengidentifikasi pola dan membuat prediksi berdasarkan data historis, yang kemudian diintegrasikan ke dalam berbagai aplikasi praktis, seperti sistem rekomendasi, diagnosis medis, pengenalan wajah dan lain-lain. Algoritma ini dapat dikategorikan menjadi tiga jenis utama: *supervised learning*, *reinforcement learning*, dan *unsupervised learning Supervised learning* menggunakan data berlabel untuk pelatihan model, sementara *unsupervised learning* mengidentifikasi pola tanpa bantuan data berlabel. Di sisi lain, *reinforcement learning* melibatkan agen yang belajar dari umpan balik dalam bentuk hadiah atau hukuman untuk mencapai tujuan tertentu[30].

2.2.3 *Natural Language Processing* (NLP)

Natural Language Processing (NLP) merupakan bidang kecerdasan buatan atau yang biasa disebut *Artificial Intelligence* (AI) yang memungkinkan mesin atau sistem komputer untuk memproses dan memahami bahasa manusia secara alami[26]. Untuk membuat komputer memahami bahasa manusia adalah hal yang sulit karena bahasa memiliki struktur yang kompleks. Dalam konteks pengembangan chatbot, NLP memegang peranan penting karena memberikan kemampuan pada *chatbot* untuk menafsirkan dan menjawab pertanyaan pengguna dalam bahasa yang mudah dipahami, menciptakan interaksi yang lebih alami dan efisien.

NLP bekerja melalui beberapa tahapan, antara lain tokenisasi, *stemming*, *stopword removal*, yang semuanya bertujuan untuk memahami dan merespons masukan pengguna secara tepat. Dalam proses ini, algoritma NLP mampu mengidentifikasi maksud atau niat dari pengguna (*intent detection*) serta mengenali entitas tertentu yang relevan (*entity recognition*), yang memungkinkan sistem untuk menghasilkan respon yang sesuai[31]. Teknologi

NLP juga mencakup pengenalan konteks, sehingga *chatbot* tidak hanya menjawab pertanyaan secara individual tetapi juga mampu memahami rangkaian percakapan yang terjadi.

2.2.4 *Chatbot*

Chatbot adalah program komputer yang dapat berinteraksi dengan pengguna secara otomatis melalui bahasa alami, menggunakan pemrosesan bahasa alami (NLP) dan kecerdasan buatan (AI). Pada dekade 1980-an, *chatbot* mengalami perkembangan penting melalui penerapan Kecerdasan Buatan. Salah satu inovasinya adalah A.L.I.C.E. (*Artificial Intelligent Internet Computer Entity*), sebuah *chatbot* yang dibangun menggunakan *Artificial Intelligence Markup Language* (AIML). AIML sendiri merupakan perluasan dari XML, yang memungkinkan *chatbot* ini memiliki kapabilitas komunikasi yang lebih canggih dan responsif[32]. Sekarang, chatbot sudah lebih canggih karena dapat dikelola atau dirancang menggunakan algoritma Deep Learning dan Transformasi (*Transformers*). *Chatbot* sangat dapat dimanfaatkan untuk segala sektor dalam kehidupan manusia, mulai dari layanan pelanggan, kesehatan hingga pendidikan, untuk meningkatkan efisiensi komunikasi. Melalui kemampuan untuk memahami dan merespons input pengguna secara langsung, *chatbot* berfungsi sebagai jembatan antara manusia dan sistem komputer, membantu dalam merespons pertanyaan atau memberikan informasi tanpa perlu keterlibatan manusia[33]. Seiring dengan meningkatnya komputasi dan ketersediaan framework AI, *chatbot* telah berkembang pesat, memungkinkannya untuk menangani percakapan yang semakin kompleks dan alami. *Chatbot* terbagi menjadi 2, antara lain: *Open-Domain Chatbot* dan *Closed-Domain Chatbot.*

1. ***Open-Domain Chatbot***

   Open-Domain Chatbot adalah *chatbot* yang dapat menangani topik yang luas tanpa batasan tertentu, memungkinkan pengguna untuk bertanya tentang berbagai hal di luar konteks tertentu. Jenis *chatbot* ini

menggunakan model AI yang besar dan biasanya dilatih menggunakan data dalam jumlah besar untuk dapat memberikan respons yang alami dalam berbagai topik percakapan. Dalam pengembangannya, *chatbot open-domain* dihadapkan pada tantangan utama untuk mempertahankan relevansi dan koherensi jawaban dalam percakapan yang bersifat umum dan acak. Ini memerlukan algoritma canggih yang dapat mengenali konteks yang sangat beragam. Terdapat beberapa teknik yang digunakan dalam membangun *chatbot open-domain*, termasuk pemanfaatan model bahasa yang telah dilatih secara luas dan menyeluruh[34].

2. *Closed-Domain Chatbot*

Berbeda dengan *open-domain*, *Closed-Domain Chatbot* bekerja pada ruang lingkup yang terbatas dan dirancang untuk menjawab pertanyaan dalam topik atau domain tertentu. Misalnya, *chatbot* yang digunakan dalam layanan kesehatan dirancang untuk menjawab pertanyaan yang berkaitan dengan kesehatan dan tidak akan memberikan tanggapan yang relevan di luar domain tersebut. *Chatbot* jenis ini lebih mudah untuk dikembangkan karena ruang lingkupnya yang terbatas memungkinkan sistem untuk fokus pada informasi spesifik. *Closed-domain chatbot* lebih akurat karena hanya bekerja dengan data dan konteks tertentu, sehingga resikonya untuk memberikan jawaban yang tidak relevan lebih rendah dibandingkan *open-domain chatbot*[35]. Dalam banyak aplikasi, *closed-domain chatbot* digunakan karena lebih mudah dipelihara dan memberikan hasil yang lebih dapat diandalkan.

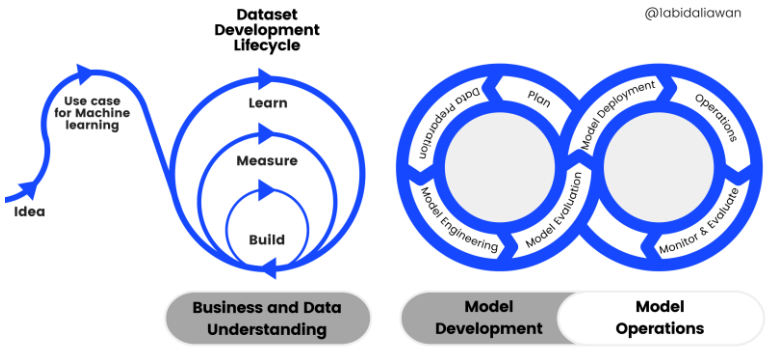
Seiring perkembangan teknologi, baik *open-domain* maupun *closed-domain chatbot* menghadapi tantangan masing-masing dalam hal akurasi dan relevansi. *Open-domain chatbot* perlu dapat menangani percakapan yang luas, sementara *closed-domain chatbot* harus menjaga konsistensi dan ketepatan dalam domain spesifiknya. Meskipun demikian, kedua jenis *chatbot* ini terus

ditingkatkan agar dapat memenuhi kebutuhan industri yang semakin kompleks dan memberikan pengalaman yang lebih baik kepada pengguna di masa sekarang dan mendatang.

2.3 Framework/Algoritma yang digunakan

2.3.1 Framework CRISP-ML

CRISP-ML *(Cross-Industry Standard Process for Machine Learning)* merupakan suatu kerangka konseptual yang disusun untuk memberikan panduan pada pengembangan penerapan *machine learning*. CRISP-ML juga perluasan dari *framework* CRISP-DM *(Cross-Industry Standard Process for Data Mining)* karena kebutuhan akan interpretabilitas dan penjelasan yang lebih tinggi dalam penerapan *machine learning.* Pada CRISP-ML lebih fleksibel dan iteratif daripada CRISP-DM, memungkinkan proyek untuk kembali ke tahap-tahap awal jika ada perubahan dalam data atau jika ada kebutuhan baru yang muncul. Ini penting dalam proyek *machine learning* yang sering kali menghadapi data kompleks dan tidak terstruktur, seperti dalam bidang properti, di mana perubahan pada data atau pendekatan dapat terjadi di tengah proyek [36].



Gambar 2.1 CRISP-ML *Life Cycle Process*[37]

Dibawah ini merupakan tahapan dari metode CRISP-ML, antara lain[38]:

1. **Business and Data Understanding**

   Tahap pertama dalam CRISP-ML adalah memahami kebutuhan bisnis serta konteks permasalahan yang akan dilakukan oleh *machine learning*. Langkah ini mencakup identifikasi tujuan bisnis, analisis kebutuhan dalam penelitian, dan pengenalan batasan-batasan penelitian. Selain itu, penting untuk memastikan bahwa data yang digunakan selaras dengan tujuan penelitian, sehingga hasil model *machine learning* dapat diukur dan dievaluasi secara efektif.

2. **Data Preparation**

   Tahap kedua dalam CRISP-ML adalah pengolahan data menggunakan berbagai metode preprocessing untuk memastikan data berkualitas sebelum memasuki tahap pemodelan. Langkah ini melibatkan pembersihan data *(data cleansing),* transformasi, dan pemilihan data yang relevan guna meningkatkan kualitas serta representasi data. Langkah ini juga mencakup penanganan data yang hilang dan normalisasi data untuk meningkatkan representasi data sehingga model yang dibangun dapat lebih akurat dan andal dalam memprediksi hasil sesuai kebutuhan penelitian.

3. **Modelling**

   Setelah dilakukan tahap *data preparation,* pada tahap ini dipilih algoritma *machine learning* yang sesuai dan dilatih menggunakan data yang telah dipersiapkan atau biasa disebut *model training*. Proses ini melibatkan eksperimen dengan berbagai model, fitur, dan parameter untuk mencapai kinerja terbaik sesuai dengan kriteria yang telah ditetapkan pada penelitian.

4. **Evaluation**

   Tahap selanjutnya, model yang telah dilatih dievaluasi untuk memastikan bahwa kinerjanya memenuhi kebutuhan bisnis dan teknis. Proses evaluasi ini mencakup pengujian model terhadap data uji, dengan penilaian kinerja menggunakan metrik seperti akurasi, presisi, dan *recall*. Metrik-metrik ini membantu menilai

seberapa baik model mampu menghasilkan prediksi yang sesuai dengan tujuan penelitian dan kebutuhan bisnis, sehingga dapat diandalkan dalam penerapannya di dunia nyata.

5. **Deployment**

   Setelah model dievaluasi dan memenuhi kriteria yang ditetapkan, model tersebut sudah dapat diterapkan. Tahap ini memastikan integrasi model dengan sistem yang ada dan mempersiapkan infrastruktur untuk operasionalisasi model. Biasanya dapat diimplementasikan dengan beberapa sistem atau *software* yang ada.

6. **Monitoring and Maintenance**

   Tahapan terakhir adalah model yang telah diimplementasikan atau diterapkan dipantau *(monitoring)* secara berkala untuk mendeteksi perubahan kinerja dalam data yang dapat mempengaruhi efektivitas model. Pemeliharaan melibatkan pembaruan model *(maintenance)* secara berkala dan penyesuaian terhadap perubahan kebutuhan bisnis atau lingkungan operasional.

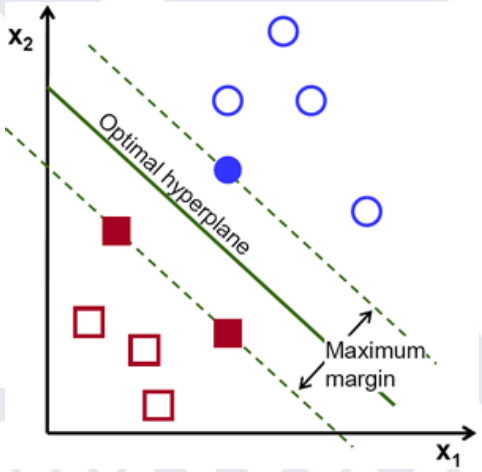
2.3.2 Algoritma Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah salah satu metode algoritma *machine learning* (pembelajaran mesin) yang *supervised learning* yang digunakan untuk klasifikasi dan, dalam beberapa kasus, analisis regresi. Metode ini dirancang untuk menangani data yang kompleks, baik linier maupun nonlinier, dan banyak digunakan dalam aplikasi seperti klasifikasi teks, klasifikasi gambar, deteksi spam, pengenalan tulisan tangan, analisis ekspresi gen, dan deteksi anomali[39]. SVM sangat andal karena merupakan algoritma yang memaksimalkan margin, memastikan hasil klasifikasi yang terbaik. Berikut adalah rumus dari SVM:

$$\boldsymbol{w \cdot x - b = 0}$$

Rumus 2.1 Rumus Support Vector Machine (SVM)[40].

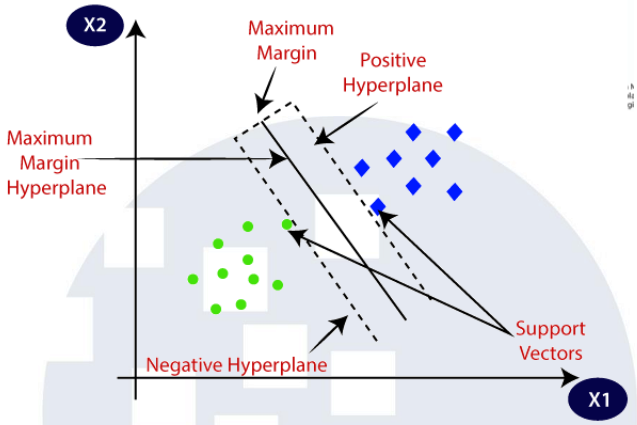
Support Vector Machine (SVM) memiliki *support vector* yang bekerja sebagai vektor terdekat dengan menemukan *hyperplane* optimal yang memisahkan kelas-kelas dalam ruang fitur. *Hyperplane* ini berupa garis atau bidang yang memisahkan kelas dengan margin maksimum. Dalam SVM, hyperplane optimal ini memaksimalkan jarak antara titik data terdekat dari masing-masing kelas, karena vektok-vektor saling mendukung *hyperplane* maka disebut *support vector*[41]. *Hyperplane* dengan margin maksimum ini disebut *hard margin* ketika data dapat dipisahkan secara linier. Margin adalah jarak antara *support vector* dari setiap kelas yang berada di sekitar *hyperplane*. Pada gambar 2.2 dan gambar 2.3, margin digambarkan sebagai jarak antara dua garis putus-putus. Margin terbesar *maximum* margin diperoleh dengan memaksimalkan jarak antara *hyperplane* dan titik terdekat dari *hyperplane* tersebut.



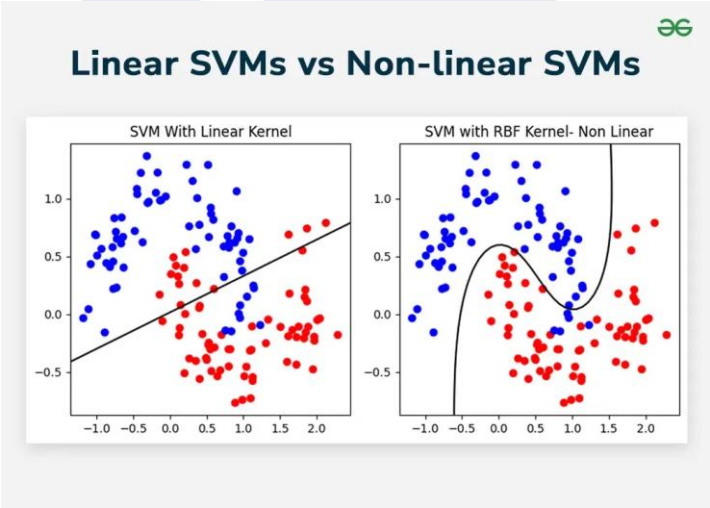
Gambar 2.2 *Hyperplane* pada SVM [42].

Dalam Support Vector Machine (SVM), *positive hyperplane* dan *negative hyperplane* adalah dua *hyperplane* yang berada di kedua sisi dari *maximum* margin atau *optimal hyperplane* (*hyperplane* utama yang memisahkan kelas-kelas)[40]. *Positive hyperplane* ini adalah garis atau bidang yang berada di sisi positif dari optimal *hyperplane*, melewati titik-titik data dari satu kelas. Selanjutnya adalah *negative hyperplane*

ini adalah garis atau bidang yang berada di sisi negatif dari optimal *hyperplane*, melewati titik-titik data dari kelas lain. *Hyperplane* positif dan negatif akan membantu menentukan margin maksimum antara dua kelas dalam SVM, yang diukur sebagai jarak antara kedua *hyperplane*.



Gambar 2.3 *Detail of Hyperplane* pada SVM [42].



Gambar 2.4 *Linear SVM vs Non-linear SVM* [43].

Berdasarkan sifat dari batas keputusan, Support Vector Machine (SVM) dapat dibagi menjadi dua bagian utama:

1. **Linear SVM:** Linear SVM menggunakan batas keputusan linier untuk memisahkan titik data dari berbagai kelas. Ketika data dapat dipisahkan secara linier dengan tepat, Linear SVM sangat cocok

digunakan. Ini berarti bahwa satu garis lurus (dalam 2D) atau *hyperplane* (dalam dimensi yang lebih tinggi) dapat sepenuhnya membagi titik data ke dalam kelas masing-masing[43]. *Hyperplane* yang memaksimalkan margin antara kelas-kelas tersebut menjadi batas keputusan.

2. **Non-Linear SVM:** Non-Linear SVM digunakan untuk mengklasifikasikan data ketika data tidak dapat dipisahkan menjadi dua kelas dengan garis lurus (dalam kasus 2D). Dengan menggunakan fungsi kernel, Non-Linear SVM dapat menangani data yang tidak dapat dipisahkan secara linier[44]. Fungsi kernel ini mentransformasi data input asli ke dalam ruang fitur berdimensi lebih tinggi, di mana titik data dapat dipisahkan secara linier. Linear SVM kemudian digunakan untuk menemukan batas keputusan nonlinier dalam ruang yang telah dimodifikasi ini.

2.3.3 Metode TF-IDF

Metode TF-IDF *(Term Frequency-Inverse Document Frequency)* adalah teknik yang banyak digunakan dalam penambangan teks dan pemrosesan bahasa alami (NLP) untuk menentukan kepentingan suatu kata dalam suatu dokumen relatif terhadap kumpulan dokumen *(corpus)*. TF-IDF membantu menyoroti kata-kata yang lebih relevan dalam dokumen tertentu, mengabaikan kata-kata umum yang sering muncul tetapi kurang relevan (seperti "dan", "atau", "itu")[45]. TF-IDF memiliki dua komponen yang utama, antara lain:

1. **TF *(Term Frequency)*** yaitu berfungsi untuk mengukur frekuensi kemunculan suatu kata dalam sebuah dokumen. TF menunjukkan seberapa sering suatu kata muncul dalam dokumen, biasanya dinyatakan dalam persentase atau rasio terhadap total kata dalam dokumen tersebut. Berikut adalah rumus dari TF:

$$TF = \frac{\text{Number of times the term appears in the document}}{\text{Total numbers of terms in the document}}$$

Rumus 2.2 Rumus *TF (Term Frequency)*[46]

2. **IDF *(Inverse Document Frequency)*** yaitu berfungsi untuk mengukur seberapa umum atau jarangnya suatu kata dalam kumpulan dokumen / *corpus*. Jika suatu kata muncul di banyak dokumen, maka IDF akan rendah, tetapi jika kata tersebut jarang muncul, IDF akan lebih tinggi, menandakan pentingnya kata tersebut dalam konteks tertentu. Penambahan "+1" dalam rumus dilakukan untuk menghindari pembagian dengan nol ketika sebuah kata muncul di seluruh dokumen[47], berikut adalah rumus dari IDF:

$$IDF = \log\left(\frac{N}{n} + 1\right)$$

Rumus 2.3 Rumus *IDF (Inverse Document Frequency)*[46]

Kombinasi keduanya menjadikan metode TF-IDF yang dapat menghitung relevansi suatu kata dengan menggabungkan informasi frekuensi kemunculan kata dalam dokumen dan seberapa jarangnya kata tersebut dalam *corpus* dengan rumus berikut:

$$TF\text{-}IDF = TF \times IDF$$

Rumus 2.4 Rumus *TF-IDF (Term Frequency-Inverse Document Frequency)*[46]

TF-IDF memiliki beberapa keunggulan dalam pemrosesan teks dan penambangan data, terutama dalam mengukur kepentingan kata atau frasa dalam dokumen. Metode ini sederhana dan efektif karena mampu memberikan bobot lebih tinggi pada kata atau frasa yang signifikan dalam konteks tertentu, sambil mengurangi pengaruh kata-kata umum yang sering muncul namun kurang relevan. Selain itu, TF-IDF juga dapat kemampuannya mengurangi dimensi data dengan mengabaikan kata-kata yang memiliki bobot rendah, sehingga meningkatkan efisiensi pemrosesan dan mengurangi *noise*, mudah diinterpretasikan,

membuatnya populer untuk ekstraksi kata kunci, pencarian informasi, serta klasifikasi teks[48]. Bobot TF-IDF juga sering digunakan sebagai fitur dalam algoritma *machine learning* yang lebih kompleks, menjadikannya fleksibel dan dapat digabungkan dengan teknik NLP lainnya.

2.3.4 Metode Evaluasi Metrik Akurasi

Dalam evaluasi model *machine learning*, metrik akurasi, precision, recall, dan F1-score adalah ukuran yang membantu menilai kinerja model, terutama pada masalah klasifikasi. Berikut adalah rumus evaluasi untuk menguji[49][50] model tersebut, antara lain:

1. ***Accuracy*** adalah metrik pengukuran yang melakukan rasio prediksi benar (negatif dan positif) dengan keseluruhan data yang ada.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Rumus 2.5 Rumus Evaluasi *Accuracy*

Legend dari rumus 2.5 adalah sebagai berikut:

a. TP *(True Positive):* Jumlah prediksi positif yang benar (model memprediksi positif, dan faktanya memang positif).
b. TN *(True Negative):* Jumlah prediksi negatif yang benar (model memprediksi negatif, dan faktanya memang negatif).
c. FP *(False Positive):* Jumlah prediksi positif yang salah (model memprediksi positif, tetapi faktanya negatif).
d. FN *(False Negative):* Jumlah prediksi negatif yang salah (model memprediksi negatif, tetapi faktanya positif).

2. ***Recall*** adalah rasio antara jumlah *true positive* dengan *false negative* berfungsi untuk mengukur kemampuan model untuk mendeteksi kelas positif. Nantinya akan membagi data benar dan data tidak benar.

$$Recall = \frac{TP}{P}$$

Rumus 2.6 Rumus Evaluasi *Recall*

3. ***Precision*** adalah metrik pengukuran yang menghitung prediksi positif benar pada model, sehingga dapat mengetahui klasifikasi nilai positif yang tidak benar serta berfungsi untuk mengukur ketepatan model dalam memprediksi kelas positif.

$$Precision = \frac{TP}{TP+FP}$$

Rumus 2.7 Rumus Evaluasi *Precision*

4. ***FI-Score*** adalah metrik evaluasi penilaian akhir sebagai sumber informasi terhadap keseimbangan nilai *Recall* dan nilai *Precision* yang dibentuk.

$$F1 - Score = \frac{2xPrecision\ x\ Recall}{Precision+Recall}$$

Rumus 2. 8 Rumus Evaluasi *F1-Score*

Keempat persamaan diatas berupa *accuracy, recall, precision,* dan *f1*-score berfungsi untuk mengevaluasi kinerja model klasifikasi, terutama dalam konteks *machine learning*.

2.3.5 Metode *Closed Beta Testing*

Closed Beta Testing adalah salah satu metode *Beta Testing* atau tahap pengujian suatu aplikasi atau sistem yang termasuk dalam *User Acceptance Test* (UAT) yang melibatkan sekelompok kecil pengguna yang diundang atau dipilih secara khusus untuk melakukan uji coba langsung terhadap aplikasi atau sistem[51]. Proses ini bertujuan untuk menguji fungsi dan kehandalan aplikasi atau sistem sebelum peluncuran secara umum. *Closed beta testing* juga bertujuan untuk mengumpulkan masukan dan menemukan *bug* atau masalah dalam aplikasi atau sistem sebelum diluncurkan ke publik.

Pengujian ini memiliki beberapa karakteristik utama, di antaranya adalah terbatasnya partisipasi hanya kepada pengguna tertentu yang

diundang atau dipilih, harapan akan umpan balik rinci mengenai fitur, kegunaan, dan kinerja produk, serta penggunaan hasil pengujian untuk memperbaiki produk sebelum peluncuran publik. *Closed beta testing* merupakan langkah penting dalam proses pengembangan produk, memastikan bahwa produk yang akan diluncurkan memenuhi standar kualitas dan keandalan yang diharapkan oleh pengembang dan pengguna[26].

2.4 Tools yang digunakan

2.4.1 Python

Sebuah bahasa pemrograman yang berorientasi pada objek *(object-oriented)* yang mudah untuk dipahami dan dipelajari. Python programming memiliki fungsionalitas *library* berstandar yang lengkap dan kemampuan mengkombinasikan sintaks kode dengan manajemen memori yang otomatis [52]. Bahasa pemrograman Python juga memiliki modul yang sangat lengkap untuk memenuhi segala kebutuhan, terdapat sekitar 300 modul internal dan sekitar 100 modul eksternal, hal ini memungkinkan pemrograman untuk mengembangkan aplikasi baru dengan cara yang sederhana, cepat, murah, hemat biaya, dan tepat waktu. Hellman menganalisis 117 modul, membaginya menjadi 19 kelas, dan menjelaskan metode dan fungsi yang masuk ke setiap kelas *library* pada Python contohnya seperti modul *software development, text processing,* tipe data, perhitungan matematika, dan model pembelajaran mesin [53].

2.4.2 Jupyter Notebook

Software yang sebagai alat bantuan untuk pengolahan data dengan kemudahan aksesibilitas berbagai bahasa pemrograman khususnya Python. Jupyter Notebook dirancang untuk membuat analisis data lebih mudah dan memungkinkan pengguna untuk menulis sel *markdown* yang berisi penjelasan tentang logika program yang ada, diikuti dengan visualisasi langsung dari hasil program tersebut [54]. Jupyter Notebook juga dapat dengan mudah dibagikan

karena disimpan sebagai file teks terstruktur (format JSON) dan memungkinkan transfer kode model dari satu *instance* ke *instance* lainnya untuk melatih ulang model [55]. Jupyter Notebook merupakan lingkungan pembembangan interaktif berbasis web yang antarmukanya fleksibel sehingga membantu pengguna dalam mengkonfigurasi dan mengatur alur kinerja dalam analitikal *data science*, *machine learning*, komputasi ilmiah, dan komputasi jurnalisme dalam mendukung ilmu interaktif data yang tebuka dan gratis untuk semua orang [56].

2.4.3 Visual Studio Code

Visual Studio Code (VS Code) adalah editor kode sumber terbuka yang dikembangkan oleh Microsoft. Aplikasi ini dirancang untuk membantu pengembang dalam menulis, mengedit, dan mengelola kode sumber dari berbagai bahasa pemrograman. VS Code sangat populer di kalangan pengembang karena ringan, fleksibel, dan memiliki banyak fitur yang mendukung berbagai workflow pengembangan perangkat lunak. Visual Studio Code mendukung berbagai bahasa pemrograman seperti JavaScript, Python, Java, C++, dan banyak lagi. Fitur seperti penyorotan sintaks, *auto-completion*, dan IntelliSense membantu pengembang menulis kode dengan lebih efisien dan mengurangi kesalahan. Dengan fitur-fitur yang kaya dan fleksibilitasnya, Visual Studio Code telah menjadi alat esensial bagi pengembang perangkat lunak di seluruh dunia, mendukung berbagai kebutuhan pengembangan dari proyek kecil hingga sistem yang kompleks[57].

2.4.4 Streamlit

Streamlit adalah *framework open-source* berbasis Python yang dikembangkan untuk memudahkan pembuatan aplikasi web secara interaktif, khususnya dalam bidang *data science* dan *machine learning*. *Framework* ini memungkinkan pengguna untuk membangun antarmuka visual dengan mudah menggunakan sintaks Python yang sederhana tanpa perlu menggunakan bahasa pemrograman *front-end* lainnya. Hal ini membuat Streamlit sangat cocok digunakan oleh peneliti atau *data scientist* yang ingin menyajikan hasil analisis data atau model *machine learning* dalam bentuk aplikasi web tanpa harus

menguasai keterampilan pengembangan web[58]. Streamlit juga memiliki sejumlah keunggulan, antara lain:

1. **Iterasi Cepat:** Streamlit dilengkapi dengan fitur *hot-reloading* yang memungkinkan pembaruan kode secara langsung terlihat pada browser, sehingga mendukung proses iterasi dan eksperimen secara cepat.
2. **Widget Interaktif:** Platform ini memungkinkan pengguna untuk menyertakan widget interaktif, seperti tombol, *slider*, dan menu *dropdown*, yang berguna untuk mengatur parameter serta tampilan aplikasi.
3. **Integrasi yang Lancar:** Streamlit dapat dengan mudah diintegrasikan dengan berbagai pustaka Python populer, seperti Pandas, Matplotlib, dan Scikit-learn.
4. **Penyebaran:** Streamlit menawarkan kemudahan dalam penyebaran aplikasi, baik secara lokal maupun di *cloud*, sehingga dapat dibagikan kepada pengguna lain.

Dengan demikian, Streamlit dapat menjadi *tools* yang sangat dalam menyajikan dan membagikan hasil analisis secara interaktif dan mudah diakses. Dengan fitur-fiturnya yang mendukung pembuatan aplikasi berbasis Python tanpa perlu pengetahuan mendalam tentang pengembangan web, Streamlit memungkinkan pengguna untuk mengatur dan memvisualisasikan data dalam tampilan yang lebih intuitif[59][60].