BAB II
KAJIAN PUSTAKA

2.1 Penelitian Terdahulu

Pengelolaan pelanggan perguruan tinggi, yang mencakup calon mahasiswa, mahasiswa, alumni, dan pengguna alumni merupakan faktor terpenting dalam meningkatkan layanan dan kualitas di suatu perguruan tinggi. Tabel 2.1 menyajikan rangkuman sekitar 45 artikel yang terkait dengan tema potensi/perfoma mahasiswa menggunakan pembelajaran mesin. Tabel 2.1 merupakan rangkuman artikel dari penelitian yang terkait tema disertasi yaitu:

Tabel 2.1 Artikel Pembelajaran Mesin Berdasarkan Metode Tinjauan Pustaka

No	Penulis, Tahun. Judul Artikel	Metodologi	Tujuan
1.	(Ofori et al., 2020), **"*Using Machine Learning Algorithms to Predict Students ' Performance and Improve Learning Outcome : A Literature Based Review*",** Journal of Information and Technology (2020), Vol.4 , March, pages 33-55, ISSN: 2616-3573	Menguraikan penelitian *machine learning* terdahulu dalam konteks pendidikan. Mengidentifikasi kombinasi algoritma atau pemanfaatan data perilaku mahasiswa. Menyintesis berbagai pendekatan sebelumnya, seperti model klasifikasi berbasis *supervised learning* (*Decision Tree*, SVM) dan model *ensemble*.	Mengevaluasi efektivitas berbagai algoritma pembelajaran mesin dalam memprediksi performa akademik mahasiswa, serta mengidentifikasi faktor-faktor utama yang berkontribusi terhadap keberhasilan atau kegagalan akademik.
2.	(Rastrollo-guerrero et al., 2020), ***"Analyzing and Predicting Students ' Performance by Means of Machine Learning : A Review"***, MDPI, doi 10.3390/app10031042	Menyajikan tinjauan terhadap penelitian *machine learning* di bidang pendidikan. Menyoroti algoritma yang paling sering digunakan.	Menganalisis faktor-faktor yang memengaruhi kinerja akademik mahasiswa dan memprediksi performa mereka menggunakan berbagai algoritma *machine learning*.

Tabel 2.1 Artikel Pembelajaran Mesin Berdasarkan Metode Tinjauan Pustaka (Lanjutan)

No	Penulis, Tahun. Judul Artikel	Metodologi	Tujuan
3.	(Alalawi et al., 2023), "***Contextualizing the current state of research on the use of machine learning for student performance prediction : A systematic literature review",*** Engineering Report Wiley, pages 1-25, doi. 10.1002/eng2.12699	*Systematic literature review* dengan mengacu pada panduan metodologis Kitchenham et al. (2007)	Meninjau secara sistematis penelitian-yang menggunakan teknik ML dalam memprediksi performa akademik mahasiswa
4.	(Fahd et al., 2021), "***Application of machine learning in higher education to assess student academic performance, at-risk,and attrition: A meta-analysis of literature"*** Education and Information Technologies (2022) 27:3743–3775 https://doi.org/10.1007/s10639-021-10741-7	*Systematic literature review* dengan PRISMA	Menyediakan gambaran penerapan ML di pendidikan tinggi untuk memprediksi performa mahasiswa
5.	(Alsariera et al., 2022), ***"Assessment and Evaluation of Different Machine Learning Algorithms for Predicting Student Performance"***, Computational Intelligence and Neuroscience (2022), doi.https://doi.org/10.1155/2022/4151487	Pendekatan *Systematic Literature Review* untuk mengkaji penelitian-penelitian terdahulu yang relevan antara tahun 2015–2021	Menilai dan mengevaluasi berbagai algoritma ML yang digunakan dalam memprediksi kinerja akademik mahasiswa, serta mengidentifikasi atribut (fitur) yang paling berpengaruh terhadap hasil belajar
6.	(Sekeroglu et al., 2021), ***"Systematic Literature Review on Machine Learning and Student Performance Prediction : Critical Gaps and Possible Remedies",*** Applied Sciences (2021), https://doi.org/10.3390/app112210907	Metode *Systematic Literature Review* dengan mengadaptasi kerangka kerja Kitchenham dan Charters (2007)	Meninjau secara sistematis penelitian-penelitian terdahulu yang menggunakan ML dalam prediksi performa mahasiswa, serta mengidentifikasi kesenjangan riset dan rekomendasi pengembangan penelitian selanjutnya
7.	(Oppong, 2023), ***"Predicting Students ' Performance Using Machine Learning Algorithms : A Review",*** Asian Journal of Research in Computer Science Volume (2023), Vol.16, Issue.3, pages.128-148 Doi. 10.9734/AJRCOS/2023/v16i3351	Mengekstrak isi artikel-artikel berupa algoritma, fitur, *dataset*, dan akurasi dari setiap penelitian.	Memprediksi performa mahasiswa dengan ML. Mengidentifikasi algoritma yang paling banyak dan paling efektif digunakan. Menganalisis atribut penting dalam data mahasiswa.

Tabel 2.1 Artikel Pembelajaran Mesin Berdasarkan Metode Tinjauan Pustaka (Lanjutan)

No	Penulis, Tahun. Judul Artikel	Metodologi	Tujuan
8.	(Enughwure & Ogbise, 2020), ***" Application of Machine Learning Methods to Predict Student Performance : A Systematic Literature Review",*** International Research Journal of Engineering and Technology (IRJET),2020, Vol.07, May, pages. 3405-3415, e-ISSN: 2395-0056 p-ISSN: 2395-0072	*Systematic, structured*, dan berbasis kuantitatif deskriptif.	Merangkum dan memetakan: metode dan tools ML yang digunakan, jenis variabel yang digunakan dalam prediksi, kontribusi penelitian dalam bidang prediksi kinerja mahasiswa.
9.	(Ahmed, 2024), ***"Student Performance Prediction Using Machine Learning Algorithms",*** Applied Computational Intelligence and Soft Computing, 2024, https://doi.org/10.1155/2024/4067721	Analisis eksperimental yang meliputi *preprocessing, feature selection* (*Random Forest), clustering (K-means)*, pembangunan model prediksi, *hyperparameter tuning* (*Grid Search + Cross-Validation*), dan evaluasi menggunakan metrik kinerja klasifikasi.	Mengidentifikasi fitur penting, menguji dan membandingkan algoritma ML, meningkatkan akurasi melalui tuning, dan menghasilkan model prediksi yang akurat untuk membantu pengambilan keputusan pendidikan.
10.	(Ouhaddou et al., 2025), ***"Predicting Student Academic Path Using Machine Learning : Systematic Review"***, IEEE Access, 2025, doi. 10.1109/IRASET64571.2025.11008352	Menggunakan *Systematic Literature Review* melalui pencarian di IEEE, Scopus, ScienceDirect, ResearchGate; pemilihan kata kunci berbasis struktur terminologi; seleksi artikel bertahap serta ekstraksi dan analisis atribut, algoritma, metode *feature selection*, dan metrik akurasi menggunakan tabel dan grafik. Sebanyak 42 artikel akhir dianalisis.	Menganalisis, mensintesis, mengelompokkan, dan mengevaluasi 42 studi terkait prediksi jalur akademik mahasiswa menggunakan ML, serta menjawab *research question* terkait *learning outcomes*, tantangan, dan dampak prediksi terhadap mahasiswa dan institusi.

Tabel 2.1 Artikel Pembelajaran Mesin Berdasarkan Metode Tinjauan Pustaka (Lanjutan)

No	Penulis, Tahun. Judul Artikel	Metodologi	Tujuan
11.	(Baashar et al., 2025), ***" Predicting student ' s performance using machine learning methods : A systematic literature review"***, International Conference on Computer & Information Sciences (ICCOINS) 2020, ISBN 9781728171531	SLR dengan penyusunan RQ berbasis PIOC, kriteria inklusi/eksklusi yang jelas, pencarian sistematis di 5 *database* internasional, screening bertahap (judul, abstrak,full text), serta ekstraksi dan sintesis data atribut, metode ML, dan akurasi dari 30 studi.	Mengidentifikasi atribut penting, metode ML yang digunakan, serta akurasi model dalam prediksi performa mahasiswa, dengan menutup gap penelitian sebelumnya tentang kurangnya mapping metode dan atribut yang relevan.
12.	(Chakrapani & D, 2022), ***"Academic Performance Prediction Using Machine Learning : A Comprehensive & Systematic Review"***, 2022 International Conference on Electronic Systems and Intelligent Computing (ICESIC), doi. 10.1109/ICESIC53714.2022.9783512	*Systematic Literature Review* berbasis panduan Okoli, meliputi: penyusunan RQ, strategi pencarian multi *database*, penggunaan kata kunci Boolean, pembatasan tahun (2015–2022), proses iteratif (*refine & re-run*), eliminasi duplikasi, *full-text screening*, dan dokumentasi artikel yang terpilih.	Mengidentifikasi penelitian, memetakan atribut, menganalisis metode ML, dan menilai akurasi model prediksi performa akademik melalui tinjauan literatur yang sistematis dan komprehensif.
13.	(Nawang et al., 2021), ***"A systematic literature review on student performance predictions",*** International Journal of Advanced Technology and Engineering Exploration, 2021. Vol.8, Issue.84, doi. http://dx.doi.org/10.19101/IJATEE.2021.874521	Metode SLR berdasarkan Kitchenham, meliputi:*Planning, Conducting, Reporting*	Memetakan fitur, metode, dan algoritma ML yang digunakan dalam prediksi performa siswa, sekaligus mengidentifikasi gap penelitian dari studi tahun 2016–2020.

Posisi penelitian disertasi terhadap penelitian pembelajaran mesin sebelumnya di artikel yang menggunakan metode tinjauan pustaka yaitu penelitian disertasi ini menempati posisi yang berbeda dari seluruh artikel pada Tabel 2.1 karena disertasi ini mengembangkan model baru, sedangkan dalam artikel-artikel

tersebut dilakukan tinjauan pustaka sistematis. Disertasi ini menggunakan pendekatan pembelajaran semi-terpandu, yang menggabungkan metode *co-training* LSTM-SVM, sementara seluruh artikel pada Tabel 2.1 hanya merangkum dan menganalisis penelitian pembelajaran mesin yang sudah ada tanpa membahas teknik pembelajaran semi-terpandu. Penelitian disertasi mengimplementasikan *pseudolabeling* pada data berlabel dan data tak berlabel, sedangkan artikel di Tabel 2.1 tidak menyinggung mekanisme pemanfaatan *pseudolabeling*. Disertasi ini menerapkan konteks 10 perguruan tinggi di Indonesia yang memiliki keterbatasan data berlabel, sedangkan seluruh tinjauan artikel penelitian sebelumnya tidak memasukkan kasus lokal di perguruan tinggi yang memiliki keterbatasan data berlabel. Penelitian disertasi ini mengisi kekosongan kajian yang belum dicakup oleh penelitian sebelumnya, karena penelitian disertasi ini mengusulkan model pembelajaran semi-terpandu yang sama sekali belum dibahas dalam penelitian terdahulu.

Tabel 2.2 Artikel Pembelajaran Mesin Terpandu/AI

No	Penulis, Tahun. Judul Artikel	Metodologi	Tujuan
1.	(Pallathadka et al., 2023), ***"Classification and prediction of student performance data using various machine learning algorithms"***, Materials Today: Proceedings	Naive Bayes, ID3, C4.5, *Support Vector Machine*	Memprediksi performa mahasiswa menggunakan algoritma ML.
2.	(Hasan et al., 2019), "***Machine Learning Algorithm for Student's Performance Prediction"*** , 10th ICCCNT 2019 July 6-8, 2019 , IIT - Kanpur, Kanpur, India	*KNN, Decision Tree*	Memprediksi hasil ujian akhir mahasiswa
3.	(Buenaño-fern & Gil, 2019), ***"Application of Machine Learning in Predicting Performance for Computer Engineering Students : A Case Study ",*** MDPI (2019):1-18 https://doi.org/10.3390/su11102833	*Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbor, Logistic Regression*	Menerapkan dan membandingkan algoritma pembelajaran mesin dalam memprediksi performa akademik.
4.	(Badal & Sungkur, 2023), ***"Predictive modelling and analytics of students' grades using machine learning algorithms",*** Education and Information Technologies (2022), https://doi.org/10.1007/s10639-022-11299-8	*Machine learning, (Decision Tree, Random Forest, SVM, kNN, Linier Regression, Neural Network)*	Memprediksi nilai mahasiswa menggunakan berbagai model ML

Tabel 2.2 Artikel Pembelajaran Mesin Terpandu/AI (Lanjutan)

No	Penulis, Tahun. Judul Artikel	Metodologi	Tujuan
5.	(Jang, Choi, Jung, & Kim, 2022), ***"Practical early prediction of students' performance using machine learning and eXplainable AI"***, Education and Information Technologies (2022), https://doi.org/10.1007/s10639-022-11120-6	Regresi Logistik, Pohon Keputusan, Hutan Acak, Perceptron Multi-Lapisan, Mesin Vektor Dukungan, XGBoost, LightGBM, VTC, dan STC	Memprediksi siswa berisiko sedari dini
6.	(Çakıt & Dağdeviren, 2021),***" Predicting the percentage of student placement: A comparative study of machine learning algorithms***", Education and Information Technologies (2022) 27:997–1022, https://doi.org/10.1007/s10639-021-10655-4	Algoritma *Machine Learning* (ML)	Mengembangkan model prediksi mahasiswa.
7.	(Bhutto et al., 2020), ***"Predicting Students' Academic Performance Through Supervised Machine Learning"***, 2020 International Conference on Information Science and Communication Technology Predicting, ISBN. 9781728168999	SVM SMO, *Logistic Regression*	Membangun model prediksi performa mahasiswa, membandingkan dua algoritma *supervised*
8.	(Sathe & Adamuthe, 2021), ***" Comparative Study of Supervised Algorithms for Prediction of Students ' Performance"***, Modern Education and Computer Science, 2021, Vol.13, Issue.1, pages 1-21, doi. 10.5815/ijmecs.2021.01.01	C5.0, C4.5 *Decision Tree*, CART, *Naive Bayes*, KNN, *Random Forest, Support Vector Machine*	Menentukan fitur yg berkorelasi dengan nilai siswa. Membandingkan performa 7 algoritma *supervised*. Menganalisis pengaruh parameter *tuning* terhadap akurasi.
9.	(Ezz & Elshenawy, 2020), "***Adaptive recommendation system using machine learning algorithms for predicting student's best academic program",*** Education and Information Technologies (2020) 25:2733–2746 https://doi.org/10.1007/s10639-019-10049-7	Modeling multi-algoritma	Membuat sistem rekomendasi adaptif untuk menempatkan mahasiswa pada jurusan teknik yang sesuai berdasarkan performa tahun persiapan

Tabel 2.2 Artikel Pembelajaran Mesin Terpandu/AI (Lanjutan)

No	Penulis, Tahun, Judul Artikel	Metodologi	Tujuan
10.	(Jiao et al., 2022)," ***Artificial intelligence-enabled prediction model of student academic performance in online engineering education"*** Artifcial Intelligence Review (2022) https://doi.org/10.1007/s10462-022-10155-y	*Artificial Neural Network, Support Vector Machine*	Mengembangkan model prediksi berbasis *evolutionary computation* untuk menilai performa mahasiswa
11.	(Iatrellis, Savvas, Fitsilis, & Gerogiannis, 2021), "***A two-phase machine learning approach for predicting student outcomes***", Education and Information Technologies (2021) 26:69–88 https://doi.org/10.1007/s10639-020-10260-x	Algoritma *K Means Clustering*	Mengembangkan pendekatan prediksi berbasis dua fase (*clustering* lalu *prediction*)
12.	(Bansal, Buckchash, & Raman, 2022), "***Computational Intelligence Enabled Student Performance Estimation in the Age of COVID-19",*** SN Computer Science (2022) 3:41 https://doi.org/10.1007/s42979-021-00944-7	*Deep learning*	Membuat sistem prediksi nilai mahasiswa berbasis CI yang efektif di masa pandemi
13.	(Kaur, Mehta, Randhawa, Sharma, & Park, 2021),*" Ensemble learning-based prediction of contentment score using social multimedia in education",* Multimedia Tools and Applications (2021) 80:34423–34440 https://doi.org/10.1007/s11042-021-10806-2	Metode *Cuckoo Search meta-heuristic* untuk memilih fitur penting dari *dataset*	Memprediksi kepuasan mahasiswa dengan pendekatan *ensemble learning*
14.	(Asselman, Khaldi, & Aammou, 2020), "***Evaluating the impact of prior required scaffolding items on the improvement of student performance prediction",*** Educational and Information Technologies. (2020) https://doi.org/10.1007/s10639-019-10077-3	*Performance Factors Analysis (PFA)* berbasis regresi logistik sebagai *baseline*	Mengembangkan dua model baru: model 1 tanpa bantuan *scaffolding* dan model 2 dengan *scaffolding*
15.	(Badal & Sungkur, 2023), ***"Predictive modelling and analytics of students' grades using machine learning algorithms",*** Education and Information Technologies (2022), https://doi.org/10.1007/s10639-022-11299-8	*Machine learning, (Decision Tree, Random Forest, SVM, kNN, Linier Regression, Neural Network)*	Memprediksi nilai mahasiswa menggunakan berbagai model ML
16.	(Abu Zohair, 2019), "***Prediction of Student's performance by modelling small dataset size",*** Educational Technology in Higher Education (2019) 16:27 https://doi.org/10.1186/s41239-019-0160-3	*SVM, LDA, MLP-ANN, Naive Bayes, KNN*	Membuat model prediksi. Mengevaluasi akurasi berbagai algoritma ML
17.	(Basnet, Johnson, & Doleck, 2022), "***Dropout prediction in Moocs using deep learning and machine learning***", Education and Information Technologies (2022) https://doi.org/10.1007/s10639-022-11068-7	Algoritma *Machine Learning vs Deep Learning*	Menilai perbedaan kinerja ML vs DL dalam prediksi *dropout*

Tabel 2.2 Artikel Pembelajaran Mesin Terpandu/AI (Lanjutan)

No	Penulis, Tahun, Judul Artikel	Metodologi	Tujuan
18.	(Abu Zohair, 2019), "***Prediction of Student's performance by modelling small dataset size",*** Educational Technology in Higher Education (2019) 16:27 https://doi.org/10.1186/s41239-019-0160-3	*SVM, LDA, MLP-ANN, Naive Bayes, KNN*	Membuat model prediksi. Mengevaluasi akurasi berbagai algoritma ML
19.	(Çakıt & Dağdeviren, 2021)**,*" Predicting the percentage of student placement: A comparative study of machine learning algorithms***", Education and Information Technologies (2022) 27:997–1022, https://doi.org/10.1007/s10639-021-10655-4	Algoritma *Machine Learning* (ML)	Mengembangkan model prediksi, membandingkan kinerja berbagai algoritma ML dan identifikasi variabel paling signifikan
20.	(Mubarak, Cao, & Hezam, 2022), "***Modeling students' performance using graph convolutional networks",*** Complex & Intelligent Systems (2022) 8:2183–2201 https://doi.org/10.1007/s40747-022-00647-3	Model *based on Graph Convolutional Network*	Mengklasifikasikan keterlibatan siswa dalam pola perilaku dalam kegiatan pembelajaran *online*
21.	(Wu et al., 2022)," ***SGKT: Session graph-based knowledge tracing for student performance prediction***", Expert Systems with Applications, 206 (2022) 117681, doi: 10.1016/j.eswa.2022.117681	*Graph Convolutional Network Gated, and Graph Neural Networks*	Mengusulkan penelusuran pengetahuan baru tentang siswa yang didasarkan: *Session Graph Based Knowledge Tracing* (SGKT).

Seluruh artikel pada Tabel 2.2 menggunakan pendekatan pembelajaran mesin terpandu sebagai dasar pengembangan model prediksi, sedangkan disertasi ini mengusulkan pendekatan melalui pemanfaatan pembelajaran semi-terpandu *co-training* LSTM–SVM. Artikel-artikel di Tabel 2.2 mengimplementasikan algoritma pembelajaran mesin terpandu seperti *Decision Tree*, *Random Forest*, SVM, KNN, *Naive Bayes*, *Logistic Regression*, atau ANN, sementara disertasi ini menggabungkan kemampuan representasi sekuensial dari LSTM dan *margin based learning* dari SVM dalam pembelajaran semi-terpandu. Penelitian disertasi juga mengatasi keterbatasan data berlabel, karena model yang dikembangkan belajar dari data berlabel sekaligus tidak berlabel melalui mekanisme *pseudolabeling*, sedangkan artikel-artikel pada Tabel 2.2 sepenuhnya bergantung pada *dataset* berlabel penuh. Disertasi ini memproses data temporal dan perilaku mahasiswa

menggunakan LSTM, suatu pendekatan yang masih jarang digunakan dalam penelitian pendidikan, khususnya pada konteks klasifikasi potensi mahasiswa. Disertasi ini menghadirkan kebaruan metodologis dan praktis, karena penelitian ini mengembangkan model pembelajaran semi-terpandu yang tidak ditemukan pada artikel-artikel pembelajaran terpandu, sekaligus mengalihkan fokus dari prediksi nilai atau risiko *dropout* menjadi klasifikasi potensi akademik mahasiswa, yang belum dibahas secara mendalam oleh penelitian sebelumnya.

Tabel 2.3 Artikel Pembelajaran Mesin dengan Metode *Hybrid/Ensemble*

No	Penulis, Tahun, Judul Artikel	Metodologi	Tujuan
1.	(Al-Tameemi et al., 2024),"***Hybrid Machine Learning Approach for Predicting Student Performance Using Multi-class Educational Datasets***",The 14th International Symposium on Frontiers in Ambient and Mobile Systems (FAMS 2024)The 14th International Symposium on Frontiers in Ambient and Mobile Systems (FAMS 2024) April 23-25, 2024, Hasselt, Belgiumt	*Decision Tree, Random Forest, SVM, dan KNN***.**	Mengembangkan pendekatan hibrid berbasis ML yang dapat meningkatkan akurasi performa akademik mahasiswa, dgn mengombinasi keunggulan dari beberapa algoritma ML.
2.	(Singh & Pal, 2020)**,"*Machine Learning Algorithms and Ensemble Technique to Improve Prediction of Students Performance"***, International Journal of Advanced Trends in Computer Science and Engineering (2020), Vol.9, May-June 2020, https://doi.org/10.30534/ijatcse/2020/221932020	*Decision Tree, Random Forest, Support Vector Machine, Naive Bayes, K-Nearest Neighbor*	Mengembangkan model prediksi performa akademik mahasiswa dengan beberapa pembelajaran mesin dan teknik *ensemble*
3.	(Butt et al., 2023), ***"Performance Prediction of Students in Higher Education Using Multi-Model Ensemble Approach",*** IEEE Access (2023), Vol.11, Issue December, pages 136091-136108, doi:10.1109/ACCESS.2023.3336987	*Decision Tree, Random Forest, Support Vector Machine, Logistic Regression, K-Nearest Neighbor*	Meningkatkan akurasi prediksi performa akademik dengan menggunakan pendekatan multi model *ensemble* berbasis pembelajaran mesin.

Tabel 2.3 Artikel Pembelajaran Mesin dengan Metode *Hybrid/Ensemble* (Lanjutan)

No	Penulis, Tahun, Judul Artikel	Metodologi	Tujuan
4.	(Sokkhey & Okazaki, 2020)**, "*Hybrid Machine Learning Algorithms for Predicting Academic Performance"***, International Journal of Advanced Computer Science and Applications (2020), Vol.11 Issue 1, pages 32-41, www.ijacsa.thesai.org 33	*Decision Tree, Random Forest, Support Vector, Machine (SVM), Logistic Regression*	Meningkatkan akurasi prediksi performa akademik mahasiswa dengan menggunakan pendekatan *hybrid machine learning,*
5.	(Fida et al., 2022)**, *"A Novel Hybrid Ensemble Clustering Technique for Student Performance Prediction"***, Journal of Universal Computer Science (2022), Vol.28, Issue 8, pages 777-798, doi 10.3897/jucs.73427	*Clustering, classification*	Membangun *smarter dataset* dengan seleksi fitur yang akurat. Mengembangkan teknik *hybrid ensemble* untuk meningkatkan hasil prediksi performa siswa. Menggabungkan keunggulan *clustering & ensemble classification* untuk menurunkan false negatif dan meningkatkan *precision.* Mengatasi gap dari dua penelitian dasar sebelumnya (Almasri 2019 & Francis 2019
6.	(Kaur et al., 2021)*," **Ensemble learning-based prediction of contentment score using social multimedia in education",*** Multimedia Tools and Applications (2021) 80:34423–34440 https://doi.org/10.1007/s11042-021-10806-2	Metode *Cuckoo Search meta-heuristic* untuk memilih fitur penting dari *dataset*	Memprediksi kepuasan mahasiswa dengan pendekatan *ensemble learning*
7.	(Mienye et al., 2022), ***"A Survey of Ensemble Learning : Concepts , Algorithms , Applications , and Prospects"***, EEE Access, 2022, Vol.10, Agustus, pages. 99129-99149, doi.10.1109/ACCESS.2022.3207287	*Random Forest*, AdaBoost, Gradient Boosting, XGBoost, LightGBM, CatBoost, CNN *ensembles*	Memberikan gambaran komprehensif tentang *ensemble learning*

Tabel 2.3 Artikel Pembelajaran Mesin dengan Metode *Hybrid/Ensemble* (Lanjutan)

No	Penulis, Tahun, Judul Artikel	Metodologi	Tujuan
8.	(Hanisah et al., 2023), ***"Comparison of ensemble hybrid sampling with bagging and boosting machine learning approach for imbalanced data"***, Indonesian Journal of Electrical Engineering and Computer Science. 2023, Vol.29, Issue 1, Doi.10.11591/ijeecs.v29.i1.pp598-608	*Ensemble Random Forest* dan *Gradient Boosting*	Menganalisis dampak ketidakseimbangan kelas model klasifikasi. Membandingkan performa model-model ML konvensional vs. ensemble (RF & GB).Mengukur efek tiga teknik sampling
9.	(Ali et al., 2024), ***"A Hybrid Deep Learning Model to Predict High-Risk Students in Virtual Learning Environments"***, IEEE Access, 2024, Juli, pages.103687-103703	*Hybrid deep learning*	Memprediksi siswa berisiko tinggi dalam lingkungan pembelajaran virtual.
10.	(Pek et al., 2023), ***"The Role of Machine Learning in Identifying Students At-Risk and Minimizing Failure"***, IEEE Access, 2023, Vol.11, Dec 2022, doi.10.1109/ACCESS.2022.3232984	Teknik *ensemble stacking*	Memprediksi siswa yang berisiko gagal (*risk students*) lebih awal.
11.	(Dianah et al., 2021), " Multiclass Prediction Model for Student Grade Prediction Using Machine Learning", IEEE Access, 2021, Vol.9, doi. 10.1109/ACCESS.2021.3093563	J48, SVM, NB, kNN, *Logistic Regression*, *Random Forest*	Mengidentifikasi model prediksi terbaik untuk mengklasifikasikan nilai akhir mahasiswa

Artikel-artikel pada Tabel 2.3 menggunakan berbagai pendekatan *hybrid* dan *ensembl*e seperti *bagging, boosting, stacking, Random Forest*, dan *Gradient Boosting* sebagai dasar dalam meningkatkan akurasi prediksi performa mahasiswa, sedangkan disertasi ini mengembangkan pendekatan *hybrid* pembelajaran semi-terpandu melalui mekanisme *co-training* LSTM–SVM. Penelitian yang tertulis di artikel Tabel 2.3 menerapkan model *hybrid* yang sepenuhnya bersifat pembelajaran terpandu, sementara disertasi ini menggabungkan *deep learning* dan *machine learning* dalam *co-training* yang mampu memanfaatkan data berlabel maupun tidak berlabel secara bersamaan. Disertasi ini juga memanfaatkan LSTM sebagai *sequence learning* model untuk menangkap pola temporal mahasiswa, sedangkan artikel-artikel sebelumnya lebih banyak menggunakan *ensemble supervised*

tradisional yang tidak dirancang untuk merepresentasikan data sekuensial. Selain itu, penelitian-penelitian pada Tabel 2.3 mengolah data melalui teknik *sampling* atau *feature selection*, sedangkan disertasi ini menggunakan *pseudolabeling* dan *iterative training* untuk memperkaya data latih secara otomatis. Dari sisi tujuan, artikel-artikel *hybrid* sebelumnya berfokus pada prediksi nilai atau risiko gagal, sedangkan disertasi ini mengalihkan orientasi menuju klasifikasi potensi mahasiswa, yang memberikan dimensi baru dalam pemanfaatan pembelajaran mesin di pendidikan tinggi. Dengan demikian, penelitian disertasi ini mengisi gap besar dalam literatur, karena belum ada studi *hybrid/ensemble* yang menerapkan pendekatan pembelajaran semi-terpandu maupun integrasi LSTM–SVM dalam skema *co-training.*

Seluruh artikel pada Tabel 2.1, 2.2, dan 2.3 menunjukkan bahwa penelitian sebelumnya didominasi oleh pendekatan pembelajaran terpandu dan pembelajaran *hybrid*, sedangkan penelitian disertasi ini mengembangkan model baru berbasis pembelajaran semi-terpandu melalui *co-training* LSTM–SVM dengan pelabelan semu. Artikel-artikel pada Tabel 2.1 berfokus pada penyusunan tinjauan pustaka dan pemetaan metode pembelajaran mesin, sedangkan disertasi ini menghasilkan arsitektur model prediksi yang utuh, sehingga berada di luar kategori penelitian berbasis tinjauan pustakan. Artikel-artikel pada Tabel 2.2 menerapkan algoritma pembelajaran terpandu seperti *Decision Tree*, *Random Forest*, SVM, KNN, *Naive Bayes*, *Logistic Regression*, dan ANN, sementara disertasi ini menggabungkan *deep learning* (LSTM) dan pembelajaran mesin (SVM) dalam teknik *co-training* yang mampu memanfaatkan data berlabel dan tidak berlabel secara simultan. Artikel-artikel pada Tabel 2.3 mengembangkan *hybrid* atau *ensemble* berbasis pembelajaran terpandu melalui *bagging, boosting, stacking,* dan kombinasi multi-model, tetapi tidak ada satupun yang menggunakan pembelajaran semi-terpandu *hybrid, co-training,* atau integrasi LSTM–SVM seperti pada penelitian disertasi ini. Penelitian sebelumnya juga berfokus pada prediksi nilai, risiko gagal, atau *dropou*t, sedangkan disertasi ini mengalihkan orientasi ke klasifikasi potensi mahasiswa, yang memberikan kontribusi baru dalam analitik pendidikan. Penelitian disertasi

ini mengatasi keterbatasan data berlabel, yang merupakan masalah umum di hampir sebagian besar perguruan tinggi Indonesia, dengan menerapkan mekanisme *pseudolabeling* dan *iterative learning*, sedangkan seluruh penelitian sebelumnya bergantung pada *dataset* berlabel penuh. Dengan demikian, disertasi ini mengisi gap yang jelas dalam literatur, karena tidak ada penelitian terdahulu pada ketiga tabel tersebut yang menggabungkan pembelajaran semi-terpandu, *co-training* LSTM -SVM, dan *pseudolabeling* untuk membangun model identifikasi potensi mahasiswa.

Tabel 2.4 menyajikan secara garis besar perbedaan (*gap*) antara penelitian disertasi dengan penelitian lainnya yang telah ditelaah di Tabel 2.1, Tabel 2.2, dan Tabel 2.3.

Tabel 2.4 Gap Penelitian Disertasi dengan Penelitian Pembelajaran Mesin Lainnya

No	**Aspek Penelitian**	**Penelitian Terdahulu (Gap yang ditemukan)**	***Research Gap***	**Kontribusi Disertasi**
1.	Jenis Pembelajaran	Semua penelitian menggunakan pembelajaran mesin terpandu atau *ensemble supervised* (DT, RF, SVM, NB, ANN, *boosting, bagging, stacking*).	Belum ada penelitian yang menerapkan pembelajaran mesin semi-terpandu dalam prediksi potensi/performa mahasiswa.	Mengembangkan Pembelajaran semi-terpandu *hybrid* melalui *co-training* LSTM–SVM.
2.	Integrasi Model	Tidak ada penelitian yang menggabungkan LSTM + SVM dalam satu proses	Diperlukan integrasi model *deep learning* dan *traditional* ML memanfaatkan kekuatan keduanya.	Mengusulkan model LSTM–SVM dalam proses *co-training* iteratif.
3.	Pemanfaatan Data Tidak Berlabel	Semua penelitian membutuhkan data berlabel penuh. Tidak ada *pseudo-labeling*.	Perguruan Tinggi Indonesia memiliki keterbatasan data berlabel, sehingga butuh model yang bisa memanfaatkan data tidak berlabel.	Disertasi menerapkan *pseudo-labeling* dan pembelajaran iteratif untuk memanfaatkan data tidak berlabel.
4.	Arsitektur *Hybrid*	Penelitian *hybrid* sebagian besar berbasis ***bagging, boosting, stacking***, bukan pembelajaran mesin semi-terpandu *hybrid*.	Belum ada model pembelajaran semi-terpandu *hybrid* di pendidikan.	Disertasi menghadirkan *hybrid co-training* antara LSTM dan SVM.

Tabel 2.4 Gap Penelitian Disertasi dengan Penelitian Pembelajaran Mesin Lainnya (Lanjutan)

No	Aspek Penelitian	Penelitian Terdahulu (Gap yang ditemukan)	*Research Gap*	Kontribusi Disertasi
5.	Jenis Data dan Sifat Fitur	Penelitian sebelumnya tidak fokus pada data temporal/sekuensial.	Data aktivitas mahasiswa bersifat sekuensial sehingga butuh *sequence* model.	Disertasi memproses data dengan LSTM untuk menangkap pola temporal.
6.	Tujuan Prediksi	Fokus pada prediksi nilai, *risk*, *dropout*, atau rekomendasi jurusan	Belum ada penelitian yang fokus pada identifikasi potensi mahasiswa.	Disertasi berkontribusi pada identifikasi potensi akademik mahasiswa untuk pengembangan diri
7.	Konteks Pendidikan Indonesia	Hampir tidak ada penelitian yang membahas konteks Perguruan Tinggi Indonesia dengan keterbatasan label.	Dibutuhkan model yang cocok untuk kondisi PT lokal	Disertasi menghadirkan solusi yang relevan untuk PT Indonesia dengan rendah label
8.	Bentuk Output	Kebanyakan penelitian hanya menghasilkan model prediksi dalam bentuk eksperimen.	Dibutuhkan model yang dapat diimplementasikan dan diuji secara nyata.	Disertasi menghasilkan web-app *co-training* LSTM-SVM yang bisa dipakai institusi secara praktis.

2.2 Keaslian (Originalitas) Penelitian

Penelitian disertasi ini menganalisis bagaimana cara mengetahui potensi mahasiswa yang dimiliki perguruan tinggi menggunakan metode pembelajaran semi-terpandu. Keaslian penelitian ini terletak pada tiga aspek utama:

1. Metode *co-training* LSTM-SVM diterapkan untuk mengklasifikasi potensi mahasiswa di perguruan tinggi, dengan menggabungkan kemampuan LSTM untuk data sekuensial dalam disertasi ini adalah data akademis IPK, prestasi, keaktifan organisasi dan SVM untuk klasifikasi dalam disertasi ini menggunakan data gaji orang tua, dan jumlah saudara.
2. *Pseudolabeling* diterapkan pada data latih sebagai jembatan agar untuk mengisi kekurangan label pada data mahasiswa agar menghasilkan model sesuai dengan karakteristik dan kebijakan masing-masing perguruan tinggi. *Pseudolabeling* dalam pembelajaran semi-terpandu digunakan untuk meningkatkan akurasi dengan memanfaatkan data tidak berlabel.

3. Nilai *confident prediction* dibutuhkan untuk memberi keyakinan pada algoritma untuk mengklasifikasi data ke kelas atau label.

2.3 Pembelajaran Mesin

Pembelajaran Mesin atau (*machine learning*) merupakan serangkaian teknik yang dapat membantu dalam menangani dan memprediksi data yang sangat besar dengan cara mempresentasikan data-data tersebut dengan algoritma pembelajaran (Wilson & Anwar, 2024). Istilah pembelajaran mesin pertama kali didefinisikan oleh Arthur Samuel pada tahun 1959. Menurut Arthur Samuel, pembelajaran mesin adalah suatu bidang ilmu komputer yang memberikan kemampuan pembelajaran kepada komputer untuk mengetahui sesuatu tanpa pemrograman yang jelas. Algoritma pembelajaran mesin dapat dilakukan dengan menggunakan berdasarkan Python, Matlab, dan bahasa lainnya, dimana sebagai inputan dapat berupa data tak terpandu (*unsupervised*), data terpandu (*supervised)*, dan data *reinforcement*.

Pembelajaran-semi-terpandu (*semi supervised*) dapat dianggap sebagai gabungan antara pembelajaran tak terpandu dengan pembelajaran terpandu. Algoritma ini akan dimanfaatkan untuk menyelesaikan permasalahan klasifikasi (*classification*) dan kluster (*clustering*). Pembelajaran mesin menggunakan data sebelumnya sebagai pengalaman untuk meningkatan performa model dan melakukan prediksi yang akurat (IBM, 2021). Dalam pembelajaran mesin, beberapa pendekatan utama yang digunakan untuk mengembangkan model prediktif (M. E. L. Hassani et al., 2025) yaitu:

1. Pembelajaran Terpandu (*Supervised Learning*)

   Supervised learning menggunakan data berlabel untuk melatih model. Model mempelajari hubungan antara input dan output yang sudah diketahui, lalu memanfaatkan pola tersebut untuk memprediksi atau mengklasifikasikan data baru. Dalam praktiknya, pendekatan ini banyak digunakan untuk tugas seperti klasifikasi email *spam* atau deteksi penyakit dari citra medis, karena model dapat belajar secara langsung dari contoh yang sudah diberi label kebenaran.
2. Pembelajaran Tak Terpandu (*Unsupervised Learning*)

Unsupervised learning bekerja dengan data tanpa label. Algoritma mencoba menemukan struktur tersembunyi dalam data dengan cara mengidentifikasi kelompok (*clustering*) atau pola yang serupa, serta melakukan reduksi dimensi untuk menyederhanakan kompleksitas data. Pendekatan ini bermanfaat saat informasi awal terbatas, misalnya dalam segmentasi pelanggan berdasarkan perilaku pembelian tanpa label eksplisit.

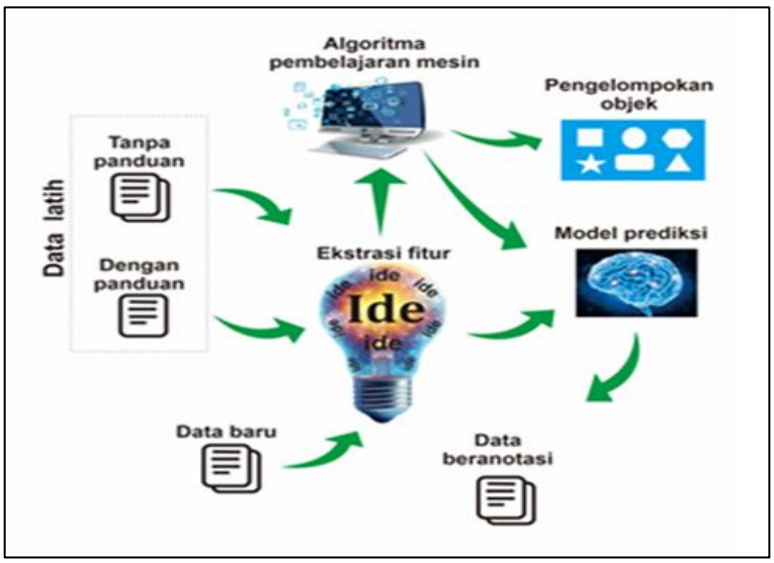
3. Pembelajaran Semi Terpandu (*Semi-Supervised Learning*)

   Semi-supervised learning mengombinasikan data berlabel dalam jumlah kecil dengan data tidak berlabel dalam jumlah besar. Model menggunakan informasi dari data berlabel untuk memandu proses pembelajaran dari data tak berlabel. Pendekatan ini sangat berguna ketika proses anotasi data mahal atau memakan waktu, karena algoritma tetap bisa memanfaatkan *dataset* besar tanpa memerlukan pelabelan manual yang banyak.

4. Pembelajaran Penguatan (*Reinforcement Learning*)

   Reinforcement learning melibatkan agen yang belajar melalui interaksi langsung dengan lingkungan. Agen mengambil keputusan, menerima *reward* atau *punishment*, lalu memperbarui strategi untuk memaksimalkan *reward* kumulatif. Pendekatan ini banyak diterapkan pada bidang robotika, *game*, dan sistem rekomendasi, karena memungkinkan model belajar secara adaptif berdasarkan umpan balik dari lingkungannya.

Gambar 2.1 menunjukan sejumlah data latih dari skenario pembelajaran terpandu, pembelajaran tak-terpandu, dan *reinforcement learning* yang akan menjadi inputan bagi mesin komputer untuk belajar dan bekerja sesuai dengan algoritma dari tiap skenario yang ada, sehingga akan menghasilkan outputan sesuai model prediksi.



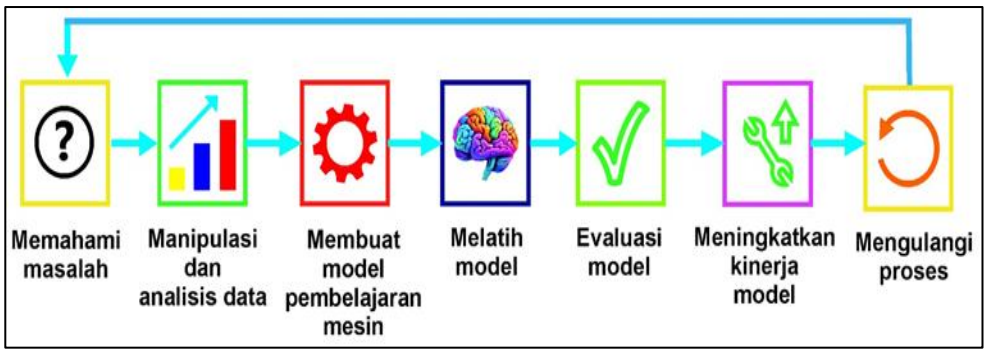
Gambar 2.1 Pembelajaran mesin (Pantech, 2018)

Gambar 2.1 menyajikan proses pembelajaran mesin menggunakan data latih sebagai sumber utama untuk membangun model. Data latih terdiri dari data dengan panduan dan data tanpa panduan yang digunakan sesuai kebutuhan metode pembelajaran. Algoritma pembelajaran mesin memproses data latih untuk melakukan ekstraksi fitur yang menghasilkan representasi informasi penting dari data. Proses ekstraksi fitur menyediakan ide-ide atau pola yang dibutuhkan untuk tahap analisis selanjutnya. Algoritma pembelajaran mesin membentuk pengelompokan objek berdasarkan kesamaan fitur yang muncul pada data. Algoritma pembelajaran mesin juga menghasilkan model prediksi dengan memanfaatkan fitur yang telah diekstraksi. Model prediksi memproses data beranotasi untuk meningkatkan kemampuan pemodelan dalam mengenali pola. Model prediksi kemudian memproses data baru untuk menghasilkan keluaran yang sesuai dengan pola pembelajaran sebelumnya. Alur keseluruhan menunjukkan bahwa proses pembelajaran mesin mengubah data mentah menjadi fitur, kemudian menjadi pengelompokan objek atau model prediksi yang siap digunakan untuk analisis lanjutan.

Salah satu skenario pembelajaran mesin yang akan digunakan dalam penelitian ini adalah penggabungan antara pembelajaran terpandu dengan

pembelajaran tak-terpandu atau yang sering disebut pembelajaran semi-terpandu atau semi *supervised learning*.

Pada Gambar 2.2 tampak proses pembelajaran mesin untuk memahami masalah yang ingin diselesaikan sebagai dasar pembangunan model (De Souza Nascimento et al., 2019). Pengumpulan dan manipulasi data agar data yang relevan dapat dianalisis sesuai kebutuhan penelitian (Roh et al., 2021). Pembuatan model pembelajaran mesin yang sesuai dengan jenis data dan tujuan klasifikasi atau prediksi (Black et al., 2023). Model yang dihasilkan akan dilatih menggunakan data latih agar parameter internal dapat diperbarui dan kesalahan prediksi dapat diminimalkan (Ilyas et al., 2022). Evaluasi model dengan menggunakan matrik seperti akurasi, *presisi*, *recall*, dan F1-*score* untuk menilai performa prediksi (Yacouby & Axman, 2020). Evaluasi model memungkinkan peningkatan kinerja model dengan teknik seperti *tuning* parameter, penambahan fitur baru, atau pemilihan algoritma alternatif (Raschka, 2020). *Tuning* parameter adalah proses mengoptimalkan parameter dari suatu model untuk meningkatkan kinerjanya. Teknik ini sangat penting dalam pembelajaran mesin karena parameter yang tepat dapat membuat perbedaan besar dalam seberapa baik model dapat menggeneralisasi dan membuat prediksi yang akurat pada data baru. Pengulangan proses pembelajaran mesin secara iteratif untuk memperbaiki hasil, memperbarui data, dan mengoptimalkan akurasi model sesuai kebutuhan aplikasi nyata (Jhaveri et al., 2022).



Gambar 2.2 Cara Kerja Pembelajaran Mesin

2.3.1 Pembelajaran Semi-Terpandu

Pembelajaran semi-terpandu dapat diartikan sebagai salah satu jenis pembelajaran mesin dengan melibatkan data berlabel dalam jumlah kecil dan data tak berlabel dengan jumlah besar. Pembelajaran mesin yang ada biasanya membutuhkan banyak waktu untuk memberi label pada data yang jumlahnya banyak. Pembelajaran semi-terpandu hanya memberikan label pada sedikit data latih sehingga proses pengolahan data akan lebih mudah berkat ketersediaan algoritma ini. Dua metode yang biasa dilakukan, yakni *self-training* dan *co-training* (AltexSoft, 2024).

Self-training merupakan metode semi-terpandu yang menggunakan satu model untuk memprediksi/mengklasifikasi label pada data tak berlabel. Model menghasilkan *pseudolabel* berdasarkan tingkat kepercayaan tertentu dan memasukkan *pseudolabel* tersebut ke dalam siklus pembelajaran berikutnya. Proses ini membuat model mengandalkan prediksi dirinya sendiri sehingga model berisiko memperkuat kesalahan apabila *pseudolabel* yang dihasilkan tidak akurat. *Self-training* bekerja efektif pada permasalahan yang memiliki struktur data sederhana dan hubungan antar fitur yang tidak kompleks (B. Chen et al., 2022).

(Amini et al., 2025) menyatakan *self-training* dapat dilakukan untuk proses klasifikasi dan regresi. Keunggulannya adalah dapat memanfaatkan data yang sudah terlabel maupun yang tidak. Jika menggunakan metode ini, maka langkah-langkahnya yang perlu dilakukan antara lain:

1. Memilih data berukuran kecil dari data yang sudah terlabel, misalnya saja data yang menunjukkan gambar secara spesifik mengenai sebuah objek. Gambar yang ada dapat menjadi model dasar untuk membantu proses dan metode selanjutnya.
2. Kemudian, dilanjutkan dengan proses yang disebut dengan *pseudolabeling*. Proses ini merupakan pengklasifikasian data yang tidak terlabel ke data yang sudah terlabel.

3. Proses terakhir adalah membuat dataset baru, di mana data ini berasal dari data yang sudah terlabel maupun dari *pseudolabeling*. Langkah dilanjutkan dengan improvisasi *dataset* hingga proses prediksi.

Self-training memulai dengan melatih model menggunakan sejumlah kecil data berlabel dan menghasilkan *pseudolabel* untuk data tak berlabel. *Pseudolabel* yang memiliki level keyakinan tinggi diintegrasikan ke dalam *dataset* latih untuk memperluas data pelatihan dan meningkatkan presisi model.

Metode *co-training*, sebenarnya lebih kompleks jika dibandingkan dengan *self-training*. Metode *co-training* cukup efektif untuk pengklasifikasian *website* (Chen & Wang, 2024). Langkah yang harus dilakukan di *co-training* adalah sebagai berikut:

1. Mengklasifikasikan model secara terpisah untuk tiap tampilan yang menggunakan sejumlah kecil data yang sudah terlabel.
2. Kemudian, sejumlah data yang tidak berlabel dan lebih besar bisa ditambahkan ke penerima label semu.
3. Proses selanjutnya adalah *pseudolabeling*. Di sinilah data yang telah diklasifikan pada tahap pertama dan kedua akan dilihat, mana yang memiliki kesalahan dan tidak. Terakhir adalah mengombinasikan prediksi dari dua klasifikasi data yang sudah *update* untuk hasil akhirnya.

Co-training mengembangkan pendekatan *self-training* dengan melatih dua *classifier* secara terpisah menggunakan fitur yang berbeda dari data berlabel. Lalu, *classifier* yang memiliki hasil paling meyakinkan dalam melabelkan data secara semu model lainnya. Proses ini diulang secara iteratif, sehingga kedua model saling memperkuat kemampuannya (Chen & Wang, 2024).

2.3.2 *Co-training*

Co-training adalah metode pembelajaran semi-terpandu yang memanfaatkan dua "*view*" independen dari data untuk memperluas set pelabelan secara bergantian (Rothenberger & Diochnos, 2024). *Co-training* menggunakan asumsi dua *view* kondisional yang saling bebas dimana masing-masing subset fitur X^A dan X^B harus

cukup untuk memprediksi label sendiri serta melakukan iterasi pelabelan dengan melatih dua *classifier* secara terpisah. Penelitian disertasi ini memilih menggunakan metode *co-training* untuk diuji cobakan *co-training* adalah teknik pembelajaran semi-terpandu yang membutuhkan dua tampilan data. Satu karakteristik utama dari metode *co-training* ini adalah penggunaan data berlabel dalam jumlah kecil dan data tak berlabel dalam jumlah besar (Nassar et al., 2021a). Teknik ini mengasumsikan bahwa setiap contoh dijelaskan menggunakan dua set fitur berbeda yang menyediakan informasi pelengkap tentang instans tersebut. Idealnya, kedua tampilan tersebut independen secara kondisional (yaitu, dua set fitur dari setiap instans independen secara kondisional berdasarkan kelasnya) dan setiap tampilan mencukupi (yaitu, kelas instans dapat diprediksi secara akurat dari setiap tampilan saja). *Co-training* akan mempelajari pengklasifikasi terpisah untuk setiap tampilan menggunakan contoh berlabel apa pun. Hasil paling meyakinkan dari setiap pengklasifikasi pada data yang tidak berlabel kemudian digunakan untuk secara berulang membangun data pelatihan berlabel tambahan. *Pseudolabelling* dalam semi-terpandu dapat menurunkan kualitas pelabelan karena tidak mewakili kelas yang ada dalam kumpulan data berlabel.

Metode *co-training* dapat memanfaatkan semantik label untuk mengatasi masalah ini (Nassar et al., 2021b) Kerangka *co-training* dapat menyelesaikan pembelajaran *multiview* dengan menggunakan *pseudolabel* sebagai representasi tambahan untuk meningkatkan hasil klasifikasi (W. Zhang et al., 2022). Studi eksperimental yang komprehensif telah dilakukan untuk menunjukkan efektivitas *co-training* pada pembelajaran semi-terpandu untuk multi label (Zhan & Zhang, 2017). *Co-training* mendapatkan popularitas di komunitas riset dan industri, dan telah berhasil diterapkan di sejumlah aplikasi dunia nyata (Zhou, 2018). Metode *co-training* juga telah digunakan untuk *Transfer Learning* (Ning et al., 2020). *Co-training* adalah topik baru dalam pembelajaran mesin yang bertujuan untuk mengekstrak pengetahuan yang diperoleh dalam tugas/domain sumber dan menggunakannya untuk memfasilitasi pembelajaran fungsi prediktif target dalam tugas/domain yang berbeda. *Co-training* memiliki kemampuan untuk

meningkatkan kemampuan generalisasi model dengan memanfaatkan informasi dari data yang tidak berlabel. *Co-training* memungkinkan mesin untuk memikirkan dari berbagai perspektif seperti manusia dengan membagi data ke dalam beberapa pandangan, merancang pembelajar secara ilmiah, dan mengestimasi kepercayaan label secara akurat. Metode ini juga dapat meningkatkan akurasi klasifikasi dan konvergensi model. Penelitian terkait tema *co-training* tahun 2016 oleh (Yang et al., 2016) yang melakukan penelitian di bidang geografi dengan memanfaatkan dua model pembelajaran terpisah data berlabel digunakan untuk memberikan label pada data yang belum dilabeli. *Co-training* membantu membantu kasus SRL (*Self Regulated Learning*) tersulit dengan data label terbatas (Ngoc et al., 2016). Penelitian pada tahun 2017 menggunakan metode *co-training* pada pembelajaran semi-terpandu untuk multi-label *learning*, diterapkan pada data yang memiliki banyak label terkait satu sampel. Metode *co-training* digunakan untuk meningkatkan akurasi dengan menggunakan dua *classifier* yang berbeda, yang masing-masing dilatih pada subset fitur yang berbeda dan kemudian saling bertukar informasi untuk meningkatkan label pada data yang tidak berlabel (Zhan & Zhang, 2017). Penelitian di bidang kesehatan menerapkan *co-training* dan pelabelan semu (Zhou, 2018). Seleksi sampel adaptif dengan aturan tetap yang menghasilkan akurasi klasifikasi meningkat (Wu, Li, & Wang, 2018). Penelitian *co-training* menggunakan *single-view* untuk menyeleksi *high confidence* (J. Chen et al., 2019). Penelitian *co-training* yang bertujuan mengurangi *noise* pada *pseudolabel* dengan menggunakan *self-paced* (Ma et al., 2020). Artikel yang menerapkan *co-training* dalam bidang ilmu data (Likhosherstov et al., 2021), (Ning et al., 2021) dalam bidang geologi, dan (C. Li, Dong et al., 2021) dalam bidang kesehatan terkait dengan Covid 19. Penelitian yang mengkombinasikan *self-supervised* dengan *co-training* (Xia et al., 2021). Penelitian *co-training* yang memperbaiki kekokohan *few/zeroshot prompting* (Lang et al., 2022). Penelitian yang membahas bagaimana menghentikan *co-training* saat keuntungan marjinal menurun untuk menghindari *error amplification* (Grolman et al., 2022). Penelitian *co-training* yang memilih *pseudolabel* yang memiliki nilai *confidence* tertinggi (Shen et al., 2022). Penelitian

co-training digunakan untuk segmentasi medis (H. Xie et al., 2023) dan untuk klasifikasi gambar (M. Chen et al., 2022). Penelitian *co-training* yang membangun views otomatis dan menghindari *retraining* penuh (Rothenberger & Diochnos, 2025). Penelitian dengan metode *co-training* dilakukan untuk meningkatkan akurasi klasifikasi dan konvergensi model *co-training* dapat digunakan dalam berbagai tugas penelitian seperti klasifikasi kesalahan, dan identifikasi orang berdasarkan audio-visual (Ning et al., 2021). Penelitian disertasi ini dimulai di tahun 2024 menerapkan *co-training* LSTM-SVM dengan menentukan nilai *confidence* 0,9 yang digunakan untuk mengklasifikasikan potensi mahasiswa di perguruan tinggi. Ringkasan penelitian tentang *co-training* ditampilkan dalam Tabel 2.5.

Tabel 2.5 Penelitian- Penelitian *Co-training*

Tahun	Penulis	Keterangan / Tujuan Penelitian
2004	Chen et al.	Menguji klasifikasi gambar
2016	Yang et al.	Menerapkan *co-training* di bidang geografi dengan memanfaatkan dua model pembelajaran terpisah untuk melabeli data tak berlabel.
2016	Do et al.	Menggunakan *co-training* untuk membantu kasus *Self Regulated Learning* (SRL) dengan data label terbatas.
2017	Zhan & Zhang	Menerapkan *co-training* pada semi-terpandu *multi-label learning* dengan dua *classifier* berbeda yang saling bertukar informasi.
2018	Zhou	Menggunakan *co-training* dan pelabelan semu di bidang kesehatan.
2018	Wu et al.	Mengembangkan seleksi sampel adaptif berbasis aturan tetap yang meningkatkan akurasi klasifikasi.
2020	J. Chen et al.	Menerapkan *co-training* dengan *single view* untuk menyeleksi data ber*confidence* tinggi.
2020	Ma et al.	Menggunakan *co-training* dengan strategi self-paced untuk mengurangi *noise* pada *pseudolabel*.
2020	Jesper et al.	Menjelaskan karakteristik utama *co-training* dengan data berlabel sedikit dan data tak berlabel besar.
2021	Likhosherstov et al.	Mengaplikasikan *co-training* dalam bidang ilmu data.
2021	Ning et al.	Menerapkan *co-training* dalam bidang geologi.
2021	C. Li et al.	Menggunakan *co-training* di bidang kesehatan terkait Covid-19.
2021	Xia et al.	Mengombinasikan *self supervised learning* dengan *co-training*.
2022	Lang et al.	Memperbaiki *robustnes co-training* untuk *few-shot* dan *zero-shot prompting.*
2022	Grolman et al.	Membahas strategi menghentikan *co-training* saat keuntungan marjinal menurun agar tidak terjadi *error amplification*.
2023	Shen et al.	Menerapkan *co-training* untuk menyeleksi *pseudolabel* dengan *confidence* tertinggi.
2023	H. Xie et al.	Menggunakan *adversarial co-training* untuk segmentasi medis.
2023	M. Chen et al.	Menerapkan *co-training* untuk klasifikasi gambar.
2023	Ning et al.	Menggunakan *co-training* untuk klasifikasi kesalahan dan identifikasi audio-visual.
2024	Rothenberger & Diochnos	Membangun views otomatis dalam *co-training* dan menghindari *retraining* penuh.
2024	Handayani et al	Menerapkan *co-training* LSTM–SVM dengan *threshold confidence* 0,9 untuk memprediksi potensi mahasiswa di perguruan tinggi.

Ringkasan penelitian tentang *co-training* ditampilkan dalam Tabel 2.5. Proses *co-training* umumnya melibatkan tiga langkah utama: *view acquisition, learner differentiation,* dan *label confidence estimation* (M.Bishop, 2019). *View acquisition* bertujuan untuk memperoleh dua tampilan independen dan cukup dari data, yang masing-masing dapat digunakan oleh model pembelajaran yang berbeda. Tampilan independen tidak tersedia secara alami, tampilan tersebut dapat dibangun menggunakan model pra-terlatih (Rothenberger & Diochnos, 2024). Tampilan selanjutnya dilakukan *learner differentiation* dimana dua model pembelajaran yang berbeda dilatih secara terpisah pada masing-masing tampilan. Pendekatan ini memastikan bahwa setiap model mempelajari representasi unik dari data, yang meningkatkan kemampuan model untuk saling melengkapi dan memperbaiki kesalahan satu sama lain (Tan et al., 2021). Pada tahap label *confidence estimation*, setiap model membuat prediksi pada data tanpa label dan menilai kepercayaan model terhadap hasil tersebut. Hasil dengan tingkat kepercayaan tinggi kemudian digunakan sebagai label semu untuk melatih model lain, memungkinkan kedua model untuk saling memperkuat dan meningkatkan akurasi secara keseluruhan (Nassar et al., 2021a).

Peran regresi logistik sangat sesuai dengan *co-training* karena fungsi *sigmoid*-nya menghasilkan probabilitas terkalibrasi, *fitting* yang cepat, dan interpretabilitas yang tinggi (Hosmer, 2014). Probabilitas terkalibrasi dalam regresi logistik disebabkan kemampuannya memilih sampel dengan *confidence* tinggi untuk output $\left(\sigma(z)\right)$ = probabilitas sesungguhnya $\approx$ 0 atau $\approx$ 1 untuk *pseudolabeling*. Regresi logistik memiliki optimasi konveks dan cepat, dimana fungsi *loss binary cross-entropy* (*negatif log-likelihood*) bersifat konveks sehingga menghasilkan solusi unik dengan konvergensi cepat. Regresi logistik memiliki peran penting saat melatih model berulang dalam iterasi *co-training*. Regresi logistik memiliki *interpretabilitas* koefisien atau koefisien yang dapat dijelaskan dari setiap β_j yang memiliki kekuatan dan arah pengaruh fitur x_j pada *log-odds*. Regresi logistik membantu menganalisis *view* dan menentukan fitur mana yang memberi *pseudolabel* paling andal. Regresi logistik memiliki integrasi mudah

sebagai *view* yang berperan sebagai *classifier* ketiga (di samping LSTM dan SVM) untuk memverifikasi *pseudolabel*. Model linear probabilistik akan menghasilkan sifat berbeda dari *margin-based* SVM atau non-linier LSTM sehingga akan memperkaya perspektif *view*. Fungsi *log-odds* dalam regresi logistik adalah cara mengubah probabilitas menjadi angka yang tak terbatas ($-\infty$ hingga $+\infty$) dengan mengambil logaritma dari rasio peluang (*odds*). Adapun rumus *log-odds* (z) seperti tersaji di persamaan (2.1):

$$z = \textit{log-odds} = \textit{logit}\,(x) = \beta_0 + \sum_{j=1}^{P} \beta_j . x_j \qquad (2.1)$$

Keterangan:

β_0 (*intercept*) : nilai *log-odds* ketika semua fitur = 0 (setelah standardisasi).

Nilai z itu sendiri bisa berkisar dari $-\infty$ sampai $+\infty$. Agar dapat menginterprestasikan hasil model dalam bentuk peluang (probabilitas) antara 0 dan 1 yang lebih intuitif untuk klasifikasi dibutuhkan fungsi *sigmoid* yang tersaji di persamaan (2.2):

$$\sigma(z) = \frac{1}{1+e^{-z}} \qquad (2.2)$$

Fungsi *sigmoid* berfungsi memetakan nilai z ke rentang [0 1], dimana $-\infty < z < +\infty$ atau $0 < \sigma(z) < 1$. Nilai ambang batas akan dijadikan penentu data masuk ke klasifikasi kelas mana sesuai dengan nilai yang ditentukan. Tanpa konversi ke probabilitas, output z kurang bermakna sebagai model yang meyakinkan sementara probabilitas dibutuhkan untuk mengevaluasi, membandingkan, dan menetapkan ambang batas keputusan.

2.4 Pembelajaran Mesin dalam Perguruan Tinggi

Perguruan tinggi mengintegrasikan pembelajaran mesin untuk mengatasi keterbatasan pengelolaan manajemen mahasiswa yang selama ini bergantung pada evaluasi administratif dan intuisi pengambil kebijakan. Perguruan tinggi memanfaatkan data transkrip, demografi, serta aktivitas *Learning Management System* (LMS) untuk meningkatkan akurasi prediksi keberhasilan akademik dan

risiko putus studi (Vaarma & Li, 2024). Analisis kritis menunjukkan bahwa tanpa dukungan pembelajaran mesin, perguruan tinggi sering gagal mendeteksi pola risiko secara dini, terutama pada mahasiswa dengan dinamika belajar yang tidak terlihat dalam laporan akademik tradisional.

Perguruan tinggi memproses data mahasiswa yang semakin besar dan heterogen, namun banyak kampus masih mengalami hambatan dalam konsolidasi data lintas sistem. (Badmus et al., 2024) menegaskan bahwa pembelajaran mesin menyediakan kemampuan analitik yang tidak dimiliki sistem konvensional, seperti identifikasi anomali perilaku belajar dan pola non-linear yang tidak dapat dipetakan oleh metode statistik biasa. Analisis ini mengkritisi bahwa keberhasilan sistem prediktif sangat bergantung pada kualitas integrasi data, bukan semata pada algoritmanya. Komunitas riset mengembangkan *learning analytics* untuk menyediakan umpan balik instruksional, namun analisis menunjukkan bahwa banyak *dashboard* gagal memberikan dampak pedagogis karena hanya menampilkan matrik kuantitatif. (Masiello et al., 2024) menunjukkan *dashboard* dapat menjadi alat pedagogis kuat jika dirancang dengan fokus pada proses belajar, bukan sekadar indikator aktivitas. (Paulsen & Lindsay, 2024) menegaskan bahwa *dashboard* yang minim konteks pedagogis justru berpotensi menyesatkan dosen dalam mengambil keputusan. Temuan ini menyoroti kebutuhan integrasi antara desain teknologi dan teori belajar.

Kajian modern menunjukkan bahwa teknik pembelajaran mesin generatif memperluas kapasitas personalisasi pembelajaran, namun teknologi ini juga menimbulkan risiko seperti distorsi rekomendasi belajar dan ketergantungan pada model yang tidak transparan (Rodríguez-Ortiz et al., 2025). Perguruan tinggi perlu mengkritisi sifat "*black box*" dari model generatif yang dapat mempengaruhi keadilan keputusan akademik. Analisis kritis merekomendasikan pengembangan sistem evaluasi internal untuk menjaga akuntabilitas penggunaan model generatif dalam pendidikan tinggi.

Perguruan tinggi menggunakan model seperti *Random Forest*, *Support Vector Machine* (SVM), dan *Naive Bayes* untuk memprediksi risiko putus studi

(De Santos et al., 2019; Vaarma & Li, 2024). Tinjauan kritis terhadap penelitian-penelitian tersebut menunjukkan bahwa sebagian besar model belum mengatasi isu bias algoritmik, terutama pada kelompok mahasiswa dari latar belakang sosial ekonomi tertentu. Institusi pendidikan perlu menyadari bahwa keakuratan model bukan satu-satunya indikator kualitas. Prediksi merupakan komponen yang tidak dapat dipisahkan dari implementasi pembelajaran mesin di pendidikan.

Perguruan Tinggi menerapkan algoritma pembelajaran mesin untuk memprediksi kinerja akademik mahasiswa berdasarkan riwayat nilai dan atribut personal, di mana algoritma seperti *Random Forest* terbukti menghasilkan akurat terkait IPK (Airlangga, 2024). Model prediktif juga *memberikan* dasar bagi perguruan tinggi untuk merancang kegiatan akademik, menyusun bimbingan belajar, serta mengalokasikan sumber daya untuk mendukung mahasiswa yang membutuhkan.

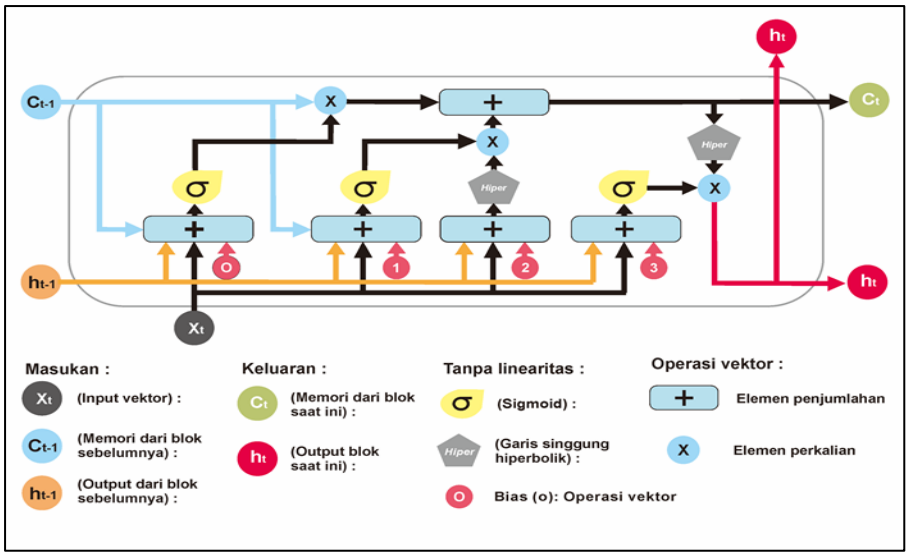
Perguruan tinggi menggunakan pembelajaran mesin untuk mengidentifikasi mahasiswa berisiko putus studi atau gagal akademik sejak awal semester. Penelitian *dropout* mengungkapkan bahwa model pembelajaran mesin yang menggabungkan data administratif, nilai awal, dan aktivitas semester pertama untuk memprediksi risiko *dropout* dengan akurasi tinggi (Segura et al., 2022). Studi lainnya menunjukkan bahwa universitas menggunakan data LMS, IPK awal, beban studi, dan frekuensi interaksi untuk mendeteksi mahasiswa yang beresiko, sehingga perguruan tinggi dapat melakukan pendampingan akademik atau konseling belajar (Hoca & Dimililer, 2025).

Perguruan Tinggi mengadopsi algoritma pembelajaran mesin untuk memprediksi masa studi, ketepatan waktu kelulusan, dan peluang menyelesaikan program tepat waktu. Model seperti *Naive Bayes*, *Random Forest*, dan SVM menunjukkan performa baik dalam mengestimasi masa kuliah berdasarkan IPK, status pekerjaan orang tua, data sosial ekonomi, jalur masuk, dan riwayat nilai dari mata kuliah (Hoca & Dimililer, 2025; Oktadiani et al., 2023). Literatur terkini menegaskan bahwa prediksi kelulusan berbasis pemnbelajaran mesin memperkuat

perencanaan akademik dan membantu institusi meningkatkan retensi mahasiswa dalam jangka panjang.

2.5 *Long Short-Term Memory* (LSTM)

Long Short-Term Memory (LSTM) diusulkan oleh Sepp Hochreiter & Jurgen Schmidhuber pada tahun 1997. LSTM mempunyai sel memori dan arsitektur dalam LSTM yang diperlihatkan pada Gambar 2.3 terdiri dari gerbang masukan, koneksi berulang, gerbang *forget*, dan gerbang keluaran.



Gambar 2.3 Arsitektur LSTM

LSTM juga mampu mengingat informasi jangka panjang. Gerbang masukan berfungsi untuk memblokir atau memasukan bagian yang akan diperbaharui. Gerbang keluaran adalah hasil dari lapisan *sigmoid* yang dijalankan untuk menentukan sel mana yang akan menjadi keluarannya. Gerbang *forget* merupakan memori-memori masa lalu untuk melupakan masa lalu (Le et al., 2019). Ide dasar dari LSTM yaitu adanya jalur yang menghubungkan antara *cell state* (C_{t-1}) sebelumnya dengan *cell state* yang sekarang (C_t). Jalur tersebut, merupakan informasi pada *cell state* dengan mudah dapat diteruskan ke *cell state* berikutnya dengan beberapa modifikasi yang diperlukan. Nilai *cell state* adalah vektor yang dirancang untuk menyimpan informasi tentang konteks sekuen data. Pada

pemrosesan kalimat, informasi yang dapat akan disimpan di dalam *cell state* adalah gender dari subjek, apakah subjek tunggal atau jamak dan sebagainya. Fitur-fitur ini akan diekstrak oleh LSTM selama proses latih. LSTM memiliki ide penggunaan gerbang *sigmoid* (disimbolkan dengan σ) yang akan mengatur informasi apakah diteruskan atau dihentikan. Output dari fungsi *sigmoid* adalah antara nol dan satu dengan arti nol adalah informasi dihentikan seluruhnya dan satu adalah informasi diteruskan seluruhnya. Keluaran dari fungsi *sigmoid* dikalikan dengan suatu nilai lain untuk menentukan seberapa besar informasi tersebut akan digunakan untuk proses berikutnya. Penentuan informasi yang akan dibuang dari C_{t-1} dengan menggunakan fungsi *sigmoid* yang disebut sebagai gerbang *forget* dilakukan di langkah pertama. Gerbang *forget* menerima nilai s_{t-1} dan x_t yang disambungkan, dan menghasilkan nilai antara nol dan satu. Nilai nol menandakan bahwa informasi akan dibuang sedangkan satu berarti informasi diteruskan. Formulasi dari nilai gerbang *forget* ditampilkan pada persamaan (2.3).

$$f_t = \sigma\big(w_f[s_{t-1} \cdot x_t]\big) \cdot b_f \tag{2.3}$$

Keterangan:

f_t : Gerbang *forget*

Wf : Berat dari tiap gerbang neuron pada waktu ke t di gerbang *forge*t

s_{t-1} : *State* sebelumnya

x_t : Masukkan di waktu sekarang

b_f : Bias untuk waktu ke t pada gerbang *forget*

Penentuan informasi apa yang akan ditambahkan dan disimpan ke *cell state*. Dilakukan di langkah kedua yang merupakan hasil dari penggabungan dari s_{t-1} dan x_t dengan dua fungsi, yaitu fungsi *sigmoid* sebagai gerbang input dan fungsi tanh sebagai gerbang *intermediate*. Informasi yang akan ditambahkan pada *cell state* merupakan hasil dari perkalian kedua fungsi tersebut. Nilai gerbang input disajikan pada persamaan (2.4) dan nilai kandidat dilakukan dengan persamaan (2.5) sebagai berikut:

$$i_t = \sigma(W_i \,.\, [h_{t-1}, x_t] + b_i) \quad (2.4)$$

$$C_t = tanh(W_c \,.[\, s_{t-1}, x_t] + b_c) \quad (2.5)$$

Keterangan:

C_t : *Cell state*

Σ : Fungsi sigmoid

i_t : Gerbang input

$tanh$: Fungsi $Tanh$

W_c : Berat tiap gerbang *neuron* pada waktu ke t di *cell state*

W_i : Berat tiap gerbang *neuron* pada waktu ke t di gerbang input

x_t : Input waktu sekarang

b_i : Bias pada waktu gerbang input

b_c : Bias pada *cell state*

State yang lama diperbaharui ke *cell state* dengan persamaan (2.6) sebagai berikut:

$$Ct = f_t \ * Ct\text{-}1 + i_t * \bar{c}_t \quad (2.6)$$

Keterangan:

C_t : *Cell State* baru yang dicari

f_t : Gerbang *forget*

C*t* -1 : *Cell state* sebelum orde ke t

i_t : Hasil gerbang input

$\bar{c}_t$: Nilai baru yang dapat ditambahkan ke *cell state*

Penambahan output dari gerbang *forget* di langkah pertama menjadi langkah selanjutnya. Penentuan output LSTM merupakan langkah terakhir. Output yang dihasilkan diperoleh dari perhitungan *sigmoid* dari gabungan s_{t-1} dan x_t yang disebut dengan gerbang output. Gerbang output menentukan berapa besar nilai dari *cell state* yang akan dihasilkan pada s_t. Lalu dihitung nilai fungsi $tanh$ dari e dan dikalikan dengan nilai dari gerbang output. Hasil perkaliannya tersebut menjadi

output dari unit LSTM di persamaan (2.7) dan (2.8) sebagai berikut (Isnain et al., 2020):

$$o_t = \sigma(W_o . [s_{t-1}, x_t] + b_o) \quad (2.7)$$

$$s_t = o_t \cdot tanh(C_t) \quad (2.8)$$

Keterangan:

o_t : Gerbang output

h_t : Hasil akhir dari LSTM

σ : Fungsi *sigmoid*

h_{t-1} : Hasil dari proses sebelumnya

x_t : Input waktu sekarang

b_o : Bias waktu pada gerbang output

$tanh$: Fungsi $tanh$

Dalam konteks pendidikan, LSTM sangat efektif untuk prediksi dan klasifikasi performa mahasiswa berdasarkan data sekuensial seperti nilai dan interaksi mereka selama satu semester atau lebih. Beberapa aplikasi LSTM dalam pendidikan tinggi termasuk prediksi kinerja mahasiswa berdasarkan data historis mahasiswa (F. Chen & Cui, 2020). Analisis sentimen mahasiswa melalui interaksi dalam *platform* pembelajaran (Z. Li & Lei, 2024).

2.5.1 *Confidence* dalam LSTM

Pada LSTM atau *Long Short-Term Memory*, digunakan untuk data sekuensial, *confidence score* dapat dihitung berdasarkan probabilitas atau ke mungkinan yang dihasilkan oleh model. Output LSTM, nilai *confidence* sering diwakili oleh kemungkinan yang dihasilkan oleh fungsi aktivasi *softmax* atau *sigmoid* pada *layer* output, tergantung pada masalah klasifikasi (Tomoya & Hitoshi, 2019). *Softmax function* dalam klasifikasi multi-kelas, output LSTM biasanya berupa kemungkinan kelas yang dihitung menggunakan fungsi softmax di persamaan (2.9):

$$P(y = c|x) = \frac{exp(z_C)}{\Sigma_j \, exp(z_j)} \quad (2.9)$$

Keterangan:

z_C : nilai yang dihitung oleh LSTM untuk kelas c_i

$exp(z_j)$: eksponensial dari nilai untuk semua kelas. Nilai ini menunjukkan *confidence* atau probabilitas data milik kelas c_i. Jika probabilitas kelas yang diprediksi tinggi (misal: 0.95), maka model sangat yakin (*confidence)*, sebaliknya jika probabilitas kelas yang diprediksi rendah (misal: 0.5), maka *confidence* lebih rendah

2.6 *Support Vector Machine* (SVM)

Support Vector Machine (SVM) dikembangkan oleh Boser, Guyon, dan Vapnik, pertama kali diperkenalkan pada tahun 1992 di Annual Workshop on Computational Learning Theory. Konsep dasar metode SVM sebenarnya merupakan gabungan atau kombinasi dari teori-teori komputasi yang telah ada pada tahun sebelumnya, seperti *marginhyperplane*, kernel diperkenalkan oleh Aronszajn tahun 1950, *Lagrange Multiplier* yang ditemukan oleh Joseph Louis Lagrange pada tahun 1766, dan demikian juga dengan konsep-konsep pendukung lain. SVM merupakan suatu teknik untuk melakukan prediksi, baik prediksi dalam kasus regresi maupun klasifikasi. SVM menghasilkan *hyperplane* sebagai garis di ruang dua dimensi atau bidang datar di ruang berdimensi tinggi, sesuai kompleksitas fitur (Guido et al., 2024). *Hyperplane* ini dapat berupa garis pada dua dimensi dan dapat berupa *flat plane* pada banyak dimensi. SVM mengatasi masalah klasifikasi non-linier dengan mentransformasikan data input ke ruang fitur berdimensi lebih tinggi melalui *kernel trick* dan mengoptimalkan *hyperplane* di ruang baru tersebut (Guido et al., 2024). Karakteristik SVM secara umum dirangkum sebagai berikut:

1. Secara prinsip SVM adalah *linear classifier*.
2. *Pattern recognition* dilakukan dengan mentransformasikan data pada ruang input ke ruang yang berdimensi lebih tinggi (*feature space*), dan optimisasi dilakukan pada ruang *vector* yang baru tersebut. Hal ini membedakan SVM dari solusi *pattern recognition* pada umumnya, yang melakukan optimisasi

parameter pada hasil transformasi yang berdimensi lebih rendah daripada dimensi ruang input.

3. Menerapkan strategi *Structural Risk Minimization* (SRM).
4. Prinsip kerja SVM pada dasarnya hanya mampu menangani klasifikasi dua kelas, namun telah dikembangkan untuk klasifikasi lebih dari dua kelas dengan adanya *pattern recognition.*

Metode *Support Vector Machine* memiliki beberapa keuntungan yaitu:

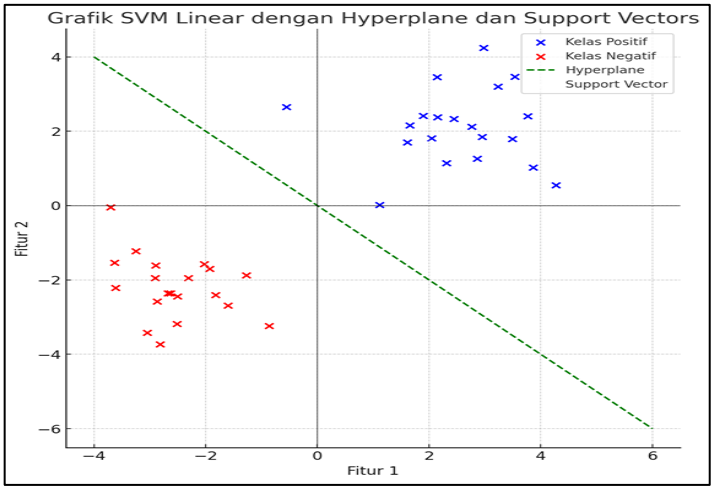
1. Generalisasi didefinisikan sebagai kemampuan suatu metode untuk mengklasifikasi suatu *pattern* atau pola, yang tidak termasuk data yang digunakan dalam fase pembelajaran metode itu.
2. *Curse of dimensionality* didefinisikan sebagai masalah yang dihadapi suatu metode *pattern recognition* dalam mengestimasikan parameter dikarenakan jumlah sampel data yang relatif lebih sedikit dibandingkan dengan dimensional ruang vektor tersebut.
3. *Feasibility* SVM dapat diimplementasikan relatif lebih mudah, karena proses penentuan *support vector* dapat dirumuskan dalam *Quadratic Programing* (QP) *problem.*

Adapun kerugian dari metode *Support Vector Machine* adalah sebagai berikut:

1. Sulit dipakai pada masalah berskala besar. Dalam hal ini dimaksudkan dengan jumlah sampel yang diolah.
2. SVM secara teoritik dikembangkan untuk masalah klasifikasi dengan dua kelas. Namun dewasa ini SVM telah dimodifikasi agar dapat menyelesaikan masalah dengan lebih dari dua kelas.

Gambar 2.4 menampilkan *hyperplane* pada *Support Vector Machine* merupakan salah satu metode dalam pembelajaran-terpandu yang biasanya digunakan untuk klasifikasi (seperti *Support Vector Classification*) dan regresi (*Support Vector Regression*). Pemodelan klasifikasi, SVM memiliki konsep yang lebih matang dan lebih jelas secara matematis dibandingkan dengan teknik-teknik klasifikasi lainnya. SVM juga dapat mengatasi masalah klasifikasi dan regresi linear

ataupun non linear. SVM banyak digunakan dalam dunia pendidikan untuk mengklasifikasikan mahasiswa berdasarkan kinerja di masa lalu, serta untuk memprediksi potensi akademik di masa depan SVM bekerja dengan cara mencari *hyperplane* terbaik yang memisahkan dua kelas dalam ruang fitur. Keunggulan SVM adalah kemampuannya untuk bekerja dengan data yang tidak terpisahkan secara linier dengan menggunakan kernel *trick.*



Gambar 2.4 *Hyperplane* yang memisahkan dua kelas positif (+1), negatif (-1)

Hyperplane adalah sebuah fungsi yang dapat digunakan untuk pemisah antar kelas. Dalam 2-D fungsi yang digunakan untuk klasifikasi antar kelas disebut sebagai *line whereas*, fungsi yang digunakan untuk klasifikasi antas kelas dalam 3-D disebut *plane similarly*, sedangan fungsi yang digunakan untuk klasifikasi di dalam ruang kelas dimensi yang lebih tinggi di sebut *hyperplane*. *Hyperplane* dalam Gambar 2.5 diperlihatkan sebagai garis lurus yang berada di tengah-tengah antara dua kelas data. Garis ini memisahkan dua kelas secara optimal dan jarak antara *hyperplane* dengan objek-objek data berbeda dengan kelas yang berdekatan (terluar) yang diberi tanda bulat kosong dan positif. Dalam SVM objek data terluar yang paling dekat dengan *hyperplane* disebut *support vector*.

Support Vector berupa beberapa titik data terluar dari kedua kelas (diberi lingkaran kosong dan positif) yang paling dekat dengan *hyperplane.* Titik-titik ini

menjadi penentu dalam menentukan letak *hyperplane*. Objek yang disebut *support vector* paling sulit diklasifikasikan dikarenakan posisi yang hampir tumpang tindih (*overlap*) dengan kelas lain. Objek *support vector* bersifatnya kritis, yang akan diperhitungkan untuk menemukan *hyperplane* yang paling optimal oleh SVM. SVM dibagi menjadi dua jenis yaitu, SVM linear digunakan untuk data yang dapat dipisahkan secara linear, yang berarti jika sebuah *dataset* dapat diklasifikasi menjadi dua kelas dengan menggunakan sebuah garis lurus tunggal, maka data tersebut disebut sebagai data yang dapat dipisahkan secara linear. *Classifier* yang digunakan disebut sebagai *Linear SVM classifier*. *SVM non-linear* digunakan untuk data yang dapat dipisahkan secara tidak linear, yang berarti jika sebuah *dataset* tidak dapat diklasifikasi menggunakan garis lurus, maka data tersebut disebut data *non*-linear. *Classifier* yang digunakan disebut sebagai *Non-linear SVM classifier*.

2.6.1 *Kernel* dalam SVM

Kekuatan utama dari SVM adalah penggunaan *kernel*, yang memungkinkan SVM untuk bekerja dengan data non-linear. *Kernel* adalah fungsi yang memetakan data ke ruang fitur yang lebih tinggi, di mana data yang awalnya tidak dapat dipisahkan secara linier menjadi dapat dipisahkan dengan mudah. *Kernel trick* memungkinkan SVM untuk melakukan pemetaan ini tanpa secara eksplisit menghitung ruang fitur yang lebih tinggi, yang menghemat waktu dan sumber daya komputasi (Khan et al., 2023). Jenis kernel yang digunakan dalam SVM di disertasi ini adalah linier kernel. Linear *kernel* digunakan ketika data dapat dipisahkan dengan *hyperplane* linier dengan rumus di persamaan (2.10) sebagai berikut:

$$K(x, y) = x^T . y \tag{2.10}$$

Keterangan :

x : vektor fitur data pertama x^T : transpos dari vektor

y : vektor fitur data kedua

Penjelasan komponen linier kernel adalah:

x adalah fitur dari data pertama, misal dalam konteks *dataset* mahasiswa, x berisi 5 fitur yaitu : nilai IPK, jumlah saudara, pendapatan orang tua, prestasi, keaktifan dalam organisasi. Maka x adalah vektor dengan 5 elemen x={3, 7, 5, 4, 2, 3}.

y adalah fitur dari data kedua, berisi data mahasiswa lainnya dengan atribut yang serupa misalnya: y={3, 5, 2, 3, 4, 2}.

x^T adalah transpos dari vektor x yang artinya mengubah vektor kolom menjadi baris. Karena x adalah vektor baris maka $x = x^T$ = {3, 7, 5, 4, 2, 3}

Produk titik $(x^T.y)$ adalah produk titik antara dua vektor x dan y adalah operasi matematika yang menghasilkan skalar. Produk titik dilakukan dengan mengalikan elemen-elemen yang sesuai dari dua vektor dan menjumlahkan hasil perkalian tersebut. Maka produk titik $K(x^T.y)$ yang diperoleh adalah:

$K(x^T y) = x_1.y_1 + x_z.y_2 + x_{3.}y_3 + x_{4.}y_4 + x_5.y_5$

$K(x^T y)$ = 3.3 + 7.5 + 5.2 + 4.3 + 2.4 + 3.2

$K(x^T y)$ = 80 (skalar)

Hasil dari nilai skalar digunakan dalam perhitungan lebih lanjut untuk menentukan posisi *hyperplane* dan margin dalam SVM.

Sifat linear kernel:

a. Tidak memiliki parameter non-linear seperti yang dimiliki oleh *kernel* jenis lain seperti *polynomial* kernel atau RBF Kernel. Hal ini membuatnya sangat efisien untuk data yang terpisah secara linier.
b. Sederhana dan cepat karena hanya melibatkan produk titik sederhana, linear kernel lebih cepat dihitung dibandingkan kernel lainnya, terutama untuk data besar.

Menurut (Lam et al., 2009) langkah-langkah untuk membangun model SVM adalah sebagai berikut:

1. Memilih *kernel* yang tepat sangat penting. *Kernel* linier digunakan ketika data dapat dipisahkan secara linier, sementara RBF kernel (*Radial Basis Function*) digunakan ketika data tidak terpisah secara linier. Dalam penelitian disertasi ini dipilih kernel linier dengan rumus di persamaan (2.11):

$$K(x_i, x_j) = x_i \cdot x_j \qquad (2.11)$$

Keterangan:

x_i dan x_j dua vektor fitur yang dibandingkan

$x_i \cdot x_j$ adalah *inner product* atau produk titik antara vektor x_i dan x_j

Matriks *kernel* ditentukan setelah memilih jenis *kernel*. Matriks kernel adalah matriks yang terdiri dari hasil *kernel* untuk setiap pasangan titik data dalam *dataset*. Matriks *kernel* menggambarkan seberapa mirip (atau seberapa dekat) data satu dengan yang lain berdasarkan *inner product*. Setiap elemen dalam matriks *kernel* $K(x_i, x_j)$ adalah hasil dari *inner product* antara titik data x_i dan x_j. Adapun rumus contoh matriks *kernel* yang memiliki titik x_1, x_2, x_3 tersaji pada persamaan (2.12):

$$\text{Matriks } kernel = \begin{bmatrix} K(x_1x_1) & K(x_1x_2) & K(x_1x_3) \\ K(x_2x_1) & K(x_2x_2) & K(x_2x_3) \\ K(x_3x_1) & K(x_3x_2) & K(x_3x_3) \end{bmatrix} \quad (2.12)$$

Keterangan:

Setiap elemen $K(x_i, x_j)$ dihitung sebagai *inner product* dari dua vektor data x_i dan x_j.

Inner product ditentukan setelah matriks *kernel* terbentuk. *Inner product* atau produk titik dari dua vektor adalah operasi dasar untuk menghitung kesamaan antara dua titik dalam ruang fitur dan bagian terpenting dalam menghitung matriks kernel. Jika x_i dan $x_j$adalah dua vektor fitur, maka *inner product* $x_i \cdot x_j$dihitung sebagai jumlah dari perkalian elemen-elemen yang sesuai pada kedua vektor seperti pada persamaan (2.13):

$$x_i \cdot x_j = \sum_{k=1}^{n} x_{i,k} \cdot x_{j,k} \quad (2.13)$$

Keterangan:

Di mana $x_{i,k}$ dan $x_{j,k}$ adalah elemen-elemen ke-k dari vektor x_i dan x_j dan n adalah jumlah fitur dalam setiap vektor.

2. Melatih model SVM menggunakan data pelatihan. SVM mencoba menemukan *hyperplane* terbaik yang memisahkan dua kelas. Proses ini melibatkan optimasi bobot dan bias untuk mendapatkan margin terbaik.
3. Mengevaluasi model setelah model dilatih, evaluasi dilakukan menggunakan data uji untuk mengukur kinerjanya, menggunakan metrik akurasi, *precision*, *recall*, dan F1-*score*. Evaluasi ini memberikan gambaran seberapa baik model dalam mengklasifikasikan data baru.
4. Prediksi setelah model selesai dilatih dan diuji, model ini digunakan untuk memprediksi kelas dari data yang belum terklasifikasi, berdasarkan informasi fitur yang telah dipelajari.

Berikut adalah rumus yang digunakan untuk prediksi kelas dalam SVM yaitu:

1. Prediksi Kelas dalam SVM

   SVM bekerja dengan memisahkan dua kelas menggunakan *hyperplane* terbaik yang memiliki margin terbesar. Fungsi keputusan untuk prediksi kelas dapat ditulis seperti persamaan (2.14) berikut:

   $$f(x) = \omega^T x + b \tag{2.14}$$

   Keterangan:

   $f(x)$: fungsi keputusan yang menentukan kelas dari data x.

   x : vektor fitur dari data yang ingin diprediksi (misal: nilai IPK, gaji orang tua, jumlah saudara, prestasi, keaktifan organisasi, dll).

   ω : faktor bobot yang ditemukan selama pelatihan.

   b : bias yang juga ditemukan selama pelatihan.

2. Langkah-langkah untuk menentukan kelas:

   Prediksi kelas Positif : jika $f(x) > 0$, maka kelas yang diprediksi adalah kelas 1 (misal, potensi tinggi).

   Prediksi kelas Negatif : jika $f(x) < 0$, maka kelas yang diprediksi adalah kelas 0 (misal, potensi rendah).

2.6.2 *Confidence* dalam SVM

Pada SVM, *confidence* dapat dihitung berdasarkan jarak margin antara titik data dan *hyperplane* yang memisahkan kelas. Dalam hal ini, jarak margin memberikan gambaran seberapa yakin model terhadap prediksinya. Margin adalah jarak antara *hyperplane* dan titik data terdekat dari kedua kelas. Titik-titik ini disebut *support vectors*. *Confidence score* dikondisikan jika data terletak jauh dari *hyperplane*, maka *confidence* model tinggi, dan sebaliknya jika data berada dekat dengan *hyperplane*, maka *confidence*-nya lebih rendah (Ning et al., 2021). Secara matematis *confidence* dapat dihitung sebagai nilai absolut dari fungsi keputusan $f(x)$ sesuai persamaan (2.15) berikut:

$$Confidence = |f(x)| = |\omega^T x + b| \tag{2.15}$$

Keterangan :

ω : vektor bobot.

x : vektor fitur data yang diuji.

b : bias.

Jika *confidence* mendekati nol, berarti model kurang yakin dengan prediksinya

2.7 Pembelajaran Semi-Terpandu dan *Pseudolabeling* (Pelabelan Semu)

Pembelajaran semi-terpandu menggabungkan data berlabel dan tak berlabel untuk meningkatkan performa model melalui strategi seperti pelabelan semu dengan ambang batas kepercayaan. Rumus pelabelan semu dengan *threshold confidence* tersaji di persamaan (2.16).

$$\hat{y} = \arg\max P(y|x) \ , \quad \text{jika} \max P(y|x) \ \geq r \tag{2.16}$$

Keterangan:

$\hat{y}$: Label prediksi yang dipilih oleh model untuk sampel x (*pseudolabel)* karena diberikan secara otomatis oleh model

$P(y|x)$: Probabilitas bahwa suatu input x termasuk ke dalam kelas y

$\arg\max P(y|x)$: Operasi untuk memilih kelas dengan probabilitas tertinggi

$\max P(y|x)$: Nilai probabilitas terbesar dari semua kelas

r = tau : *Threshold* (ambang batas kepercayaan)

Jika $\max P(y|x) \geq r$ = maka label prediksi $\hat{y}$ dianggap cukup yakin untuk dipakai sebagai *pseudolabel*

Model menggunakan fungsi *Loss* gabungan memastikan model tidak hanya bagus di data berlabel (jumlahnya sedikit), tetapi juga bisa menggeneralisasi dengan baik menggunakan data tak berlabel (jumlahnya besar) secara simultan (Sosea & Caragea, 2023). Fungsi *loss* gabungan yang dioptimalkan model tersaji di persamaan (2.17).

$$L = L\sup(X_1, Y_1.) + \lambda Lunsup(Xu, \hat{Y}u) \quad (2.17)$$

Keterangan:

L :Total fungsi *loss* gabungan yang dioptimalkan model (kombinasi *loss supervised* + *unsupervised*)

$L\,sup(X_1, Y_1.)$: *Loss supervised* (*loss* terpandu)

λ : Parameter bobot (*weighting factor*) untuk mengatur kontribusi *loss unsupervised* terhadap total *loss*

$Lunsup(Xu, \hat{Y}u)$: *Loss unsupervised* (*loss* tak terpandu)

Model akan beradaptasi tanpa bergantung pada pelabelan manual dan mempertahankan akurasi dalam memprediksi. Metode *DYMatch* diterapkan dalam pelabelan semu dinamis dan *feature consistency*, sehingga meningkatkan efektivitas penggunaan data tak berlabel secara adaptif. Pelabelan semu dikembangkan dengan *threshold adaptif* serta *contrastive loss* untuk sampel tidak yakin untuk mempertahankan serta memanfaatkan informasi dari data tak berlabel yang memiliki *confidence* rendah (X. Zhang et al., 2024). Pendekatan SoftMatch untuk mengatasi *trade off* jumlah kualitas pelabelan semu dengan menggunakan fungsi bobot Gaussian yang terpotong untuk menyeimbangkan pemilihan sampel

berdasarkan *confidence* untuk meningkatkan generalisasi pembelajaran semi-terpandu (H. Chen et al., 2023).

2.8 Matrik Evaluasi Model

Evaluasi model dalam pembelajaran mesin sangat penting untuk memastikan keandalan dan efektivitas model yang digunakan. Beberapa matrik yang umum digunakan dalam evaluasi model klasifikasi, seperti *Accuracy*, *Precision*, *Recall*, dan *ROC Curve*, memberikan gambaran umum tentang kinerja model. Matrik evaluasi model juga memiliki kekurangan dan tidak selalu mencerminkan performa model secara komprehensif, terutama jika terdapat ketidakseimbangan kelas atau kesalahan dalam data (Dj Novakovi et al., 2017)**.**

Accuracy adalah metrik yang paling sederhana dan sering digunakan, tetapi seringkali tidak mencerminkan kinerja model yang sebenarnya, terutama dalam kasus ketidakseimbangan kelas. *Precision* dan *Recall* lebih disarankan untuk menilai kemampuan model dalam mengklasifikasikan data positif dan negatif (Vujović, 2021). ROC (*Receiver Operating Characteristic*) *curve* adalah alat yang lebih tepat untuk mengevaluasi model klasifikasi, karena menunjukkan *trade-off* antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR) di berbagai *threshold* keputusan. Area *Under the Curve* (AUC) dari ROC memberikan gambaran tentang seberapa baik model dalam memisahkan kelas positif dan negatif.

Dalam penelitian disertasi, model yang dihasilkan dievaluasi menggunakan metrik *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Adapun rumus – rumus yang akan digunakan untuk menghitung metrik terdapat di persamaan (2.18), persamaan (2.19), persamaan (2.20), dan persamaan (2.21) (Opitz, 2024; Powers, 2011; Sathyanarayanan & Tantri, 2024).

1. Akurasi (*accuracy*), mengukur prediksi yang benar terhadap total data.

$$\text{Akurasi} = \frac{\text{Jumlah prediksi benar}}{\text{Jumlah total data}} \tag{2.18}$$

2. Presisi (*precision*), mengukur ketepatan prediksi positif.

$$\text{Presisi} = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (2.19)$$

3. Sensitivitas (*recall*), mengukur kemampuan model untuk menemukan semua prediksi positif yang benar.

$$\text{Sensitivitas} = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (2.20)$$

4. *F1-Score*, merupakan rata-rata harmonis dari *precision* dan *recall*.

$$F1\text{-}Score = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall} \quad (2.21)$$

Keterangan:

True Positives (TP) adalah jumlah contoh yang benar-benar positif dan diprediksi sebagai positif oleh model. Artinya, model benar-benar memprediksi bahwa suatu sampel termasuk dalam kelas positif dan ternyata sampel tersebut memang benar positif.

False Positives (FP) adalah jumlah contoh yang sebenarnya negatif, tetapi model memprediksi bahwa contoh tersebut adalah positif. Ini dikenal juga sebagai *Type I error* atau *False Alarm*.

True Negatives (TN) adalah jumlah contoh yang benar-benar negatif dan diprediksi sebagai negatif oleh model. Artinya, model dengan benar memprediksi bahwa suatu sampel tidak termasuk dalam kelas positif.

False Negatives (FN) adalah jumlah contoh yang sebenarnya positif, tetapi diprediksi oleh model sebagai negatif. Ini dikenal sebagai *Type II error* atau *Miss*.