

AI PLAY BOOK

Series 2 Thriving Deep Into AI
for Central Banking



Hak cipta © pada penulis dan dilindungi Undang-Undang

Dilarang memperbanyak sebagian atau seluruh bagian buku ini tanpa izin tertulis dan resmi dari Bank Indonesia

**AI PLAYBOOK SERIES 2
THRIVING DEEP INTO AI FOR CENTRAL BANKING**

[di sini nama siapa]



Pengantar Gubernur BI

Dalam era transformasi digital yang semakin dinamis, kecerdasan artifisial (*Artificial Intelligence/AI*) menjadi salah satu teknologi kunci dalam memperkuat ketahanan dan daya saing ekonomi nasional. Bagi Bank Indonesia, AI berperan strategis dalam mendukung inisiatif *Integrated Digital Central Bank* (IDCB), sebuah visi jangka panjang untuk menjadikan BI sebagai bank sentral digital terdepan. Implementasi AI secara tepat akan memperkuat pengambilan keputusan secara *decision-led intelligence* yang didukung dengan data, meningkatkan efisiensi operasional, serta memperkuat kapabilitas pengawasan dan regulasi yang adaptif terhadap perkembangan teknologi.

Pendekatan *leapfrogging* melalui pengembangan *use case* AI yang berdampak tinggi menjadi krusial bagi Bank Indonesia. Strategi ini memungkinkan BI untuk melompati tahapan-tahapan konvensional dan langsung mengadopsi solusi AI yang relevan, inovatif, dan berorientasi masa depan. Dengan mengidentifikasi dan mengeksekusi *use case* yang strategis, BI dapat mempercepat pencapaian IDCB dan memberikan nilai tambah yang signifikan dalam mendukung kebijakan moneter, sistem pembayaran, serta stabilitas sistem keuangan nasional.

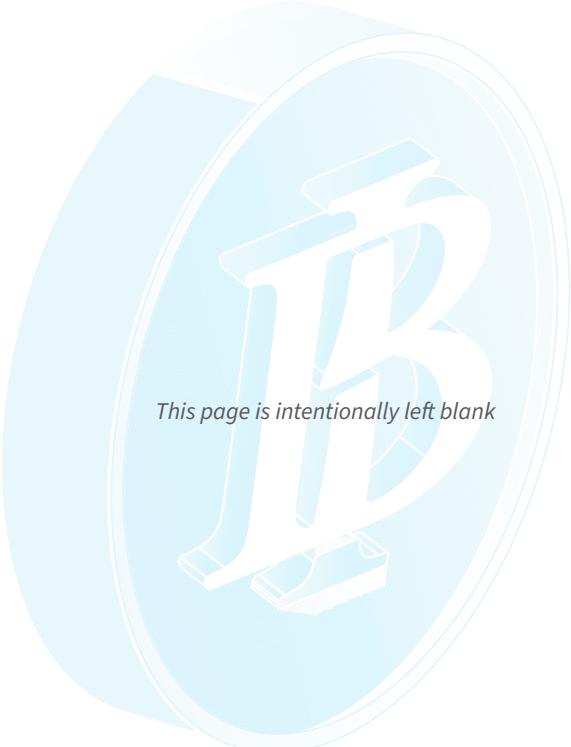
Untuk mendukung keberhasilan strategi ini, AI *Playbook Series* 2 hadir sebagai panduan praktis dalam mengakselerasi pengembangan *use case* AI terutama pada *use case* yang bersifat *leapfrog*. Buku ini menyajikan kerangka kerja, metodologi, dan referensi *best practices* yang relevan bagi seluruh satuan kerja di Bank Indonesia dalam mengidentifikasi, merancang, dan mengimplementasikan solusi AI secara terarah dan terukur. Dengan hadirnya *playbook* ini, diharapkan proses transformasi digital Bank Indonesia dapat berjalan lebih efisien, adaptif dan selaras dengan tujuan strategis IDCB.

Selamat membaca, semoga buku ini dapat menjadi sumber semangat dan inspirasi bagi seluruh insan Bank Indonesia dalam mengintegrasikan AI ke dalam proses bisnis dan pengambilan kebijakan.

Gubernur Bank Indonesia

PERRY WARJIYO





This page is intentionally left blank

Pengantar Deputi Gubernur BI

Kecerdasan artifisial (*Artificial Intelligence/AI*) merupakan salah satu pilar utama dalam mendukung transformasi digital Bank Indonesia menuju visi *Integrated Digital Central Bank* (IDCB). Dalam konteks ini, AI bukan hanya sekadar teknologi pendukung, tetapi menjadi *enabler* strategis dalam memperkuat efektivitas kebijakan, efisiensi operasional, dan akurasi pengambilan keputusan. Dengan potensi besar yang dimiliki AI, Bank Indonesia perlu memastikan bahwa setiap pengembangan dan penerapan teknologi ini berjalan secara terstruktur, berkelanjutan, dan selaras dengan arah strategis IDCB.

AI *Playbook Series 2* yang merupakan kelanjutan dari AI *Playbook Series 1*, disusun secara lebih teknis dan ditujukan bagi para praktisi internal yang akan secara langsung terlibat dalam pengembangan dan implementasi AI di lingkungan Bank Indonesia. Buku ini membahas lanskap AI, pemodelan, metodologi evaluasi, hingga tahapan *deployment*. Dengan pendekatan tersebut, buku ini diharapkan menjadi panduan praktis yang aplikatif dalam mendorong adopsi AI yang efektif dan berdampak.

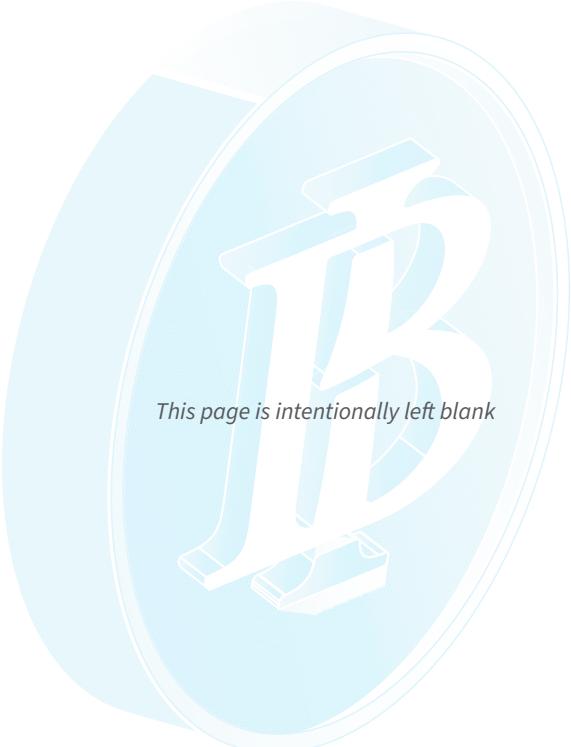
Sebagai institusi yang memegang peranan penting dalam menjaga stabilitas ekonomi dan keuangan nasional, Bank Indonesia menempatkan prinsip kehati-hatian dan tata kelola sebagai landasan penting dalam setiap penerapan teknologi baru. Oleh karena itu, AI *Playbook Series 2* juga memberikan perhatian khusus terhadap aspek *responsible AI*, yang juga pada bagian akhir buku ini. Pendekatan yang etis, transparan, dan akuntabel menjadi kunci dalam memastikan bahwa pengembangan AI tidak hanya canggih secara teknis, tetapi juga selaras dengan nilai dan prinsip tata kelola yang dijunjung tinggi oleh Bank Indonesia.

Semoga AI *Playbook Series 2* ini dapat memberikan manfaat nyata dari sisi praktis bagi seluruh insan Bank Indonesia dalam mengembangkan dan mengimplementasikan solusi AI secara tepat guna.

Deputi Gubernur Bank Indonesia

JUDA AGUNG





This page is intentionally left blank

How to Use This Book?

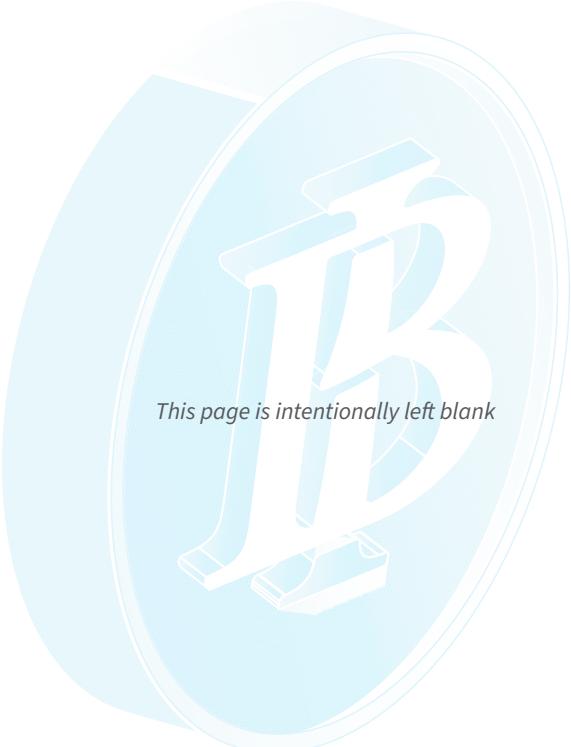
Pertama, buku ini dirancang bagi mereka yang ingin memanfaatkan AI secara teknis untuk membantu proses bisnis. Pembaca diasumsikan telah memahami apa itu AI beserta hal-hal dasar terkait, sebagaimana telah dibahas pada *AI Playbook Series 1: The First Step*. Idealnya, pembaca buku ini memiliki pemahaman yang cukup baik terkait matematika yang diaplikasikan pada *continuous space*, terutama *probability theory*, *linear algebra*, dan *calculus*, meskipun penulis berupaya memberikan penjelasan sesederhana mungkin untuk mengilustrasikan beberapa konsep yang lebih kompleks.

Kedua, buku ini membahas hal-hal yang lebih bersifat teknis mengenai lanskap AI, evaluasi, serta pemanfaatannya. Perlu diketahui bahwa tidak semua hal terkait AI/*machine learning* dibahas dalam buku ini. Penekanan pembahasan dalam buku ini lebih pada dasar, satu atau dua penjelasan contoh umum yang *applicable*, serta isu-isu populer yang sering dijumpai dalam pengembangan AI/*machine learning*. Dengan demikian, pembaca dipersilakan untuk melakukan pengembangan sendiri terkait *state of the art* dan hal-hal di luar pembahasan.

Ketiga, agar dapat secara efektif dan efisien dalam memahami isi buku ini, pembaca disarankan membaca secara urut 3 bab pertama. Ketiga bab pertama tersebut bersifat fundamental yang mencakup semesta terkait AI dan *machine learning*, bagaimana komputer belajar. Selain itu, ketiga bab tersebut juga sebagai dasar untuk mengikuti diskusi yang disajikan pada Bab 4, 5, dan 6. Selanjutnya pada Bab 7 buku ini bersifat sedikit *advance* terkait isu-isu pada proses *learning*. Bab 8 berisi *technical notes*, tips, dan trik untuk mengembangkan AI dan ML secara teknis.

Get your hands dirty!

Terakhir, buku ini menganjurkan agar pembaca juga dapat secara langsung berekspeten pada pemanfaatan AI dan ML, baik dengan data sintetis maupun data *real*.



This page is intentionally left blank

Daftar Isi

PENGANTAR GUBERNUR BI.....	III
PENGANTAR DEPUTI GUBERNUR BI	V
BAGIAN 1. INTRODUCTION: AI REFRESHMENT	1
1.1 <i>Artificial Intelligence and Machine Learning</i>	5
1.2 <i>Machine Learning and Econometrics</i>	7
1.3 <i>Current Trends and Breakthrough in AI</i>	9
1.4 <i>AI Landscape</i>	12
BAGIAN 2. MACHINE LEARNING: A QUICK CONCEPT EXAMPLE	13
2.1 <i>Linear Regression: Ide Mesin Belajar</i>	14
2.2 <i>Training and Test Sets and Generalisation Error</i>	15
BAGIAN 3. LEARNING LANDSCAPE.....	19
3.1 <i>Supervised Learning</i>	20
3.2 <i>Unsupervised Learning</i>	23
3.3 <i>Reinforcement Learning</i>	26
BAGIAN 4. MODEL EVALUATION	31
4.1 Memilih Model: Performa vs. <i>Interpretability</i>	32
4.2 <i>Machine Learning Evaluation</i>	33
4.2.1 Tujuan dan Tahapan Umum.....	33
4.2.2 Metriks.....	34
4.2.3 Hal-Hal Terkait Evaluasi.....	43
BAGIAN 5. POPULAR APPLIED MACHINE LEARNING	47
5.1 <i>Natural Language Processing (NLP)</i>	48
5.2 <i>Computer Vision & Image Recognition</i>	52
5.3 <i>Speech and Signal Processing</i>	56
5.4 <i>Data Mining and Time Series Analysis</i>	60
5.5 <i>Generative AI (GenAI)</i>	64
BAGIAN 6. UTILISING AI IN BANK INDONESIA	71
6.1 <i>AI for Monetary Policies</i>	72
6.2 <i>AI for Macroprudential Policies</i>	75
6.3 <i>AI for Payment System</i>	80
6.4 <i>AI for HR</i>	82
BAGIAN 7. ADVANCED TOPICS ON MACHINE LEARNING:	
<i>PUSHING THE BOUNDARY</i>	85
7.1 <i>Bayesian Learning</i>	86

7.2	<i>Probabilistic Graphical Models</i>	88
7.3	<i>Mixture Models</i>	90
7.4	<i>Ensemble Models</i>	92
7.5	<i>Gaussian Processes</i>	93
7.6	<i>Neural Networks</i>	94
7.7	<i>Approximation: Markov Chain Monte Carlo vs.</i> <i>Variational Inference</i>	98
BAGIAN 8. SOME TECHNICAL NOTES ON PRACTICING AI		103
8.1	<i>Tips on Data Preparation</i>	104
8.2	<i>Tips on Machine Learning Modelling</i>	104
8.3	<i>Tips on AI/Machine Learning Evaluation, Deployment and</i> <i>Monitoring</i>	105
REFERENCES ..		107
APPENDIX ..		111
	Appendix 1 Daftar Machine Learning Models	111
	Appendix 2 Glossary (Terminologi Umum)	117
	Appendix 3 Rekomendasi Tools Dasar untuk Lab/Eksperimen	123
	Appendix 4 List Use Case AI di BI	125

BAGIAN 1

INTRODUCTION

AI Refreshment



"A computer would deserve to be called intelligent if it could deceive a human into believing that it was human" – Alan Turing



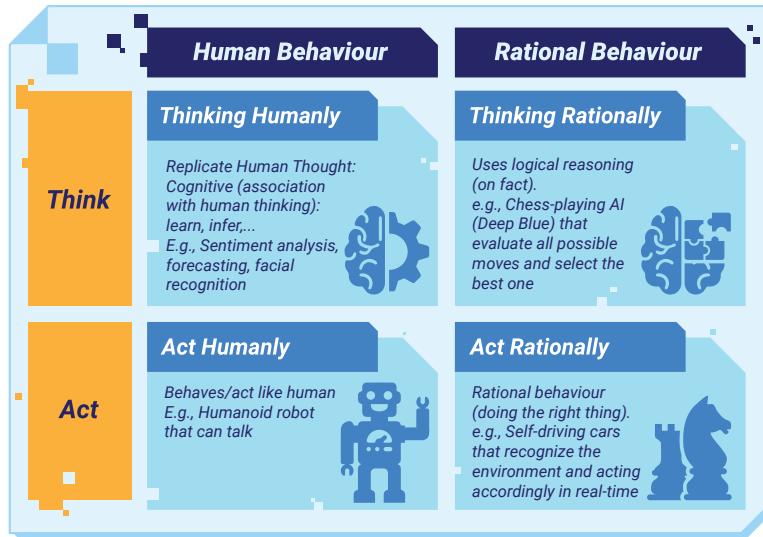
What is AI?

Kecerdasan artifisial atau *artificial intelligence* (AI) telah didefinisikan dengan berbagai cara oleh para ahli selama bertahun-tahun. Beragam pandangan ini mencerminkan luasnya cakupan AI, mulai dari pembelajaran mesin (*machine learning*), pemrosesan bahasa alami (*natural language processing*), hingga *deep learning*. Setiap definisi memiliki pendekatan yang berbeda, tetapi semuanya berfokus pada upaya untuk menciptakan mesin yang mampu meniru atau bahkan melampaui kemampuan kognitif manusia.

Dalam buku *Artificial Intelligence: A Modern Approach* (Russell dan Norvig, 2003, 2021), Stuart Russell dan Peter Norvig mendefinisikan AI sebagai suatu agen cerdas (dalam hal ini merujuk pada mesin) yang mampu mengindera lingkungannya, beroperasi secara mandiri, beradaptasi terhadap perubahan, dan merespons dengan tindakan atau keputusan. Agar dianggap sebagai AI, sistem tersebut harus mampu menunjukkan kecerdasan yang terukur, seperti:

- **Kemampuan untuk belajar dan beradaptasi:** Sistem AI harus bisa memperbaiki dirinya sendiri seiring dengan akumulasi pengalaman atau data baru.
- **Kemampuan untuk menalar atau mengambil keputusan:** Sistem AI harus mampu menyelesaikan masalah, memprediksi hasil, atau membuat keputusan yang optimal berdasarkan informasi yang tersedia.
- **Otonomi dalam bertindak:** AI seharusnya bisa bertindak tanpa intervensi manusia secara terus-menerus, mengerjakan tugas-tugas yang kompleks, seperti agen otonom atau robot.
- **Generalitas dalam penerapan:** Sistem AI tidak hanya bekerja untuk satu tugas spesifik, tetapi dapat diadaptasi untuk tugas lain tanpa modifikasi yang besar.

Russell dan Norvig menggunakan empat kuadran untuk mengelompokkan berbagai cara AI dapat dipahami dan dikembangkan sebagaimana ilustrasi pada Gambar 1 berikut.



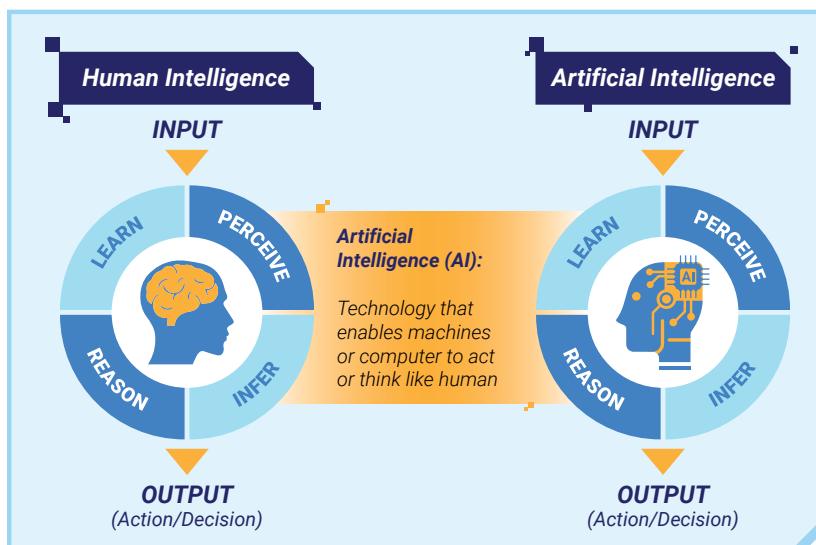
Gambar 1 Four Quadrants of AI

- **Thinking Humanly** — Meniru proses berpikir manusia (psikologi kognitif). Pendekatan ini berfokus pada **model kognitif** yang meniru cara manusia berpikir. Di sini, AI dirancang untuk meniru proses mental yang terjadi dalam otak manusia, seperti persepsi, ingatan, penalaran, dan pengambilan keputusan. Pendekatan ini membutuhkan pemahaman psikologi kognitif manusia untuk membangun sistem AI yang dapat meniru cara manusia berpikir.
- **Thinking Rationally** — Berpikir berdasarkan **logika formal**, deduktif. Pendekatan ini lebih fokus pada **logika formal** dan **penalaran deduktif**. AI yang dirancang dengan pendekatan ini bertujuan untuk melakukan penalaran yang benar secara logika, terlepas dari bagaimana manusia berpikir. Dalam hal ini, AI menggunakan prinsip logika matematis untuk menarik kesimpulan dari fakta-fakta yang diketahui.
- **Acting Humanly** — Meniru perilaku manusia dalam tindakan. Pendekatan ini berfokus pada **perilaku**. Sistem AI yang dirancang untuk bertindak seperti manusia akan mencoba meniru cara manusia merespons atau bertindak dalam situasi tertentu. Pendekatan ini sering digunakan dalam pengembangan **chatbot** atau **Turing Test**, di mana AI harus menunjukkan perilaku yang menyerupai manusia agar tidak dapat dibedakan dari manusia sejati.
- **Acting Rationally** — Bertindak secara **rasional** dan **optimal** untuk mencapai tujuan. Pendekatan ini fokus pada **agen yang rasional**, yang berarti AI bertindak untuk mencapai tujuan tertentu dengan cara yang optimal. Agen

rasional ini membuat keputusan yang maksimal untuk mencapai hasil terbaik berdasarkan informasi yang tersedia. Pendekatan ini tidak perlu meniru cara manusia berpikir, tetapi lebih menekankan pada **optimalitas tindakan** dalam lingkungan tertentu.

Meski terdapat berbagai definisi yang muncul dari para pakar (misalnya: Brynjolfsson et al (2014), Russell & Norvig (2021), Hyman (1982), Mitchell & Melanie (2019), Nilsson (2010), dan Domingos (2015)), semua pendapat tersebut memiliki benang merah yang sama: AI adalah upaya untuk meniru dan memperluas kemampuan manusia dalam hal berpikir, belajar, dan mengambil keputusan. AI berusaha mereplikasi kecerdasan manusia dalam bentuk algoritma dan program komputer yang mampu melakukan tugas-tugas yang membutuhkan kecerdasan jika dilakukan oleh manusia. Oleh karena itu, AI dapat didefinisikan sebagai teknologi atau sistem yang memungkinkan suatu agen/mesin/komputer dapat bertindak atau berpikir cerdas (seperti manusia) a.l. belajar (*learn*), mengindra (*perceive*), menyimpulkan (*infer*), dan berpikir (*reason*). Sederhananya, ilustrasi grafis mengenai apa itu AI dapat dilihat pada Gambar 2.

Pada dasarnya suatu fungsi kognitif manusia (Gambar sisi kiri) dapat dimanifestasikan ke dalam 4 (empat) jenis, yaitu *learn*, *perceive*, *reason*, serta *infer*. Keempat komponen ini dapat ditirukan ke dalam mesin sehingga membentuk AI. Singkatnya perbedaan antara kecerdasan alami manusia vs. kecerdasan artifisial ditinjau dari 4 komponen tersebut, yaitu:



Gambar 2 Human Intelligence vs. Artificial Intelligence

Tabel 1 Human Intelligence vs Artificial Intelligence

Komponen	Kecerdasan Alami Manusia	Kecerdasan Artifisial (AI)
Learn	Merujuk pada proses memperoleh pengetahuan, keterampilan, atau pemahaman melalui pengalaman, studi, atau pengajaran. Belajar pada manusia melibatkan sejumlah mekanisme biologis, kognitif, dan psikologis yang memengaruhi bagaimana informasi baru diserap, diproses, dan diterapkan.	"Learn" atau "belajar" merujuk pada kemampuan sistem atau model untuk meningkatkan performa atau pengambilan keputusan dari pengalaman atau data yang diterimanya.
Perceive	Melalui indra, manusia menerima informasi dari lingkungan, seperti penglihatan, pendengaran, dan sentuhan.	Kemampuan sistem AI untuk mengumpulkan, memahami, dan memproses informasi sensoris dari lingkungannya, seperti gambar, suara, atau teks. Kemampuan ini memungkinkan AI untuk mendeteksi pola, menengahai objek, bahkan memahami konteks dari yang masuk.
Reason	Setelah informasi disimpan, manusia menganalisis dan memahami hubungan antar-informasi, menggunakan logika dan penalaran untuk membuat keputusan atau menyelesaikan masalah.	Kemampuan AI untuk menggunakan logika, aturan, atau penalaran berbasis pengetahuan untuk membuat keputusan atau menarik kesimpulan dari data yang ada. Hal ini mencerminkan proses berpikir yang lebih kompleks dibandingkan dengan sekadar mengumpulkan atau memproses informasi.
Infer	Merujuk pada kemampuan untuk menyimpulkan atau membuat kesimpulan dari informasi yang ada tanpa memerlukan penjelasan eksplisit. Dalam konteks ini, manusia menggunakan pengetahuan yang telah dimiliki sebelumnya dan petunjuk kontekstual untuk memahami sesuatu yang secara langsung dikatakan atau dijelaskan. Proses inferensi melibatkan logika, penalaran, dan pengalaman masa lalu.	Proses pengambilan keputusan atau keputusan berdasarkan data, model, atau aturan yang telah dipelajari oleh sistem. AI menggunakan inferensi untuk menganalisis input dan menghasilkan output yang relevan atau memprediksi hasil berdasarkan pola dan hubungan yang telah dilatih sebelumnya.

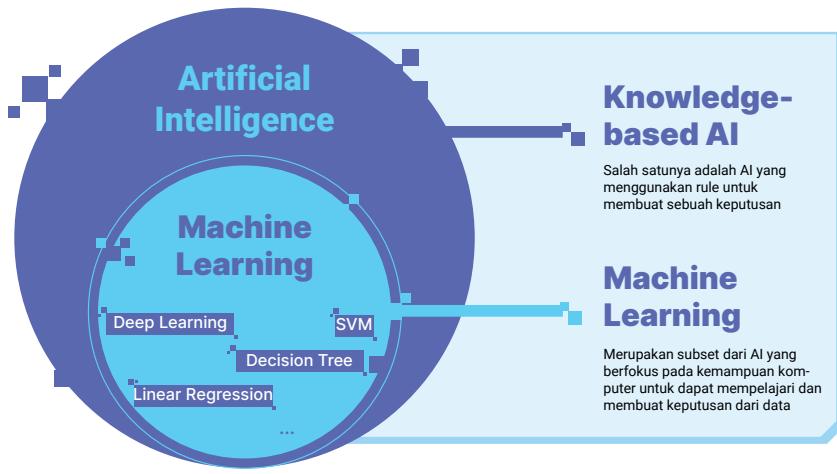
AI sering kali dibandingkan dengan metode-metode analisis lain seperti *causal modelling*, ekonometrik, atau statistik karena pada beberapa kasus dapat digunakan untuk menyelesaikan masalah yang sama.

1.1 Artificial Intelligence and Machine Learning

Saat ini, sering kali definisi AI dan *Machine Learning* saling dipertukarkan satu sama lain. Kenyataannya, AI merupakan cakupan yang lebih luas dan dalam hal ini, *machine learning* adalah bagian dari AI. Tentu saja tidak semua AI adalah *machine learning*, tetapi *machine learning* merupakan salah satu manifestasi dari AI berdasarkan kuadran yang terdefinisi oleh Russell dan Norvig (2003, 2021). Pada dasarnya, terdapat karakteristik utama yang membedakan AI *machine learning* dengan AI *non-machine learning* atau sering disebut sebagai *knowledge-based* AI terutama dalam pemrosesan data atau informasi. *Knowledge-based* AI memungkinkan pemrosesan

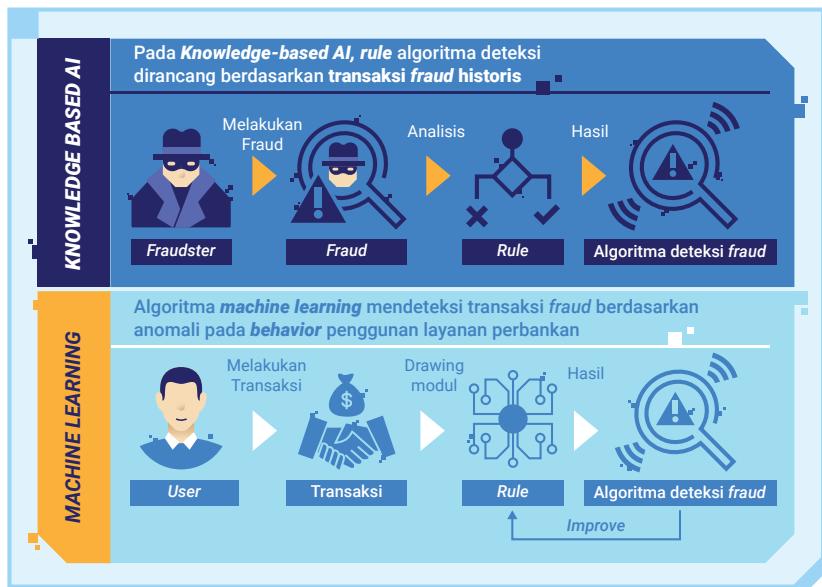
informasi dilakukan secara eksplisit dengan memanfaatkan pengetahuan yang dimiliki oleh pengembang model. Sistem ini, walaupun relatif *straightforward* dan bersifat deterministik, memiliki keterbatasan dalam hal fleksibilitas.

Di sisi lain, jenis AI yang kedua (*machine learning*) merupakan sistem yang memungkinkan pemrosesan informasi dilakukan secara implisit oleh mesin itu sendiri berdasarkan data yang diberikan. Sifat AI jenis ini lebih probabilistik dibandingkan jenis pertama. Jenis ini merupakan yang paling popular saat ini mengingat kemampuannya untuk mempelajari pola data *input* secara adaptif, memiliki fleksibilitas terhadap berbagai jenis data, tanpa memerlukan intervensi yang diprogram secara eksplisit dari manusia. Secara umum hubungan antara AI dan ML dapat diilustrasikan melalui diagram Venn pada Gambar 3 berikut.



Gambar 3 Al vs. ML

Sebagai ilustrasi untuk membedakan antara *machine learning* dan *knowledge-based AI* (AI *non-machine learning*) dapat digambarkan dalam pemanfaatannya untuk mendeteksi aktivitas *fraud* (amatil Gambar 4). Dalam pendekatan *knowledge-based*, deteksi penipuan bergantung pada *rule* yang dibentuk dari pengalaman manusia sebelumnya. *Fraudster* melakukan tindakan penipuan, yang kemudian diidentifikasi oleh analisis berdasarkan perilaku masa lalu. Dari hasil analisis ini, aturan-aturan tertentu diterapkan untuk mendeteksi penipuan di masa mendatang. Keterbatasan pendekatan ini terletak pada sifatnya yang statis; hanya perilaku yang telah dikenali sebelumnya yang dapat diidentifikasi sebagai penipuan, dan aturan-aturan ini tidak mudah disesuaikan dengan perubahan pola atau teknik penipuan baru.



Gambar 4 Machine Learning vs. Knowledge Based AI

Sebaliknya, dalam pendekatan berbasis *machine learning*, sistem dibangun dengan mempelajari pola transaksi pengguna dari data yang dihasilkan. *Machine learning* memungkinkan model untuk dilatih menggunakan transaksi yang dilakukan oleh manusia, sehingga model dapat mempelajari pola normal dan mendeteksi perilaku yang dianggap anomali atau tidak biasa. Pendekatan ini lebih dinamis, karena model *machine learning* dapat terus diperbarui dan disempurnakan berdasarkan data baru, sehingga deteksi penipuan menjadi lebih fleksibel dan mampu beradaptasi dengan berbagai pola baru yang dilakukan oleh penipu.

1.2 Machine Learning and Econometrics

Pada dasarnya, baik *machine learning* maupun *econometrics* memiliki kesamaan, yaitu melakukan prediksi dan atau analisis berdasarkan data (Charpentier *et al*, 2018). Dalam konteks ilmu ekonomi, *machine learning* umumnya dilakukan untuk proses prediktif, deskriptif, maupun preskriptif.

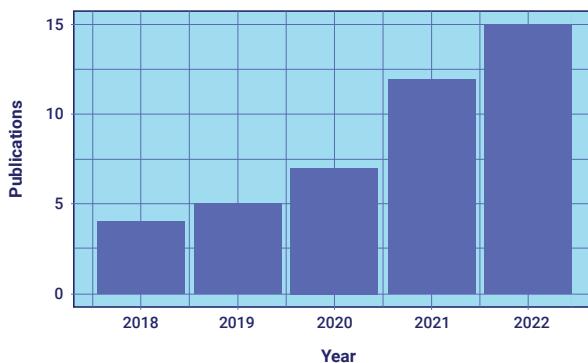
Pemodelan dengan *econometrics* umumnya memberikan penekanan untuk memahami fenomena ekonomi, dengan beberapa asumsi-asumsi ekonomi yang juga telah didefinisikan. Di sisi lain, *machine learning* lebih bertujuan pada pengembangan algoritma berdasarkan data dengan melakukan pelatihan agar

meminimalisasi eror. Secara lebih rinci, beberapa perbandingan antara *Machine Learning* dan *Econometrics* dapat disimak pada Tabel 2.

Tabel 2 Perbandingan ML dan *Econometrics*

	<i>Machine Learning</i>	<i>Econometrics</i>
1 Filosofi tujuan	<i>Prediction, Pattern Recognition</i>	<i>Causal Inferences, Hypothesis Testing</i>
2 Modelling	<i>Data Driven</i> , sehingga lebih fleksibel	<i>Theory & assumption driven</i> , sehingga lebih kaku
3 Interpretability	<i>Tradeoff: Interpretability vs Complexity</i>	Secara alami <i>interpretable</i>
4 Jumlah Data	<i>Data quality/quantity vs performance</i>	Dapat bekerja pada data yang lebih kecil
5 Proses Estimasi	Algoritmik dan aproksimasi (hampiran)	<i>Closed Form</i> (lebih eksak): OLS, MLE
6 Causality Inference	Pada model tertentu (kausal ML)	Penekanan pada <i>causality</i> lebih kuat
7 Overfitting Handling	<i>Cross Validation, Regularisation</i>	Penambahan batasan berbasis hipotesis
8 Nonlinearity	Lebih fleksibel terhadap nonlinearitas	Umumnya mengasumsikan linear
9 Aplicability	Lebih fleksibel: s.d. <i>unstructured data</i>	Diaplikasikan <i>structured data</i>

Penggunaan *machine learning* telah masif digunakan dalam bidang keilmuan ekonomi. Sebagai contoh, statistik yang dirangkum oleh Desai (2023) menunjukkan bahwa jumlah riset ekonomi yang memanfaatkan *machine learning* meningkat setiap tahun, setidaknya s.d. tahun 2022.



Gambar 5 Jumlah publikasi bidang ekonomi yang memanfaatkan *machine learning*, berdasarkan 10 top jurnal di bidang ekonomi.

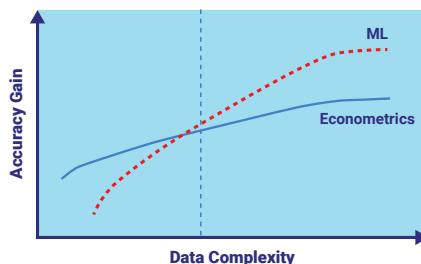
Secara umum, kedua pendekatan dapat digunakan untuk menganalisis data terkait ekonomi. Penggunaan *machine learning* lebih ditekankan pada pemanfaatan untuk analisis data nontradisional, nonlinear, berukuran besar, kompleks, tidak terstruktur

dengan *forecast horizon* yang disarankan umumnya bersifat *shorter term*. Sementara itu, pendekatan ekonometrik lebih dititikberatkan pada permasalahan yang lebih struktural dengan data yang relatif kecil, tradisional, dengan *forecast* yang disarankan umumnya bersifat *longer term*.

Lalu bagaimana peran ML dalam melengkapi ekonometrik untuk memperkuat analisis ekonomi? Setidaknya terdapat dua penguatan (Hsiao, 2024), yaitu:

- ML dapat memungkinkan analisis berdasarkan multimodal data dengan granularitas yang berbeda sehingga meningkatkan kualitas analisis, misalnya kausalitas.
- ML dapat memungkinkan analisis dengan data yang berdimensi tinggi.
- ML, terutama *unsupervised* memungkinkan agregasi informasi, misalnya melalui *clustering*. Hal ini memudahkan analis untuk memahami data ataupun fenomena berdasarkan pola secara garis besar.

Saat ini kita berada di era *big data*. Pertanyaan kemudian, bagaimana pengaruh kompleksitas terhadap performa dilihat dari kedua sudut pandang baik *machine learning* dan *econometrics*?



Gambar 6 Perbandingan Performa ML dan *Econometrics*

Secara umum, dapat dikatakan bahwa *machine learning* mungkin lebih cocok untuk menganalisis data dengan kompleksitas yang lebih tinggi dibandingkan dengan ekonometrik. Sebaliknya, bila kompleksitas data cenderung rendah, asumsi yang kuat telah diketahui, maka ekonometrik lebih cocok karena terdapat potensi memiliki akurasi yang lebih unggul dari *machine learning*.

1.3 Current Trends and Breakthrough in AI

Mengapa AI begitu penting? Mengapa AI semakin mendominasi dalam pemanfaatan teknologi digital? Jawabannya sederhana: **AI mampu menyelesaikan tantangan yang tidak bisa diatasi oleh manusia dengan cara tradisional.**

Pertama, AI memiliki kemampuan untuk mengolah dan menganalisis *big data*. Kita hidup dalam era yang dibanjiri data yang diciptakan secara masif (*data deluge*), mulai dari transaksi keuangan hingga rekam medis digital. Manusia akan kesulitan untuk memproses semua informasi ini secara manual. Di sinilah AI hadir sebagai solusi dengan memanfaatkan berbagai algoritma, a.l: *machine learning*, untuk menganalisis data secara *real-time*, menemukan pola yang tersembunyi, dan menghasilkan *insight*.

Selain itu, AI dapat meningkatkan efisiensi dan produktivitas. Di sektor bisnis, misalnya, AI telah mengotomatisasi banyak proses yang sebelumnya membutuhkan tenaga kerja manusia, seperti analisis data, layanan pelanggan, serta mendeteksi *fraud* secara cepat (Russell, McAfee, etc). *Chatbot* berbasis AI mampu menangani pertanyaan pelanggan dengan cepat. *Fraud detection system* dapat membantu perusahaan mengidentifikasi transaksi mencurigakan/*fraud* dengan cepat dan akurat.

Perkembangan AI diyakini akan semakin pesat ke depannya (Jensen, 2024, Jordan & Mitchell, 2015). Perkembangan AI saat ini berada di tengah momentum besar yang dipicu oleh sejumlah inovasi signifikan. Teknologi seperti **Generative AI** dan **Large Language Models (LLM)**, **Quantum Computing**, **Responsible AI**, dan **AI Agents** mulai mengubah cara kita berinteraksi dengan teknologi, memberikan solusi bagi berbagai permasalahan, dan membuka berbagai peluang baru.

Salah satu tren yang sudah sering kita dengar adalah pesatnya perkembangan *Generative AI* dan *Large Language Models (LLM)*. **Generative AI** adalah cabang AI yang memungkinkan mesin menciptakan konten baru, seperti teks, gambar, dan bahkan musik, berdasarkan data yang telah dipelajarinya. Contoh yang paling terkenal adalah GPT (*Generative Pre-trained Transformer*), yang dikembangkan oleh OpenAI, Gemini yang dibangun oleh Google, dan Llama oleh Meta (Jensen, 2024). Dengan model seperti GPT-3, GPT-4, Gemini 1.5, dan Llama 3.1, AI kini mampu menghasilkan teks yang sangat menyerupai hasil manusia, menulis artikel, menjawab pertanyaan, hingga menciptakan kode pemrograman (Brown, 2020). LLM ini telah merevolusi berbagai aplikasi, mulai dari asisten virtual, *chatbot*, hingga generasi artikel secara otomatis. Namun, di balik keunggulannya, ada kekhawatiran mengenai penyalahgunaan teknologi ini, terutama dalam menciptakan disinformasi atau manipulasi konten.

Quantum computing adalah teknologi komputasi yang memanfaatkan prinsip-prinsip mekanika kuantum untuk melakukan komputasi yang jauh lebih cepat daripada komputer klasik. Dalam sistem komputasi klasik, informasi diproses dalam bit yang hanya bisa berada dalam dua keadaan, yaitu 0 atau 1. Sebaliknya, dalam *quantum computing*, informasi diproses dalam qubit, yang memungkinkan superposisi, di mana sebuah qubit bisa berada dalam keadaan 0 dan 1 secara bersamaan. Hal ini memungkinkan *quantum computers* untuk melakukan banyak perhitungan sekaligus. Sehingga, secara teoretis, dapat menyelesaikan masalah yang sangat kompleks jauh lebih cepat daripada komputer klasik.

Bagi dunia AI, *quantum computing* menawarkan potensi yang sangat besar. Saat ini, model jaringan saraf tiruan (*neural network*) yang kompleks seperti *deep learning* (Bengio et al, 2015), membutuhkan daya komputasi yang sangat besar untuk melatih model dengan data berukuran besar. *Quantum computing* bisa menjadi solusi untuk mengurangi waktu pelatihan. Dalam aplikasi seperti pengenalan gambar (*image processing*), pemrosesan bahasa alami (*natural language processing*), dan simulasi yang kompleks, *quantum computing* dapat membantu mempercepat pengolahan dan analisis data dibandingkan teknologi komputasi klasik.

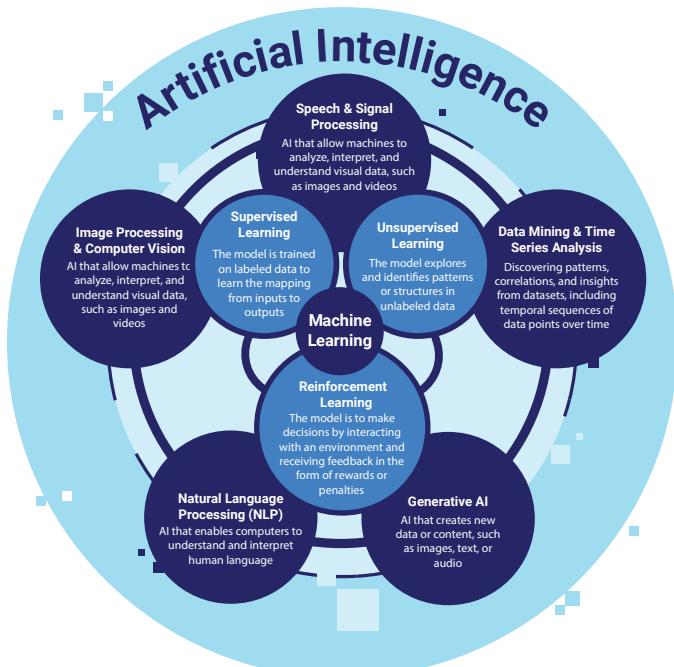
Di samping itu, perkembangan **AI agents** juga tak kalah penting. **AI agents** adalah sistem yang mampu bertindak secara mandiri berdasarkan data yang mereka terima dan keputusan yang mereka buat sendiri. Contoh pemanfaatan teknologi ini adalah kendaraan otonom yang dapat beroperasi tanpa pengemudi, serta robot dalam industri manufaktur yang bekerja secara otomatis dalam lini produksi. Di masa depan, **AI agents** diproyeksikan akan semakin terintegrasi dalam kehidupan sehari-hari, dari rumah pintar yang mampu mengatur konsumsi energi secara efisien hingga sistem transportasi yang diatur oleh **AI agents** untuk mengurangi kemacetan dan meningkatkan keselamatan berlalu lintas.

Tren lainnya yang semakin berkembang adalah konsep **Responsible AI**, yaitu pengembangan AI yang menekankan pada etika, transparansi, akuntabilitas, dan keadilan. Di era ketika AI semakin banyak dimanfaatkan dalam pengambilan keputusan, seperti dalam sistem rekrutmen, penentuan kelayanan kredit perbankan, dan layanan kesehatan, risiko bias, dan pelanggaran privasi perlu menjadi perhatian serius. Algoritma yang dilatih pada data yang tidak representatif dapat menciptakan hasil yang bias, misalnya diskriminasi rasial dalam identifikasi wajah atau bias *gender* dalam pemrosesan aplikasi pekerjaan. Oleh karena itu, **Responsible AI** menjadi sangat penting untuk memastikan bahwa teknologi AI bekerja secara adil dan tidak menguntungkan atau merugikan kelompok tertentu. Selain itu, transparansi juga menjadi kunci, di mana masyarakat perlu memahami bagaimana dan mengapa keputusan diambil oleh sebuah sistem AI. Inisiatif ini akan membantu membangun kepercayaan publik terhadap AI dan memastikan bahwa teknologi ini benar-benar membawa manfaat bagi semua pihak.

Ke depan, AI akan semakin terintegrasi dengan kehidupan kita sehari-hari. Kita akan melihat AI yang lebih kolaboratif dan lebih cerdas. Namun, dengan semua inovasi ini, penting untuk terus mendorong pengembangan **Responsible AI** untuk memastikan bahwa teknologi ini membawa manfaat maksimal bagi semua orang, tanpa mengorbankan nilai-nilai fundamental seperti keadilan, privasi, dan transparansi.

AI bukan sekadar teknologi; ia adalah refleksi dari kemampuan manusia untuk mendorong batasan intelektualnya. Dengan pemanfaatan yang bertanggung jawab, AI akan terus berkembang memberikan dampak positif yang membentuk masa depan yang lebih baik bagi umat manusia.

1.4 AI Landscape



Gambar 7 *Landscape: Machine Learning, AI, dan Aplikasinya*

Lanskap AI yang dirangkup pada buku ini diilustrasikan sebagaimana pada Gambar 8. Sejauh ini terdapat berbagai macam AI dan *machine learning* yang dikembangkan setiap saat. Dapat dikatakan bahwa *machine learning* merupakan suatu bagian dari AI. Artinya, tidak semua AI adalah *machine learning*, sementara *machine learning* itu sendiri adalah AI (AI yang dapat belajar dari data). Perlu diketahui bahwa *machine learning* adalah jenis AI yang popular akhir-akhir ini karena fleksibilitasnya dalam belajar dari data.

Pada layer *learning*, buku ini merangkum *landscape learning* menjadi tiga jenis utama, yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning* (Murphy (2012) dan Murphy (2024)). Penjelasan masing-masing jenis *learning* tersebut dapat disimak pada Bagian 2.

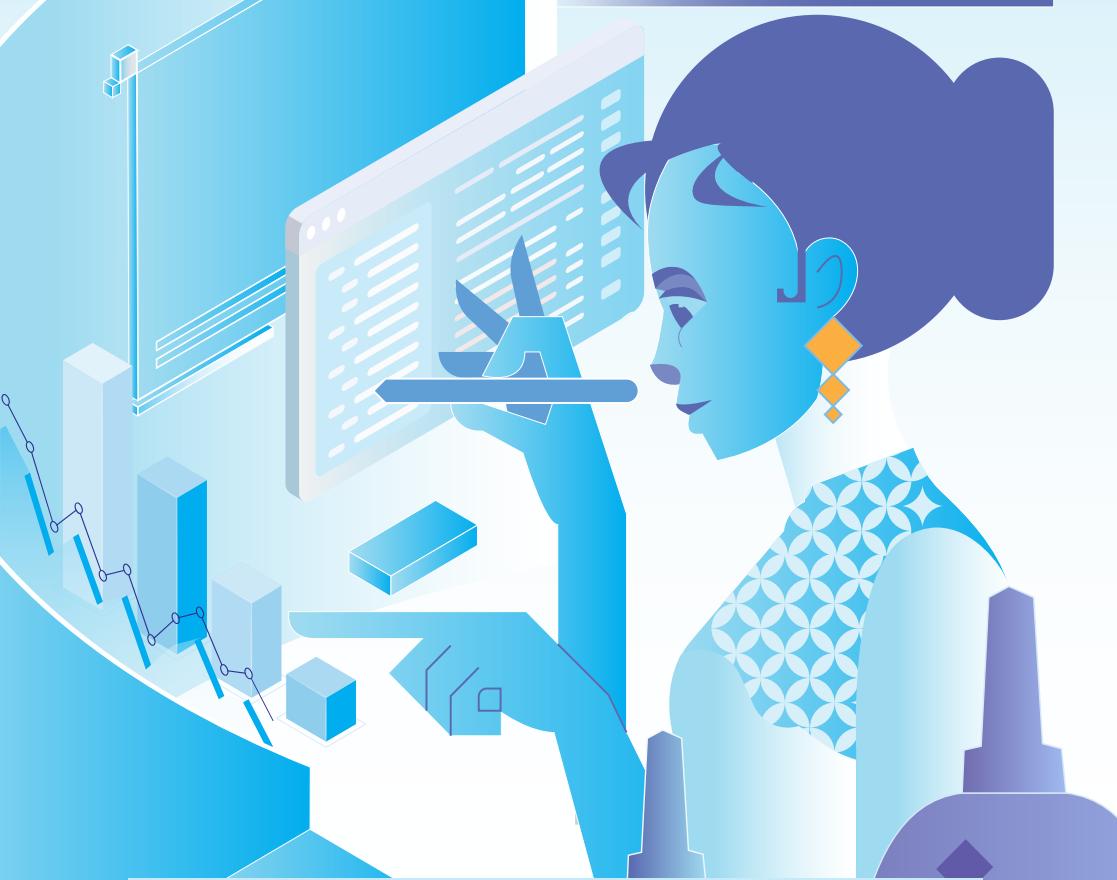
Di layer berikutnya, buku ini merangkum jenis pengaplikasian *machine learning* ke dalam lima kategori aplikasi berdasarkan jenis data dan/atau *task*, yaitu *natural language processing (NLP)*, *Image Processing & CV*, *speech & signal processing*, *Data Mining & time series analysis*, serta *Generative AI*. Hubungan antar-layer dapat dirangkum sebagaimana pada diagram venn Gambar 7 tersebut.

BAGIAN 2

MACHINE LEARNING



a quick concept example



"The real power of machine learning is not just about prediction, but about understanding the underlying patterns." - Andrew Ng.

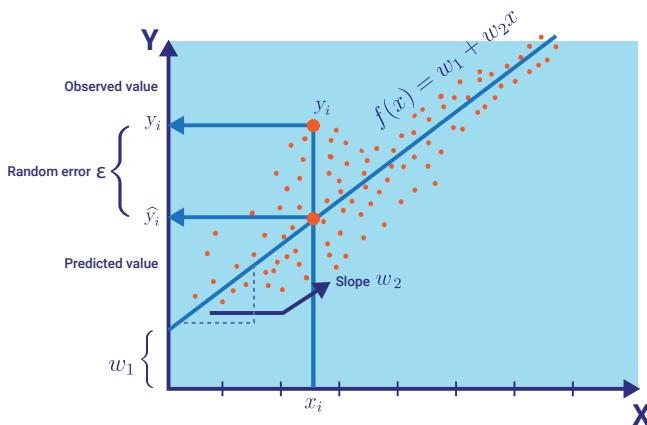
2.1 Linear Regression: Ide Mesin Belajar

Umumnya *machine learning* dapat dikatakan sebagai proses pemodelan aproksimasi fungsi berdasarkan data. Dengan demikian tujuan dari aktivitas pembelajaran mesin adalah menghasilkan suatu fungsi yang cukup representatif terhadap fenomena. *Linear regression* umum digunakan sebagai *starting point* memahami *machine learning*. Hal ini disebabkan karena *linear regression* memberikan ide dasar dan inspirasi bagaimana mesin belajar dari data. Bahasan pada bagian ini akan mendiskusikan *linear regression* dalam perspektif *machine learning*. ***Linear (in parameter) regression*** bertujuan memodelkan hubungan antara variabel independen (*input*) dan variabel dependen (*output*) – berupa nilai *real (continuous)* dengan asumsi bahwa hubungan antara parameter dan *input* bersifat linear, atau dengan kata lain, *output* dapat dijelaskan sebagai kombinasi linear dari parameter dan *input*. Secara matematis, suatu *linear regression* yang sederhana dapat diformulasikan sebagai berikut.

$$\hat{y}_i = w_1 + w_2 x_i + \varepsilon_i$$

dependent variable intercept slope coefficient independent variable random error term
 komponen linear komponen random error

Dengan ilustrasi grafis adalah seperti pada Gambar 8 berikut.



Gambar 8 Ilustrasi Linear Regression

Dalam hal ini w_1 dan w_2 adalah *latent variable (unknown variable)* yang perlu diestimasi berdasarkan data (X, y) .

Bentuk yang lebih umum

Suatu *linear regression* menitikberatkan hubungan linear antara *output* dengan parameter, meskipun bisa jadi hubungan antara *input* dan *output* tidak linear. Secara umum, ekspresi *linear regression* dapat direpresentasikan sebagai berikut:

$$y \approx f = \mathbf{w}^T \phi(\mathbf{x}) \quad (2.1)$$

Dengan $\phi(\mathbf{x})$ adalah *basis function*, yang bertujuan memetakan x ke dalam suatu ruang, baik itu linear maupun nonlinear. Misalnya pada contoh formula pada Persamaan (2.1) di atas *basis function* yang digunakan adalah polynomial dengan derajat $K-1$, maka bentuk persamaannya menjadi:

$$y \approx f = \mathbf{w}^T \phi(\mathbf{x}) = w_1 + w_2 x + w_3 x^2 + \dots + w_K x^{K-1} \quad (2.2)$$

Pilihan *basis function* selain polynomial tersebut misalnya adalah *radial basis function* (RBF), atau *logistic sigmoid function*.

2.2 Training and Test Sets and Generalisation Error

- *Training*: Pada tahap ini, model mempelajari pola dan hubungan dari data *input* yang sudah diberi label. Data dilengkapi dengan jawaban atau hasil yang diinginkan, yang membantu model mempelajari cara mengklasifikasikan atau memprediksi *output* berdasarkan *input*.
- Proses *fitting* dilakukan dengan menghitung *discrepancy*, maupun pengukuran seberapa dekat hasil prediksi dengan nilai yang sebenarnya. Estimasi parameter tersebut dapat dilakukan secara *closed form solution*/ solusi eksak misalnya dengan *ordinary least square* (OLS), atau secara aproksimasi misalnya *gradient descent*.
- *Testing*: Setelah model selesai dilatih, model diuji dengan menggunakan data baru yang tidak dilihat selama pelatihan. Data ini digunakan untuk menilai seberapa baik model dapat membuat prediksi berdasarkan apa yang telah dipelajari sebelumnya.

$$\text{Generalization Error} = E_{p(x,y)} [L(y, f(x))] = \int L(y, f(x)) p(x, y) dx dy \quad (2.3)$$

Generalisation error tersebut dapat diaproksimasi berdasarkan sampel data. Hal ini perlu mengasumsikan bahwa kita telah memiliki data yang distribusinya telah diketahui, atau dengan kata lain sampel yang kita dapatkan telah merepresentasikan fenomena. Secara matematis, ekspresi *error* kemudian menjadi:

$$\text{Average test error} = \frac{1}{M} \sum_{m=1}^M \left(y^{(m)}, f(\mathbf{x}^{(m)}) \right) \quad (2.4)$$

dengan

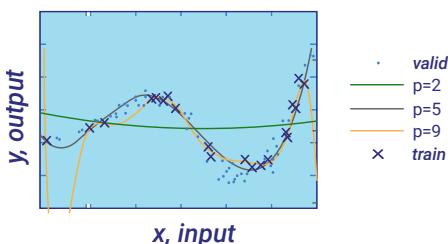
$$\mathbf{x}^{(m)}, y^{(m)} \sim p(\mathbf{x}, y) \quad (2.5)$$

Sebagai contoh, berikut *objective function* (fungsi tujuan) ketika proses pelatihan model:

$$\begin{aligned} E_\lambda(w; y, \Phi) &= \sum_{n=1}^N \left[y^{(n)} - f(x^{(n)}; w) \right]^2 + \lambda \sum_{k=1}^K w_k^2 \\ &= (y - \Phi w)^T (y - \Phi w) + \lambda w^T w \end{aligned} \quad (2.6)$$

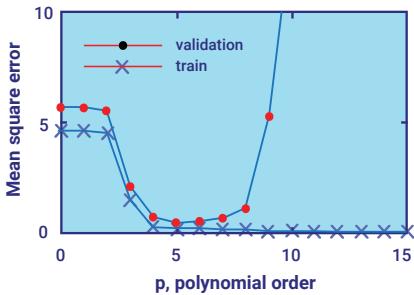
Catatan: Kualitas pelatihan diukur dengan *Euclidean distance* (L2-norm). Perhatikan bahwa dalam hal term $\lambda \sum_{k=1}^K w_k^2$ ditambahkan untuk mengantisipasi adanya solusi ekstrem yang menyebabkan *overfitting*. Variabel λ merupakan konstanta yang merepresentasikan bobot penalti. Dalam literasi *machine learning*, penalti semacam ini memiliki berbagai nama, a.l. Tikhonov Regularisation, *Ridge Regression*, atau L2-norm penalty (L2-norm regularization). Sebagai refresher: *overfitting* merupakan kondisi ketika model terlalu kompleks, sehingga gagal melakukan generalisasi. Lawan dari *overfitting* adalah kondisi ketika model terlalu sederhana, sehingga gagal juga melakukan generalisasi.

Sebagai ilustrasi pada contoh di atas, berikut ini grafik yang merepresentasikan pengaruh kompleksitas model.

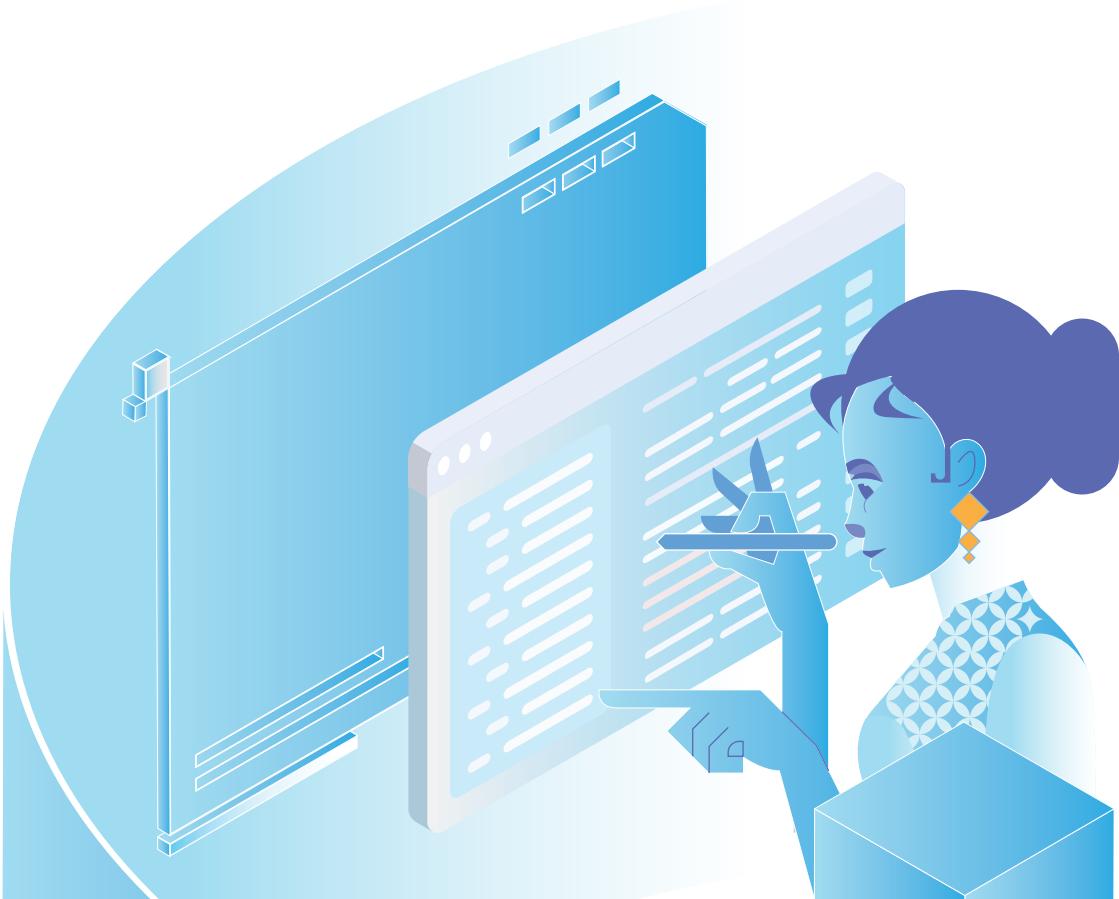


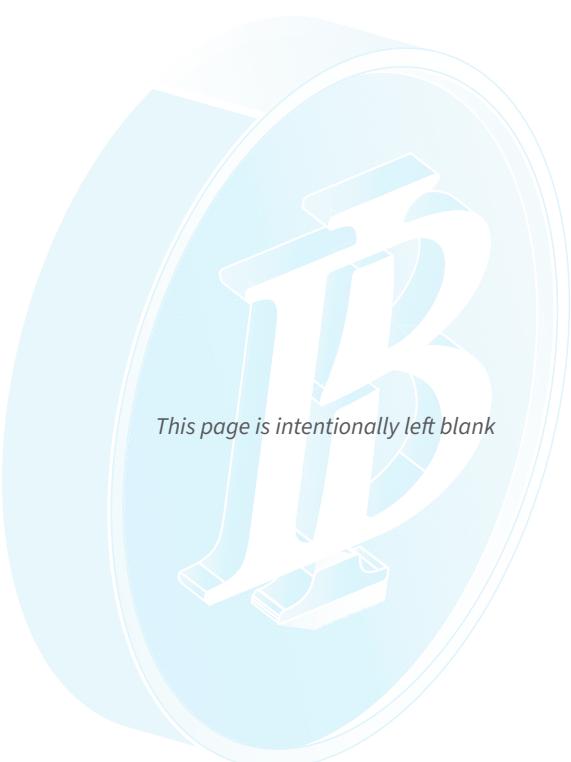
Gambar 9 Plot pengaruh *degree* (p) model pada *training data*. (Img src: Arno Onken/Iain Murray)

Selanjutnya, apabila kita lakukan analisis lebih lanjut terkait kompleksitas model ketika diujikan pada *test data (validation data)*, maka kemungkinan kurva dapat terlihat sebagaimana pada Gambar 10 berikut.



Gambar 10 Plot pengaruh *degree (p)* terhadap error apabila diujikan pada *test data (validation data)*. (img src: Arno Onken/Iain Murray)





This page is intentionally left blank

BAGIAN 3

LEARNING LANDSCAPE

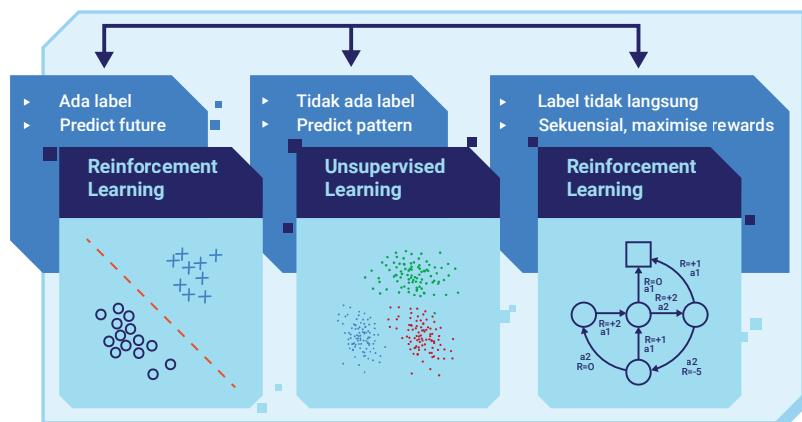


"Hundreds of new learning algorithms are invented every year."

– Pedro Domingos, The Master Algorithm.



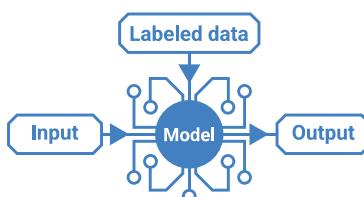
Machine Learning merupakan suatu jenis AI yang memungkinkan komputer dapat memiliki pengetahuan dengan belajar sendiri dengan cara menemukan pola berdasarkan data tanpa melalui pemrograman secara eksplisit. Dalam dunia *machine learning*, terdapat beberapa jenis pendekatan untuk mencapai tujuan ini, tetapi secara umum dapat digolongkan menjadi *supervised learning*, *unsupervised learning*, dan *reinforcement learning*.



Gambar 11 Perbandingan Pendekatan ML

3.1 Supervised Learning

Supervised learning atau *supervised machine learning* adalah salah satu pendekatan *machine learning* di mana model dilatih menggunakan data berlabel atau jawaban yang benar.



Gambar 12 Ilustrasi Supervised Learning

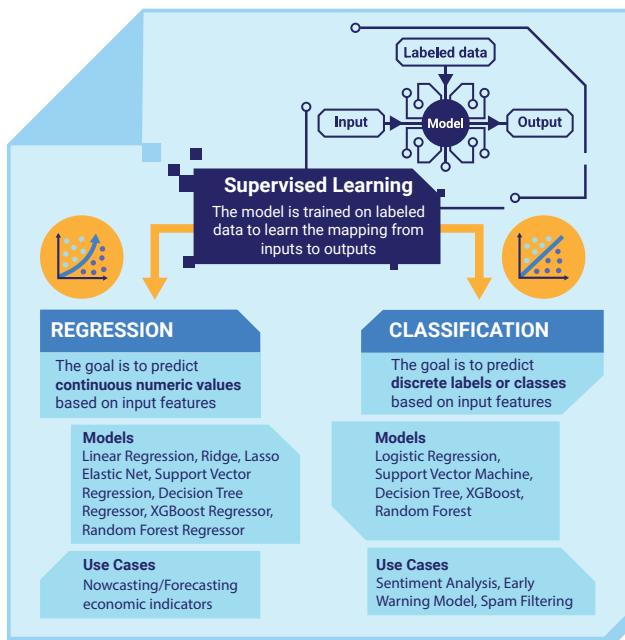
Dalam *supervised learning*, algoritma mempelajari hubungan antara *input* dan *output* menggunakan data yang telah diberi label, sebagaimana digambarkan pada Gambar 12. Model yang dilatih ini diharapkan dapat memprediksi *output* dengan akurat saat diberikan data baru yang belum dilihat sebelumnya.

Tujuan proses pembelajaran ini adalah untuk meminimalkan perbedaan antara prediksi model dan nilai sebenarnya. Oleh karena itu, evaluasi model dilakukan dengan membandingkan hasil prediksi dengan nilai aktual. Untuk lebih mudah memahami, *supervised learning* bisa dianalogikan seperti seorang guru yang memberikan soal (*input*) beserta jawabannya (*label*), selanjutnya siswa (*machine learning*) akan belajar untuk menjawab ketika diberikan soal baru berdasarkan contoh-contoh sebelumnya.

Regression vs. Classification

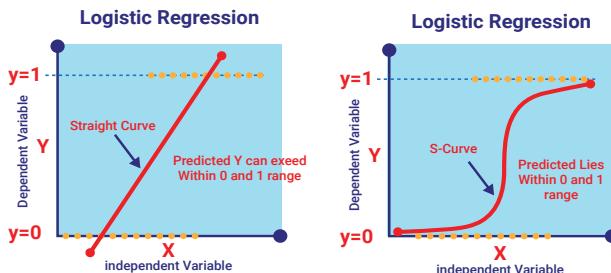
Lebih lanjut, *supervised learning* terbagi ke dalam 2 (dua) kategori berdasarkan jenis *dependent variable* sebagaimana diilustrasikan pada Gambar 13:

1. **Regresi (Regression):** Digunakan ketika *output* yang diinginkan adalah nilai kontinu, seperti memprediksi harga rumah berdasarkan berbagai faktor. Dalam konteks harga rumah, model regresi mencari pola hubungan antara fitur *input* (misal luas tanah, jumlah kamar, dan keterangan lokasi) dengan *output* berupa angka (harga rumah). Misalnya, model regresi dilatih menggunakan data historis rumah-rumah yang pernah dijual, di mana setiap data memiliki fitur *input* (misal: luas = 150 m², kamar = 3, lokasi = pusat kota) dan label *output* (harga = Rp1,2 miliar). Model belajar hubungan matematis antara fitur-fitur ini dengan harga rumah. Setelah dilatih, model bisa memprediksi harga rumah baru yang belum pernah dilihat sebelumnya. Contohnya, jika ada rumah dengan luas 200 m², 4 kamar, dan berlokasi di dekat stasiun, model mungkin memperkirakan harganya Rp1,8 miliar.
2. **Klasifikasi (Classification):** Digunakan ketika *output* yang diinginkan merupakan kategori atau kelas. Contohnya, mengklasifikasikan email sebagai *spam* atau *bukan spam*. Dalam konteks ini, model dilatih menggunakan email yang sudah dilabeli dengan fitur kata-kata dalam email (seperti “hadiah”, “klik di sini”, “anda menang”), alamat pengirim, atau pola penulisan lain. Berarti, data dengan kata-kata atau pola mencurigakan akan dimasukkan ke dalam label *spam*, sedangkan yang lainnya berarti non-*spam*. Hasilnya, model akan bisa memprediksi email baru. Contoh: Email dengan subjek “Anda menang lotre berikut, klik untuk mengetahui lebih lanjut!” model akan memasukkan email tersebut ke dalam kategori *spam*, sedangkan pada email dengan subjek “Laporan keuangan tahunan PT XXX”, model akan memasukkan email tersebut ke dalam kategori non-*spam*.



Gambar 13 Jenis dan contoh penerapan *Supervised Learning*

Contoh klasifikasi: *logistic regression* digunakan untuk memperkirakan *discrete value* berdasarkan satu set variabel independen yang diberikan. Pada dasarnya *logistic regression* merupakan bentuk dari *linear regression* yang *output*-nya dipetakan ke dalam fungsi *logit*. Dalam hal ini, kombinasi *linear* dari set *input* tersebut akan memprediksi probabilitas, sehingga nilai *output*-nya akan terletak antara 0 (minimal) dan 1 (maksimal). Berikut ini adalah ilustrasi perbedaan antara *logistic regression* dan *linear regression* sederhana.



Gambar 14 Linear Regression vs. Logistic Regression

Secara matematis formula umum *logistic regression* adalah sebagai berikut.

$$p(x) = \frac{1}{1 + e^{-(w_0 + w_1 x)}} \quad (3.1)$$

Pertanyaan berikutnya adalah bagaimana ekspresi *loss function*? Hal ini menjadi *concern* karena y (target/dependen variabel) dalam hal ini bukanlah lagi *continuous*, tetapi menjadi *discrete* {0,1}. Perlu diketahui bahwa pada Gambar 14 di atas adalah *probability* $f(x; \mathbf{w}) = p(y=1|x, \mathbf{w})$. Dengan demikian, *maximum likelihood* (i.e. estimasi berdasarkan data) pada saat *fitting* model bertujuan untuk memaksimalkan *probability*:

$$L(\mathbf{w}) = \prod_{n=1}^N p(y^{(n)} | x^{(n)}, \mathbf{w}) \quad (3.2)$$

Dalam *machine learning*, perolehan solusi umumnya dilakukan dengan optimisasi melalui *minimization* daripada *maximization*. Agar minimisasi dapat dilakukan ekspresi sebagaimana pada Gambar 14 di atas, kemudian dapat direpresentasikan dalam *negative log likelihood* sebagai berikut.

$$\text{NLL} = -\log L(\mathbf{w}) = -\sum_{n=1}^N \log \left[\sigma(\mathbf{w}^T \mathbf{x}^{(n)})^{y^{(n)}} \left(1 - \sigma(\mathbf{w}^T \mathbf{x}^{(n)})^{1-y^{(n)}} \right) \right] \quad (3.3)$$

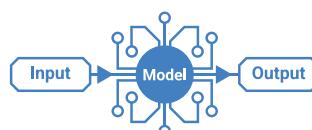
Bentuk lain dari Gambar 14 di atas adalah *cross entropy loss*, dengan ekspresi sebagai berikut.

$$L = y \log \hat{y} + (1-y) \log(1-\hat{y}) \quad (3.4)$$

Berbeda dengan *linear regression*, proses pencarian solusi *logistic regression* dilakukan dengan melakukan aproksimasi misalnya melalui *gradient descent*.

3.2 Unsupervised Learning

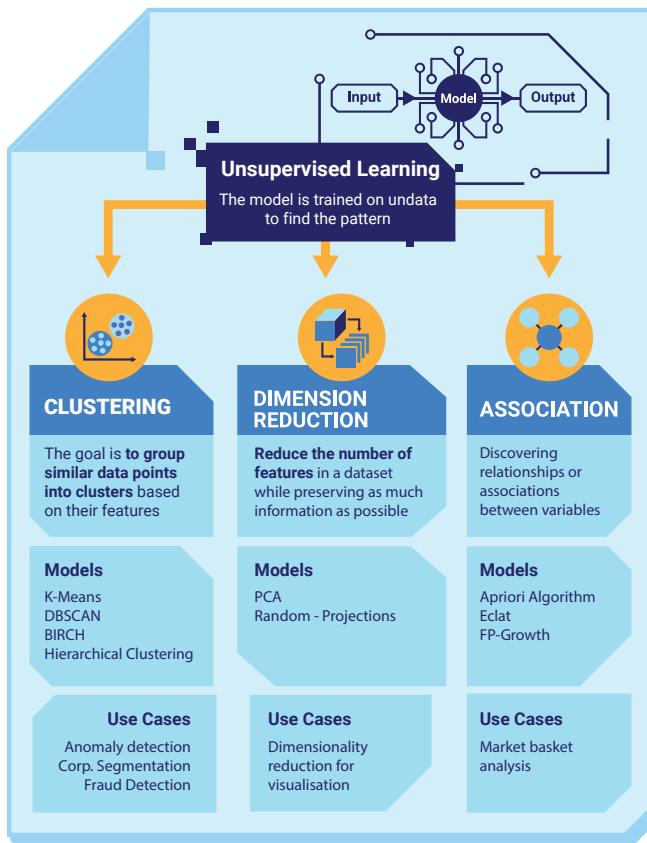
Unsupervised learning adalah kategori kedua *machine learning* yang digunakan untuk menganalisis data tanpa label atau target yang didefinisikan. Pendekatan ini bertujuan untuk menemukan pola atau struktur tersembunyi pada data secara lebih bebas. Dibandingkan dengan *supervised learning*, *unsupervised learning* memberikan fleksibilitas lebih besar dalam eksplorasi data untuk memperoleh wawasan sebelum proses analisis lebih lanjut. Hal ini dapat diilustrasikan pada Gambar 15 berikut.



Gambar 15 Ilustrasi *Unsupervised Learning*

Meskipun terdapat berbagai macam teknik dalam *unsupervised learning*, secara garis besar teknik popular mencakup a.l. *clustering*, *dimensionality reduction*, dan *association mining* sebagaimana pada Gambar 16.

Sebagai contoh, *clustering* biasanya dilakukan pada tahap *exploratory data analysis* untuk melihat a.l. gambaran umum segmentasi data, adanya kemungkinan *outlier* (penculan data), dsb.. Contoh lain misalnya *dimensionality reduction* antara lain untuk keperluan visualisasi data ke dalam 2 atau 3 dimensi, serta *association mining* untuk melihat pola keterhubungan.

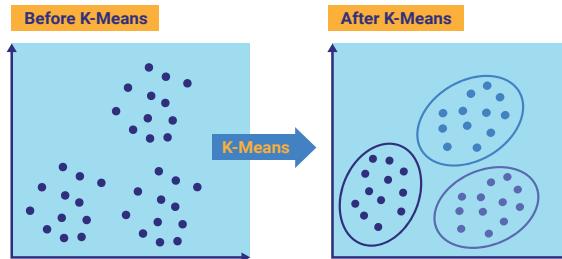


Gambar 16 Jenis dan contoh penerapan *unsupervised learning*

Bagian ini akan membahas teknik k-Means *Clustering*, salah satu teknik *unsupervised learning* yang popular, untuk mendemonstrasikan bagaimana model belajar tanpa label. Teknik ini memungkinkan pembagian dataset secara otomatis menjadi beberapa kelompok (*cluster*) berdasarkan kesamaan fitur atau karakteristik dalam

data. Contoh penggunaan teknik ini adalah pada *customer segmentation*, yang bertujuan untuk mengelompokkan pelanggan berdasarkan karakteristik yang serupa, seperti demografi, lokasi geografis, atau perilaku belanja.

K-Means clustering adalah algoritma *unsupervised machine learning* yang umum digunakan untuk pengelompokan data dan identifikasi pola. Prosesnya dimulai dengan memilih sejumlah titik pusat (*centroid*) secara acak. Algoritma kemudian menempatkan data ke dalam kelompok yang paling sesuai dengan pusat tersebut, dan secara iteratif memperbarui posisi pusat hingga mencapai pengelompokan yang optimal.



Gambar 17 Ilustrasi k-Means Clustering

Secara teknis, metode pembagian data dengan *k-Means* diilustrasikan sebagaimana pada Gambar 17. Diberikan himpunan pengamatan (x_1, x_2, \dots, x_n) dengan tiap pengamatan berupa vektor riil berdimensi d . Pengklasteran k rata-rata bertujuan untuk membagi n pengamatan ke dalam k ($\leq n$) kelompok $S = \{S_1, S_2, \dots, S_k\}$ sedemikian hingga ragam dalam kelompok minimum. Secara matematis, tujuan algoritma ini adalah:

$$\arg \min_S \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2 = \arg \min_S \sum_{i=1}^k |S_i| Var S_i \quad (3.5)$$

dengan μ_i adalah titik tengah (rata-rata) dalam himpunan S_i .

Secara garis besar, pemelajaran *k-Means clustering* terdiri atas beberapa tahap mencakup:

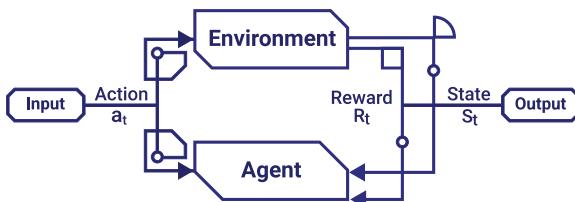
1. Pilih k buah titik tengah secara acak.
2. Kelompokkan data sehingga terbentuk k buah kelompok dengan titik tengah tiap kelompok merupakan titik tengah yang telah dipilih sebelumnya.
3. Perbarui nilai titik tengah tiap kelompok.

$$\mu_k = \frac{1}{N_k} \sum_{j=1}^{N_k} x_j \quad (3.6)$$

4. Ulangi langkah 2 dan 3 sampai titik tengah semua kelompok tidak lagi berubah.

3.3 Reinforcement Learning

Reinforcement learning (RL) adalah salah satu jenis *machine learning* yang bertujuan untuk membuat serangkaian keputusan untuk mendapatkan hasil yang optimal (Mnih et al, 2015). *Reinforcement learning* pada umumnya diaplikasikan pada permasalahan-permasalahan yang bersifat **sekuensial**, dalam hal ini menentukan strategi berturut-turut untuk memaksimalkan hasil, misalnya aplikasi *Chess* (Catur). Sebagai ilustrasi, Gambar 18 berikut merepresentasikan bagaimana RL memaksimalkan *output* berdasarkan *input* dan *environment* yang diberikan:



Gambar 18 Cara kerja Reinforcement Learning

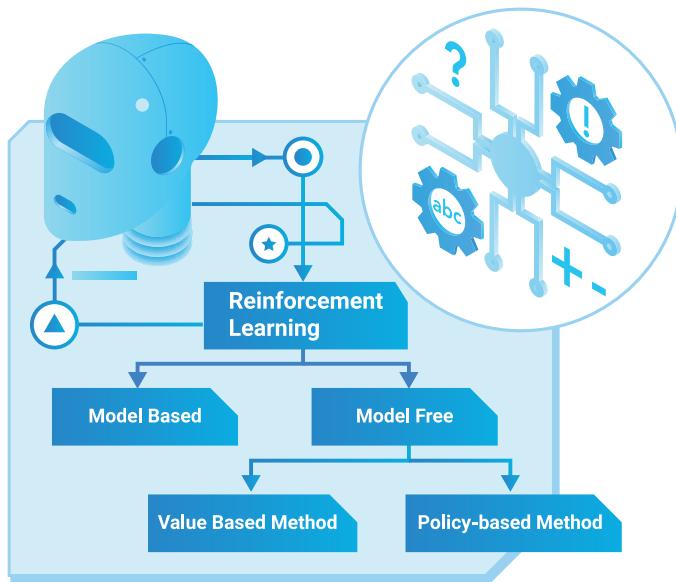
Saat pemelajaran, label termasuk *rewards* dan *punishment* umumnya tidak diberikan secara langsung. Fokus RL adalah penyeimbangan antara eksplorasi (wilayah yang belum dipetakan) dan eksplorasi (berdasarkan pengetahuan yang dimiliki dari historis).

Agent akan belajar untuk mencapai tujuan, memaksimalkan *rewards* dalam lingkungan yang kompleks dan tidak pasti (*complex and uncertain environment*). Dalam pembelajaran ini, agen atau mesin akan melakukan berbagai percobaan, pemelajaran, dan evaluasi performa. Agen akan menentukan tindakan yang paling sesuai berdasarkan situasi yang sedang dihadapi.

Untuk memanifestasikan *intelligent behaviour* sebagaimana tersebut, formulasi matematika dan komputasi serta *cognitive science* dikembangkan sehingga agen dapat memiliki kemampuan untuk beradaptasi dan menentukan langkah terbaik berdasarkan kondisi yang sedang dihadapi. Hal inilah yang membedakan *reinforcement learning* dari *supervised learning* dan *unsupervised learning*. Jenis pemelajaran ini kini mulai banyak diimplementasikan, meski masih belum sebanyak *supervised learning* dan *unsupervised learning*.

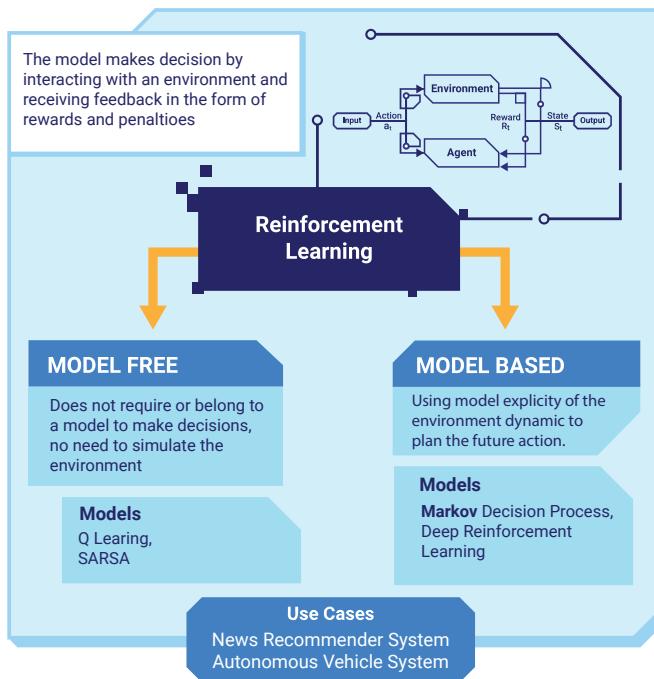
Saat ini, terdapat berbagai macam penggolongan RL berdasarkan berbagai parameter/kacamata. Dalam buku ini kami menggolongkan bahwa secara garis besar terdapat 2 (dua) jenis RL, yaitu *Model Free* dan *Model Based RL* dengan

mempertimbangkan pemodelan *environment*, dengan contoh model dan penerapan terlihat pada Gambar 19 dan Gambar 20 berikut.



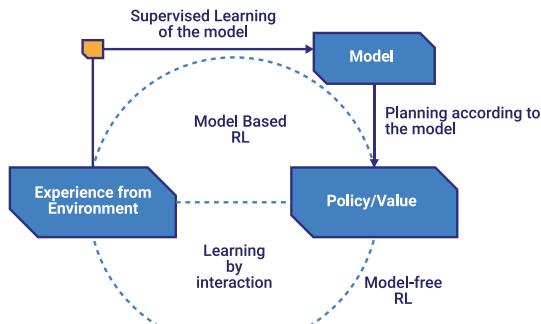
Gambar 19 Penggolongan Reinforcement Learning: Model Based vs. Model Free

1. **Tanpa Model (Model Free):** Menggunakan fungsi kebijakan dan/atau nilai, tetapi tidak memiliki model. Hal ini berarti bahwa agen akan belajar dengan tanpa pemodelan *environment*. Dengan kata lain, agen akan belajar sangat bergantung pada *environment* secara langsung. Jenis ini cocok untuk kondisi dengan data yang relatif banyak dan *environment* yang cukup kompleks. *Model Free* sendiri dapat dibagi menjadi dua kategori, di antaranya adalah:
 - a. **Berbasis Nilai (Value Based):** Tidak memiliki kebijakan (*policy*), memilih tindakan secara rakus (*greedily*) berdasarkan nilai-nilai keadaan.
 - b. **Berbasis Kebijakan (Policy Based):** Tidak memiliki fungsi nilai, menggunakan fungsi kebijakan untuk memilih tindakan.
2. **Berbasis Model (Model Based):** Menggunakan fungsi kebijakan dan/atau nilai serta memiliki model. Model yang dimaksud di sini berperan dalam penggambaran kondisi *environment*. Dengan adanya *environment modelling*, agen dapat lebih terbantu dalam beradaptasi untuk membuat keputusan/*action*. Jenis *reinforcement learning* ini sangat mementingkan perencanaan, dan sangat membantu bila jumlah berada pada kondisi dengan data yang relatif tidak terlalu banyak jika dibandingkan *model-free reinforcement learning*.



Gambar 20 Jenis dan contoh penerapan *reinforcement learning*

Sebagai ilustrasi sederhana, Gambar 21 berikut menggambarkan bagaimana *model based* dan *model free reinforcement learning* bekerja.

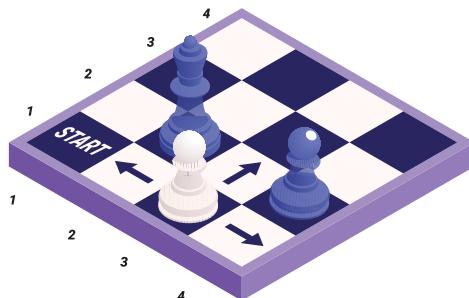


Gambar 21 Ilustrasi Model free vs. Model based Reinforcement Learning

Secara umum, terdapat dua masalah mendasar dalam pengambilan keputusan berurutan:

- 1. Pemelajaran:** Lingkungan awalnya tidak diketahui dan agen perlu berinteraksi dengan lingkungan untuk memperbaiki kebijakannya.
- 2. Perencanaan:** Jika lingkungan diketahui, agen melakukan perhitungan dengan modelnya dan kemudian memperbaiki kebijakannya.

Ilustrasi dari bagaimana *Reinforcement Learning* (Markov Decision Process) bekerja adalah sebagai berikut.



Gambar 22 Ilustrasi Markov Decision Process

Misalkan pada sebuah lingkungan dengan skema grid 3×4 . Sebuah pion memiliki *objective* untuk berpindah dari state START (1,1) menuju state FINISH (4,3). Dalam situasi tersebut, pion tidak dapat melewati grid (2,2), serta harus menghindari rintangan yang terdapat pada grid (4,2). Apabila menyentuh grid (4,2), maka pion akan mendapatkan *penalty* atau *punishment*. Pion atau *agent* dalam lingkungan tersebut hanya dapat melakukan satu *actions* untuk setiap state: **UP, DOWN, LEFT, RIGHT.**

Untuk menempuh rute terdekat dari START hingga END, *agent* dapat menempuh rute sebagai berikut.

- RIGHT RIGHT UP UP RIGHT
- UP UP RIGHT RIGHT RIGHT

Dalam lingkungan di atas, *stochastically*, 80% dari *actions* dapat berjalan menuju arah yang tepat. Akibatnya, terdapat 20% kemungkinan pion tersebut berjalan menuju arah yang salah, dalam kasus ini menuju LEFT atau RIGHT. Dalam hal ini *agent* menerima *rewards* untuk setiap *time step* sebagai berikut.

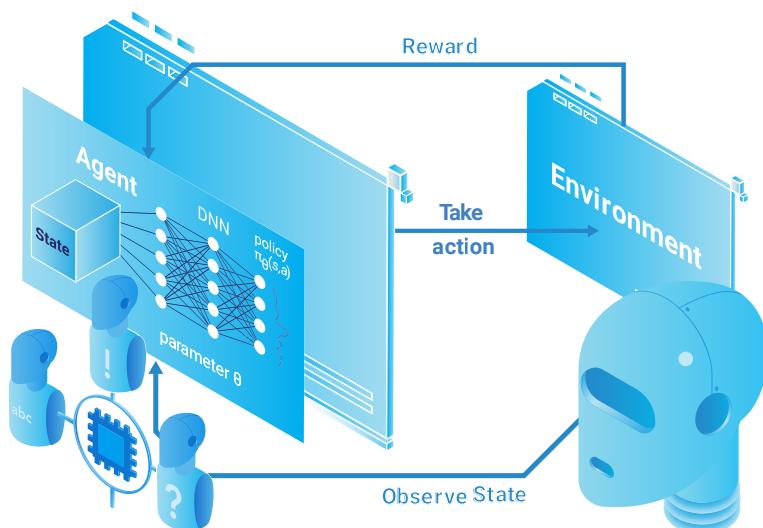
- *Small rewards* untuk setiap langkah yang tepat, dan *small punishment* untuk langkah yang salah.
- Jika mencapai *obstacle* seperti pada grid (4,2) akan mendapat *punishment* dengan skor yang besar.

- Skor *rewards* terbesar didapatkan pada grid FINISH (4,3).

Berdasarkan informasi mengenai lingkungan, objektif dari *agent*, serta *policy function* yang ada, model *Reinforcement Learning* yang memanfaatkan algoritma *Markov Decision Process* memiliki objektif utama untuk maksimisasi jumlah dari *rewards* yang didapatkan.

Box 3.1 Simple Reinforcement Learning

- *Deep Reinforcement Learning* memanfaatkan *deep learning* untuk mempelajari pola yang memungkinkan *agent* memaksimalkan *rewards*.
- Pemelajaran dilakukan untuk mendapatkan parameter (θ) agar dapat diperoleh *policy* terbaik (*action* (*a*) berdasarkan *state* (*s*)) $\pi\theta(s,a)$, yang dapat memaksimalkan *rewards*.



Gambar 23 Simple Reinforcement Learning

BAGIAN 4

MODEL EVALUATION



"There's not been enough focus on the real problem, which is building planetary-scale machine learning-based systems that actually work, deliver value to humans, and do not amplify inequities." – Michael I Jordan

4.1 Memilih Model: Performa vs. Interpretability

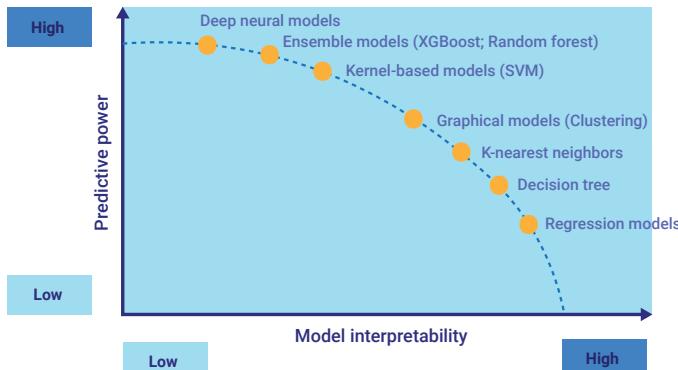
Sebelum menjawab pertanyaan pada judul bab di atas, penting untuk memahami definisi dari performa dan *interpretability* terlebih dahulu. Performa didefinisikan sebagai kemampuan model dalam menghasilkan prediksi. Adapun metrik yang digunakan sebagai satuan pengukuran adalah akurasi, presisi, *F1-score*, dan lain sebagainya. Selanjutnya, interpretabilitas merupakan kemampuan yang dimiliki model untuk menjelaskan bagaimana model mengambil keputusan. Contoh aspek *interpretability* dapat ditemukan pada koefisien regresi linear.

Selanjutnya, pemahaman konteks bisnis juga penting untuk menentukan model apa yang harus dipilih berdasarkan performa dan *interpretability*. Sebagai ilustrasi, performa lebih penting saat objektif (tujuan) model membutuhkan tingkat akurasi yang tinggi, misalnya pada model untuk deteksi kanker. Di lain sisi, terdapat situasi di mana akurasi tidak menjadi prioritas utama, melainkan pemahaman bagaimana sebuah keputusan dibuat oleh model. Hal ini dapat ditemukan pada studi kasus perumusan kebijakan perekonomian.

Oleh karena itu, tidak ada jawaban “one-size-fits-all”, di mana setiap permasalahan bisnis, memiliki spektrum yang berbeda pada kebutuhan akan aspek performa dan *interpretability*. Sebagai tambahan, setiap model pun memiliki spektrum performa dan *interpretability* yang berbeda. Tabel 3 dan Gambar 24 di bawah merepresentasikan karakteristik model, ditinjau dari dimensi performa dan *interpretability*.

Tabel 3 Perbandingan Model

Model	Performa	Interpretability	Contoh Penggunaan
Linear Regression	Moderate	High	Analisis dampak, prediksi harga
Decision Tree	Moderate	High	Diagnosis medis
Random Forest	High	Moderate	Prediksi cuaca
Gradient Boosting (e.g., XGBoost)	High	Moderate	Analisis keuangan
Support Vector Machine	High	Low	Klasifikasi teks
Neural Networks	Very High	Low	Pengenalan wajah, NLP



Gambar 24 Penjelasan mengenai performa vs. *interpretability*

Untuk merangkum penjelasan di atas, terdapat dua aspek penting untuk menentukan model apa yang akan digunakan, berdasarkan aspek performa dan *interpretability*, yaitu:

1. **Permasalahan bisnis** bisa jadi memiliki kebutuhan akan performa dan *interpretability* dengan spektrum yang berbeda.
2. **Karakteristik model** yang akan digunakan.

4.2 Machine Learning Evaluation

4.2.1 Tujuan dan Tahapan Umum

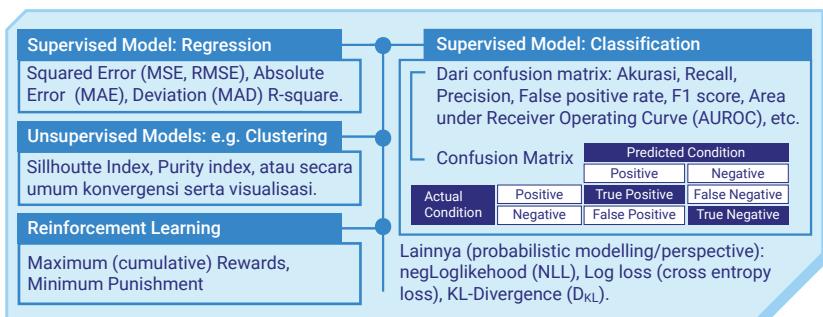
Evaluasi Machine Learning adalah proses penting yang menentukan seberapa baik suatu model dapat memprediksi atau mengklasifikasikan data yang belum pernah dilihat sebelumnya. Ini melibatkan penggunaan berbagai teknik dan metrik untuk menilai performa model, mengidentifikasi potensi pengembangan model, serta memastikan model tersebut sesuai untuk tugas tertentu. Berikut adalah penjelasan komprehensif mengenai evaluasi *machine learning*.

1. Tujuan evaluasi *machine learning*:
 - a. Mengukur performa model: Apakah model dapat memprediksi dengan akurasi tinggi pada data yang belum pernah dilihat sebelumnya?
 - b. Menguji generalisasi: Seberapa baik model dapat menghindari *overfitting* (memori berlebihan terhadap data latih) dan *underfitting* (kurang memahami pola dalam data)?
 - c. Memperbaiki model: Membantu menentukan apakah perlu dilakukan tuning model lebih lanjut, seperti mengubah hiperparameter atau arsitektur model.

2. Tahapan evaluasi:
 - a. *Training Set*: Dataset yang digunakan untuk melatih model.
 - b. *Validation Set*: Dataset terpisah yang digunakan untuk melakukan konfigurasi parameter model seperti regularisasi, *learning rate*, dll.
 - c. *Test Set*: Dataset yang benar-benar baru untuk mengevaluasi kinerja akhir model tanpa bias.

4.2.2 Metriks

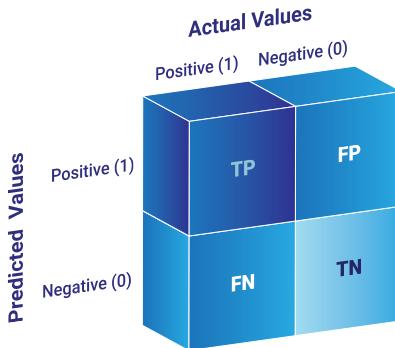
Pemilihan metrik yang tepat tergantung pada jenis masalah (klasifikasi, regresi, dll.). Secara umum, klasifikasi metriks evaluasi secara kuantitatif dapat dilihat pada Gambar 25 berikut.



Gambar 25 Rangkuman Metriks Evaluasi Model

A. Evaluasi untuk Klasifikasi

1. **Confusion Matrix**: Umumnya untuk memulai suatu evaluasi klasifikasi, diperlukan suatu confusion matriks. Pada bentuk yang sangat sederhana Gambar 28, matriks ini merupakan matriks 2x2 yang merangkum hasil prediksi *true positive*, *true negative*, *false positive*, dan *false negative*. Adapun penjelasan mengenai keempat metrik tersebut adalah sebagai berikut.
 - a. *True Positive*: Model memprediksi hasil positif dengan benar.
 - b. *True Negative*: Model memprediksi hasil negatif dengan benar.
 - c. *False Positive*: Model salah dalam memprediksi hasil positif.
 - d. *False Negative*: Model salah dalam memprediksi hasil negatif.



Gambar 26 Confusion Matrix

- Akurasi:** Persentase prediksi yang benar dari keseluruhan prediksi.

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

- Precision:** Rasio prediksi positif yang benar terhadap total prediksi positif.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4.2)$$

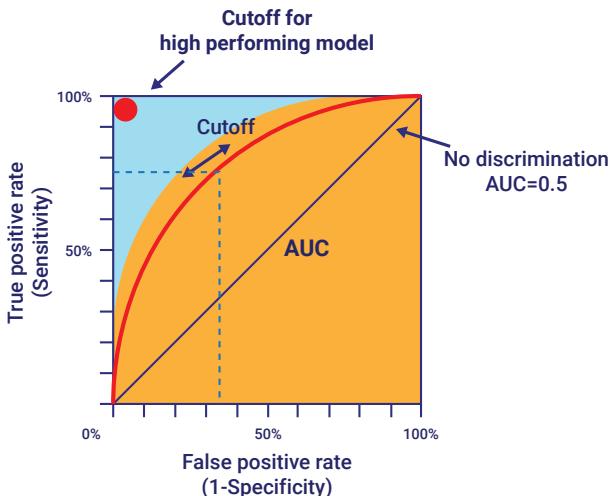
- Recall (Sensitivitas):** Rasio deteksi yang benar untuk semua kelas positif.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4.3)$$

- F1-Score:** Harmonik rata-rata *precision* dan *recall*, cocok untuk dataset yang tidak seimbang.

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4.4)$$

- ROC-AUC:** Plot antara *True Positive Rate (Recall)* dan *False Positive Rate* untuk berbagai *threshold*, di mana AUC (*Area Under the Curve*) mengukur performa model secara keseluruhan (ilustrasi terlihat pada Gambar 27).



Gambar 27 Ilustrasi AUCROC

Dalam hal ini, terdapat 2 komponen penyusun AUC ROC, yaitu:

1. **True Positive Rate (TPR)** atau **Recall**, yaitu rasio prediksi benar untuk kelas positif dibandingkan total data positif. $TPR = TP / (TP + FN)$.
2. **False Positive Rate (FPR)**, yaitu rasio prediksi salah untuk kelas positif dibandingkan total data negatif. $FPR = FP / (FP + TN)$.

AUC mengukur luas area di bawah kurva ROC. Nilai **AUC** berkisar antara 0 hingga 1, dengan nilai yang lebih besar menunjukkan performa model yang lebih baik dalam memisahkan kelas positif dan negatif.

- **AUC = 1** berarti model mampu memisahkan kelas positif dan negatif secara sempurna.
- **AUC = 0.5** menunjukkan model tidak lebih baik daripada tebakan acak.
- **AUC < 0.5** berarti model bekerja lebih buruk dari tebakan acak.

B. Evaluasi untuk Regresi

1. **Mean Absolute Error (MAE)**: Nilai rata-rata dari eror absolut antara hasil prediksi dan nilai sebenarnya.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (4.5)$$

2. **Mean Squared Error (MSE)**: Nilai rata-rata kuadrat dari selisih antara hasil prediksi dan nilai sebenarnya.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (4.6)$$

3. **Root Mean Squared Error (RMSE):** Akar dari MSE, memberikan kesalahan dalam unit yang sama dengan target.
4. **R-squared (R^2):** dikenal sebagai **koefisien determinasi**, adalah metrik statistik yang digunakan untuk mengevaluasi seberapa baik model regresi memprediksi hasil berdasarkan variabel bebas (*independent variables*). R^2 mengukur proporsi variasi dalam variabel dependen (*dependent variable*) yang dapat dijelaskan oleh variabel independen dalam model regresi

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (4.7)$$

dalam hal ini

- SS_{res} adalah **sum of squared residuals** (jumlah kuadrat sisa atau kesalahan prediksi).

$$\sum_i (y_i - f_i)^2 = \sum_i e_i^2 \quad (4.8)$$

- SS_{tot} adalah **total sum of squares**, yaitu variasi total dari data asli.

$$\sum_i (y_i - \bar{y})^2 \quad (4.9)$$

Beberapa interpretasi terkait *R-squared*, yaitu:

- **$R^2 = 0$:** Model tidak menjelaskan sama sekali variasi dalam data. Dengan kata lain, prediksi model sama buruknya dengan rata-rata variabel dependen.
- **$R^2 = 1$:** Model sempurna dalam menjelaskan semua variasi dalam data. Ini berarti prediksi model sesuai dengan nilai observasi.
- **$0 < R^2 < 1$:** Model menjelaskan sebagian variasi, di mana nilai yang lebih tinggi menunjukkan bahwa model lebih baik dalam menjelaskan variasi data.

C. Evaluasi untuk *Unsupervised Learning*

Pada umumnya, matriks evaluasi untuk *unsupervised learning* cenderung bergantung pada tipe *task* dan tujuan pada *unsupervised learning* itu sendiri. Berikut ini beberapa evaluasi umum untuk *unsupervised learning*, khususnya *clustering* dan *dimensionality reduction*.

1. Internal Evaluation Metrics

Metode ini mengukur kinerja model berdasarkan data itu sendiri tanpa menggunakan label atau referensi eksternal.

- Silhouette Score:** Mengukur seberapa baik titik-titik data dikelompokkan.

Ini berdasarkan jarak rata-rata antara titik dalam *cluster* dan *cluster* terdekat lainnya. Nilai berkisar dari -1 (*clustering* buruk) hingga 1 (*clustering* baik).

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (4.10)$$

Di mana $a(i)$ adalah jarak rata-rata ke anggota lain dalam *cluster* yang sama, dan $b(i)$ adalah jarak rata-rata ke anggota *cluster* terdekat lainnya.

- Davies-Bouldin Index:** Mengukur kesamaan rata-rata antara *cluster*. Nilai yang lebih rendah menunjukkan *clustering* yang lebih baik, dengan nilai minimal mendekati 0.
- Inertia (Within-Cluster Sum of Squares):** Digunakan dalam metode seperti *K-Means* untuk mengevaluasi seberapa dekat titik data dalam *cluster*. Nilai lebih rendah menunjukkan *clustering* lebih rapat.

2. External Evaluation Metrics

Jika ada *ground truth* atau label dalam data, meskipun jarang dalam *unsupervised learning*, matriks eksternal bisa digunakan:

- Adjusted Rand Index (ARI):** Matriks ini menghitung kesamaan antara *clustering* hasil dan *ground truth*, memperhitungkan kemungkinan *clustering* acak. Nilai berkisar dari 0 hingga 1, dengan nilai lebih tinggi menunjukkan *clustering* yang lebih akurat.
- Mutual Information (MI):** Mengukur seberapa banyak informasi yang dibagikan antara hasil *clustering* dan *ground truth*.
- Fowlkes-Mallows Index:** Mengukur *precision* dan *recall* pasangan data untuk mengevaluasi *clustering* terhadap label *ground truth*.

3. Relative Evaluation Metrics

Matriks ini digunakan untuk membandingkan performa beberapa model atau konfigurasi.

- Elbow Method:** Menilai jumlah optimal *cluster* berdasarkan perubahan inertia atau *within-cluster sum of squares*. *Elbow point* adalah jumlah *cluster* optimal di mana penurunan *inertia* mulai berkurang.
- Gap Statistic:** Mengukur perbedaan antara *within-cluster sum of squares* dari model dan model acak. Nilai gap yang lebih besar menunjukkan *clustering* yang lebih baik.

4. Visual Evaluation

Visualisasi adalah alat penting dalam mengevaluasi hasil dari *unsupervised learning*, terutama untuk data yang bisa diproyeksikan ke ruang dimensi rendah (2D atau 3D).

- a. **Scatter Plots:** Digunakan untuk memvisualisasikan hasil *clustering* jika dimensi data bisa dikurangi (misalnya dengan PCA atau t-SNE).
- b. **t-SNE dan PCA:** Teknik reduksi dimensi ini membantu memvisualisasikan data dalam 2D atau 3D, membantu melihat distribusi *cluster* secara visual.

D. Evaluasi untuk Reinforcement Learning

Evaluasi untuk **Reinforcement Learning (RL)** berbeda dari *supervised* dan *unsupervised learning* karena melibatkan agen yang belajar melalui interaksi dengan lingkungan. Agen RL berusaha untuk memaksimalkan *reward* kumulatif dengan membuat serangkaian keputusan. Evaluasi RL berfokus pada bagaimana kinerja agen dalam mencapai tujuan setelah belajar. Berikut adalah beberapa metode utama evaluasi dalam *reinforcement learning*:

1. Reward Kumulatif

Reward kumulatif adalah matriks evaluasi paling dasar dalam RL. Ini mengukur *total reward* yang diperoleh agen selama episode atau periode waktu tertentu. Agen RL mencoba untuk memaksimalkan *reward* ini melalui kebijakan yang dipelajari.

- a. **Return:** Jumlah total *reward* yang diperoleh agen dalam satu episode.

$$G_t = R_{t+1} + R_{t+2} + \dots + R_T \quad (4.11)$$

Di mana G_t adalah *return* pada waktu t , dan R_t adalah *reward* pada waktu t .

- b. **Discounted Return:** Banyak algoritma RL menggunakan *return* yang didiskontokan untuk menghitung nilai sekarang dari *reward* masa depan.

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \quad (4.12)$$

Di mana $0 < \gamma < 1$ adalah faktor diskon yang menentukan seberapa jauh *reward* masa depan diperhitungkan.

2. Convergence

Mengukur seberapa cepat agen RL mencapai kebijakan optimal dan stabilitas selama pelatihan adalah matriks evaluasi penting. Konvergensi menunjukkan bahwa agen telah berhenti memperbarui kebijakan secara signifikan dan telah menemukan solusi optimal.

3. Learning Efficiency

Efisiensi pembelajaran mengukur seberapa cepat agen belajar dari lingkungan untuk mencapai *reward* yang lebih tinggi. Hal ini bisa dievaluasi melalui **learning curves** yang menunjukkan bagaimana *reward* kumulatif meningkat seiring waktu.

- a. **Sample Efficiency:** Jumlah sampel interaksi yang dibutuhkan agen untuk mencapai performa tertentu. Metode *RL model-free* umumnya memiliki *sample efficiency* yang lebih rendah dibandingkan dengan *model-based RL*.

- b. **Computational Efficiency:** Evaluasi juga mempertimbangkan berapa banyak waktu komputasi yang dibutuhkan agen untuk mencapai kebijakan optimal.

4. **Stability vs. Exploration**

Pada *reinforcement learning*, agen harus menyeimbangkan **eksplorasi** (mencoba tindakan baru untuk menemukan *reward* yang lebih baik) dan **eksloitasi** (menggunakan pengetahuan yang ada untuk memaksimalkan *reward*). Evaluasi stabilitas agen RL berfokus pada bagaimana agen menyeimbangkan keduanya.

Exploration Efficiency: Seberapa baik agen menjelajahi tindakan baru dan seberapa efisien dalam menemukan kebijakan optimal tanpa terjebak di solusi suboptimal.

5. **Policy Robustness**

Kebijakan yang dipelajari agen harus **robust** terhadap variasi dalam lingkungan. Evaluasi ini mengukur seberapa baik agen bisa mempertahankan kinerja ketika menghadapi kondisi baru atau tidak terduga di luar data pelatihan.

Generalization: Kemampuan agen untuk mempertahankan performa yang baik di lingkungan baru atau dengan gangguan yang tidak ada selama pelatihan.

6. **Safety and Constraints**

Dalam beberapa aplikasi, agen *reinforcement learning* harus mematuhi batasan tertentu (misalnya, keselamatan dalam sistem otonom seperti mobil *self-driving*). Evaluasi matriks ini melihat seberapa baik agen mematuhi batasan-batasan tersebut selama eksplorasi dan eksloitasi.

Safe RL: Memastikan bahwa agen mematuhi batasan keselamatan saat mengeksplorasi dan mempelajari kebijakan.

7. **Average Reward vs. Total Reward**

Terdapat dua pendekatan dalam mengukur *reward*:

- Average Reward:** Menilai rata-rata *reward* per tindakan atau per episode selama pelatihan. Berguna untuk mengevaluasi kinerja dalam jangka panjang.
- Total Reward:** Total *reward* yang diperoleh selama pelatihan atau episode. Cocok untuk aplikasi di mana fokus adalah pada optimasi *reward* kumulatif.

8. **Comparison to a Baseline**

Evaluasi sering dilakukan dengan membandingkan performa agen RL dengan **baseline** atau agen lain yang sudah memiliki kebijakan optimal, seperti:

- Random Policy:** Membandingkan kebijakan agen terhadap kebijakan acak.
- Optimal Policy:** Membandingkan kinerja agen terhadap agen yang menggunakan kebijakan optimal yang diketahui (jika tersedia).

E. Metriks Evaluasi yang Lebih General

Berikut ini 2 (dua) metriks yang lebih umum dengan melihat secara perspektif probability, yaitu *Kullback-Leibler (KL) Divergence* dan *Kolmogorov-Smirnov Test*.

1. **Kullback-Leibler (KL) Divergence** adalah ukuran ketidaksesuaian atau perbedaan antara dua distribusi probabilitas. *KL Divergence* menggambarkan seberapa banyak satu distribusi probabilitas P menyimpang dari distribusi referensi Q .

Secara matematis, *KL Divergence* dapat dirumuskan sebagai berikut untuk kasus diskret:

$$D_{KL}(P \parallel Q) = \sum_i P(x_i) \log \frac{P(x_i)}{Q(x_i)} \quad (4.13)$$

Sementara itu, untuk distribusi kontinu adalah sebagai berikut.

$$D_{KL}(P \parallel Q) = \int P(x) \log \frac{P(x)}{Q(x)} dx \quad (4.14)$$

KL Divergence sering dianggap sebagai ukuran “ketidakseragaman” informasi. Dengan kata lain, ini mengukur seberapa banyak informasi yang hilang saat menggunakan distribusi Q sebagai pendekatan untuk distribusi sebenarnya P . *KL Divergence* **tidak simetris**, artinya $D_{KL}(P||Q)$ biasanya tidak sama dengan $D_{KL}(Q||P)$.

- **Jika $D_{KL}(P||Q)=0$** , ini berarti bahwa distribusi P dan Q identik.
- **Nilai yang lebih besar dari nol** menunjukkan semakin banyak ketidaksesuaian antara distribusi P dan Q .

Contoh Penggunaan

KL Divergence banyak digunakan dalam berbagai bidang, seperti:

- a. **Teori Informasi:** Untuk mengukur perbedaan antara dua distribusi probabilitas, misalnya dalam pengodean informasi.
- b. **Machine Learning:** Digunakan dalam optimasi model probabilistik, seperti dalam *Variational Inference*.
- c. **Natural Language Processing (NLP):** Untuk mengevaluasi ketidaksesuaian antara distribusi kata dalam teks.
- d. **Pengolahan Gambar:** Membandingkan distribusi histogram warna atau tekstur dari dua gambar.

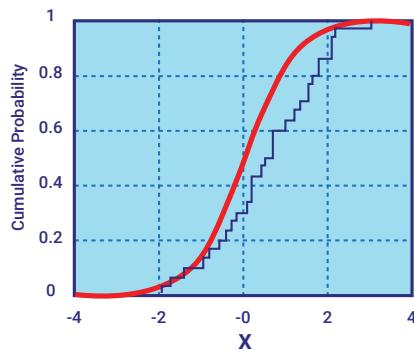
2. Kolmogorov Smirnov (K-S) Test

Kolmogorov-Smirnov (K-S) Test adalah uji statistik non-parametrik yang digunakan untuk membandingkan distribusi probabilitas antara dua sampel atau untuk membandingkan distribusi sampel dengan distribusi referensi (teoretis). Tujuannya adalah untuk menentukan apakah dua distribusi tersebut berbeda secara signifikan atau apakah data sampel berasal dari distribusi tertentu.

Jenis Kolmogorov-Smirnov Test

- One-sample K-S Test:** Menguji apakah sampel berasal dari distribusi teoretis tertentu (seperti distribusi normal, eksponensial, dll.).
- Two-sample K-S Test:** Menguji apakah dua sampel independen berasal dari distribusi yang sama.

K-S Test membandingkan fungsi distribusi kumulatif (CDF) dari dua distribusi dan mengukur jarak maksimum antara keduanya. Uji ini menghasilkan **statistik D**, yang merupakan nilai maksimum perbedaan antara dua CDF. Nilai statistik ini kemudian dibandingkan dengan nilai kritis untuk menentukan apakah hipotesis nol (bahwa kedua distribusi adalah sama) dapat diterima atau ditolak. Secara ilustrasi, KS test ini dapat dilihat pada Gambar 28 berikut.



Gambar 28 Ilustrasi Komogorov-Smirnov (K-S) test

Statistik K-S dapat diformulasikan sebagai berikut.

$$D = \sup_x |F_1(x) - F_2(x)| \quad (4.15)$$

Dalam hal ini:

- $F_1(x)$ dan $F_2(x)$ adalah fungsi distribusi kumulatif dari dua sampel yang dibandingkan.
- $\sup_x (x)$ adalah supremum (nilai maksimum) dari perbedaan absolut antara dua CDF.

Beberapa interpretasi terkait *K-S Test*, yaitu:

- **D kecil:** Jika nilai D kecil, maka distribusi dua sampel sangat mirip dan hipotesis nol tidak bisa ditolak.
- **D besar:** Jika nilai D besar, maka distribusi dua sampel berbeda signifikan dan hipotesis nol ditolak.

K-S Test digunakan dalam berbagai bidang seperti:

- **Keuangan:** Menguji apakah distribusi harga saham mengikuti distribusi tertentu.
- **Statistika:** Untuk mengevaluasi kesesuaian model dengan data observasi.
- **Data science:** Membandingkan distribusi sampel untuk analisis eksploratif data.

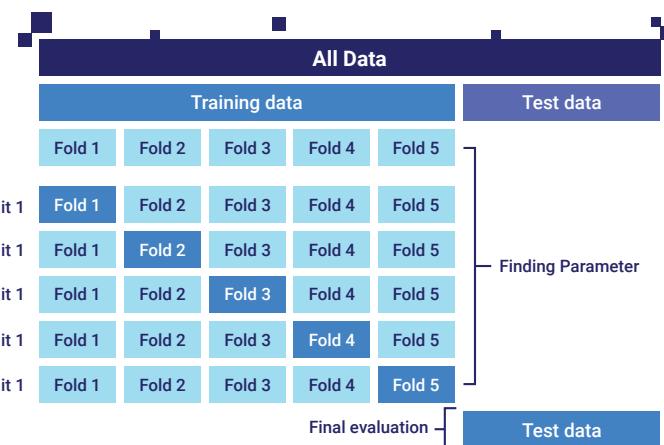
4.2.3 Hal-Hal Terkait Evaluasi

A. Cross-Validation

Cross-validation adalah teknik untuk menghindari *overfitting* dan menilai generalisasi model. Teknik yang umum adalah ***k-fold cross-validation***:

- Dataset dibagi menjadi **k** bagian yang setara.
- Model dilatih pada **k-1** bagian dan diuji pada satu bagian.
- Proses ini diulang **k** kali, dan hasil rata-rata dihitung.

Ilustrasi mengenai *cross validation* dapat disimak pada Gambar 29 berikut.



Gambar 29 Ilustrasi *Cross k-fold Validation*, dengan $k=5$ (sumber: scikit learn)

Cross-validation memberikan gambaran performa yang lebih *robust*, terutama pada dataset kecil.

B. Bias-Variance Tradeoff

- **Bias:** Kesalahan yang disebabkan oleh model yang terlalu sederhana, sehingga gagal menangkap pola penting dalam data (*underfitting*).
- **Variance:** Kesalahan yang disebabkan oleh model yang terlalu kompleks, sehingga *overfitting* terhadap data latih dan tidak generalisasi dengan baik pada data baru.

Tradeoff bias-variance adalah keseimbangan antara bias rendah dan variansi rendah. Model yang ideal mampu menangkap pola data tanpa *overfitting*.

C. Overfitting dan Underfitting

- **Overfitting:** Ketika model terlalu cocok dengan data latih dan performa buruk pada data baru.
- **Underfitting:** Ketika model gagal menangkap pola dalam data latih, sehingga hasil prediksinya buruk baik pada data latih maupun data uji.

Salah satu cara untuk mengatasi *overfitting* adalah dengan teknik **regularisasi** seperti L1 dan L2 *regularization*, atau dengan menggunakan **dropout** pada neural networks.

D. Regularisasi dan Tuning Hyperparameter

- **Regularisasi:** Metode untuk mencegah *overfitting* dengan menambahkan penalti pada ukuran kompleksitas model, seperti **L1 regularization** (Lasso) atau **L2 regularization** (Ridge).
- **Hyperparameter Tuning:** Menyetel parameter eksternal seperti *learning rate*, jumlah layer, atau parameter pohon keputusan melalui metode seperti **grid search** atau **random search**.

E. Handling Imbalanced Data

- **Resampling:** Melakukan **oversampling** pada kelas minoritas atau **undersampling** pada kelas mayoritas.
- **Class Weighting:** Menyesuaikan bobot kelas untuk memberikan penalti lebih besar pada kesalahan prediksi di kelas minoritas.
- **Synthetic Minority Over-sampling Technique (SMOTE):** Membuat data sintetis untuk kelas minoritas

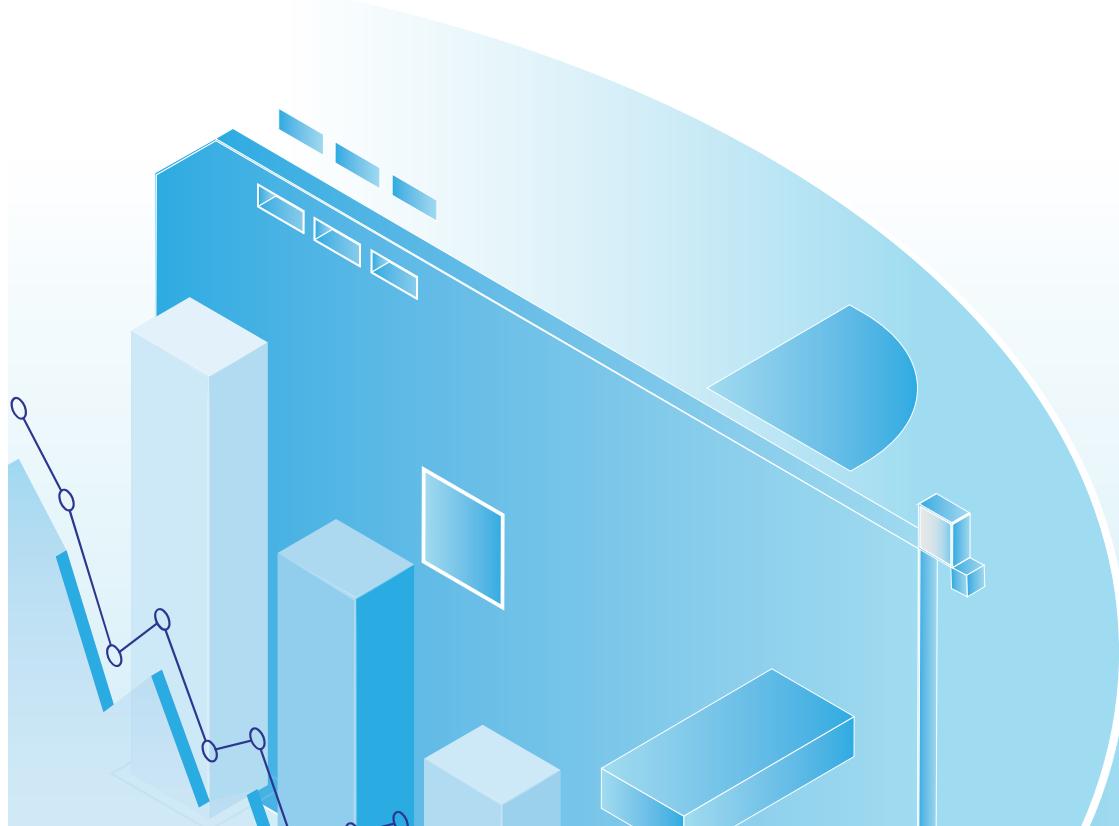
F. Explainability and Interpretability

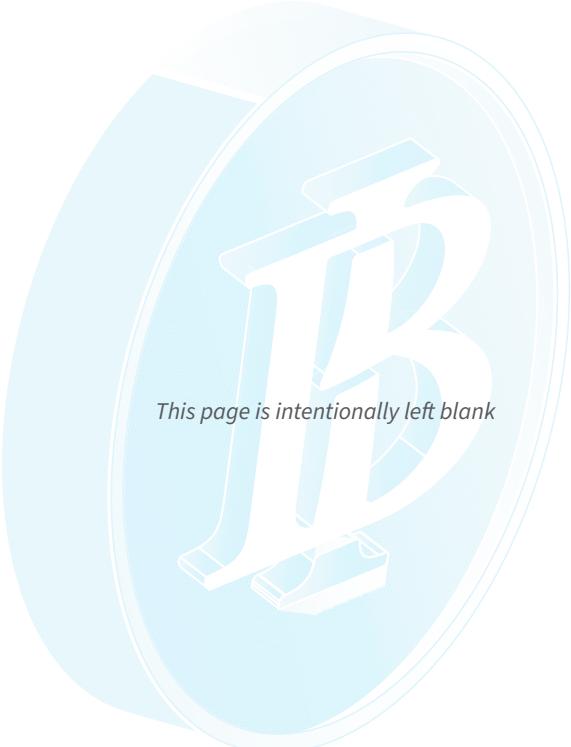
Dalam konteks aplikasi *machine learning* yang penting, seperti kesehatan dan keuangan, interpretabilitas model sangat krusial. Metode seperti **SHAP** dan **LIME** membantu memahami bagaimana fitur-fitur tertentu memengaruhi prediksi model, yang membuat model lebih transparan dan dapat dipahami oleh pengguna.

G. Evaluasi Berkelanjutan dan Ethical AI

- **Audit Berkala:** Model harus dievaluasi secara berkala untuk memastikan bahwa model masih sesuai dengan lingkungan atau data yang berubah.
- **Fairness and Bias:** Penting untuk mengevaluasi apakah model bias terhadap kelompok tertentu. Hal ini dapat dilakukan dengan memonitor distribusi prediksi untuk berbagai subkelompok populasi.

Evaluasi *machine learning* merupakan proses yang komprehensif dan berlapis-lapis, mulai dari pemilihan metrik yang tepat, penerapan *cross-validation*, hingga interpretasi hasil model. Proses evaluasi juga harus memperhatikan aspek etika, fairness, dan generalisasi model agar hasilnya dapat diandalkan serta aman untuk diterapkan di berbagai konteks.





This page is intentionally left blank

BAGIAN 5

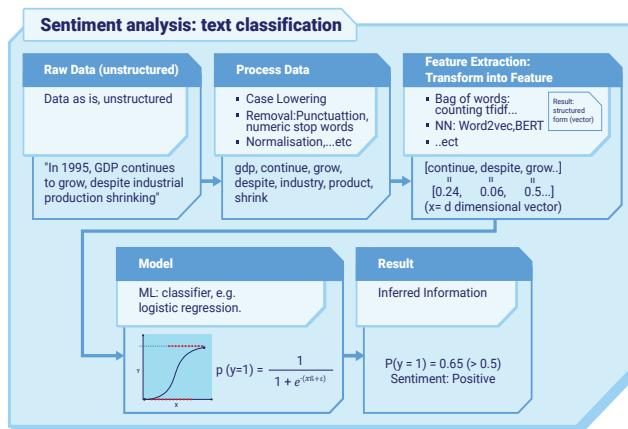
POPULAR APPLIED MACHINE LEARNING



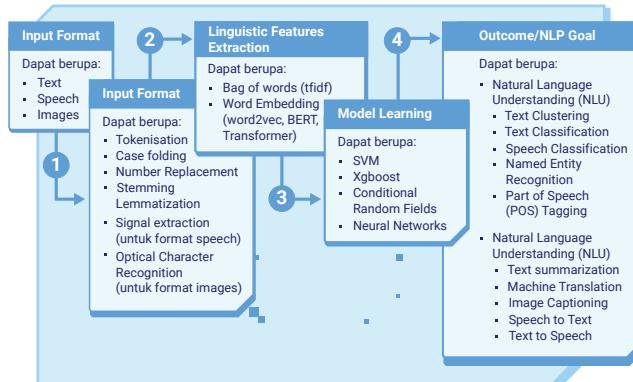
5.1 Natural Language Processing (NLP)

Natural Language Processing (NLP) adalah cabang dari kecerdasan buatan (AI) yang berkaitan dengan interaksi antara komputer dan bahasa manusia (*natural language*). NLP memungkinkan komputer untuk memahami, menafsirkan, dan menghasilkan bahasa manusia dengan cara yang bermanfaat. NLP melibatkan berbagai teknik dari linguistik komputasi, pembelajaran mesin, dan analisis teks.

Gambar 30 mengilustrasikan contoh implementasi sederhana dari NLP, yaitu *text classification* berupa *sentiment analysis*. Secara umum proses NLP dapat digeneralisasi ke dalam tahapan-tahapan utama, yaitu meliputi prapemprosesan data, ekstraksi fitur, pemodelan, dan evaluasi.



Gambar 30 Contoh sederhana NLP untuk *Sentiment Analysis* (klasifikasi teks)



Gambar 31 Gambaran umum tahapan dalam popular NLP

Secara umum proses dalam NLP dengan AI memiliki ciri, yaitu adanya proses *linguistic feature extraction* dan proses pembangunan representasi pengetahuan dengan AI, misalnya saat ini yang popular, yaitu *machine learning*. Hal ini dapat digambarkan pada Gambar 31.

Aspek umum analisis dengan NLP

- **Sintaksis:** Analisis struktur kalimat, tata bahasa, dan aturan sintaksis. Teknik seperti *part-of-speech tagging* dan *parsing* digunakan untuk mengenali hubungan antara kata-kata.
- **Semantik:** Fokus pada pemahaman makna dari kalimat dan kata-kata. Semantik mencakup penafsiran konteks dan arti di balik teks.

Teknik Unik yang terdapat dalam NLP

1. Preprocessing

- **Tokenisasi:** Proses memecah teks menjadi unit-unit kecil seperti kata atau frasa.
- **Stemming maupun Lemmatization:** Stemming mengubah kata ke bentuk dasarnya, misalnya mengubah “running” menjadi “run” atau “berlari” menjadi “lari”, atau lemmatisation mengubah kata “drunk” menjadi “drink”

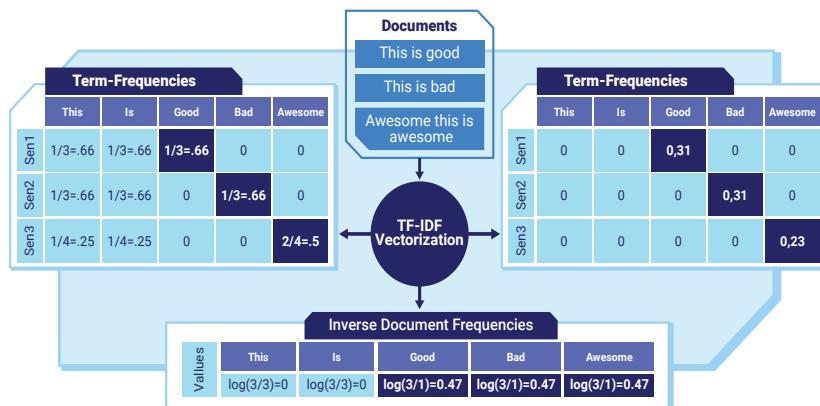
2. Linguistic Feature Extraction

Ekstraksi fitur linguistik dapat berupa token itu sendiri (misalnya dalam direpresentasikan terdistribusi multinomial (diskret) dalam kasus *named entity recognition* dengan *conditional random field*), atau berupa vektor yang merepresentasikan token tersebut. Contoh ekstraksi fitur token ke dalam vektor misalnya tfidf dan neural network based word embedding:

- **Feature Extraction via tfidf.** Teknik yang cukup klasik (namun efektif dalam beberapa kasus) ini bersifat frekuatif dan mengandalkan *bag of words* (*bow*). Sederhananya, suatu kata akan dilihat berdasarkan frekuensi dalam suatu dokumen dan dibobot secara berbanding terbalik dengan popularitas kata tersebut pada keseluruhan dokumen dalam koleksi. Secara matematis, pembobotan ini dapat direpresentasikan sebagai:

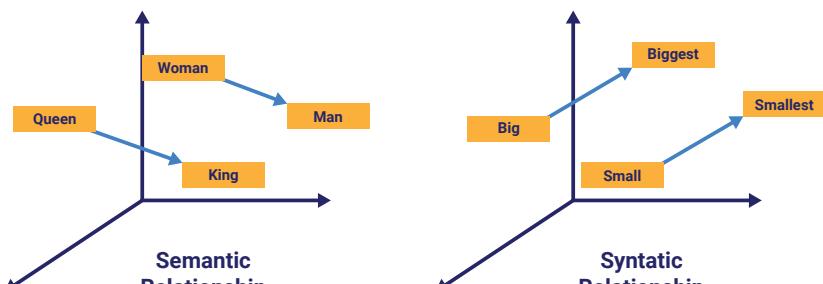
$$tfidf_{w,d} = tf_{w,d} \times \log \frac{N}{df_w} \quad (5.1)$$

Dengan $tf_{w,d}$ adalah frekuensi kata w dalam dokumen d , lalu df_w jumlah dokumen dalam koleksi yang mengandung kata w , serta N adalah jumlah keseluruhan dokumen dalam koleksi (korpus). Ilustrasi proses ekstraksi menggunakan pembobotan tfidf dapat dilihat pada Gambar 32 berikut.



Gambar 32 Ilustrasi ekstraksi fitur dengan pembobotan tfidif

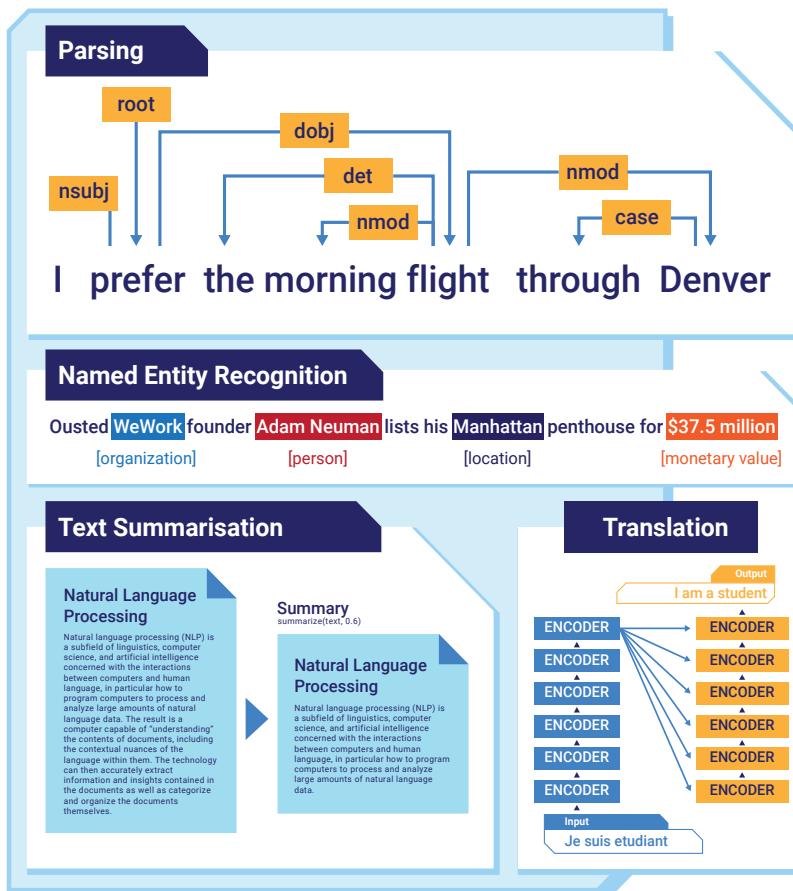
- **Feature extraction via neural network embedding:** Teknik seperti Word2Vec atau GloVe yang mengubah kata menjadi vektor angka, memungkinkan komputer untuk menangkap makna semantik dan hubungan antarkata. Gambar 33 berikut merepresentasikan visualisasi *word embedding* dalam koordinat 3 dimensi.



Gambar 33 Visualisasi Word Embedding

Aplikasi NLP

Beberapa contoh popular NLP di antaranya *Named Entity Recognition*, *Parsing*, *Summarisation*, maupun *translation* sebagaimana dapat dilihat pada Gambar 34 berikut. Tentunya masih terdapat berbagai macam NLP bergantung pada tujuan.



Gambar 34 Contoh beberapa popular NLP task

- **Document Classification:** NLP dapat dimanfaatkan untuk mengategorikan dokumen dengan menganalisis konten melalui transformasi *word* ke *vector* lalu diklasifikasikan menggunakan *machine learning classifier*.
- **Pencarian Informasi (Search Engines):** NLP digunakan oleh mesin pencari seperti Google untuk memahami kueri pengguna dan memberikan hasil yang relevan.
- **Chatbot dan Virtual Assistant:** Siri, Alexa, dan *chatbot* lainnya menggunakan NLP untuk memahami permintaan pengguna dan memberikan tanggapan yang sesuai.
- **Machine Translation:** Aplikasi seperti Google Translate menggunakan NLP untuk menerjemahkan teks antara berbagai bahasa.

- **Analisis Sentimen:** Digunakan untuk menilai sentimen dari teks, misalnya apakah ulasan produk positif atau negatif.
- **Summarization:** Merangkum teks panjang menjadi versi yang lebih singkat, namun tetap mempertahankan makna utamanya.
- **Named Entity Recognition (NER):** Mengenali nama entitas seperti orang, lokasi, organisasi, tanggal, dll..
- **Text Generation:** Komputer menghasilkan teks baru berdasarkan *input* yang diberikan (Brown et al, 2020). Misalnya, digunakan dalam aplikasi *chatbot* atau *machine translation*.
- **Speech-to-Text dan Text-to-Speech:** Konversi suara menjadi teks atau sebaliknya, yang memungkinkan aplikasi seperti asisten virtual dan perangkat pengenalan suara.

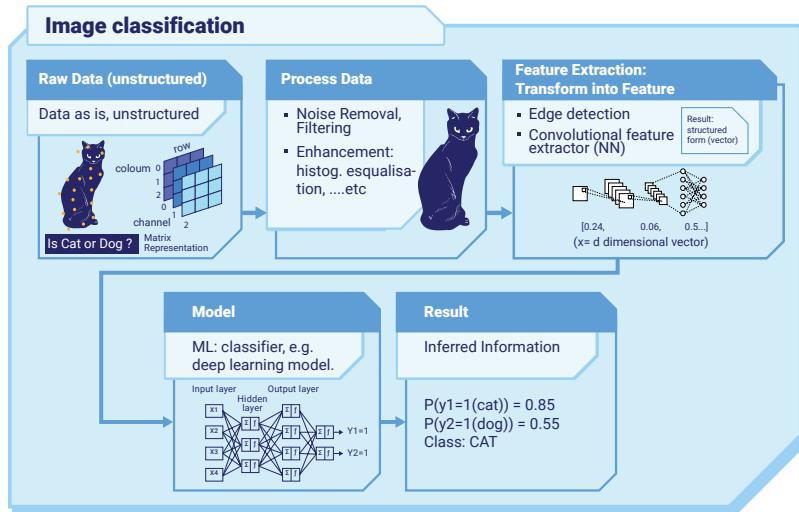
Tantangan dalam NLP

- **Ambiguitas:** Bahasa manusia sering kali ambigu, yang berarti satu kata atau kalimat bisa memiliki banyak arti tergantung konteksnya.
- **Polisemi dan Homonimi:** Satu kata dapat memiliki beberapa arti (polisemi), atau dua kata yang terdengar sama tetapi memiliki arti yang berbeda (homonimi).
- **Konteks:** NLP membutuhkan kemampuan untuk memahami konteks, karena arti dari sebuah kata atau frasa bisa berbeda tergantung pada kalimat sekitarnya.

NLP telah berkembang pesat berkat kemajuan dalam pembelajaran mesin dan ketersediaan data besar, yang membuatnya semakin efektif dalam menangani tugas-tugas berbasis bahasa alami.

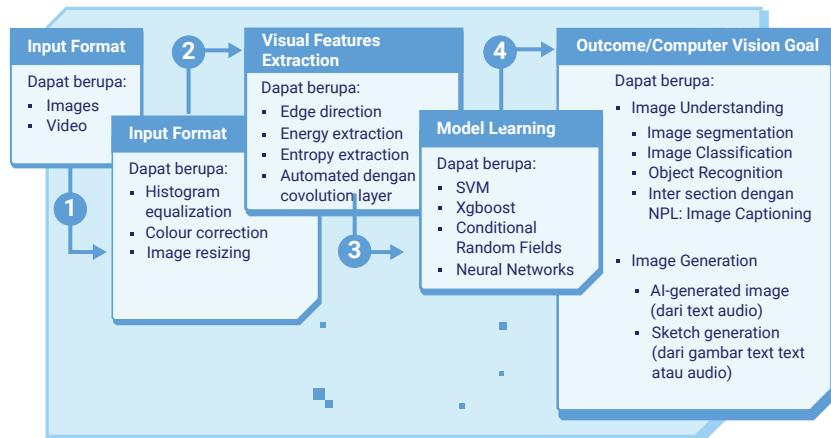
5.2 Computer Vision & Image Recognition

Computer Vision (CV) adalah cabang dalam kecerdasan buatan (AI) yang berfokus pada pengenalan, pemrosesan, dan pemahaman gambar serta video digital oleh komputer. Tujuan utama dari *computer vision* adalah memungkinkan komputer untuk meniru penglihatan manusia dengan menafsirkan *visual input* dan membuat keputusan berdasarkan data visual tersebut (Szeliski, 2021). Sebagai contoh, Gambar 35 menggambarkan bagaimana suatu gambar diolah dalam suatu proses *image classification*.



Gambar 35 Contoh tahapan pada *image classification* sederhana

Computer vision secara umum mencakup beberapa tahapan sebagaimana pada Gambar 36. Sebagaimana NLP, dalam computer vision terdapat tahapan yang mencirikan AI *computer vision*, yaitu adanya *visual feature extraction* dan proses pembangunan representasi pengetahuan.

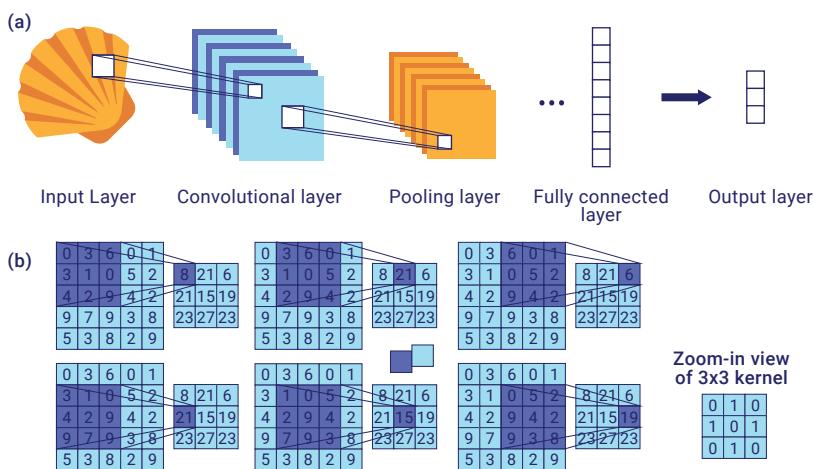


Gambar 36 Tahapan *Computer Vision* Secara Umum

Teknik Unik yang terdapat dalam Computer Vision

1. Preprocessing

- **Histogram Equalisation (HE)** adalah teknik pemrosesan citra yang bertujuan untuk meningkatkan kontras gambar dengan meratakan distribusi nilai intensitas piksel. Teknik ini sangat berguna dalam meningkatkan detail gambar yang memiliki kontras rendah atau pencahayaan yang kurang merata.
- **Image Resizing** adalah proses mengubah ukuran suatu gambar, baik dengan memperbesar (*upsampling*) maupun memperkecil (*downscaling*), tanpa mengubah konten utama gambar. Teknik ini umumnya dilakukan agar gambar memiliki ukuran yang lebih kecil sehingga pemrosesan menjadi lebih ringan.



Gambar 37 Convolutional Neural Network untuk ekstraksi fitur spasial suatu gambar

2. Feature extraction

Beberapa fitur gambar diekstraksi baik secara tradisional (misalnya *edge detection*, *entropy extraction*, *energy extraction*), maupun otomatis melalui mekanisme *convolutional neural network*. Contoh umum metode yang klasik misalnya:

- **Edge Detection:** Algoritma ini digunakan untuk menemukan tepi dalam gambar, yang sangat penting untuk mengidentifikasi bentuk objek. Metode umum termasuk algoritma Sobel dan Canny.
- **Entropy Extraction:** Proses pengukuran ketidakteraturan atau kompleksitas dalam suatu sinyal atau gambar. Semakin tidak teratur suatu area maka entropy akan semakin tinggi, begitu pula sebaliknya.

- **Extraction dengan Convolutional Layer:** Merupakan proses mengekstraksi abstraksi fitur spasial gambar menggunakan *neural network*. Gambar 37 mengilustrasikan bagaimana suatu *convolutional neural network* bekerja untuk sebuah gambar.

Aplikasi Computer Vision

Sebagaimana NLP, penggunaan *computer vision* sebagai bagian dari AI juga semakin masif. Gambar 38 mengilustrasikan beberapa contoh lain pemanfaatan *computer vision*.

1. Pengenalan Gambar (*Image Recognition*)

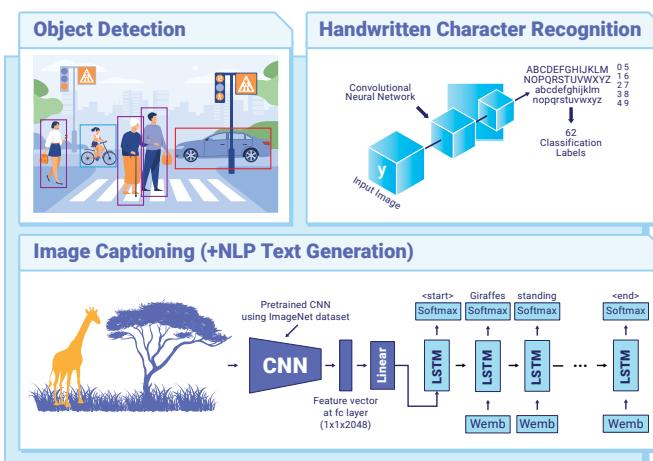
Teknik ini melibatkan identifikasi objek, fitur, atau pola dalam gambar. Contohnya, mengenali wajah manusia, nomor plat kendaraan, atau jenis hewan dalam gambar.

2. Segmentasi Gambar (*Image Segmentation*)

Teknik ini memecah gambar menjadi beberapa segmen untuk menganalisis objek secara lebih detail. Segmentasi bisa dilakukan pada level piksel untuk mengenali tepi atau batas dari objek di dalam gambar. Lebih lanjut, segmentasi dilakukan untuk menandai objek yang berbeda, bergantung pada kedalaman segmentasi (a.l. *semantic*, *instance*, maupun *panoptic*).

3. Klasifikasi Gambar (*Image Classification*)

Mengkategorikan gambar berdasarkan kelas yang telah ditentukan sebelumnya. Misalnya, dalam sebuah dataset gambar anjing dan kucing, sistem CV akan menentukan apakah gambar yang diberikan adalah anjing atau kucing.



Gambar 38 Contoh Pemanfaatan *Computer Vision*

4. Pemrosesan Video (*Video Analysis*)

Menangkap dan menganalisis informasi dari urutan gambar (video). Ini bisa digunakan untuk mengenali gerakan, melacak objek bergerak, atau melakukan analisis perilaku.

5. Pengenalan Wajah (*Facial Recognition*)

Digunakan untuk verifikasi identitas, misalnya dalam sistem keamanan atau perangkat seperti *smartphone*.

6. Mobil Otonom (*Autonomous Vehicles*)

Kendaraan tanpa pengemudi menggunakan CV untuk mendeteksi rambu lalu lintas, pejalan kaki, dan kendaraan lain di jalan.

7. Augmented Reality (AR)

CV digunakan untuk mendeteksi dan melacak objek di dunia nyata dan menambahkannya dengan informasi atau objek virtual.

8. Aplikasi Pemantauan Kesehatan

Digunakan dalam analisis citra medis seperti MRI dan X-ray untuk mendeteksi penyakit.

9. Aplikasi pada Retail

Pengenalan produk dan pelacakan pembeli di toko-toko fisik menggunakan CV untuk meningkatkan pengalaman belanja dan analisis perilaku konsumen.

Tantangan dalam *Computer Vision*

- **Variasi Visual:** Gambar yang diambil dalam kondisi pencahayaan berbeda, dari sudut yang berbeda, atau dengan *noise* bisa membuat tugas CV menjadi sulit.
- **Derau/Noise:** Ketika sebagian objek tertutup oleh objek lain, pengenalan objek bisa menjadi lebih kompleks.
- **Resolusi Rendah:** Gambar dengan resolusi rendah atau buram bisa menyebabkan kesalahan dalam analisis objek.

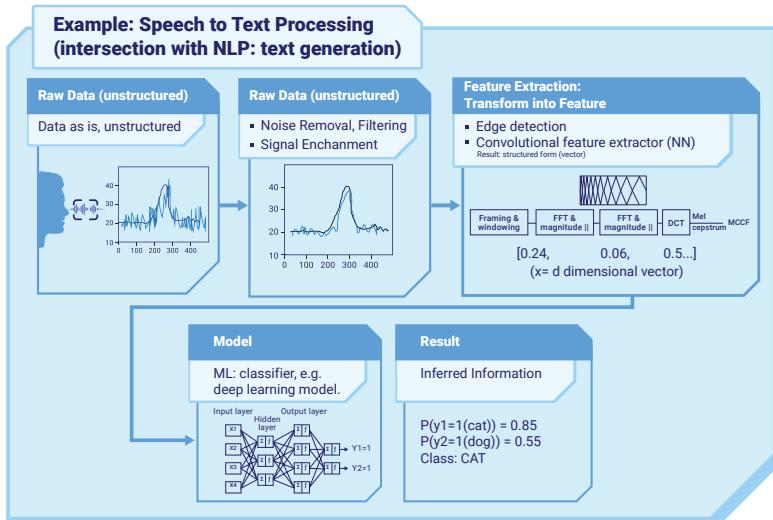
Computer Vision adalah teknologi kunci yang memiliki dampak besar di berbagai industri, termasuk kesehatan, otomotif, keamanan, dan hiburan. Dengan kemajuan teknik *deep learning*, CV semakin efisien dalam menangani tantangan visual kompleks dan terus berkembang dalam memahami dunia visual, mendekati kemampuan penglihatan manusia.

5.3 *Speech and Signal Processing*

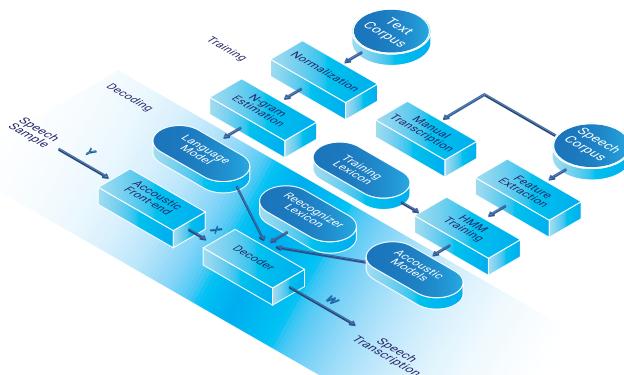
Signal Processing adalah bidang yang mempelajari analisis, modifikasi, dan pemrosesan sinyal-sinyal seperti suara, gambar, atau data biologis. Dengan semakin

berkembangnya algoritma **Machine Learning (ML)**, kemampuan pemrosesan sinyal mengalami peningkatan signifikan karena ML memungkinkan komputer untuk mengekstraksi pola, fitur, atau informasi penting dari sinyal yang kompleks atau tidak terstruktur.

Tahapan *signal processing* secara umum terlihat sebagaimana pada Gambar 39. Contoh pada Gambar tersebut merepresentasikan pemrosesan *speech to text*.

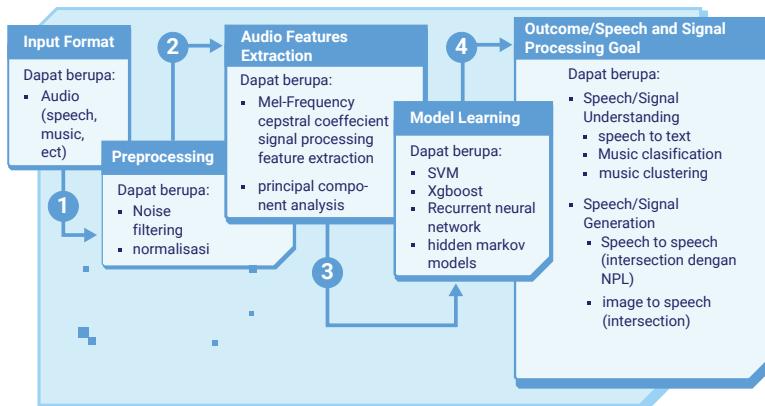


Gambar 39 Tahapan Signal Processing



Gambar 40 Contoh tahapan *signal processing* sederhana: *speech to text*

Sebagai generalisasi pada Gambar 40 di atas, secara umum, tahapan dalam *speech and signal processing* dapat dirangkum pada Gambar 41. Secara umum, terdapat tahapan yang mencirikan AI *speech and signal processing*, yaitu adanya *signal feature extraction* dan proses pembangunan representasi pengetahuan (AI *modelling*).



Gambar 41 Tahapan Speech and Signal Processing

Teknik Unik dalam *Signal Processing* dengan *Machine Learning*

1. *Preprocessing*

Sinyal yang diterima sering kali mengandung *noise* atau gangguan. Oleh karena itu, langkah pertama adalah mempersiapkan data sinyal dengan memfilter *noise*, normalisasi, atau transformasi ke domain yang lebih mudah dianalisis (seperti *Fourier Transform*). Contohnya, pada sinyal audio, *preprocessing* bisa melibatkan penghapusan frekuensi yang tidak relevan.

2. *Feature Extraction*

Sinyal mentah sulit digunakan langsung untuk pelatihan model *machine learning*. *Feature extraction* bertujuan mengubah sinyal mentah menjadi representasi yang lebih sederhana dan informatif, seperti spektrum frekuensi atau pola tertentu. Pada sinyal gambar, fitur yang umum adalah tepi dan tekstur. Berikut adalah beberapa teknik ekstraksi fitur.

- Fourier Transform*: Teknik ini mengubah sinyal dari domain waktu ke domain frekuensi. Ini sangat berguna untuk mengekstrak informasi frekuensi, seperti dalam analisis audio atau sinyal biologis (misalnya, EKG). Secara matematis, *Fourier Transform* direpresentasikan dalam formula berikut.

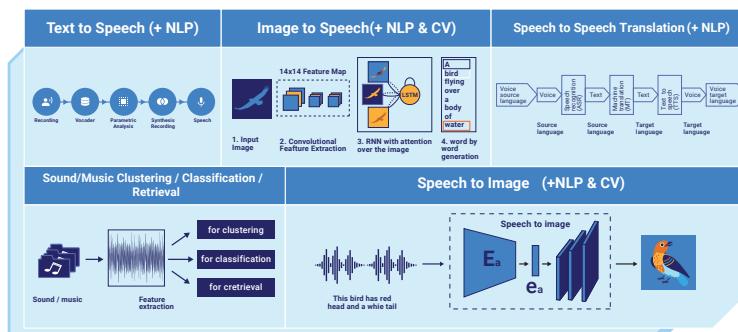
$$(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \quad (5.2)$$

- Fourier Transform* mengubah sinyal waktu kontinu $x(t)$ ke dalam domain frekuensi $X(f)$.
- c. *Principal Component Analysis* (PCA): Metode ini mengurangi dimensi sinyal dengan menemukan arah di mana variabilitas terbesar terjadi dalam data. Ini sering digunakan untuk kompresi dan ekstraksi fitur dalam sinyal gambar dan suara.
 - d. *Mel-Frequency Cepstral Coefficients* (MFCC): Dalam pengenalan suara, MFCC adalah teknik yang banyak digunakan untuk mengekstrak fitur yang merepresentasikan karakteristik suara dalam domain frekuensi.

Fitur-fitur yang telah diekstraksi tersebut kemudian digunakan sebagai *input* dalam pemrosesan berikutnya oleh *machine learning* sesuai *task/tujuan* yang diharapkan.

Aplikasi Signal Processing dengan Machine Learning

Signal dan *signal processing* telah dimanfaatkan di berbagai bidang. Sebagaimana *signal processing* juga dapat dimanfaatkan secara bersama dengan NLP (misal *text to speech*), atau *image* (*voice to content based image retrieval*). Gambar 42 mengilustrasikan beberapa contoh pemanfaatan *speech and signal processing*.



Gambar 42 Contoh Pemanfaatan *Speech and Signal Processing*

Beberapa contoh penerapan lainnya, yaitu:

1. Pengenalan Suara (*Speech Recognition*)
Digunakan dalam aplikasi seperti asisten virtual (contoh: Siri, Google Assistant). Sinyal suara dikonversi menjadi representasi fitur yang kemudian diproses oleh model *machine learning* untuk mengenali kata-kata.
2. Deteksi Sinyal Biomedis (*Biomedical Signal Processing*)
Dalam diagnosis medis, sinyal seperti EEG (elektroensefalografi) atau EKG (elektrokardiogram) diproses dan dianalisis menggunakan teknik *machine learning* untuk mendeteksi gangguan seperti epilepsi atau aritmia jantung.

3. Pengolahan Sinyal Radar dan Sonar
Digunakan dalam sistem navigasi, deteksi objek bawah air, dan dalam militer. Sinyal radar atau sonar dianalisis untuk mendeteksi objek atau pola dalam data yang sulit dilihat.
4. Pemrosesan Sinyal Audio (*Audio Signal Processing*)
Digunakan dalam aplikasi seperti pengenalan musik, penghapusan *noise*, atau peningkatan kualitas suara dalam perangkat telekomunikasi dan hiburan.

Tantangan dalam *Signal Processing* dengan *Machine Learning*

1. **Data yang tidak stabil:** Beberapa sinyal, terutama sinyal biologis seperti EKG atau EEG, bisa sangat bervariasi antar-individu, yang membuat *modeling* lebih menantang.
2. **Dimensi tinggi:** Sinyal bisa memiliki dimensi yang sangat tinggi (misalnya gambar dengan resolusi tinggi), sehingga diperlukan teknik *feature extraction* yang efisien.
3. **Noise:** Sinyal sering kali tercemar oleh *noise* atau gangguan lain, sehingga *preprocessing* yang baik sangat penting.

Signal processing dengan *machine learning* adalah pendekatan yang sangat kuat untuk mengolah data sinyal yang kompleks, dan memiliki aplikasi yang luas di banyak bidang, termasuk medis, telekomunikasi, dan pengenalan gambar serta suara. Dengan memanfaatkan teknik *machine learning*, kita dapat membuat model yang lebih adaptif dan efisien dalam menangani tantangan pemrosesan sinyal yang rumit.

5.4 Data Mining and Time Series Analysis

Data Mining adalah proses menemukan pola, tren, atau informasi berharga dari sejumlah besar data mentah. Teknik ini digunakan dalam berbagai bidang seperti pemasaran, kesehatan, keuangan, dan lain sebagainya untuk mengubah data yang disimpan menjadi pengetahuan yang dapat ditindaklanjuti. Oleh karena itu, *data mining* sering juga disebut *Knowledge Discovery in Database* (KDD).

Tujuan Data Mining

1. **Ekstraksi Pengetahuan:** Menggali informasi tersembunyi yang berguna dari data.
2. **Prediksi:** Memodelkan perilaku atau tren di masa depan berdasarkan data historis.
3. **Pengenalan Pola:** Mengidentifikasi pola tersembunyi di dalam data.

Jenis Data Apa yang Dapat Ditambang?

Data mining tidak terbatas pada satu jenis data atau media. Setiap bentuk penyimpanan data dapat dipertimbangkan untuk dapat dilakukan *data mining*. Namun, masing-masing jenis data tentunya memiliki metode dan metologi yang bervariasi ketika akan dilakukan *data mining*. Berbagai jenis data yang dapat digunakan untuk *data mining* meliputi (Macskassy, 2007):

1. *Flat File*: File teks atau biner yang sederhana dan sering digunakan untuk *data mining* skala kecil
2. *Relation Database*: Kumpulan tabel yang memiliki karakteristik nilai objek yang dapat terhubung antartabel.
3. *Data Warehouse*: Kumpulan data yang telah dibersihkan dan diolah untuk memenuhi kebutuhan bisnis.
4. *Transaction Database*: Kumpulan dokumen yang memiliki *unique identifier*, *timestamp*, dan sekumpulan *object*.
5. *Multimedia Database*: Data berupa audio, video, gambar, dan teks yang biasanya disimpan dalam bentuk *file systems*.
6. *Spatial Database*: Mengandung informasi geografis.
7. *Time-Series Database*: Data yang selalu bergerak setiap saat, seperti *stock market*.
8. *World Wide Web*: Mencakup berbagai jenis konten seperti teks, video, musik, *hyperlink*, hingga konten yang terus berubah.

Umumnya, langkah-langkah *data mining* mengikuti tahapan sebagaimana pada framework Cross Industrial Standard Process for Data Mining (CRISP-DM) yang mencakup *business understanding* s.d. *deployment* sebagaimana dijelaskan pada *AI Playbook Series 1: The First Step* sebelumnya.

Teknik Data Mining

Pada dasarnya berbagai teknik inferensi model dapat digunakan dalam *data mining* karena tujuan umum *data mining* adalah perolehan informasi serta pola dari sekumpulan data yang ada. Secara umum, terdapat dua pendekatan utama dalam *data mining*, yaitu prediktif dan deskriptif. Prediktif membuat estimasi atau prediksi berdasarkan data yang ada, sedangkan deskriptif menjelaskan karakteristik utama dari data yang sedang dianalisis (Fayyad et al., 1996). Beberapa contoh teknik yang umum digunakan dalam *data mining* yaitu a.l.:

1. **Classification**: Mengategorikan data ke dalam kelompok yang telah ditentukan. Contoh: memprediksi apakah pelanggan baru akan membeli produk tertentu berdasarkan perilaku pembelian sebelumnya.
2. **Clustering**: Mengelompokkan data yang serupa ke dalam kategori yang tidak diketahui sebelumnya. Contoh: segmentasi pelanggan.

3. **Association Rule Mining:** Mencari hubungan antarvariabel dalam data. Contoh: “Market Basket Analysis” untuk menemukan barang-barang yang sering dibeli bersama.
4. **Regression:** Memprediksi nilai numerik berdasarkan hubungan antarvariabel. Contoh: memprediksi harga rumah berdasarkan faktor seperti ukuran dan lokasi.

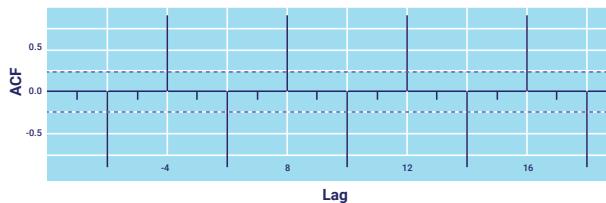
Contoh penerapan teknik di atas di antaranya:

1. **Anomaly Detection:** Mengidentifikasi data yang tidak biasa atau anomali. Contoh: mendeteksi transaksi penipuan dalam data keuangan.
2. **Prediction - Forecasting:** Memperkirakan nilai atau tren masa depan berdasarkan data historis. Contoh: Memprediksi tren penjualan produk berdasarkan data penjualan sebelumnya.
3. **Prediction - Nowcasting:** Memperkirakan nilai atau tren masa saat ini berdasarkan data historis, serta data terkini yang dapat dijadikan sebagai dasar. Contoh: Memprediksi level inflasi berdasarkan beberapa indikator historis dan indikator bulan berjalan.

Time Series Analysis dalam Data Mining

Time series analysis merupakan teknik data mining yang berfokus pada data yang berubah seiring waktu, digunakan untuk mengekstrak pola, tren, dan wawasan. Biasanya teknik ini sering digunakan dalam berbagai bidang seperti keuangan, cuaca, kesehatan, dan penjualan. Adapun beberapa konsep dasar terkait *time series analysis* adalah:

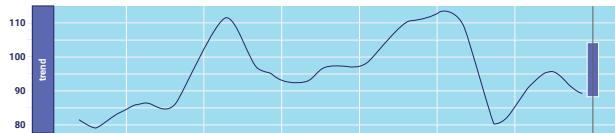
1. Karakteristik Data *Time Series*
 - a. **Autocorrelation:** Nilai data saat ini memiliki hubungan dengan nilai data di masa lalu.



Gambar 43 Autocorrelation

Terdapat *spike* pada *lag* 4, 8, 12, dan 16 yang menunjukkan adanya pola berulang pada sekitar setiap 4 titik waktu(simak Gambar 43).

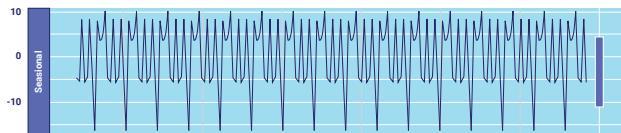
- b. *Trend*: Peningkatan atau penurunan nilai data dalam jangka panjang



Gambar 44 Trend

Pada Gambar 44, *trend* mengilustrasikan 4 potensi makna pergerakan data, yaitu data mengalami kenaikan yang stabil di awal, pertumbuhan yang stagnan di tengah, penurunan tajam, dan pemulihannya parsial di akhir.

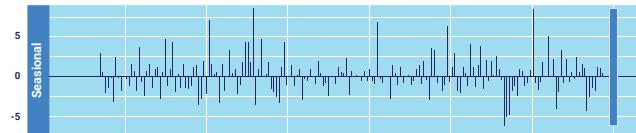
- c. *Seasonality*: Pola berulang dalam interval waktu tertentu (harian, mingguan, bulanan)



Gambar 45 Seasonality

Sebagaimana ilustrasi pada Gambar 45, nilai musiman berosilasi antara sekitar -10 hingga +10 menandakan adanya pola musiman dalam data yang cukup stabil pada frekuensi tertentu.

- d. *Residual / Noise*: Variasi acak dalam data yang tidak mencerminkan tren utama.



Gambar 46 Plot ACF/PACF

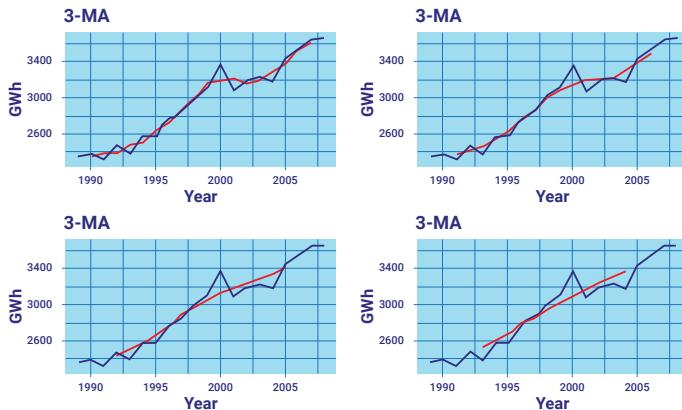
Gunakan plot ACF/PACF (ilustrasi pada Gambar 46) untuk memastikan residual tidak memiliki korelasi jangka panjang.

2. Metode Time Series Analysis

- a. *Moving Average*: Menghaluskan fluktuasi data untuk mengidentifikasi tren dengan menghitung rata-rata nilai dalam rentang waktu tertentu di sekitar titik waktu t .

$$\hat{T}_t = \frac{1}{m} \sum_{j=-k}^k y_{t+j} \quad (5.3)$$

\hat{T}_t adalah estimasi tren-siklus pada waktu t.
 m adalah adalah orde *moving average* ($m=2k+1$).
 k adalah jumlah periode untuk menghitung rata-rata.
 y_{t+j} adalah nilai deret waktu pada periode $t+j$.



Gambar 47 Metode Moving Average

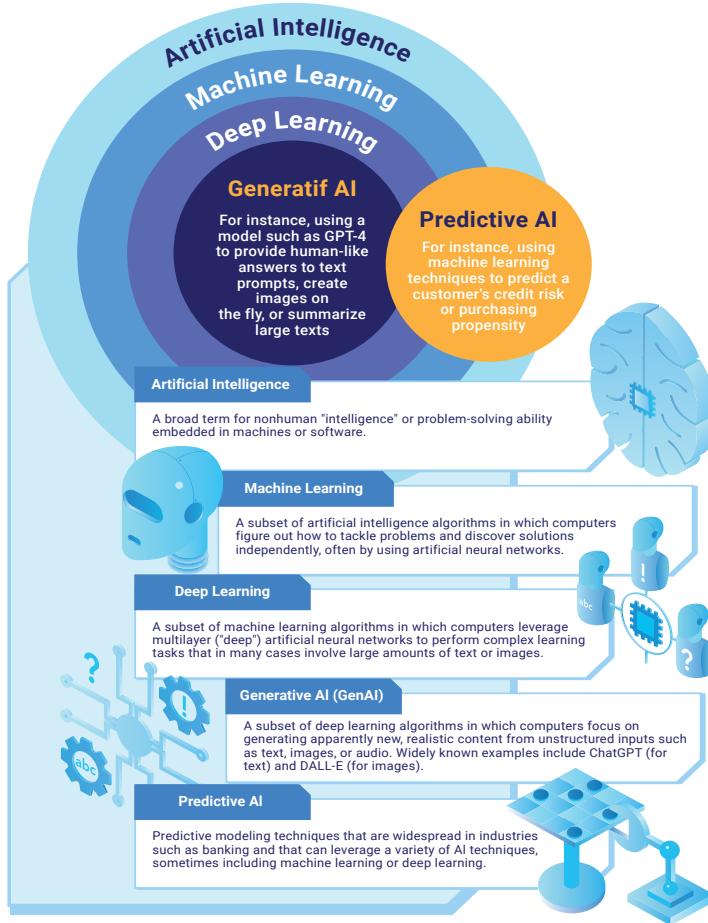
Garis merah menunjukkan tren-siklus hasil *smoothing* untuk rata-rata orde 3, 5, 7 dan 9, sedangkan garis hitam menunjukkan data asli. Hal ini terlihat sebagaimana pada Gambar 47.

- b. *Classical Decomposition*: diperkenalkan pada tahun 1920-an, merupakan teknik dasar untuk membagi data *time series* menjadi beberapa komponen utama, yaitu tren, *seasonality*, dan *residual*. Terdapat dua bentuk utama dari *classical decomposition*, yaitu *additive* dan *multiplicative*.
- c. *Exponential Smoothing*: Memberikan bobot lebih besar pada data terbaru untuk memprediksi tren.
- d. *ARIMA (AutoRegressive Integrated Moving Average)*: Model statistik yang banyak digunakan dalam prediksi data *time series*.
- e. *Long Short-Term Memory (LSTM)*: Model *deep learning* yang digunakan untuk memprediksi data *time series* dengan pola kompleks.

5.5 Generative AI (GenAI)

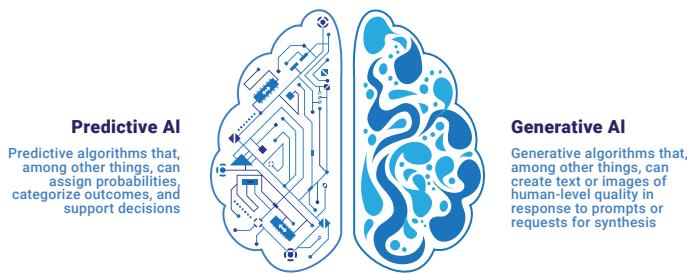
Munculnya teknologi *Generative AI* memperluas dan memperkaya pemanfaatan AI dalam berbagai sektor industri dengan berbagai nilai tambah baru. Kemampuan model *Generative AI* untuk memahami (*understanding*) dan menghasilkan (*generate*)

dengan bahasa sehari-hari memberikan nilai aksesibilitas untuk AI sehingga dapat dimanfaatkan oleh pengguna teknis maupun non-teknis. *Generative AI* memungkinkan interaksi linguistik dan memahami *input* dan perintah pengguna untuk kemudian menghasilkan teks, gambar, audio, dan bahkan video secara otomatis. Meski terdapat berbagai sudut pandang dalam penempatan GenAI, hubungan GenAI dapat diilustrasikan dalam diagram ven pada Gambar 48 berikut.



Gambar 48 Al vs. GenAI

Berdasarkan Riemer, et al (2023) *Generative AI* dan *Predictive AI* masing-masing memiliki kapabilitas yang kuat dengan tujuan yang berbeda. *Generative AI* dan *Predictive AI* saling melengkapi satu sama lainnya yang dapat dianalogikan sebagai dua bagian otak manusia sebagaimana tampak pada Gambar 49.

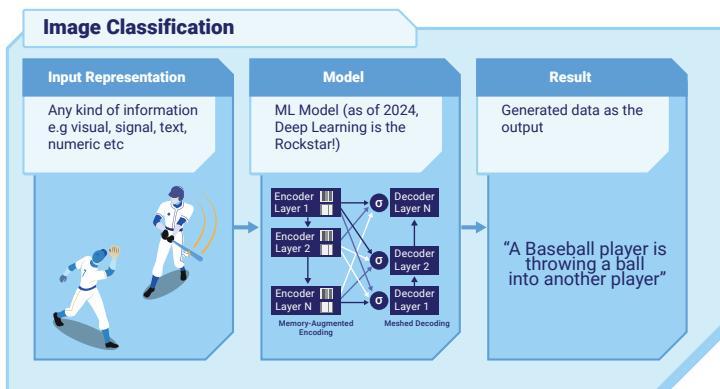


Gambar 49 Predictive vs. Generative AI

Predictive AI dapat dianggap sebagai otak kiri yang bersifat logika, pengukuran, dan perhitungan. Otak kiri ini terdiri atas algoritma seperti probabilitas, klasifikasi, dan mendukung pengambilan keputusan. Di sisi lain, *Generative AI* bertindak sebagai otak kanan yang unggul dalam kreativitas, memahami dan berekspresi, serta pemahaman perspektif secara holistik. Salah satu contoh implementasi kapabilitas otak kanan ini seperti generasi respons *Chatbot* yang menyerupai respons manusia.

Implementasi teknologi *Generative AI* tidak hanya terbatas pada *use case* seperti *Chatbot* layanan pelanggan. Akan tetapi, *Generative AI* juga dapat dimanfaatkan untuk hal lain seperti membantu analis dalam proses analisis keuangan, generasi kode pengembangan, generasi konten publikasi secara otomatis, dan sebagainya. *Generative AI* juga sudah dieksplorasi dan diimplementasi secara global oleh bank sentral seperti Bank of England, European Central Bank (ECB), dan Deutsche Bundesbank, serta institusi keuangan internasional seperti Deutsche Bank, American Express, dan Wells Fargo.

Gambar 50 mengilustrasikan salah satu contoh sederhana pemanfaatan GenAI.



Gambar 50 Contoh sederhana GenAI: Image to Text

Pemanfaatan Umum *Generative AI*

Dalam skenario percakapan berbasis *Chatbot*, model *Generative AI* dapat menerima perintah berupa teks sebagai *input* dan merespons dengan jawaban sesuai konteks yang relevan dalam bahasa selayaknya respons manusia. Di bidang lain, seperti desain, model *Generative AI* dapat menciptakan desain grafis, logo, atau karya digital berdasarkan *input* spesifik dari pengguna. Kapabilitas ini mempermudah pengguna non-teknis untuk memanfaatkan teknologi *AI* dalam menyelesaikan pekerjaan sehari-hari khususnya yang membutuhkan kreativitas.

Model *Generative AI* juga dapat dimanfaatkan dalam analisis data, di mana model *Generative AI* mampu menghasilkan narasi secara otomatis dari dataset yang kompleks, atau digunakan untuk simulasi medis untuk permodelan skenario penyakit dan potensi pengobatannya. Hal ini dapat dilakukan dengan kapabilitas model *Generative AI* dalam pemahaman konteks dan menghasilkan konten yang bersifat baru dan relevan berdasarkan pola data yang ada.

Generative AI in the Market

Generative AI yang tersedia umumnya hadir dalam dua bentuk utama: *Public Generative AI* (seperti ChatGPT), dan *Enterprise Generative AI*. *Enterprise Generative AI* disesuaikan untuk kebutuhan institusi dan organisasi, dengan menggunakan teknik *Retrieval Augmented Generation* (RAG) yang dibangun di atas model dasar *Generative AI*. Keduanya memiliki kapabilitas yang kuat namun dengan tujuan dan risiko masing-masing.

Public Generative AI dilatih dengan data dari korpus (kumpulan teks) publik yang sangat luas, seperti artikel berita, media sosial, dan literatur seperti publikasi saintifik. Korpus yang luas ini memungkinkan model tersebut memahami dan menghasilkan dari berbagai topik dan konteks. Hal ini memberikan keunggulan berupa fleksibilitas dari berbagai domain. Namun, dalam konteks profesional atau institusional, model ini memiliki keterbatasan dalam hal memahami bahasa khusus, istilah, aturan, atau makna spesifik dari institusi terkait. Selain itu, *Public Generative AI* juga terdapat kemungkinan untuk melakukan pelatihan ulang model berdasarkan *input* atau data dari pengguna. Hal ini mengakibatkan risiko pemanfaatan data pribadi atau institusi yang bersifat spesifik untuk diakses secara umum.

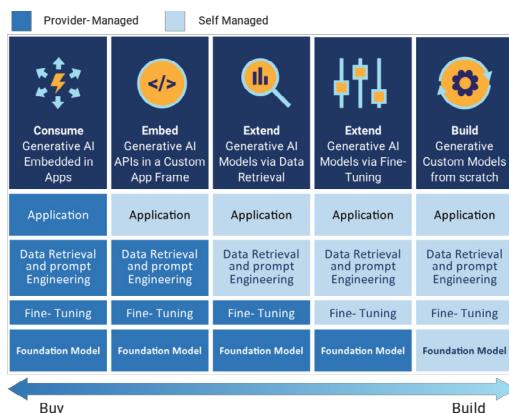
Di sisi lain, *Enterprise Generative AI* dikembangkan dan dikelola secara mandiri oleh instansi. Hal ini dapat mengisi keterbatasan yang dimiliki *Public Generative AI* seperti pemahaman yang lebih akurat dan kontekstual sesuai institusi, kemanan data, dan kapabilitas pelatihan ulang (*finetuning*) dengan data spesifik instansi. Hal ini dapat dilakukan dengan *Retrieval Augmented Generation* (RAG).

Strategi Adopsi Generative AI

Box 5.1 General Consideration

- **Fine tune on domain-specific data** (Howard & Ruder, 2018).
Pastikan model dilatih pada domain spesifik yang dihadapi.
- **Implement Transfer Learning** (Raffel *et al.*, 2020)
Memungkinkan pelatihan lebih efisien, model telah dilatih dengan *knowledge* serupa.
- **Apply Bias Mitigation Techniques** (Bolukbasi *et al.*, 2016).
- Penting untuk menghindari adanya bias.
- **Human-in-the-Loop (HITL) Approach** (Amershi *et al.*, 2019) & *Anticipate Hallucination* (Hinton, 2023)
HITL penting untuk memastikan akurasi dan *realibility*, dan mengantisipasi halusinasi
- **Ensure Privacy and Data Security** (Brundage *et al.*, 2020)
Menjamin data *security* dan privasi, baik dari sisi data *input* s.d. *output*.
- **Ensure Explainability** (Ribeiro *et al.*, 2016)
Sebisa mungkin, proses inferensi oleh Al harus dapat dijelaskan.

Apabila sebuah korporasi berencana untuk mengadopsi Solusi Generative AI, terdapat beberapa alternatif pilihan strategi adopsi berdasarkan kebutuhan masing-masing pengguna. Menurut Gartner, sebagaimana pada Gambar 51, terdapat lima pilihan strategi, di antaranya adalah *Consume*, *Embed*, *Extend (Data Retrieval)*, *Extend (fine-tuning)*, dan *Build*. Namun, secara garis besar, kelima strategi tersebut terbagi menjadi dua spektrum, yaitu *Buy* dan *Build*.



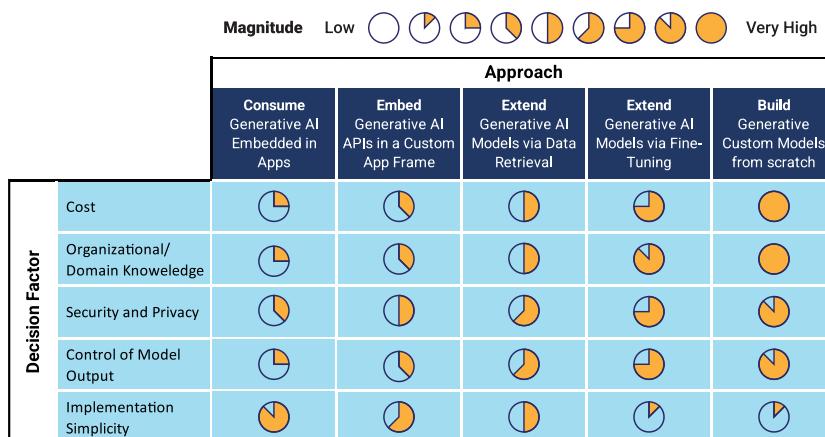
Gambar 51 Strategi Adopsi Generative AI

Strategi *Consume* menekankan penggunaan solusi *Generative AI* yang sudah tersedia pada aplikasi eksisting, contohnya seperti penggunaan fitur AI pada Adobe Firefly. Opsi ini sudah menyediakan paket lengkap mulai dari aplikasi, mekanisme *data retrieval*, hingga *foundation model*. Opsi ini tidak memiliki aspek fleksibilitas, sehingga tidak membuka kemungkinan untuk melakukan pengembangan yang terkostumisasi. Opsi kedua adalah pemanfaatan Solusi *Generative AI* melalui API. Pengguna terlebih dahulu harus memiliki aplikasi sehingga memerlukan mekanisme komunikasi antara Solusi *Generative AI* dan aplikasi eksisting melalui sebuah API.

Selanjutnya adalah opsi *extend* yang telah menyediakan paket *foundational model* yang sudah di-*fine tuned* menggunakan dataset milik penyedia Solusi. Perbedaan mendasar antara pilihan *Extend (data retrieval)* dan *Extend (fine-tuning)* adalah mekanisme *data retrieval* tidak membutuhkan *model training*, melainkan hanya perlu melakukan *data retrieval* dan *prompt engineering* terhadap *knowledge base* yang sudah disiapkan oleh pengguna. Hal ini cukup efisien mengingat proses *training* LLM membutuhkan biaya komputasi yang cukup mahal. Opsi terakhir merupakan opsi membangun Solusi *Generative AI* “*from scratch*” di mana pengguna harus membangun *foundation model* terlebih dahulu, melakukan proses *model training*, optimasi *model fine tuning* hingga menyiapkan aplikasi *front-end* dari model tersebut.

Konsideran Pemilihan Strategi Adopsi *Generative AI*

Pengembang model dapat memilih salah satu dari kelima opsi strategi sebelumnya menggunakan lima konsideran yang merepresentasikan *resource* eksisting yang dimiliki oleh tiap-tiap institusi. Kelima konsideran tersebut di antaranya adalah *cost*, *organizational domain knowledge*, *security & privacy*, *control of model output*, serta aspek kepraktisan dalam proses pengembangan solusi. Hal ini diilustrasikan sebagaimana pada Gambar 52 berikut.



Gambar 52 Konsideran Pemilihan Strategi Adopsi *Generative AI*

Opsi strategi adopsi *Generative AI* dapat dipilih menggunakan panduan pada Gambar 52 di atas. Masing-masing konsideran memiliki spektrum *magnitude* yang berbeda, bergantung pada tiap-tiap opsi strategi. Salah satu contohnya adalah pada opsi strategi *Consume* membutuhkan spektrum biaya yang lebih rendah, *organizational domain knowledge* yang rendah, namun memiliki kemudahan dalam proses pengembangan solusi.





BAGIAN 6

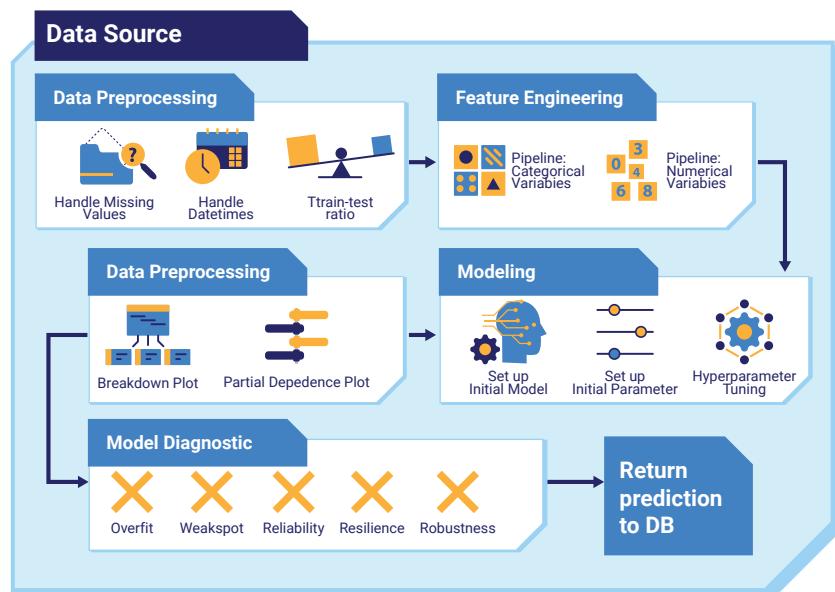
UTILISING AI IN
BANK INDONESIA

AI telah dimanfaatkan dalam area ekonomi, melalui *big data* yang mencakup tradisional data maupun non-tradisional data. Berbagai pemanfaatan untuk *tracking* makroekonomi, sistem pembayaran dan area terkait telah dilakukan misalnya pada Buell, *et al* (2021), Bok, *et al* (2017), Bruno, *et al* (2016), Buell, *et al* (2021), Chakraborty, *et al* (2017), Chapman & Desai (2020), Daas & Puts, (2014), Gensler (2020), McLaren, N., & Shanbhogue, R. (2011), Woloszko, N., (2020). Bank Indonesia juga memanfaatkan AI di beberapa sektor utama a.l. Moneter, Makroprudensial, Sistem Pembayaran, serta *Human Resources* (HR).

6.1 AI for Monetary Policies

Economic Nowcasting

Gambar 53 di bawah ini mengilustrasikan contoh metodologi untuk melakukan *nowcasting* dan *forecasting* pada indikator perekonomian menggunakan metode *machine learning*. Setidaknya terdapat dua tahap sebelum sampai pada tahap *modelling*, yaitu *data preprocessing* dan *feature engineering*.



Gambar 53 Tahapan Nowcasting

Data *preprocessing* memiliki tujuan utama untuk melakukan standardisasi data, meliputi konfigurasi *date time*, penentuan *split ratio* pada dataset *training* dan *testing*, serta meng-handle *missing values*. Selanjutnya, *feature engineering* berperan untuk

memastikan bahwa fitur atau variabel yang digunakan dalam model memberikan kontribusi terbaik pada performa model. Pengembangan model di Bank Indonesia memiliki standar dalam meng-*handle variable* yang berbentuk kategori dan numerik. Hal ini diperlukan untuk memastikan standardisasi dataset dan variabel sebelum menjadi *input* pada model.

Box 6.1 Tools Pengembangan

Data storage: Hadoop HDFS

Analytics Tools:

Programming Language: Python

Libraries:

Machine learning & Scientific : scikit-learn, xgboost, numpy, Hyperopt

Interpretability: shap

Visual: pyplot, seaborn

Structured data analysis: pandas

Optimisasi parameter dalam proses *training* juga didesain dalam sebuah *pipeline* model. Pengembang model terlebih dahulu menyediakan *parameter space* pada *pipeline* sesuai dengan karakteristik data dan *use case* terkait. Pemilihan *parameter space* menjadi kritisik karena memiliki dilema berupa *resource* komputasi pada proses *training* dan performa model. Semakin banyak jumlah parameter bisa jadi membuat model memiliki karakteristik *overfit* atau cenderung menghafal pola sehingga tidak memiliki *predictive power* yang baik.

Selanjutnya, salah satu tahapan yang cukup penting dalam proyek ini adalah evaluasi model, meliputi aspek *model interpretability* dan *model diagnostic*. Bagi Bank Indonesia, model yang baik tidak hanya yang memiliki akurasi yang baik saja, namun juga memiliki aspek *interpretability*, sehingga kebijakan yang dibuat dapat menyarai variabel penting yang berhasil diidentifikasi dalam sebuah indikator perekonomian. Selain *model interpretability*, *model diagnostic* juga digunakan untuk mengevaluasi performa model seperti *overfit test*, *weakspot test*, dan tes lainnya. Terakhir, hasil prediksi akan dikirim menuju *data mart* sebelum divisualisasikan melalui *dashboard Power BI*.

CB Credibility Index Measurements

Kedua, salah satu metodologi utama yang digunakan dalam berbagai *use case*, yaitu *text mining/text analytics*. Sebagai contoh, penggunaan *text mining* dalam sektor moneter adalah untuk mengidentifikasi persepsi publik terkait kredibilitas Bank Indonesia mengenai kebijakan moneter. Metodologi serupa juga pernah dikembangkan oleh Tobback, et al (2017), dengan menganalisis teksstual data untuk mengukur sebearapa efektif komunikasi terkait kebijakan bank sentral.

Secara umum, Gambar 54 di bawah ini mengilustrasikan metodologi bagaimana suatu indeks kredibilitas kebijakan BI yang diukur melalui 4 aspek tersebut.



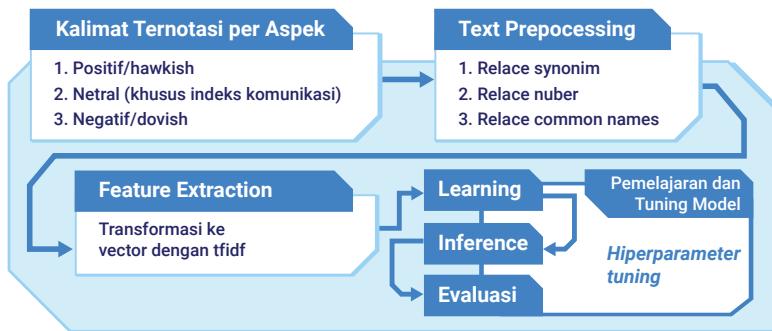
Gambar 54 Ilustrasi Metodologi pada Indeks Kredibilitas Kebijakan BI

Pertama data berupa teks diperoleh dari portal berita nasional berbahasa Indonesia. Kemudian pada berita tersebut dilakukan pemecahan ke dalam bentuk kalimat untuk kemudian dilakukan *filtering* agar diperoleh kalimat-kalimat yang mengandung *keywords* yang telah terdefinisi. Setelahnya, dilakukan 2 tahap klasifikasi, yaitu klasifikasi untuk memperoleh kalimat-kalimat yang terindikasi terkait kredibilitas, serta klasifikasi sentimen pada kalimat-kalimat yang terindikasi terkait kredibilitas. *Output* setiap aspek klasifikasi akan memberikan hasil yang *mutually exclusive*, dengan tiga kemungkinan *output* yaitu:

1. Positif/hawkish
2. Netral (khusus untuk aspek komunikasi)
3. Negatif/dovish.

Tahap terakhir yaitu agregasi pada kalimat yang muatan *sentiment* telah diprediksi oleh *machine learning* model.

Untuk *classifier* baik itu untuk klasifikasi kalimat kredibilitas maupun klasifikasi *sentiment*, secara garis besar alurnya dapat diilustrasikan pada Gambar 55.



Gambar 55 Alur Metodologi

Pada setiap aspek, model dilatih dengan memanfaatkan data latih yang dianotasi. Tujuan akhir pelatihan adalah agar model dapat seakurat mungkin mengklasifikasikan

kalimat kredibilitas atau mengklasifikasikan *sentiment* pada kalimat kredibilitas (tergantung tahapan klasifikasi apakah terkait kredibilitas, atau terkait sentimen).

Box 6.2 Tools Pengembangan

Data storage: Hadoop HDPS

Analytics Tools:

Programming Language: Python

Libraries:

Machine learning & Scientific: scikit-learn, xgboost, numpy,

Language toolkit: NLTK

Visual: pyplot, seaborn

Structured data analysis: pandas

Praproses pada teks dilakukan dengan menyiapkan teks agar bersih, konsisten, serta terhindar dari *noise*. Beberapa tahapan yang dilakukan misalnya menyeragamkan sinonim, mengganti angka, serta menyeragamkan nama/entitas. Selanjutnya, representasi teks diubah ke dalam vektor melalui transofrmasi tfidf. Setelah teks (dalam hal ini, kalimat) berhasil direpresentasikan ke dalam bentuk vektor, langkah selanjutnya adalah pelatihan model sebagaimana biasanya.

Model yang telah berhasil tersebut kemudian digunakan untuk *running* pada data agar dapat digunakan untuk membantu menganalisis *sentiment* dalam rangka pembentukan indeks kredibilitas moneter.

6.2 AI for Macroprudential Policies

Stabilitas Sistem Keuangan adalah kondisi di mana sistem keuangan berfungsi secara efektif, efisien, dan mampu bertahan dari gejolak domestik maupun global. Stabilitas ini penting untuk memastikan kelancaran intermediasi dan layanan keuangan, sehingga mendukung pertumbuhan ekonomi nasional. Bank Indonesia merumuskan dan mengimplementasikan kebijakan makroprudensial dengan mendorong intermediasi yang seimbang, berkualitas, dan berkelanjutan; mengelola serta mengurangi risiko sistemik; serta memperluas inklusi ekonomi, inklusi keuangan, dan keuangan berkelanjutan.

Dalam upaya memperkuat stabilitas sistem keuangan, Bank Indonesia terus mengembangkan pemanfaatan AI dalam mendukung perumusan kebijakan makroprudensial. Berikut adalah sejumlah *use case* AI yang telah dikembangkan untuk mendukung asesmen kebijakan makroprudensial Bank Indonesia.

Forecasting Macroprudential Key Indicators

Dalam menjaga stabilitas sistem keuangan, Bank Indonesia perlu mengantisipasi potensi risiko dan ketidakseimbangan di sektor perbankan serta pasar keuangan. Salah satu pendekatan yang dilakukan adalah dengan mengembangkan model prediksi (*forecasting*) terhadap indikator-indikator makroprudensial guna memperoleh gambaran tentang kondisi sistem keuangan dalam enam bulan ke depan dengan memanfaatkan AI/ML. Indikator utama kebijakan makroprudensial yang telah dikembangkan mencakup Alat Likuid Dana Pihak Ketiga (ALDPK), *Interest Coverage Ratio* (ICR), Kredit Konsumsi (KK), Kredit Modal Kerja (KMK), Kredit Pemilikan Rumah (KPR), Kredit Total (dari sisi *demand* dan *supply*), *Non-Performing Loan* (NPL), Suku Bunga Deposito, Suku Bunga Kredit, dan *Yield* Surat Berharga Negara (SBN). Pengembangan model *forecasting* ini bertujuan untuk meningkatkan kemampuan deteksi dini terhadap potensi risiko pada likuiditas perbankan, stabilitas kredit, serta keseimbangan pasar keuangan.

Box 6.3 Tools Pengembangan

Data storage: Hadoop HDFS/Excel.

Analytics Tools:

Programming Language: Python

Libraries:

Machine learning & Scientific: scikit-learn, xgboost, numpy,

Interpretability toolkit: SHAP

Visual: pyplot, seaborn

Structured data analysis: pandas

End user visualisation: Power BI

Data yang digunakan dalam pengembangan model *forecasting* indikator utama mencakup data:

1. Kebijakan, seperti suku bunga acuan dan kebijakan makro-prudensial.
2. Korporasi, termasuk pertumbuhan pengeluaran modal (*Capital Expenditures*), laporan keuangan perusahaan dan rasio keuangan.
3. Makroekonomi, seperti pertumbuhan ekonomi, inflasi, nilai tukar, dan kondisi perdagangan internasional.
4. Perbankan, seperti alat likuid dan dana pihak ketiga, pertumbuhan kredit, rasio kredit bermasalah, serta suku bunga kredit dan deposito.
5. Rumah tangga, seperti indeks penghasilan, ketersediaan lapangan kerja, konsumsi serta utang rumah tangga.
6. Survei dan *sentiment*, seperti indeks keyakinan konsumen serta *sentiment* kredibilitas kebijakan makroprudensial.

Early Warning Model (EWM) Likuiditas Perbankan

Pengembangan *Early Warning Model* (EWM) Likuiditas Perbankan dilatarbekangi oleh kebutuhan akan alat prediktif dalam pengawasan untuk mendukung kerangka *Dynamic Systemic Risk Surveillance* (DSRS), khususnya dalam memantau risiko likuiditas perbankan. Model ini diharapkan mampu mendeteksi tekanan likuiditas pada masing-masing bank, terutama pada tahap awal sebelum krisis berkembang, sehingga dapat memberikan sinyal dini terhadap potensi permasalahan likuiditas di masa mendatang dan memungkinkan penyusunan langkah mitigasi yang tepat.

Box 6.4 Tools Pengembangan

Data storage: Hadoop HDFS/Excel.

Analytics Tools:

Programming Language: Python

Libraries:

Machine learning & Scientific: scikit-learn, xgboost, numpy,

Interpretability toolkit: SHAP

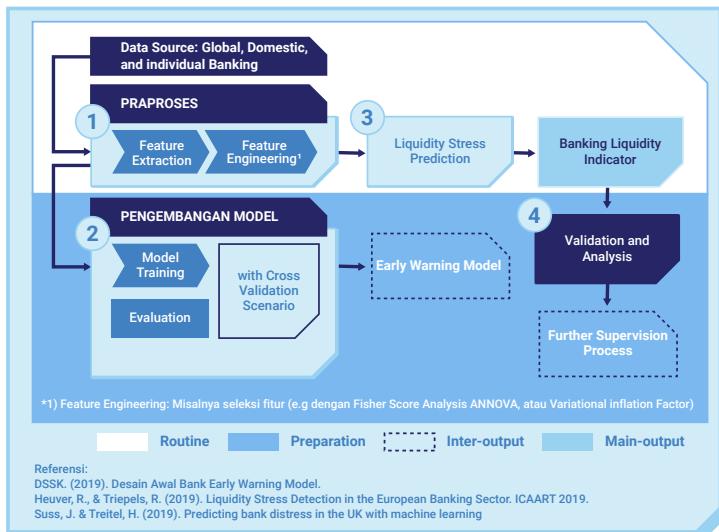
Visual: pyplot, seaborn

Structured data analysis: pandas

Secara umum, metodologi EWM ini dapat diilustrasikan pada Gambar 56. Data yang digunakan dikategorikan ke dalam tiga kelompok utama, yaitu indikator global, indikator domestik, dan indikator spesifik perbankan (*bank-specific*). Sumber data berasal dari berbagai platform, termasuk CEIC, publikasi statistik BI (SEKI/SSKI), dan BI-ANTASENA. Untuk melatih model *machine learning*, data diberi label *stress/non-stress* secara manual oleh pengawas berdasarkan kriteria seperti *run-off* Dana Pihak Ketiga (DPK), pertumbuhan DPK, atau analisis peristiwa tertentu.

Langkah-langkah data *pre-processing* mencakup:

1. Imputasi missing value menggunakan metode *forward* dan *backward filling*.
2. Mengatasi ketidakseimbangan data dengan undersampling untuk kondisi *non-stress* dan *oversampling* untuk kondisi *stress* menggunakan *Synthetic Minority Oversampling Technique* (SMOTE).
3. Penambahan variabel tambahan, seperti klasifikasi bank berdasarkan modal inti (KBMI), kategori bank sistemik, serta periode krisis (misalnya Global Financial Crisis 2007–2008 dan pandemi COVID-19 2020–2022), yang diasumsikan berpengaruh terhadap pola likuiditas bank.
4. Penyesuaian lag data dengan variasi 1 hingga 4 triwulan untuk meningkatkan akurasi prediksi.

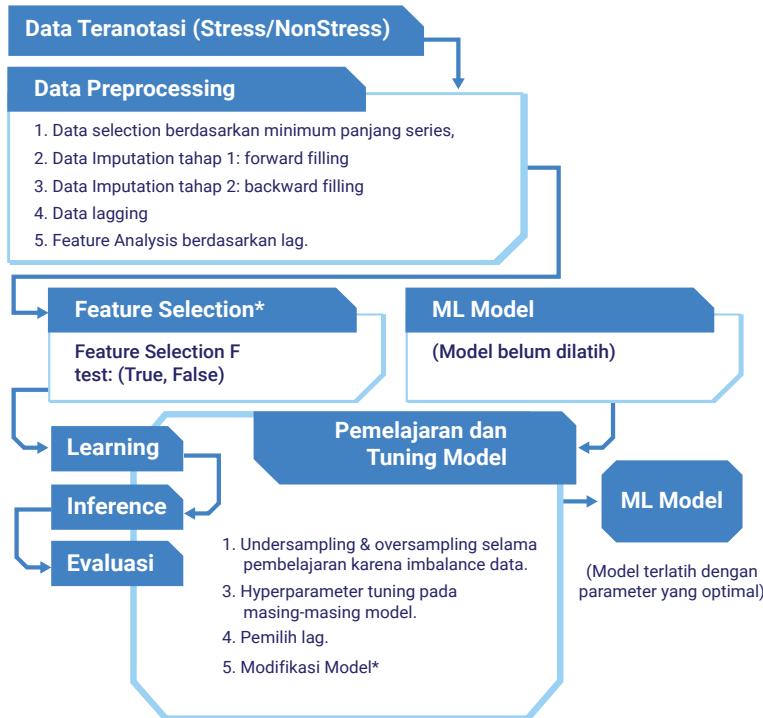


Gambar 56 Alur Use Case EWM

Feature selection dilakukan untuk memilih variabel utama yang akan digunakan untuk meningkatkan performa model dengan metode:

1. **Analysis of Variance (ANOVA)** – Menggunakan *F-test* untuk mengukur signifikansi variabel dalam membedakan kondisi *stress/non-stress*.
2. **Shapley Value Analysis** – Menilai kontribusi masing-masing variabel terhadap performa model berdasarkan nilai marginal rata-rata.

Sebagaimana pada Gambar 57, model dikembangkan dikembangkan menggunakan teknik *supervised machine learning (classification)* dengan algoritme Decision Tree, Logistic Regression, Support Vector Machine (SVM), dan Extreme Gradient Boosting (XGBoost). Proses hyperparameter tuning dilakukan untuk mendapatkan konfigurasi optimal, misalnya pada model XGBoost dilakukan pencarian parameter seperti splitting criteria, depth, learning rate, dan regularisasi.



Gambar 57 Alur Pengembangan Model EWM

Evaluasi model dilakukan menggunakan metriks *classification*, termasuk *recall*, *precision*, *accuracy*, *F1-score*, dan ROC-AUC. Mengingat adanya ketidakseimbangan data serta prioritas DSSK untuk meminimalkan *false negative* (bank yang diprediksi *non-stress* padahal sebenarnya *stress*), maka *recall* dan ROC-AUC dipilih sebagai metriks utama. Setelah melakukan ±388 percobaan menggunakan *stratified cross-validation*, algoritme XGBoost dengan lag 3 triwulan menunjukkan performa terbaik dengan *recall* 67% dan ROC-AUC 80%, sehingga direkomendasikan sebagai model utama untuk mendeteksi kerentanan likuiditas perbankan. Secara umum, metodologi ini mengadopsi penerapan **machine learning** di BoE (Suss & Treitel, 2019) untuk kasus yang serupa.

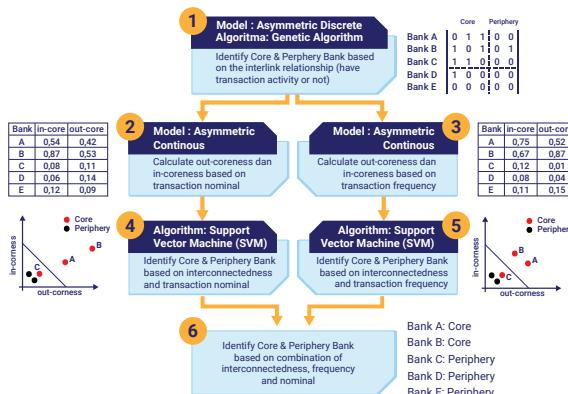
6.3 AI for Payment System

Sebagian besar transaksi ekonomi saat ini sudah dilakukan secara *cashless*, baik dalam bentuk pembayaran ritel seperti transaksi berbasis kartu, QR, dan internet, maupun transaksi bernilai besar melalui *Real Time Gross Settlement* (RTGS). Tidak seperti pembayaran ritel yang bernilai kecil dengan frekuensi tinggi, transaksi melalui sistem RTGS memiliki nilai besar dengan frekuensi rendah, meskipun demikian RTGS memiliki peran krusial dalam menjaga stabilitas keuangan mengingat hampir 90% total transaksi pembayaran di Indonesia sehingga dapat dikategorikan sebagai sistem pembayaran nasional yang memiliki peranan signifikan (*Systemically Important Payment System*).

Krisis keuangan global menjadi pengingat akan pentingnya memahami interkoneksi dalam sistem keuangan. Jaringan keuangan sering kali memiliki sejumlah kecil pemain utama yang sangat terhubung satu sama lain. Namun, gangguan pada salah satu pemain ini dapat menimbulkan risiko bagi seluruh jaringan. Pemahaman mengenai bagaimana guncangan likuiditas menyebar dalam pasar antarbank sangat bergantung pada topologi jaringan, yang dapat dibangun menggunakan data sistem pembayaran.

Beberapa studi menunjukkan bahwa sistem pembayaran antarbank dan pasar antarbank cenderung memiliki struktur *core-periphery*, di mana bank beroperasi dalam hierarki. Jaringan ini dapat dimanfaatkan untuk memprioritaskan pemantauan likuiditas pada bank yang diidentifikasi sebagai inti (*core*) serta menganalisis dampak penyebaran risiko (*contagion effect*).

Untuk mengidentifikasi struktur *core-periphery* dalam jaringan pembayaran, pada studi ini, sebagaimana diilustrasikan pada Gambar 58, pendekatan machine learning dikembangkan dengan memanfaatkan algoritma Model Asymmetric Discrete, Model Asymmetric Continuous, dan Support Vector Machine (SVM).



Gambar 58 Alur Metodologi Core-Periphery

Preprocessing Data

1. Menyaring peserta non-bank dalam BI-RTGS.
2. Mengelompokkan cabang bank ke dalam bank induknya.
3. Menganonimkan nama bank dengan kode tertentu.
4. Mengonversi data ke dalam matriks biner (0/1) dan matriks kontinu (berbasis nominal transaksi dan frekuensi).

Ekstraksi Data

1. Model Asimetris Diskrit: Menggunakan algoritme genetika untuk menentukan struktur *core-periphery* berdasarkan koneksi antarbank (1: terhubung, 0: tidak)
2. Model Asimetris Kontinu: Menghitung nilai *out-coreness* dan *in-coreness* untuk menilai kekuatan hubungan antar bank.
3. Model *Machine Learning* (SVM): Menggabungkan hasil dari dua model sebelumnya untuk mengklasifikasikan bank sebagai *core* atau *periphery* berdasarkan pola transaksi.

Box 6.5 Tools Pengembangan

Data storage: Hadoop HDFS

Analytics Tools:

Programming Language: Python

Libraries:

Skicit-learn, asymmetric discrete, asymmetric continuous

Support vector machine (SVM)

Graph & Network analysis: NetworkX, igraph

Visual: pyplot, seaborn

Structured data analysis: pandas

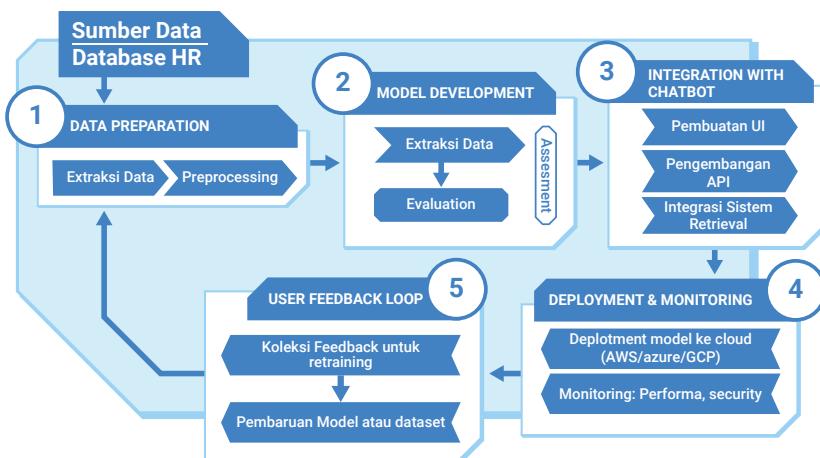
Dalam jaringan BI-RTGS, terdapat 10 bank yang secara konsisten dikategorikan sebagai bank inti (*core banks*) selama 24 bulan berdasarkan klasifikasi *Support Vector Machine* (SVM). Sesuai dengan model *core-periphery* Craig & Peter (2010), bank inti ini saling terhubung dan berfungsi sebagai perantara bagi bank-bank di level bawah (*periphery*), menjadikannya bank tier atas dalam struktur jaringan.

Meskipun penelitian ini tidak menemukan struktur *core-periphery* yang sepenuhnya sesuai dengan model ideal Craig & Peter—misalnya, tidak ada bank *periphery* yang hanya terhubung dengan sesama *periphery*—hasil analisis menunjukkan bahwa struktur *core-periphery* dalam BI-RTGS cukup stabil sepanjang periode observasi.

6.4 AI for HR

HR Super Apps (GenAI Platform)

Dalam konteks *Human Resources* (HR), penggunaan *Artificial Intelligence* (AI) dapat dimanfaatkan melalui berbagai cara salah satunya HR Super Apps. Salah satu teknologi kunci dalam HR Super Apps ini adalah *Generative AI*, atau yang lebih dikenal sebagai *Chatbot*. *Chatbot* HR memungkinkan pegawai untuk bertanya seputar berbagai hal terkait HR, mulai dari peraturan di lingkungan kantor, kebijakan manajemen karier, pengembangan kompetensi pegawai, informasi cuti, tunjangan, prosedur pengajuan dokumen, dan lain-lain. Dengan adanya *Chatbot* berbasis Generative AI, pegawai dapat mengakses informasi HR secara cepat, akurat, dan mandiri. Hal ini tidak hanya meningkatkan efisiensi operasional HR, tetapi juga meningkatkan pengalaman pegawai dalam berinteraksi dengan sistem HR. Untuk membangun chatbot HR yang efektif tersebut, diperlukan beberapa tahapan sebagaimana pada Gambar 59 berikut.



Gambar 59 Alur Metodologi dalam GenerativeAI Platform

1. Data Collection & Preparation

Fase ini merupakan fondasi dari *GenAI Platform*. Data dikumpulkan dari berbagai sumber, seperti *database* HR (dokumen kebijakan, dokumen ketentuan, FAQ, dll.) dan data eksternal. Sebelum digunakan, data harus diproses untuk memastikan kualitas dan kepatuhan terhadap regulasi privasi. Proses ini meliputi anonimisasi data sensitif, pembersihan data, dan pembuatan struktur yang sesuai.

2. Model Development

Setelah data siap, langkah berikutnya adalah pengembangan model GenAI. Pada fase ini, model seperti GPT-4, Llama 2, atau model *open-source* lainnya di-assess untuk menemukan model terbaik. Pengujian dan *prompt engineering*

juga dilakukan agar model dapat menyesuaikan terhadap berbagai skenario HR. Evaluasi dilakukan untuk memastikan kesesuaian model, relevansi jawaban, dan mitigasi bias. Hasilnya adalah model GenAI yang siap diintegrasikan dengan sistem *Chatbot*.

3. *Integration with Chatbot*

Model GenAI yang sudah siap kemudian diintegrasikan dengan antarmuka *Chatbot*. Ini melibatkan pengembangan API yang menghubungkan model dengan *frontend Chatbot*. API bertanggung jawab untuk menerima pertanyaan dari pengguna, memprosesnya menggunakan model GenAI, dan mengembalikan jawaban yang sesuai. Selain itu, sistem *Retrieval Augmented Generation (RAG)* diimplementasikan untuk memungkinkan chatbot mengakses dokumen atau file yang relevan dari *knowledge base*. Integrasi ini memastikan chatbot tidak hanya memberikan jawaban teks, tetapi juga bisa merujuk ke dokumen resmi yang tersedia.

4. *Deployment & Monitoring*

Setelah integrasi selesai, sistem siap di-deploy ke lingkungan produksi, biasanya menggunakan platform *cloud* seperti AWS, Azure, atau *Google Cloud*. *Deployment* ini dirancang untuk memastikan *Chatbot* dapat menangani banyak permintaan secara bersamaan. Setelah *deployment*, sistem terus dipantau untuk memastikan performa optimal. *Monitoring* mencakup pengecekan kecepatan respons, akurasi jawaban, dan keamanan sistem.

5. *User Feedback Loop*

Fase terakhir adalah *User Feedback Loop*, di mana umpan balik dari pengguna dikumpulkan dan dianalisis. *Feedback* ini bisa berasal dari komentar langsung dari pengguna atau analisis interaksi pengguna. Umpan balik ini sangat penting untuk mengidentifikasi beberapa perbaikan, seperti jawaban yang kurang akurat atau fitur yang kurang intuitif. *Feedback* kemudian bisa digunakan untuk perluasan atau perbaikan dataset, melakukan *finetuning* model, atau meningkatkan antarmuka *Chatbot*. Proses ini memastikan sistem terus berkembang dan tetap relevan dengan kebutuhan HR.

Box 6.6 Tools Pengembangan

Data storage: PostgreSQL, AWS S3, Azure SQL Database, Azure Cosmos DB, Azure AI Search

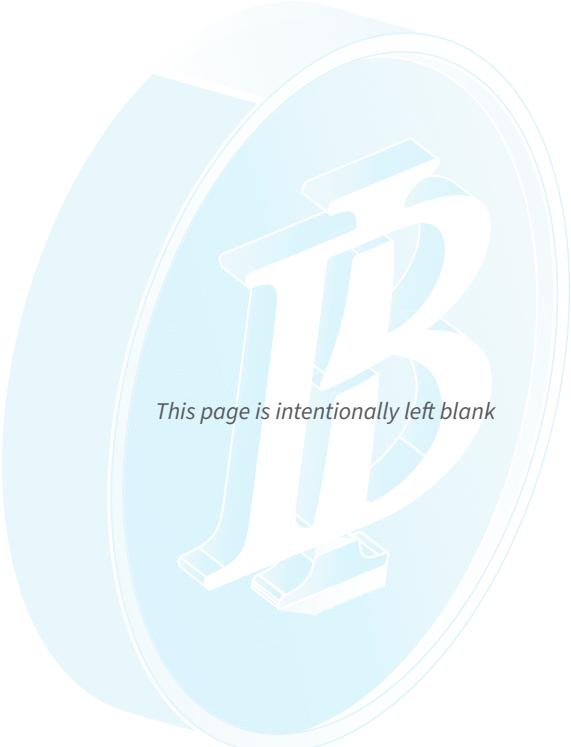
Analytics Tools:

Programming Language: Python, SQL, JavaScript
Libraries:

Hugging Face Transformers, spaCy, TensorFlow, FastAPI

Azure Specific Tools & Libraries:

Azure Open AI Service (Akses model GPT-4, GPT-3.5, dll)
Azure Cognitive Services (Text Analytics, Doc. Intelligence)



This page is intentionally left blank

BAGIAN 7

ADVANCED TOPICS ON MACHINE LEARNING

Pushing the Boundary

“..you can not do inference without making assumptions..” – David MacKay

7.1 Bayesian Learning

Sebelum membahas lebih dalam, mari kita coba simak contoh permasalahan:

Misalnya terdapat suatu penyakit langka dalam populasi, dengan probabilitas seorang sakit adalah 10 dari 100 orang. Sebanyak 40% pada populasi tersebut mengalami memang gejala penyakit yang dimaksud meski tidak seluruhnya benar-benar positif. Sementara itu, 95% orang yang benar-benar sakit menunjukkan adanya gejala. Apabila seseorang dari populasi tersebut datang dan mengalami gejala, sebenarnya berapa peluang benar-benar positif menderita penyakit yang dimaksud?

Untuk menghitungnya secara matematis, terdapat *prior information* yang kita miliki, yaitu $p(\text{sakit}) = 0,1$. *Prior information* ini menjadi salah satu ciri perspektif Bayesian dalam penghitungan *probability*. *Prior* tersebut dielaborasikan dengan *likelihood*, melalui formulasi:

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{normalisation constant}} \quad (7.1)$$

Dalam hal ini,

- *prior* = $p(\text{sakit}) = 0,1$
- *likelihood* = $p(\text{gejala}|\text{sakit}) = 0,95$
- *normalisation constant* = $p(\text{gejala}) = p(\text{gejala} | \text{sakit}) + p(\text{gejala} | \text{tidak sakit}) = 0,4$

dengan demikian, peluang orang tersebut menderita penyakit yang dimaksud adalah:

$$p(\text{sakit}|\text{gejala}) = \frac{p(\text{gejala}|\text{sakit})p(\text{sakit})}{p(\text{gejala})}$$

$$p(\text{sakit}|\text{gejala}) = \frac{0,95 \times 0,1}{0,4} = 0,2375$$

Bayesian Learning merupakan suatu konsep atau paradigma belajar mesin dengan mengimplementasikan teorema Bayes. Singkatnya, paradigma ini melakuakan estimasi dengan mempertimbangkan **prior knowledge**. Secara matematis, *Bayesian learning* memanfaatkan formulasi sebagai berikut:

$$p(w|X, y) = \frac{P(X, y|w) \cdot P(w)}{P(X, y)} = p(w|X, y) = \frac{P(X, y|w) \cdot P(w)}{\int P(X, y|w) \cdot P(w) dw} \quad (7.2)$$

di mana

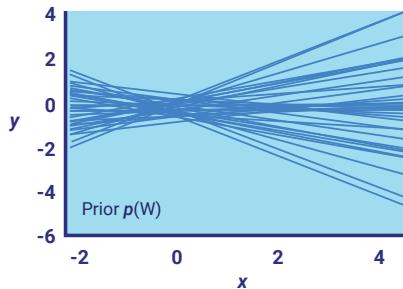
- w : point of interest (yang akan diinferensi)
- X, y : data
- $P(w | X, y)$: Posterior distribution
- $P(w)$: Prior
- $P(X, y | w)$: Likelihood
- $P(X, y)$: Konstanta normalisasi agar $p(X | y)$ adalah distribusi yang valid (sum to 1).

Pada contoh linear (*in parameter*) regression di atas (subbab 2.1), bentuk Bayesian kemudian dapat diubah ke dalam perspektif probabilistik menjadi Bayesian (*posterior*) prediction sebagai berikut.

$$p(y^* | D) = \int p(y^* | x^*, w) \cdot p(w | D) dw \quad (7.3)$$

Dengan $D = \{(x, y)\}$ merupakan *training set*, x^* data baru dan distribusi $p(w | D)$ diperoleh melalui *learning* sebagaimana pada ekspresi 7.2 di atas.

Beberapa grafik berikut ini mengilustrasikan bagaimana proses *learning (function fitting)* melalui *Bayesian framework*.

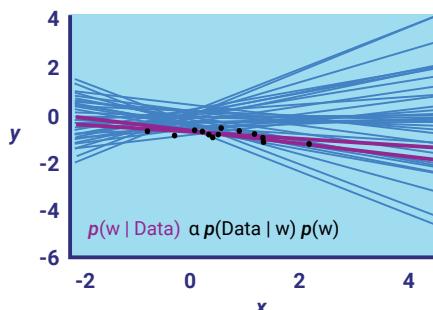


Gambar 60 Representasi Prior distribution dari w . (img src: Arno Onken/Iain Murray)

Asumsi yang digunakan untuk distribusi pada Gambar 60 di atas adalah:

$$p(\mathbf{w}) = N(\mathbf{w}; 0, 0.4^2 I) \quad (7.4)$$

Seiring bertambahnya data latih $D = (X, y)$, maka kemungkinan nilai w akan mengerucut ke suatu area (simak Gambar 61). Proses *Bayesian fitting* ini memberikan ruang untuk relaksasi pada saat analisis, sehingga dapat mengantisipasi adanya *overconfident*.



Gambar 61 Representasi Posterior Distribution dari w , setelah mempertimbangkan observed data
(img src: Arno Onken/Iain Murray)

Contoh lain, misalnya **Naïve Bayes Classifier/Regressor** yang merupakan sebuah metode klasifikasi/regresi yang berkerangka *probabilistic* dan berdasar pada teorema *Bayes*. Ciri utama dari *Naïve Bayes Classifier* ini adalah asumsi yg sangat kuat (*naïf*) akan independensi dari masing-masing kondisi/kejadian. Dalam konteks *machine learning*, *independent/naïf* di sini adalah bahwa setiap fitur *independent* satu sama lain. Selain itu, salah satu komponen yang cukup penting dalam *Bayesian machine learning* adalah adanya fasilitas untuk menguantifikasi *prior probability* dalam proses inferensi/*learning*.

$$p(y|X) = \frac{P(X|y) \cdot P(y)}{P(X)} \quad (7.5)$$

Asumsi “*naïve*” *independent*:

$$P(X|y) = P(x_1|y) \cdot P(x_2|y) \dots P(x_d | y) \quad (7.6)$$

di mana

- y : point of interest (yang akan diinferensi)
- X : data
- $P(y|X)$: Posterior
- $P(y)$: Prior
- $P(X|y)$: Likelihood
- $P(X)$: Konstanta normalisasi agar $p(y | X)$ adalah distribusi yang valid (*sum to 1*).

7.2 Probabilistic Graphical Models

Probabilistic Graphical Models (PGMs) adalah kerangka kerja untuk memodelkan distribusi probabilitas multivariat yang menggabungkan teori probabilitas dan teori graf. PGM menggunakan representasi grafis untuk memvisualisasikan hubungan

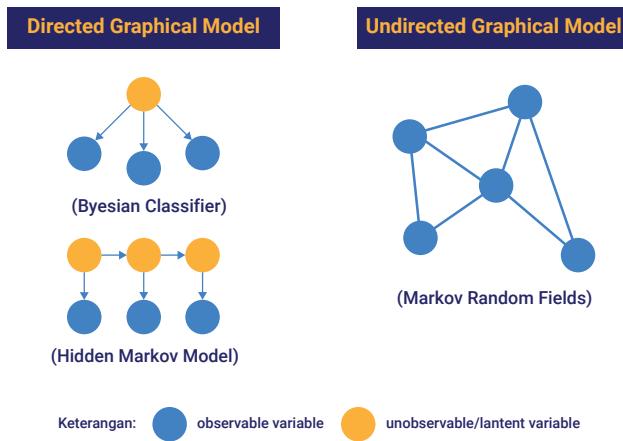
ketergantungan antarvariabel acak, di mana simpul mewakili variabel dan sisi mewakili hubungan di antara mereka.

PGM ini mengakomodasi pemanfaatan graf teori dan *probability theory*, sehingga memudahkan *interpretabilitas probabilistic* dalam setiap model. Hal ini penting karena *machine learning* secara umum dapat dipandang sebagai pendekatan fungsi yang bersifat probabilistik.

PGM sangat fleksibel karena inferensi dan pemelajaran dapat menggunakan struktur yang sama, terlepas dari *supervised* maupun *unsupervised* (Jordan, 1997). Selain itu, penanganan *missing data* juga dapat dimodelkan, dirunut dan diinterpretasikan. PGM dapat menjelaskan hubungan *conditional dependence* antarvariabel. Konsep yang mendasari PGMs mencakup:

1. **Conditional dependence:** Ketergantungan antara variabel dapat dengan mudah disimpulkan dari struktur graf.
2. **Factorization:** Distribusi probabilitas yang kompleks dapat difaktorkan menjadi produk dari distribusi yang lebih sederhana.

Secara umum, terdapat dua jenis PGM berdasarkan hubungan *dependency*, yaitu *directed* dan *undirected*, dengan ilustrasi sebagaimana pada Gambar 62.



Gambar 62 Ilustrasi Graphical Models

1. **Bayesian Networks (Directed Graphical Models):** Menggunakan graf berarah untuk mewakili hubungan kausal antarvariabel. Berikut ini representasi matematis dari *Directed graphical models*:

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | \text{parent}(X_i)) \quad (7.7)$$

2. ***Markov Networks (Undirected Graphical Models)/Markov Random Fields:***

Menggunakan graf tak berarah, biasanya digunakan untuk hubungan simetris. Berikut ini representasi matematis dari *undirected graphical models*:

$$P(X_1, X_2, \dots, X_n) = \frac{1}{Z} \prod_c \psi_c(X_c) \quad (7.8)$$

Dengan $\psi_c(\cdot)$ merupakan *potential function* yang memiliki nilai positif dan merepresentasikan hubungan interaksi antarnode dalam suatu *clique*. Z adalah *normalization constant* untuk memastikan $P(\cdot)$ merupakan distribusi yang valid (sum to 1).

Inference Methods

- ***Exact Inference:*** Metode eksak dilakukan dengan penghitungan probabilitas melalui metode seperti *Variable Elimination* atau *Junction Tree Algorithm*. Namun, metode ini menjadi sangat sulit dihitung ketika jumlah variabel atau kompleksitas graf meningkat.
- ***Approximate Inference:*** Metode perkiraan (*approximation*) seperti *variational inference* atau MCMC (*Monte Carlo Markov Chain*) dan *Loopy Belief Propagation* digunakan untuk inferensi secara efisien ketika inferensi eksak terlalu mahal secara komputasi.

PGM digunakan secara luas dalam *machine learning*, AI, dan statistik untuk mempermudah memahami hubungan kompleks antarvariabel. Contoh lain bentuk PGM yang populer adalah *conditional random fields* (CRF) dan *Restricted Boltzmann Machine* (RBM).

7.3 Mixture Models

Mixture Models adalah metode dalam ***Machine Learning*** yang digunakan untuk memodelkan data yang berasal dari beberapa distribusi berbeda. Salah satu model yang paling umum adalah ***Gaussian Mixture Model (GMM)***, yang digunakan untuk melakukan ***clustering, density estimation, and anomaly detection***.

probabilitas suatu data X muncul dihitung sebagai kombinasi dari beberapa distribusi:

$$P(X) = \sum_{i=1}^k \pi_i P(X | \theta_i) \quad (7.9)$$

di mana

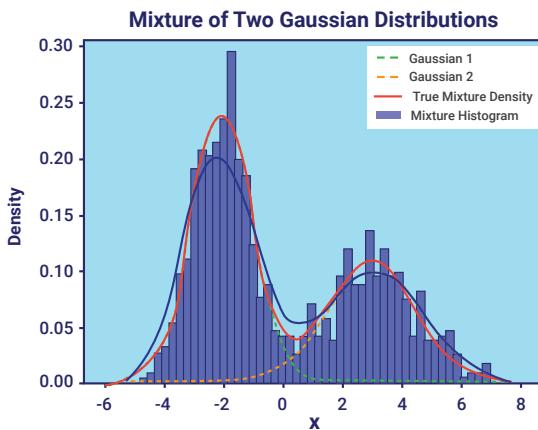
- k adalah jumlah komponen di dalam *mixture*.
- π_i adalah proporsi (*weight*) dari komponen i (asumsi $\sum \pi_i = 1$) merepresentasikan distribusi probabilitas dari komponen i dengan parameter θ_i, w_i

- θ_i merepresentasikan *mean* dan *covariance* dari setiap distribusi (terutama dalam GMM)

Apabila distribusi yang digunakan adalah *Gaussian (Normal Distribution)*, maka model disebut dengan *Gaussian Mixture Model (GMM)*.

$$\begin{aligned} z^{(n)} &\sim \text{Discrete}(\pi) \\ p(x^{(n)} | z^{(n)} = k, \theta) &= N(x^{(n)}; \mu^{(k)}, \Sigma^{(k)}) \\ \log p(D|\theta) &= \sum_{n=1}^N \log p(x^{(n)} | \theta) \end{aligned} \quad (7.10)$$

Contoh ilustrasi mixture model sederhana berupa *Gaussian Mixture Models (GMM)* dapat dilihat sebagaimana pada Gambar 63 berikut.



Gambar 63 Contoh ilustrasi mixture model

GMM dapat digunakan untuk *clustering (unsupervised learning)*, misalnya untuk Gambar 72 di atas, data terdiri atas 2 cluster dengan masing-masing cluster memiliki parameter mean μ_1 dan σ_1^2 untuk cluster 1 serta μ_2 dan σ_2^2 untuk cluster 2.

Pada prinsipnya tidak ada solusi eksak untuk estimasi parameter *mixture model*. Namun demikian, salah satu algoritma yang umum digunakan adalah *expectation-maximisation (EM)*. Sederhananya, langkah-langkah EM algorithm dalam GMM adalah sebagai berikut.

1. Inisiasi: tentukan jumlah cluster k dan nilai awal parameter (π, μ, σ)
2. *Expectation step (E-step)*: Hitung probabilitas bahwa setiap data berasal dari setiap distribusi Gaussian, mengikuti *Bayes theorem*.

3. *Maximization Step* (M-Step): Perbarui nilai parameter π, μ, σ berdasarkan probabilitas yang dihitung di E-step.
4. Iterasi: ulangi langkah 2 dan 3 hingga konvergen.

7.4 Ensemble Models

Dalam *machine learning*, penggabungan beberapa model dapat memberikan efek yang positif terhadap performa model, mengantisipasi *underfitting* dan *overfitting* sekaligus. Secara filosofis, menggabungkan beberapa model di samping meningkatkan performa, juga dapat menghindari adanya *overcertain* atas suatu prediksi.

Secara matematis, ensemble model dapat direpresentasikan sebagai posterior prediction dalam framework Bayesian posterior prediction

$$p(y|x, D) = \int p(y|x, w)p(w|D)dw \approx \frac{1}{S} \sum_{s=1}^S p(y|x, w_s) \quad (7.11)$$

dengan

$$w^s \sim p(w|D)$$

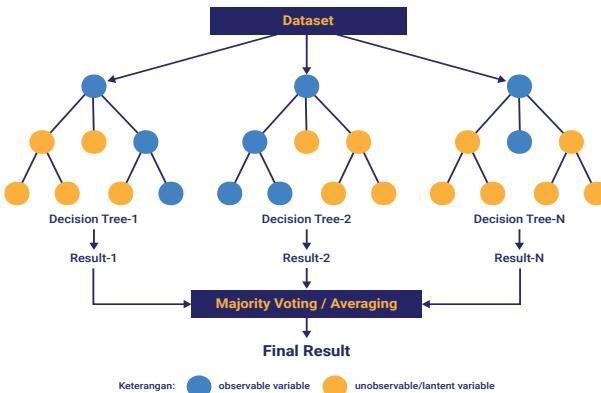
Dalam hal ini masing-masing w^s dapat dipandang sebagai model tersendiri, meskipun berasal dari distribusi yang sama (iid).

Jika diformulasikan secara lebih general, maka secara matematis *ensemble* ini dapat mengakomodasi “*mixture of experts*” sebagaimana pada ekspresi berikut.

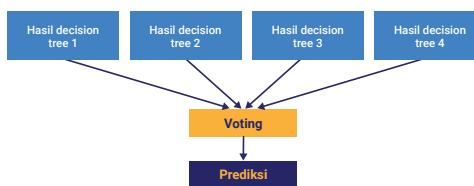
$$p(y|x, \theta) = \sum_{k=1}^K p(y|x, z=k, \theta) p(z=k|\theta) \quad (7.12)$$

Contoh *ensemble model* yang umum misalnya adalah ***random forest***. Model ini membutuhkan gabungan beberapa *decision tree* untuk memprediksi hasil yang akurat. Algoritma ensemble ini digunakan baik untuk tujuan klasifikasi dan regresi. ‘Forest’ atau ‘hutan’ mereferensikan kumpulan *decision trees* yang tidak berkorelasi, yang kemudian digabungkan bersama untuk mengurangi varians dan membuat prediksi data yang lebih akurat. Hal ini diilustrasikan sebagaimana pada Gambar 64.

Konsep sederhana dari *random forest* adalah beberapa *decision tree* yang tidak berkorelasi akan bekerja lebih baik sebagai kelompok dibandingkan individu. Saat menggunakan *random forest* sebagai pengklasifikasi, satu *decision tree* menyumbang satu suara. Setiap *decision tree* bisa menghasilkan jawaban yang sama atau berbeda satu sama lain. Hal ini diilustrasikan sebagaimana pada Gambar 65. Misalnya *decision tree* 1, 2, dan 3 memprediksi hasil 1. Sementara *decision tree* 4 dan D memprediksi hasil 0.



Gambar 64 Ilustrasi Random Forest secara umum



Gambar 65 Ilustrasi Random Forest untuk Klasifikasi

Oleh karena terdapat banyaknya alternatif jawaban dalam *decision tree* dan kemungkinan bias yang tinggi, *random forest* mengambil prediksi hasil dari beberapa *decision tree* berdasarkan suara mayoritas dan memprediksi hasil yang lebih akurat.

7.5 Gaussian Processes

Gaussian Process merupakan model statistik non-parametrik berbasis Bayesian. Artinya, model ini tidak memiliki jumlah parameter yang pasti, melainkan bergantung pada data secara langsung. Model ini dapat digunakan untuk *use case* regresi dan klasifikasi. Lalu, apa perbedaan dengan model lainnya?

Pada regresi linear, model mengasumsikan fungsi linear dan melakukan optimisasi untuk mendapatkan parameter optimal, di mana proses menjadi lebih rigid. Hal ini menegaskan bahwa metodologi *Gaussian Process* menjadi lebih fleksibel karena dapat memodelkan berbagai macam *function*. Hal ini berarti bahwa jika kita memiliki sekumpulan titik data, kita dapat membuat prediksi pada titik-titik baru dengan mempertimbangkan hubungan antara titik-titik tersebut menggunakan fungsi kovarians (kernel).

Secara umum, *Gaussian Process* hanya ditentukan oleh dua *function*, di antaranya adalah:

1. Fungsi rata-rata (*mean function*): $\mu(x)$. Jika dataset sudah dinormalisasi, maka dapat diasumsikan nol.
2. Fungsi Kernel (*Kovariansi*): $\kappa(x, x')$. Dalam kasus ini, fungsi kernel berperan sebagai *constraint* dan dapat digunakan untuk mengukur kemiripan atau hubungan antara dua titik *input* x dan x' . Dengan kata lain, Kernel juga berperan untuk menentukan sifat fungsi yang akan dimodelkan. Contoh Kernel yang umum digunakan adalah:
 - a. *Radial Basis Function (RBF)* → Cocok untuk fungsi dengan karakteristik *smoothness* yang tinggi
 - b. *Matern* → Lebih fleksibel dan cocok untuk data dengan perubahan tajam.
 - c. *Linear/Polinomial* → digunakan untuk model yang memiliki hubungan linear antara *input* dan *output*.

Dalam regresi menggunakan *Gaussian Process*, langkah utama adalah:

1. **Menentukan kernel yang sesuai** untuk menangkap pola dalam data.
2. **Menghitung matriks kovarians** berdasarkan kernel dan data *training*.
3. **Memprediksi nilai baru** dengan menghitung distribusi posterior menggunakan formula:

$$f_* | X, y, X_* \sim N(\mu_*, \Sigma_*) \quad (7.13)$$

di mana f_* adalah prediksi pada *data testing* X_* , dengan nilai rerata μ_* dan kovariansi Σ_* .

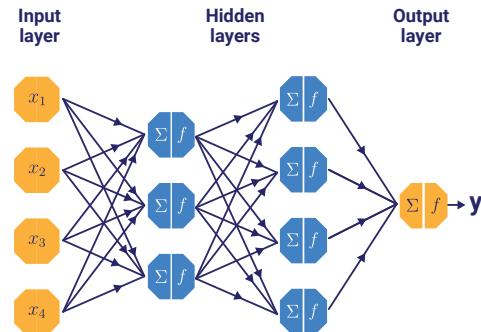
Gaussian Process memiliki beberapa keunggulan dibandingkan model lainnya, di antaranya adalah mampu menangani data dengan jumlah sampel yang kecil, memberikan estimasi yang bersifat probabilistik, serta sangat fleksibel dalam menangani *pattern* non-linear. Sayangnya, terdapat kekurangan berupa kompleksitas dan kebutuhan komputasi yang tinggi, ketika harus melakukan proses invers matriks kovarians. Selain itu, pemilihan kernel harus sesuai dengan karakteristik data, karena apabila tidak, akan menyebabkan model *overfit* maupun *underfit*.

7.6 Neural Networks

Neural network terinspirasi dari otak manusia dan meniru cara kerja sel saraf (*neuron*) dalam mengirim pesan satu sama lain. Pada manusia, *neuron* menerima sinyal atau rangsangan dari luar kemudian mengirimkannya ke otak untuk diproses. Setelahnya tubuh akan bertindak sesuai perintah yang dikirimkan otak melalui *neuron*.

Neuron manusia terdiri atas beberapa bagian, begitu juga dengan *neural network* yang terdiri atas beberapa lapisan *node* (*neuron* buatan), yaitu lapisan *input*, satu atau lebih lapisan tersembunyi, dan lapisan *output*. Hal ini membuat *neural network* dapat melakukan perhitungan yang jauh lebih kompleks (Goodfellow, et al, 2016).

Setiap *node* saling terhubung untuk menerima dan mengirimkan informasi. Sama seperti cara kerja *neuron* dalam mengirimkan informasi ke otak dan tubuh merespons sesuai dengan perintah dari otak.

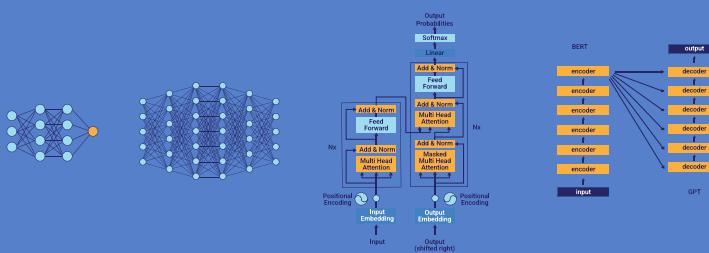


Gambar 66 Neural Network Regression Model sederhana

Gambar 66 di atas merepresentasikan sebuah *neural network* model untuk regresi. Setiap *node* atau *neuron* direpresentasikan dengan fungsi aktivasi (*activation function*) dan antar-*node* saling terhubung dengan *bobot*.

Box 7.1 From Neural Network to Deep Learning to ChatGPT

- Saat ini, *neural network* merupakan fondasi struktur yang digunakan pada *deep learning*. Salah satu variasi atau *advancement deep learning* adalah LLM (a.l. BERT, Llama), serta transformer yang kini digunakan sebagai arsitektur pada ChatGPT.
- Adanya akses terhadap *big data* dan sumber daya komputasi yang besar memberikan peluang *neural network* untuk belajar dengan baik sehingga dapat diaplikasikan pada berbagai hal, a.l. data teks, *visual*, *signal* dan lainnya.



Neural Network

Deep Learning

Transformer

BERT VS GPT

Saat ini, *neural network* merupakan model yang cukup popular yang memungkinkan pemelajaran secara mendalam (*Deep Learning*) (Lecun *et al*, (2015), Scmidhuber (2015), Chollet (2017)). Dengan kata lain, *deep learning* yang merupakan salah satu metode penerapan *machine learning* memang menggunakan *neural network* dalam memproses data.

Mekanisme **learning** dalam **neural network** adalah proses di mana jaringan saraf buatan menyesuaikan bobot dan biasnya untuk meminimalkan perbedaan antara prediksi dan hasil sebenarnya. Proses ini berlangsung secara iteratif dengan menggunakan algoritma optimasi, seperti **gradient descent**. Berikut adalah penjelasan langkah-langkahnya.

1. Inisialisasi

- Bobot (*weights*) dan bias di inisialisasi secara acak atau menggunakan metode khusus seperti inisialisasi Xavier atau He untuk memastikan proses pelatihan berjalan lancar tanpa masalah seperti *vanishing gradient*.

2. Forward Propagation

- *Input* data dimasukkan ke jaringan melalui lapisan-lapisan yang ada.
- Setiap lapisan melakukan operasi matematis (seperti perkalian matriks dan penambahan bias), kemudian hasilnya dilewatkkan melalui fungsi aktivasi (misalnya ReLU atau sigmoid) untuk menambah non-linearitas.
- Lapisan terakhir memberikan hasil prediksi berdasarkan input yang telah diproses.

3. Hitung error dengan Fungsi Loss (*Loss Function*)

- Fungsi *loss* digunakan untuk mengukur seberapa jauh hasil prediksi jaringan dari nilai sebenarnya. Contoh fungsi *loss* adalah:
 - **Mean Squared Error (MSE)** untuk permasalahan regresi.
 - **Cross-Entropy Loss** untuk permasalahan klasifikasi.
- Hasil dari fungsi *loss* ini adalah nilai skalar yang menunjukkan total *error* jaringan.

4. Backpropagation

- *Backpropagation* adalah proses utama dalam *learning*. Proses ini menghitung **gradien** dari fungsi *loss* terhadap setiap bobot dan bias dalam jaringan.
- Menggunakan **aturan rantai** (*chain rule*) dalam kalkulus, jaringan menghitung bagaimana perubahan pada setiap bobot memengaruhi *error* total.
- Gradien tersebut digunakan untuk menentukan arah dan besar perubahan bobot agar *error* dapat diminimalkan.

Adapun beberapa hal terkait *learning* (pemelajaran) *Neural Network* mencakup:

1. Gradient Descent dan Optimisasi

- *Gradient descent* adalah teknik optimasi yang digunakan untuk memperbarui bobot dan bias berdasarkan gradien yang telah dihitung.
- Mekanisme pembaruan bobot:

$$W_{\text{baru}} = W_{\text{lama}} - \eta \cdot \nabla L(W) \quad (7.14)$$

di mana

- W_{baru} adalah bobot yang diperbarui.
- W_{lama} adalah bobot saat ini.
- η adalah ***learning rate*** (skala kecil yang mengontrol seberapa besar perubahan yang dilakukan).
- $\nabla L(W)$ adalah gradien dari *loss* terhadap bobot.

Optimizer seperti ***Stochastic Gradient Descent (SGD)***, ***Adam***, atau ***RMSProp*** digunakan untuk meningkatkan efisiensi dan stabilitas pelatihan.

2. Learning Rate

- ***Learning rate*** mengontrol seberapa besar langkah perubahan bobot di setiap iterasi. *Learning rate* yang terlalu kecil akan membuat proses pembelajaran lambat, sementara *learning rate* yang terlalu besar dapat menyebabkan jaringan tidak konvergen.

3. Proses Iteratif

- Proses *forward* dan *backward pass* ini diulang untuk setiap *batch data* dalam banyak iterasi (*epoch*). Jaringan akan terus memperbarui parameternya sampai prediksinya menjadi lebih akurat dari waktu ke waktu.

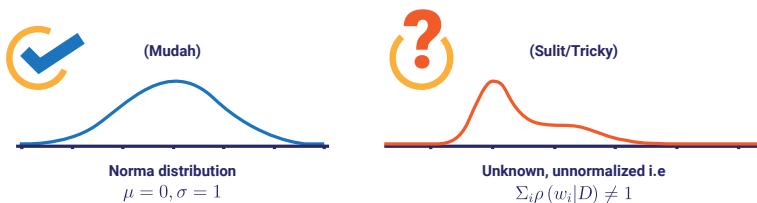
4. Konvergensi

- Proses *learning* akan berhenti ketika model mencapai konvergensi, yaitu ketika pembaruan bobot menghasilkan perubahan kecil pada *loss* atau ketika kriteria penghentian (misalnya, jumlah epoch atau ***early stopping***) tercapai.

Secara keseluruhan, mekanisme ***learning*** pada neural network adalah proses iteratif di mana jaringan memperbaiki bobotnya untuk memperkecil *error* antara prediksi dan data sebenarnya, melalui metode optimasi dan penyesuaian bobot secara bertahap.

7.7 Approximation: Markov Chain Monte Carlo vs. Variational Inference

Problem:



Gambar 67 Problem Inferensi

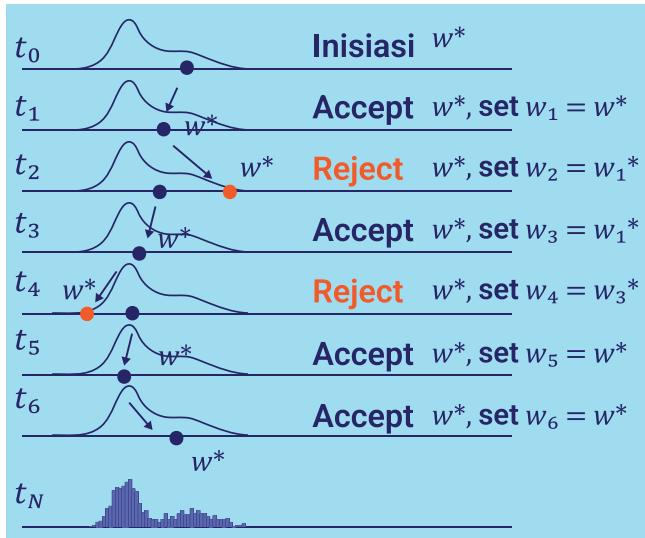
Dalam proses inferensi maupun *learning*, apabila model terlalu kompleks (ilustrasi pada Gambar 67) maka *sampling feasibility*, komputasi dan waktu menjadi isu yang perlu dipertimbangkan (MacKay, 2003). Sebagai konsekuensi, proses inferensi tersebut memerlukan mekanisme yang relatif efisien dengan tetap mempertimbangkan kualitas inferensi misalnya diukur dengan seberapa akurat proses prediksi. Bagian ini akan membahas dua jenis aproksimasi, yaitu aproksimasi yang bersifat *fully sampling*, serta aproksimasi yang bersifat deterministik.

Sebagai contoh inferensi untuk parameter:

$$p(w|X, y) = \frac{P(X, y|w) \cdot P(w)}{P(X, y)} = \frac{P(X, y|w) \cdot P(w)}{\int P(X, y|w) \cdot P(w) dw} \quad (7.15)$$

Dalam praktiknya, perhitungan konstanta normalisasi $p(X, y) = p(X, y | w) \cdot p(w) dw$ sering kali *intractable* karena melibatkan integrasi seluruh kemungkinan dari w . Dengan demikian, perlu trik tersendiri untuk menghitung posterior $p(w|X, y)$.

Markov Chain Monte Carlo (MCMC)



Gambar 68 Ilustrasi MCMC sederhana

Salah satu trik tradisional yang umum dilakukan adalah dengan meng-implementasikan Markov Chain Monte Carlo (MCMC), sebagaimana diilustrasikan pada Gambar 68. Singkatnya, MCMC ini berupaya menghasilkan sample yang dapat diterima (*acceptable*) dengan menyimulasikan *Monte Carlo methods* secara sekuenial mengikuti kaidah *Markov chain*. Kata kunci di sini yaitu “Monte Carlo” dan “Markov Chain”. Monte Carlo itu sendiri dapat diformulasikan sebagai:

$$\mathbb{E}[f(w)] \approx \frac{1}{N} \sum_{t=1}^N f(w_t) \quad (7.16)$$

Dalam hal ini, bisa jadi $f(w)$: $w \rightarrow w$.

Kemudian proses Monte Carlo tersebut dilakukan mengikuti kaidah Markov Chain:

$$p(w_t | w_{t-1}, w_{t-2} \dots w_1) = p(w_t | w_{t-1}) \quad (7.17)$$

Salah satu teknik MCMC yang cukup populer adalah Metropolis Hastings, dengan tahapan sebagai berikut.

1. Ambil sembarang nilai w_0
2. Sampling baru w^* dari proposal distribution, katakanlah q , sehingga $w^* \sim q(w^* | w_t)$.

3. Hitung *acceptance probability*,

$$A = \min\left(1, \frac{p(w^*|X,y)q(w_t|w^*)}{p(w_t|X,y)q(w^*|w_t)}\right) \quad (7.18)$$

4. *Accept/reject sample* tersebut:

- Accept w^* dengan probability A , dan set $w_{t+1} = w^*$
- Reject w^* dan set $w_{t+1} = w_t$

5. Ulangi tahapan 1 s.d. 4 sampai diperoleh sampel yang cukup banyak. Sampel dapat digunakan sebagai aproksimasi dari sampling $w \sim q(w|X,y)$.

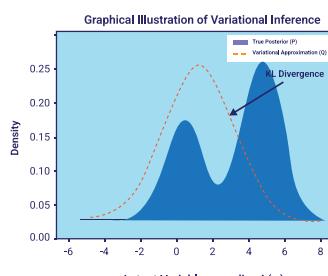
Variational Inference (VI)

Teknik yang kedua ini adalah *variational inference*, dengan mengimplementasikan optimisasi untuk meminimalkan *discrepancy* (*simak ilustrasi pada Gambar 69*). Dalam hal ini terdapat kata kunci penting, yaitu “*discrepancy*” dan “*optimisasi*”. Pertama, *discrepancy* atau *closeness* merepresentasikan seberapa dekat proposal *distribution* dengan *target distribution*, yang dalam hal ini dimanifestasikan dengan Kulback-Leibler (KL) *divergence*:

$$KL(q(w) \| p(w|D)) = \sum q(w) \log \frac{q(w)}{p(w|D)}; \quad (7.19)$$

Dengan $D=\{X,y\}$. Oleh karena menghitung $p(w|D)$ relatif sulit, maka perlu ada solusi lain yang digunakan agar optimisasi dapat dilakukan. Kita turunkan kembali ekspresi KL *divergence*:

$$\begin{aligned} KL(q(w) \| p(w|D)) &= E_{q(w)}(\log q(w)) - E_{q(w)} \log p(w,D) + E_{q(w)} \log p(D) \\ E_{q(w)} \log p(D) - KL(q(w) \| p(w|D)) &= E_{q(w)} \log p(w,D) - E_{q(w)}(\log q(w)) \end{aligned} \quad (7.20)$$



Gambar 69 Ilustrasi *Variational Inference*

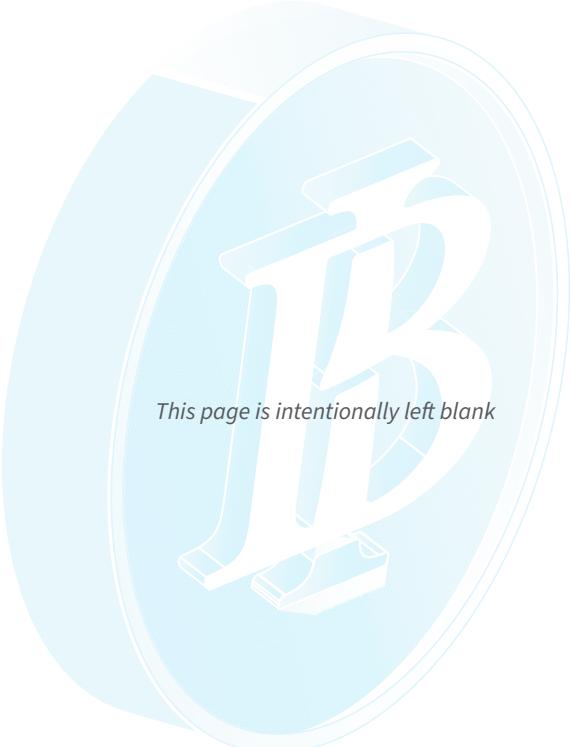
Dalam hal ini $KL \geq 0$ dan $p(D)$ merupakan konstanta, sehingga

$$\log p(D) \geq E_{q(w)} \log p(w, D) - E_{q(w)}(\log q(w)) \quad (7.21)$$

Term di bagian kanan persamaan umumnya disebut dengan *Evidence Lower Bound* (ELBO) = $E_{q(w)} \log p(w, D) - E_{q(w)}(\log q(w))$. Dengan demikian, memaksimalkan berarti meminimalkan KL , sehingga diperoleh bahwa $q(w)$ akan mendekati $p(w|D)$.

Beberapa contoh studi kasus yang memerlukan aproksimasi baik MCMC maupun VI misalnya estimasi GMM, inferensi pada FHMM (Ghahramani & Jordan, 1997), estimasi AFHMM (Kolter & Jaakkola, 2012).





This page is intentionally left blank

BAGIAN 8

SOME TECHNICAL NOTES ON PRACTICING AI



"...success in creating AI, could be the biggest event in the history of our civilisation.." – Stephen Hawking

Terdapat beberapa hal yang perlu diperhatikan dalam pengembangan maupun pemanfaatan AI/*machine learning* secara teknis. Hal ini berlaku secara umum termasuk pada *framework* ViTAL-based CRISP-DM (CRISP-DM (Shearer C. (2000)) yang berpedoman pada prinsip ViTAL) sebagaimana yang dijelaskan pada *AI Playbook Series 1: The First Step*. Pada prinsipnya beberapa hal berikut ini juga menekankan beberapa hal yang disampaikan pada *AI Playbook Series 1: The First Step*, tetapi kali ini lebih kepada aspek teknis.

8.1 Tips on Data Preparation

1. **Security and Ethics:** Data perlu disimpan secara aman (*secure*) dan dalam eksplorasinya perlu memperhatikan *ethic/privacy* dalam hal ini termasuk informasi yang bersifat sensitif.
2. **Consistency dan missing value handling:** Data harus konsisten, dan terbebas dari *bias* dan *missing value* (lebih lanjut simak Tukey (1977)).
3. **Feature engineering:** Dalam pengembangan model, perlu dipastikan agar *engineering* terhadap fitur telah sesuai. Hal ini misalnya termasuk, tetapi tidak terbatas pada:
 - a. **Feature Scaling:** Sebisa mungkin representasi *input* telah dilakukan scaling/normalization agar pemelajaran terhindar isu pemelajaran a.l. bias pembobotan pada fitur yang memiliki skala besar.
 - b. **One hot encoding (dummy variables):** Pada fitur bertipe diskret yang tidak terpengaruh order (urutan), dapat dilakukan pembuatan *dummy variable* melalui *one hot encoding*.

8.2 Tips on Machine Learning Modelling

1. **Model Parsimony: Start from the simplest:** Pertama, gunakan model dari yang paling sederhana. Hal ini penting untuk memudahkan evaluasi, serta meminimalisasi *effort* yang tidak perlu.
2. **Pelatihan dan Pengujian:** Untuk dataset latih yang berukuran kecil, pelatihan dan pengujian model hendaknya dilakukan melalui skenario *cross validation* atau *bootstrapping*, dengan proses randomisasi. Apabila memungkinkan, lakukan pengujian pada *test data* (data uji) yang merupakan *outsample*.
3. **Isu pemelajaran:** Pastikan dalam pengembangan model tidak terdapat data *leakage*, *undefit*, *overfit*, maupun *infinite loop/process*. Pastikan pemodelan mengikuti kaidah *learning from data* secara umum (simak e.g. MacKay (2003)).
4. **Curse of dimensionality:** Pastikan jumlah atribut tidak melebihi jumlah data untuk menghindari isu *curse of dimensionality* (misal jumlah data minimal adalah e^d dengan d adalah jumlah dimensi, jumlah data sebanyak n kali jumlah dimensi, dsb.). Jumlah data juga perlu disesuaikan dengan kompleksitas model. Sebagai contoh untuk *deep learning* yang kompleks, jumlah data yang

diperlukan lebih banyak dari model yang lebih sederhana seperti *logistic regression*.

5. **Pemodelan Matematika:** Apabila mengharuskan modifikasi model ataupun algoritme, lakukan reviu literatur secara menyeluruh terlebih dahulu. Pemodelan matematika harus melalui pembuktian sehingga diperoleh formulasi yang absah dan *robust*.
6. **Konsep vs. Kode:** Lakukan *catch up* konsep matematika dan implementasinya, misalnya di Python dapat dilakukan dengan mencoba *exercise library* a.l. *numpy*, *linalg* (*linear algebra*), atau *scipy* (*probability & statistics*), dll..
7. **Reproducability:** Untuk keperluan *reproducability*, pastikan telah dilakukan inisiasi *random seed* (*seeding*) pada komponen yang secara *default* terdapat randomisasi, sehingga ketika dijalankan kembali di lain waktu proses akan memberikan hasil yang konsisten. Contoh perlakuan penambahan seeding adalah pada saat inisialisasi model, *splitting train-test*, maupun *cross-validation*.
8. **Pipelining:** Upayakan gunakan *pipeline* bila memang *tools/library* tersebut menyediakan fasilitas *pipelining*.
9. **Rule Based Model:** Apabila ternyata model yang dikembangkan bersifat *rule-based*, maka perlu dilakukan asesmen mendetail dan perumusan yang komprehensif terkait representasi pengetahuan yang akan dibentuk.

8.3 Tips on AI/Machine Learning Evaluation, Deployment and Monitoring

1. **Evaluasi Kuantitatif:** Contoh asesmen kuantitatif untuk *supervised learning* mencakup di antaranya perhitungan F1 score, accuracy score, recall, precision, dll. untuk klasifikasi serta RMSE, MAE untuk regresi. Contoh asesmen kuantitatif untuk *unsupervised learning* misalnya *silhouette analysis* pada *clustering*, dll..
2. **Evaluasi Kualitatif:** Evaluasi secara kualitatif juga perlu dilakukan a.l. melalui asesmen pada grafik, *event analysis*, dll., serta analisis *interpretability* (Molnar, 2023).
3. **Asesmen dengan Teori:** Evaluasi terhadap hasil secara keseluruhan juga perlu dilakukan untuk melihat kesesuaian dengan teori ekonomi, *finance*, dan teori-teori lain yang terkait.
4. **Robustness:** Pengembangan AI dan ML harus memiliki kemampuan untuk *trust the result* dan melakukan *proper justification*. Hal ini dibuktikan melalui serangkaian pengujian baik dari seluruh tahapan *data processing*, hasil yang didapatkan, s.d. evaluasi terhadap teori domain terkait (d.h.i *econometric/finance*).
5. **Available for Review:** Setiap proses harus dapat dijelaskan melalui *review code*. Oleh karena itu, penting agar setiap proses dapat dilakukan *checkpointing* a.l. melalui *print*, *tqdm*, dll..
6. **Interpretability:** Pastikan setiap proses maupun model yang dikembangkan mendukung *interpretability*. *Interpretability* dalam hal ini dapat berupa model *agnostic*/model **specific** maupun *local vs. global interpretability*. Beberapa

alternatif metode untuk mendukung *interpretability* dapat dilihat pada buku Molnar (2023).

7. **Concept Drift Handling/Changes in Trend:** Identifikasi adanya perubahan karakteristik pada *input*. Hal ini penting karena model melakukan analisis berdasarkan data yang diberikan. Apabila terdapat perubahan karakteristik data, maka dapat berimplikasi pada hasil analisis yang *misleading*.
8. **Logging and Alerting System:** Implementasikan mekanisme *logging* dan *alerting* untuk memudahkan *tracking* dan *awareness* terkait *performance*

Box 8.1 In Finance? What are several AI potential issues?

- **Lack of Transparency**
Oleh karena AI yang powerful saat ini cenderung blackbox, transparansi dan explainability masih merupakan isu yang perlu diperhatikan.
- **Misalignment with Financial Regulation**
Saat ini AI memiliki kecenderungan berdasar pada correlation, sementara Financial Regulation berpondasi pada causation.
- **Bias and Discrimination**
AI terutama dengan machine learning, dilatih dari data. Adanya bias pada algoritma dan data dapat berimplikasi pada bias dan diskriminasi dalam implementasi analisis.
- **Systemic Risks and Market Instability**
Kebergantungan pada AI yang serupa dapat menyebabkan efek herding (kecenderungan pasar mengikuti pola yang sama). Hal ini dapat berakibat negatif terkait asumsi, dan dapat meningkatkan risiko sistemik.
- **Operational Risks and Dependence on Third Parties**
Adanya ketergantungan adopsi dengan pihak ketiga tentu meningkatkan risiko dependensi dengan vendor.
- **Challenges in Supervision and Fraud Detection**
Aktivitas keuangan yang seluruhnya berbasis AI berdampak pada sulitnya mengidentifikasi risiko/kondisi real. Selain itu, terdapat risiko fraud akibat manipulasi data.
- **Data Manipulation and Model Collapse**
Oleh karena AI berdasarkan data, adanya penurunan kualitas data maupun manipulasi data dapat berimplikasi pada performa model yang menurun atau hasil yang misleading.
- **Legal and Ethical Challenges**
Keputusan financial yang berdasarkan AI bisa jadi tidak berdasar pada regulasi/framework finansial. Tanpa regulasi yang kuat AI dapat digunakan untuk tindakan manipulative/misconduct.

Referensi: Artificial intelligence in finance: how to trust a black box? (Philipponnat, 2025)

References

- Amershi *et al.* 2019. Guidelines for Human-AI Interaction. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. URL: <https://dl.acm.org/doi/10.1145/3290605.3300233>
- Bengio, Y., LeCun, Y., & Hinton, G. (2015). *Deep learning*. Nature, 521(7553), 436-444. URL: <https://doi.org/10.1038/nature14539>.
- Bok, B., Caratelli, D., Giannone, D., Sbordone, A., & Tambalotti, A. (2017). *Macroeconomic nowcasting and forecasting with big data*. Federal Reserve Bank of New York.
- Bolukbasi *et al.* 2016. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. Proceeding of 30th Conference on Neural Information Processing Systems. URL: https://proceedings.neurips.cc/paper_files/paper/2016/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf
- Brown, Tom B., *et al.* (2020). *Language Models are Few-Shot Learners*. Advances in Neural Information Processing Systems.
- Brundage *et al.* 2020. Toward Trustworthy AI Development: Mechanisms for Supporting Verifiable Claims. Available on Centre for the Governance of AI, URL: <https://cdn.governance.ai/Toward-Trustworthy-AI-Development.pdf>
- Bruno, G. (2016). *Text mining and sentiment extraction in central bank documents*. 2016 IEEE International Conference on Big Data (Big Data), (pp. 1700-1708).
- Brynjolfsson, E., & McAfee, A. (2014). *The second machine age: Work, progress, and prosperity in a time of brilliant technologies*. W. W. Norton & Company.
- Buell, B., Chen, C., Cherif, R., Seo, H.-J., Tang, J., & Wendt, N. (2021). *Impact of COVID-19: Nowcasting and Big Data to Track Economic Activity in Sub-Saharan Africa*. IMF Working Paper.
- Campbell, Murray, A. J. Hoane Jr., and Feng-hsiung Hsu. (2002). Deep Blue. Artificial Intelligence, 134(1-2), pp. 57-83.
- Chakraborty, C., & Joseph, A. (2017). *Machine learning at central banks*. Bank of England.
- Chapman, J. T., & Desai, A. (2020). *Using Payments Data to Nowcast Macroeconomic Variables During the Onset of Covid-19*. Bank of Canada.
- Charpentier A, Flachaire E, Ly Antonie. 2018. Econometrics and Machine Learning. *Economie et Statistique / Economics and Statistics*, 505-506, pp. 147-169. URL: https://www.insee.fr/en/statistiques/fichier/3706234/505-506_Charpentier-Flachaire-Ly-EN.pdf
- Chollet, F. (2017). *Deep learning with Python*. Manning Publications.
- Daas, P. J., & Puts, M. J. (2014). *Social media sentiment & consumer confidence*. European Central Bank.

- Desai, A., 2023. Machine Learning for Economic Research: When, What, and How? Available at SSRN: URL: <https://ssrn.com/abstract=4404772>
- Domingos, P. (2015). *The master algorithm: How the quest for the ultimate learning machine will remake our world*. Basic Books.
- G Gensler, L. B. (2020). *Deep Learning and Financial Stability*. MIT Artificial Intelligence Global Policy Forum.
- G Güngör, M. D. (2016). *Corporate sector financials from financial stability perspective*. Irving Fisher Committe on Central Bank Statistics IFC-ECCBSO-CBRT.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- Ghahramani Z, Jordan M I. 1997. Factorial Hidden Markov Models. Machine Learning, URL: <https://mlg.eng.cam.ac.uk/zoubin/papers/fhmmML.pdf>
- Heaven WD. 2023. Geoffrey Hinton tells us why he's now scared of the tech he helped build. Available online on MIT Technology Review. URL: <https://www.technologyreview.com/2023/05/02/1072528/geoffrey-hinton-google-why-scared-ai/>
- Hyman, Anthony (1982). *Charles Babbage: Pioneer of the Computer*. Princeton University Press,,
- Howard & Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. URL: <https://aclanthology.org/P18-1031.pdf>
- Jensen, K. (2024). *The Future of AI: From Big Data to Generative Models*. Cambridge University Press.
- Jordan, M. I., & Mitchell, T. M. (2015). *Machine learning: Trends, perspectives, and prospects*. Science, 349(6245), 255-260. <https://doi.org/10.1126/science.aaa8415>.
- Kolter J Z, Jaakkola T. 2012. Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation. Proceeding of the Fifteenth International Conference on Artificial Intelligence and Statistics. URL: <https://proceedings.mlr.press/v22/zico12.html>
- Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. (2012). *ImageNet Classification with Deep Convolutional Neural Networks*. Advances in Neural Information Processing Systems.
- Leavitt, David. (2006). *The Man Who Knew Too Much: Alan Turing and the Invention of the Computer*. W. W. Norton & Company.
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). *Deep learning*. Nature, 521(7553), 436-444. <https://doi.org/10.1038/nature14539>.
- MacKay, David. 2003. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press. URL: <https://www.inference.org.uk/itprnn/book.pdf>
- McAfee, Andrew, and Erik Brynjolfsson. (2017). *Machine, Platform, Crowd: Harnessing Our Digital Future*. W. W. Norton & Company.

- McCorduck, Pamela. (2004). *Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence*. A. K. Peters.
- McLaren, N., & Shanbhogue, R. (2011). *Using internet search data as economic indicators*. Bank of England.
- Mitchell, Melanie. (2019). *Artificial Intelligence: A Guide for Thinking Humans*. Farrar, Straus and Giroux.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., & Hassabis, D. (2015). *Human-level control through deep reinforcement learning*. *Nature*, 518(7540), 529-533. <https://doi.org/10.1038/nature14236>
- Molnar, Christoph. 2023. *Interpretable Machine Learning – A Guide for Making Black Box Models Explainable*. Buku Online, URL: <https://christophm.github.io/interpretable-ml-book/>
- Murphy, P Kevin. 2012. *Machine Learning: Probabilistic Perspectives*. MIT Press.
- Murphy, P Kevin. 2024. *Probabilistic Machine Learning*. MIT Press. URL: <https://probml.github.io/pml-book/book1.html>.
- Nilsson, Nils J. (2010) *The Quest for Artificial Intelligence: A History of Ideas and Achievements*. Cambridge University Press.
- Raffel et al. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21 (2020) 1-67. URL: <https://jmlr.org/papers/volume21/20-074/20-074.pdf>
- Ribeiro et al. 2016. “Why Should I Trust You?” Explaining the Predictions of Any Classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. URL: <https://dl.acm.org/doi/10.1145/2939672.2939778>
- Riemer S, Straus M, Rabener E, Bickford JK, Hilbers P, Kalra N, Kapoor A, King J, Palumbo S, Pardasani N, Pauly M, Rulf K, Widowitz M. 2023. A Generative AI Roadmap for Financial Institutions, Boston Consulting Group (BCG) News Article. URL: <https://www.bcg.com/publications/2023/a-genai-roadmap-for-fis>.
- Russell SJ & Norvig Peter. 2021. *Artificial Intelligence: A Modern Approach* (4th ed.). Hoboken: Pearson.
- Schmidhuber, J. (2015). *Deep learning in neural networks: An overview*. *Neural Networks*, 61, 85-117. <https://doi.org/10.1016/j.neunet.2014.09.003>.
- Shearer C. (2000). *The CRISP-DM model: the new blueprint for data mining*. *Journal of Data Warehousing*, Vol 5:13–22. URL (alternatif): <https://mineracaodedados.files.wordpress.com/2012/04/the-crisp-dm-model-the-new-blueprint-for-data-mining-shearer-colin.pdf>
- Suss J & Treitel H. 2019. Predicting Bank Distress in the UK with Machine Learning. Bank of England Working Paper No. 831. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3465753
- Szeliski, R. (2021). *Computer vision: Algorithms and applications* (2nd ed.). Springer.

- Tobback *et al.* 2017. Between hawks and doves: measuring central bank communication. ECB Working Paper Series no. 2085. URL: <https://www.ecb.europa.eu/pub/pdf/scpwps/ecb.wp2085.en.pdf>
- Tukey, John W. (1977). *Exploratory Data Analysis*. Pearson.
- Tumala, M. M., & Omotosho, B. S. (2019). *A text mining analysis of central bank monetary policy communications in Nigeria*. Central Bank of Nigeria.
- Wirth R & Hipp. (2000). *CRISP-DM: Towards a Standard Process Model for Data Mining*. Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining, pp. 29-40. URL (alternatif): <https://www.cs.unibo.it/~danilo.montesi/CBD/Beatrix/10.1.1.198.5133.pdf>
- Woloszko, N. (2020). *Tracking activity in real time with Google Trends*. OECD.

Appendix

Appendix 1 Daftar Machine Learning Models

Appendix ini berisi list machine learning, disertai dengan tautan ke sumber rujukan untuk penjelasan lebih lanjut secara mendetail.

Instance-based algorithm

- [K-nearest neighbors algorithm](#) (KNN)
- [Learning vector quantization](#) (LVQ)
- [Self-organizing map](#) (SOM)

Regression analysis

- [Logistic regression](#)
- [Ordinary least squares regression](#) (OLSR)
- [Linear regression](#)
- [Stepwise regression](#)
- [Multivariate adaptive regression splines](#) (MARS)
- Regularization algorithm
 - [Ridge regression](#)
 - [Least Absolute Shrinkage and Selection Operator](#) (LASSO)
 - [Elastic net](#)
 - [Least-angle regression](#) (LARS)
- [Classifiers](#)
 - [Probabilistic classifier](#)
 - [Naive Bayes classifier](#)
 - [Binary classifier](#)
 - [Linear classifier](#)
 - [Hierarchical classifier](#)

Dimensionality reduction

Dimensionality reduction

- [Canonical correlation analysis](#) (CCA)
- [Factor analysis](#)
- [Feature extraction](#)
- [Feature selection](#)
- [Independent component analysis](#) (ICA)
- [Linear discriminant analysis](#) (LDA)
- [Multidimensional scaling](#) (MDS)
- [Non-negative matrix factorization](#) (NMF)

- Partial least squares regression (PLSR)
- Principal component analysis (PCA)
- Principal component regression (PCR)
- Projection pursuit
- Sammon mapping
- t-distributed stochastic neighbor embedding (t-SNE)

Ensemble learning

Ensemble learning

- AdaBoost
- Boosting
- Bootstrap aggregating (Bagging)
- Ensemble averaging
- Gradient boosted decision tree (GBDT)
- Gradient boosting machine (GBM)
- Random Forest
- Stacked Generalization (blending)

Meta-learning

Meta-learning

- Inductive bias
- Metadata

Reinforcement learning

Reinforcement learning

- Q-learning
- State-action-reward-state-action (SARSA)
- Temporal difference learning (TD)
- Learning Automata

Supervised learning

Supervised learning

- Averaged one-dependence estimators (AODE)
- Artificial neural network
- Case-based reasoning
- Gaussian process regression
- Gene expression programming
- Group method of data handling (GMDH)
- Inductive logic programming
- Instance-based learning
- Lazy learning
- Learning Automata
- Learning Vector Quantization
- Logistic Model Tree
- Minimum message length (decision trees, decision graphs, etc.)

- Nearest Neighbor Algorithm
- Analogical modeling
- Probably approximately correct learning (PAC) learning
- Ripple down rules, a knowledge acquisition methodology
- Symbolic machine learning algorithms
- Support vector machines
- Random Forests
- Ensembles of classifiers
 - Bootstrap aggregating (bagging)
 - Boosting (meta-algorithm)
- Ordinal classification
- Conditional Random Field
- ANOVA
- Quadratic classifiers
- k-nearest neighbor
- Boosting
 - SPRINT
- Bayesian networks
 - Naive Bayes
- Hidden Markov models
 - Hierarchical hidden Markov model

Bayesian

Bayesian statistics

- Bayesian knowledge base
- Naive Bayes
- Gaussian Naive Bayes
- Multinomial Naive Bayes
- Averaged One-Dependence Estimators (AODE)
- Bayesian Belief Network (BBN)
- Bayesian Network (BN)

Decision tree algorithms

Decision tree algorithm

- Decision tree
- Classification and regression tree (CART)
- Iterative Dichotomiser 3 (ID3)
- C4.5 algorithm
- C5.0 algorithm
- Chi-squared Automatic Interaction Detection (CHAID)
- Decision stump
- Conditional decision tree
- ID3 algorithm
- Random forest
- SLIQ

Linear classifier

Linear classifier

- Fisher's linear discriminant
- Linear regression
- Logistic regression
- Multinomial logistic regression
- Naive Bayes classifier
- Perceptron
- Support vector machine

Unsupervised learning

Unsupervised learning

- Expectation-maximization algorithm
- Vector Quantization
- Generative topographic map
- Information bottleneck method
- Association rule learning algorithms
 - Apriori algorithm
 - Eclat algorithm

Artificial neural networks

Artificial neural network

- Feedforward neural network
 - Extreme learning machine
 - Convolutional neural network
- Recurrent neural network
 - Long short-term memory (LSTM)
- Logic learning machine
- Self-organizing map

Association rule learning

Association rule learning

- Apriori algorithm
- Eclat algorithm
- FP-growth algorithm

Hierarchical clustering

Hierarchical clustering

- Single-linkage clustering
- Conceptual clustering

Cluster analysis

Cluster analysis

- BIRCH
- DBSCAN

- Expectation–maximization (EM)
- Fuzzy clustering
- Hierarchical clustering
- k-means clustering
- k-medians
- Mean-shift
- OPTICS algorithm

Anomaly detection

Anomaly detection

- k-nearest neighbors algorithm (k -NN)
- Local outlier factor

Semi-supervised learning

Semi-supervised learning

- Active learning –
- Generative models
- Low-density separation
- Graph-based methods
- Co-training
- Transduction

Deep learning

Deep learning

- Deep belief networks
- Deep Boltzmann machines
- Deep Convolutional neural networks
- Deep Recurrent neural networks
- Hierarchical temporal memory
- Generative Adversarial Network
 - Style transfer
- Transformer
- Stacked Auto-Encoders

Other machine learning methods and problems

- Anomaly detection
- Association rules
- Bias-variance dilemma
- Classification
 - Multi-label classification
- Clustering
- Data Pre-processing
- Empirical risk minimization
- Feature engineering
- Feature learning

- Learning to rank
- Occam learning
- Online machine learning
- PAC learning
- Regression
- Reinforcement Learning
- Semi-supervised learning
- Statistical learning
- Structured prediction
 - Graphical models
 - Bayesian network
 - Conditional random field (CRF)
 - Hidden Markov model (HMM)
- Unsupervised learning
- VC theory

Appendix 2 Glossary (Terminologi Umum)

Accuracy

Persentase prediksi benar yang dibuat oleh model.

Algorithm

Suatu metode, fungsi, atau serangkaian instruksi yang digunakan untuk menghasilkan model pembelajaran mesin. Contohnya meliputi regresi linier, pohon keputusan, mesin vektor pendukung, dan jaringan saraf.

Attribute

Kualitas yang menggambarkan suatu pengamatan (misalnya warna, ukuran, berat). Dalam istilah Excel, ini adalah tajuk kolom.

Bias metric

Mengukur selisih rata-rata antara prediksi dan yang sebenarnya (*groundtruth*).

- **Low bias (bias rendah)** mengindikasikan setiap prediksi benar, atau memiliki sedikit perbedaan antara nilai estimasi dan aktual pada saat prediksi. Bias rendah dengan *variance* tinggi terjadi ketika model mengalami *overfitting*.
- **High bias (with low variance)** mengindikasikan kondisi ketika terdapat perbedaan yang cukup besar antara nilai, terdapat banyak *noise*, dan model yang terlalu sederhana. Bias tinggi (dengan *variance* rendah) terjadi ketika model mengalami *underfitting*.

Bias term

Memungkinkan model untuk merepresentasikan pola yang tidak melalui titik asal (origin). Misalnya, jika semua fitur saya bernilai 0, apakah *output* juga akan nol? Apakah mungkin ada nilai dasar yang menjadi dasar pengaruh fitur? *Bias term* biasanya menyertai bobot dan melekat pada neuron atau filter.

Categorical Variables

Variabel dengan serangkaian nilai yang mungkin bersifat diskrit. Dapat berupa ordinal (urutan penting) atau nominal (urutan tidak penting).

Classification

Proses memprediksi *output* yang berupa kategorikal/diskret.

- **Binary classification** memprediksi satu di antara dua kemungkinan *outcome*, misalnya memprediksi apakah suatu termasuk email spam atau bukan spam.
- **Multi-class classification** memprediksi satu di antara beberapa kemungkinan *outcome* misalnya memprediksi suatu foto apakah *cat*, *dog*, atau *human*.

Classification Threshold

Nilai probabilitas terendah di mana kita merasa nyaman untuk menyatakan klasifikasi positif. Misalnya, jika probabilitas prediksi seseorang menderita diabetes > 50%, maka hasilnya True (benar), jika tidak, False (salah).

Clustering

Pengelompokan data secara tidak diawasi (*unsupervised*) ke dalam beberapa kelompok.

Confusion Matrix

Tabel yang menggambarkan kinerja model klasifikasi dengan mengelompokkan prediksi ke dalam empat kategori, sebagai contoh dalam kasus prediksi diabetes:

- **True Positives (Positif Benar):** kita dengan benar memprediksi bahwa mereka menderita diabetes.
- **True Negatives (Negatif Benar):** kita dengan benar memprediksi bahwa mereka tidak menderita diabetes.
- **False Positives (Positif Salah):** kita salah memprediksi bahwa mereka menderita diabetes (kesalahan Tipe I).
- **False Negatives (Negatif Salah):** kita salah memprediksi bahwa mereka tidak menderita diabetes (kesalahan Tipe II).

Continuous Variables

Variabel dengan rentang nilai yang mungkin didefinisikan oleh skala angka (misalnya, penjualan, masa hidup).

Convergence

Kedua yang dicapai selama pelatihan model ketika *error/loss* berubah sangat sedikit di antara setiap iterasi.

Deduction

Pendekatan dari atas ke bawah untuk menjawab pertanyaan atau memecahkan masalah. Teknik logika yang dimulai dengan teori dan menguji teori tersebut dengan observasi untuk mendapatkan kesimpulan. Contohnya: Kita mencurigai X, tetapi kita perlu menguji hipotesis sebelum menarik kesimpulan.

Deep Learning

Deep learning berasal dari algoritma pembelajaran mesin yang disebut *perceptron* atau *multi-layer perceptron* yang semakin mendapat perhatian karena keberhasilannya dalam berbagai bidang, seperti penglihatan komputer, pengolahan sinyal, diagnosis medis, hingga mobil otonom. Seperti algoritma AI lainnya, *deep learning* didasarkan pada penelitian selama beberapa dekade. Saat ini, kita memiliki lebih banyak data dan daya komputasi yang lebih murah yang membuat algoritma ini sangat kuat dalam mencapai akurasi terkini. Dalam dunia modern, algoritma ini dikenal sebagai jaringan saraf tiruan. *Deep learning* jauh lebih akurat dan kuat dibandingkan dengan jaringan saraf tiruan tradisional, tetapi sangat dipengaruhi oleh jaringan saraf dan jaringan perceptron dari pembelajaran mesin.

Dimension

Dimensi dalam pembelajaran mesin dan ilmu data berbeda dengan fisika. Di sini, dimensi data berarti berapa banyak fitur yang dimiliki dalam dataset. Misalnya,

dalam aplikasi deteksi objek, ukuran gambar yang datar dan saluran warna (misalnya 28x28) adalah fitur dari set *input*. Dalam prediksi harga rumah (mungkin) ukuran rumah adalah dataset, sehingga kita menyebutnya data berdimensi 1.

Epoch

Epoch menggambarkan jumlah waktu algoritme melihat seluruh set data.

Extrapolation

Membuat prediksi di luar jangkauan dataset

Feature

Dalam konteks dataset, fitur mewakili kombinasi atribut dan nilai. Warna adalah atribut. "Warna biru" adalah fitur. Dalam istilah Excel, fitur mirip dengan sel. Istilah fitur memiliki definisi lain dalam konteks yang berbeda.

Feature Selection

Seleksi fitur adalah proses memilih fitur yang relevan dari dataset untuk membuat model pembelajaran mesin

Feature Vector

Daftar fitur yang menggambarkan sebuah observasi dengan beberapa atribut. Dalam Excel, kita menyebut ini sebagai baris.

Gradient Accumulation

A mechanism to split the batch of samples—used for training a neural network—into several mini-batches of samples that will be run sequentially. This is used to enable using large batch sizes that require more GPU memory than available.

Hyperparameters

Hyperparameters adalah properti tingkat tinggi dari sebuah model, seperti seberapa cepat ia bisa belajar (*learning rate*) atau kompleksitas sebuah model. Kedalaman pohon dalam *Decision Tree* atau jumlah lapisan tersembunyi dalam Jaringan Saraf adalah contoh *hyperparameter*.

Induction

Pendekatan dari bawah ke atas untuk menjawab pertanyaan atau memecahkan masalah. Teknik logika yang dimulai dari observasi menuju teori. Contohnya: Kita terus mengamati X, sehingga kita menyimpulkan bahwa Y harus benar.

Instance

Sebuah titik data, baris, atau sampel dalam dataset. Istilah lain untuk observasi.

Label

Bagian "jawaban" dari sebuah observasi dalam pembelajaran terawasi (*supervised learning*). Misalnya, dalam dataset yang digunakan untuk mengklasifikasikan bunga

menjadi berbagai spesies, fiturnya mungkin termasuk panjang dan lebar kelopak, sedangkan labelnya adalah spesies bunga.

Learning Rate

Ukuran dari langkah pembaruan yang diambil selama *loop* optimasi seperti *Gradient Descent*. Dengan *learning rate* tinggi, kita dapat mencakup lebih banyak area setiap langkah, tetapi berisiko melewati titik terendah karena kemiringan bukit terus berubah. Dengan learning rate yang sangat rendah, kita dapat dengan yakin bergerak ke arah gradien negatif karena sering menghitungnya kembali. *Learning rate* rendah lebih presisi, tetapi menghitung gradien memakan waktu lama, sehingga akan memakan waktu lama untuk mencapai titik terendah.**Loss**

$\text{Loss} = \text{true_value}$ (dari dataset) - predicted value (dari model ML). Semakin rendah *loss*, semakin baik modelnya (kecuali model *overfitting* pada data pelatihan). *Loss* dihitung pada pelatihan dan validasi dan interpretasinya adalah seberapa baik model melakukannya untuk kedua set ini. Tidak seperti akurasi, *loss* bukanlah persentase. Ini adalah penjumlahan dari kesalahan yang dilakukan untuk setiap contoh dalam set pelatihan atau validasi.

Machine Learning

pembelajaran mesin adalah bidang di mana algoritma buatan manusia memiliki kemampuan untuk belajar sendiri atau memprediksi masa depan untuk data yang belum pernah dilihat.

Model

Struktur data yang menyimpan representasi dari dataset (bobot dan bias). Model dibuat/dipelajari ketika melatih algoritme pada dataset.

Neural Networks

Jaringan Saraf adalah algoritme matematika yang dimodelkan setelah arsitektur otak, dirancang untuk mengenali pola dan hubungan dalam data.

Normalization

Pembatasan belajar misalnya (nilai bobot dalam regresi) untuk menghindari *overfitting* dan meningkatkan kecepatan komputasi.

Noise

Informasi yang tidak relevan atau acak dalam dataset yang mengaburkan pola yang mendasarinya.

Null Accuracy

Akurasi dasar yang dapat dicapai dengan selalu memprediksi kelas yang paling sering ("B memiliki frekuensi tertinggi, jadi mari kita tebak B setiap saat").

Observation

Sebuah titik data, baris, atau sampel dalam dataset. Istilah lain untuk *instance*.

Outlier

Observasi yang menyimpang secara signifikan dari observasi lain dalam dataset.

Overfitting

Overfitting terjadi ketika model mempelajari data pelatihan terlalu baik dan memasukkan detail serta *noise* yang spesifik untuk dataset. *Overfitting* dapat diketahui ketika model memiliki performa baik pada set pelatihan/validasi, tetapi buruk pada set pengujian atau data dunia nyata yang baru.

Parameters

Parameter adalah properti dari data pelatihan yang dipelajari melalui pelatihan model pembelajaran mesin atau klasifikasi. Mereka disesuaikan menggunakan algoritme optimasi dan unik untuk setiap percobaan. Contoh parameter meliputi:

- bobot dalam jaringan saraf tiruan
- vektor pendukung dalam *support vector machine*
- koefisien dalam regresi linear atau logistik

Regression

Memprediksi *output* kontinu (misalnya harga, penjualan).

Regularization

Regulasi adalah teknik yang digunakan untuk mengatasi masalah *overfitting*. Ini dicapai dengan menambahkan istilah kompleksitas ke fungsi *loss* yang memberikan nilai *loss* lebih besar untuk model yang lebih kompleks.

Reinforcement Learning

Melatih model untuk memaksimalkan *reward* melalui percobaan dan kesalahan yang berulang.

ROC (Receiver Operating Characteristic) Curve

Plot dari *true positive rate* terhadap *false positive rate* di semua ambang batas klasifikasi. Ini digunakan untuk mengevaluasi kinerja model klasifikasi pada berbagai ambang batas klasifikasi. Area di bawah kurva ROC dapat diartikan sebagai probabilitas bahwa model dapat dengan benar membedakan antara observasi positif acak (misalnya “spam”) dan observasi negatif acak (misalnya “bukan spam”).

Segmentation

Proses mempartisi dataset menjadi beberapa set yang berbeda. Pemisahan ini dilakukan sedemikian rupa sehingga anggota dari satu set serupa satu sama lain dan berbeda dari anggota set lainnya.

Supervised Learning

Melatih model menggunakan dataset yang diberi label.

Test Set

Data yang digunakan pada setelah pelatihan model untuk menilai kekuatan prediksi model dalam rangka mengukur kemampuan generalisasi model terhadap data yang belum pernah dilihat.

Training Set

Sekumpulan observasi yang digunakan untuk menghasilkan model pembelajaran mesin.

Transfer Learning

Metode pembelajaran mesin di mana model yang dikembangkan untuk tugas tertentu digunakan kembali sebagai titik awal untuk model pada tugas kedua. Dalam pembelajaran transfer, kita menggunakan bobot yang sudah dilatih dari model yang sudah dilatih sebelumnya (model yang telah dilatih pada jutaan gambar dengan ribuan kelas, pada beberapa GPU dengan daya tinggi selama beberapa hari) dan menggunakan fitur yang sudah dipelajari ini untuk memprediksi kelas baru.

Type 1 Error

False Positives. Misalnya, perusahaan yang mengoptimalkan praktik perekrutan untuk mengurangi kesalahan positif palsu dalam tawaran pekerjaan. Kesalahan Tipe 1 terjadi ketika kandidat tampak baik dan mereka mempekerjakannya, tetapi ternyata sebenarnya buruk.

Type 2 Error

False Negatives. Kandidat sebenarnya hebat tetapi perusahaan melewatkannya.

Underfitting

Underfitting terjadi ketika model terlalu menggeneralisasi dan gagal memasukkan variasi yang relevan dalam data yang dapat memberikan model lebih banyak kekuatan prediksi. Model *underfitting* terjadi ketika kinerjanya buruk pada set pelatihan dan set pengujian.

Unsupervised Learning

Latih model untuk menemukan pola dalam dataset yang tidak diberi label (misalnya pengelompokan).

Validation Set

Sekumpulan observasi yang digunakan selama pelatihan model untuk memberikan umpan balik tentang seberapa baik parameter saat ini menggeneralisasi di luar set pelatihan. Jika *error* pelatihan menurun, tetapi *error* validasi meningkat, model mungkin mengalami *overfitting* dan sebaiknya menghentikan pelatihan.

Appendix 3 Rekomendasi Tools Dasar untuk Lab/Eksperimen

Beberapa *tools/komponen/library* berikut merupakan contoh yang umum digunakan, dan masih bersifat dasar/awalan. Dalam hal ini penulis tidak memberikan penekanan kelebihan dan kekurangan. Perlu diketahui bahwa saat ini terdapat berbagai macam pilihan di luar daftar yang dicantumkan berikut.

Tools	Kategori	Penekanan/Keunggulan	URL
Python	Tools Dasar	Python: <i>Flexibility</i> .	Python: https://www.python.org/
Alternatif: R		R: <i>Simplification</i> .	R: https://www.r-project.org/
Anaconda	Tools Dasar	Environment Management	https://www.anaconda.com/
Scikit-Learn	Library terapan	Tools yang memberikan kemudahan dan kepraktisan, serta komprehensif untuk mengimplementasikan model Machine Learning umum/dasar.	https://scikit-learn.org/stable/
PyTorch	Library terapan	Matrix Operation & Neural Network/Deep Learning	https://pytorch.org/
Keras	Library terapan	Easy Neural Network/Deep Learning Implementation	https://keras.io/
Tensorflow	Library terapan	Matrix Operation & Neural Network/Deep Learning Modelling	https://www.tensorflow.org/
Octoparse	Tools Intermediate	Web scraping berbasis GUI untuk mengekstrak data dari situs web secara otomatis.	https://www.octoparse.com/
Dataiku	Tools Intermediate	Platform kolaborasi untuk melakukan analitik dalam pembangunan model machine learning, pengolahan data, serta otomatisasi pipeline data.	https://www.dataiku.com/
FNA	Tools Intermediate	Platform analitik berbasis graph yang digunakan untuk pemodelan jaringan keuangan dan analisis risiko dalam sistem keuangan, pasar modal dan sistem pembayaran.	https://g20monitor.com/home.html
Impala	Tools Intermediate	Query open-source yang dikembangkan oleh Cloudera untuk menjalankan analisis data secara real-time pada environment Hadoop dengan kecepatan tinggi menggunakan syntax SQL	https://impala.apache.org/
Cloudera <i>Machine Learning</i>	Tools Intermediate	Platform untuk mengembangkan data analytics dan model machine learning hingga skala besar.	https://www.cloudera.com/products/machine-learning/cdp-tour-machine-learning.html

Tools	Kategori	Penekanan/Keunggulan	URL
Cloudera Data Engineering	Tools Advance	<i>Platform</i> untuk membangun, mengelola, dan mengotomatiskan pipeline data dalam skala besar.	https://www.cloudera.com/products/data-engineering/cdp-tour-data-engineering.html

Appendix 4 List Use Case AI di BI

Moneter Market

No	Indikator	Tahun	Sumber Data	Metodologi
1	Indikator demand tenaga kerja dari data iklan lowongan kerja	2015	Portal Lowongan Kerja	Text Mining
2	Indikator persepsi/sentimen terhadap kebijakan BI dari pemberitaan	2016	Cyber Library	Text Mining
3	Indikator Economic Policy Uncertainty (EPU) dari pemberitaan	2017	Cyber Library	Text Mining
4	Indikator ekspektasi terhadap suku bunga kebijakan BI dari pemberitaan	2018	Cyber Library	Text Mining
5	Pemanfaatan data geospasial untuk pemantauan sektor konstruksi dan perumahan	2018	Citra Satelit	Image Analysis, Deep Learning
6	Indikator supply pasar properti dari data iklan properti online	2018	Portal Properti Online	Text Mining
7	Network analysis & interconnectedness supply side korporasi dari data BI-RTGS	2018	BI-RTGS	Network Analysis, Text Mining
8	Indikator kredibilitas kebijakan moneter dari pemberitaan	2019	Cyber Library	Text Mining
9	Indikator kredibilitas kebijakan moneter dari SBNI	2019	SBNI	Machine Learning
10	Indikator review wisatawan dari situs travel online	2019	Forum Travel Online	Text Mining
11	Pemanfaatan data geospasial untuk produksi pertanian	2019	Citra Satelit	Image Analysis, Deep Learning
12	Indikator aliran DPK terkait IKNB	2023	SBNI	Text Mining
12	Behavior modeling dan proyeksi flows investor	2020	BSSS, Bloomberg	Machine Learning

Stabilitas Sistem Keuangan

No	Publikasi / Presentasi	Forum	Tahun
1	Applying Grab Job Fit in BI	8 th Biennial International Conference on Operations Research and Statistics (BIORS)	2023
2	High-frequency indicators for financial instability: a methodology utilizing big data	15 th International Conference on Data Envelopment Analysis	2023
3	Big Data Analytics on the Payment System Data for Measuring Household Consumption	12 th Irving Fisher Committee (IFC) Conference	2022
4	Big Data Analytics on Regional Economic Directions with the Combination of Several Big Data Sources	63 rd ISI World Statistical Congress	2023
5	Measuring Macro-prudential Policy Credibility Using Machine Learning	12 th Irving Fisher Committee (IFC) Conference	2022
6	Sentiment Analysis of Investors from Non- bank Payment Service Apps	12 th Irving Fisher Committee (IFC) Conference	2022
7	Measuring Payment System Policy Credibility Using Machine Learning	63 rd ISI World Statistical Congress	2023
8	Big Data Analytics on RTGS Data for Tracking Corporate Activity	12 th Irving Fisher Committee (IFC) Conference	2022

No	Publikasi / Presentasi	Forum	Tahun
9	Big Data Analytics in RTGS Transaction: Network Topology of Financial System & Systemic Risk	11 th Irving Fisher Committee (IFC) Conference	2020
10	Measuring Economic Policy Uncertainty Using Big Data: Case Study in Indonesia	10 th Irving Fisher Committee (IFC) Conference	2018
11	Measuring Policy Credibility Using Big Data	61 st ISI World Statistical Congress	2017
12	Big Data Analytics for Bank Indonesia's Policy Communication Strategy	10 th Irving Fisher Committee (IFC) Conference	2018
13	Text Mining for Corporate Risk Analysis: Indonesian News as a Case Study	6 th OECD-Asia Regional Conference on Statistics	2017
14	Network Analysis of Bank Interconnectedness	60 th ISI World Statistics Congress	2015
15	Machine Learning untuk sistem klasifikasi korporasi	ISI World Statistics Congress	2023
16	Measuring employment growth based on job vacancies	BigD&AI	2020
17	Big data analytics in Bank Indonesia	OECD	2020
18	Nowcasting inflation using big data	IMF's Conference	2019
19	Big data analytics on news sentiment for monitoring central bank credibility	IFC's Biennial Conference	2019
20	Big data analytics to analyze investor perception on central bank credibility	IFC's Biennial Conference	2019
21	Nowcasting inflation using big data: a practitioner's reality note	61 st ISI World Statistical Congress	2017
22	Big data analytics on real time transaction data	60 th ISI World Statistical Congress	2015
23	Big data analytics on financial transactions and job market	Malaysia Statistics Conference (MyStats)	2015

Sistem Pembayaran

No	Indikator	Tahun	Sumber Data	Metodologi
1	Indikator e-commerce & on-demand delivery service	2017	Portal e-commerce & on-demand delivery service	Data Mining
2	Network analysis & interconnectedness PSP (a.l. Penerbit UE, Payment Gateway) dari data BI-RTGS & SBNI	2018	BI-RTGS & SBNI	Text Mining, Network Analysis
3	Indikator aktivitas transaksi pembayaran penyelenggara e-commerce dari data BI-RTGS	2021	BI-RTGS	Text Mining
4	Indikator aktivitas transaksi pembayaran FinTech P2PL dari data BI-RTGS	2021	BI-RTGS	Text Mining
5	Indikator aktivitas transaksi penyelenggara e-commerce	2021	SBNI	Text Mining
6	Indikator aktivitas transaksi pembayaran pedagang aset crypto	2021	SBNI	Text Mining
7	Indikator kredibilitas kebijakan SP dari pemberitaan	2021	Cyber Library	Text Mining
8	Payment ID	2022	Data NIK, Piloting UE, BI, ANTASENA, SLIK	Text Mining
9	Indikator agregat transaksi BI-FAST	2022	BI-FAST	Data Mining
10	Pemanfaatan data piloting pencobaan data UE harian untuk mendukung pengawasan SP	2022	Piloting pencobaan data UE harian (nonbank)	Data Mining

Publikasi

No	Indikator	Tahun	Sumber Data	Metodologi
1	Indikator harga properti di pasar sekunder dari data iklan properti online	2016	Portal Properti Online	Text Mining, Web Scraping
2	Network analysis & interconnectedness perbankan dari data BI-RTGS	2017	BI-RTGS	Network Analysis
3	Network analysis & interconnectedness korporasi (grup konglomerasi, BUMN, IKNB) dari data SBN	2018	BN	Network Analysis
4	Deteksi dini instabilitas sistem keuangan dari pemberitaan	2019	Cyber Library	Text Mining
5	Network analysis & interconnectedness perbankan dari data transaksi valas di BI- RTGS	2020	BI-RTGS	Network Analysis
6	Indikator aktivitas korporasi sekuritas dari data BI-RTGS	2020	BI-RTGS	Text Mining
7	Indikator flight-to-quality simpanan perbankan dari data BI-RTGS	2020	BI-RTGS	Text Mining
8	Indikator kerentanan tenaga kerja digital	2021	Cyber Library	Text Mining
9	Indikator ketidakstabilan belanja masyarakat dari pemberitaan	2021	Cyber Library	Text Mining
10	Indikator sentimen layanan mobile banking	2021	Review App Store & Play Store	Text Mining
11	Indikator sentimen investor terhadap FinTech P2PL	2021	Review App Store & Play Store	Text Mining
12	Indikator aliran DPK terkait IKNB	2023	SBNI	Text Mining
13	Indikator sentimen layanan keuangan digital PJ	2022	Review App Store & Play Store	Text Mining
14	Indikator Flight-to-quality simpanan perbankan	2022	BI-FAST	Text Mining
15	Indikator data HI harian (SupTech)	2022	Data Piloting HI Harian	Text Mining
16	Indikator aliran DPK Pedagang Aset Kripto	2023	SBNI & ASNBI	Network Analysis, Text Mining
17	Indikator early signal stres korporasi pangan	2023	IEIS dan SBNI	Text Mining
18	Early warning variabel pemberitaan dengan ma	2023	Cyber Library	Machine Learning
19	Behaviour analysis perbankan	2023	IEUT	Machine Learning

Contributors

The Coach:

- Diah Rosdiana (Kepala DIDD)
- Agung Bayu Purwoko (Kepala GKDD-DIDD)

The Geeks:

- Heru Praptono
- Luqman Yusuf Bharoto
- Rudy Hardiyanto
- Irfan Sampe
- Renardi Ardiya Bimantoro
- Amin Endah Sulistiawati
- Michael Giorgio Wirawan