**Pengenalan Machine Learning**

Pada modul ini, kita akan mempelajari seluk-beluk dari Machine Learning (ML) mulai dari pengenalan, tipe-tipe ML, ML workflow, model maintenance, batasan ML saat ini, hingga penerapan ML di industri sekarang. Seperti yang sudah dijelaskan pada materi taksonomi AI, secara singkat machine learning (ML) merupakan salah satu cabang dari AI. Machine learning menggunakan algoritma untuk menganalisis data dalam jumlah yang besar, belajar dari pengetahuan yang ada pada data, dan akhirnya akan memberikan keputusan berdasarkan pengalaman yang dimilikinya.

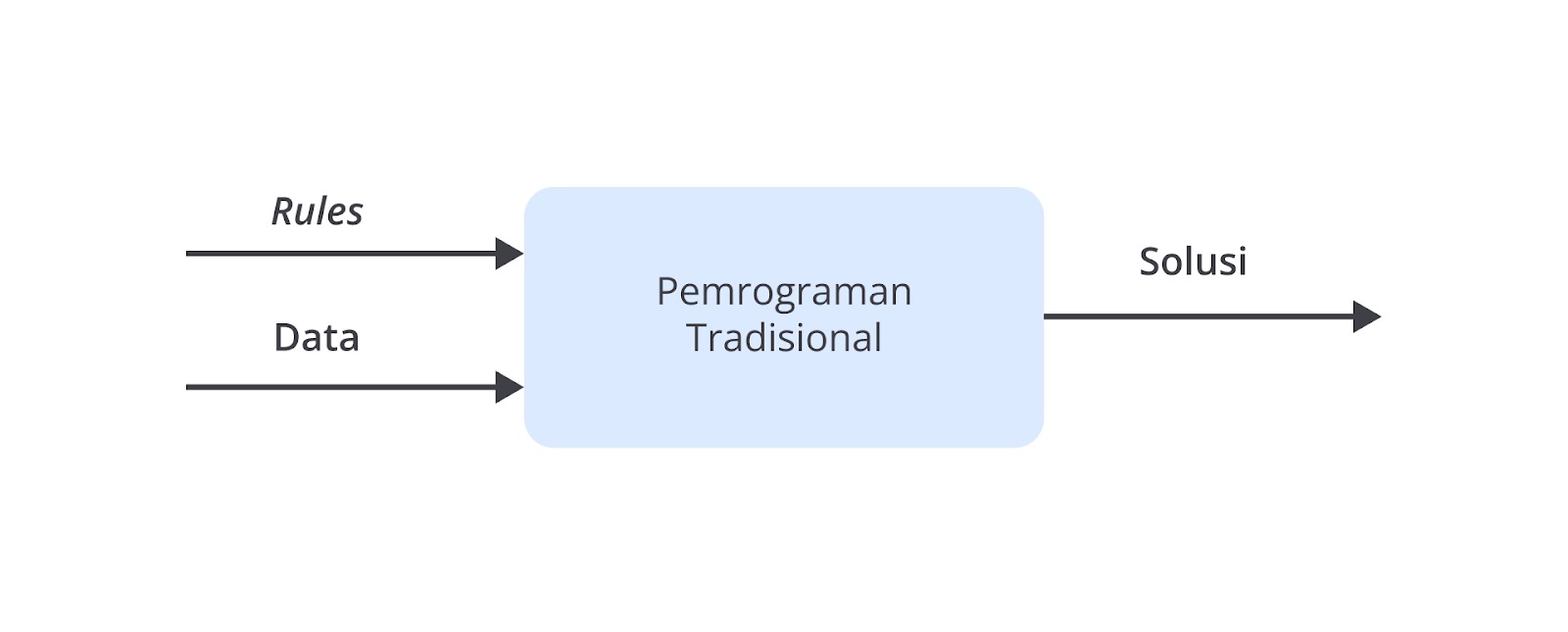
Dalam satu dekade terakhir, machine learning menjelma sebagai bidang yang berkembang sangat pesat dan terus dikembangkan para ilmuwan seluruh dunia. Inilah inti dari berbagai macam produk berteknologi tinggi terkini. Perannya sangat signifikan dalam disrupsi industri 4.0 yang sarat dengan transformasi digital.

**Traditional Programming vs Machine Learning**

Jika Anda melakukan pencarian di mesin pencari (*search engine*) seperti *google*, *bing*, *DuckDuckGo* dan lain sebagainya, Anda akan mendapatkan definisi yang beragam tentang machine learning. Hal ini tidak jadi masalah karena secara umum definisi tersebut memiliki kesamaan. Pada modul ini, kita akan membahas definisi machine learning yang disampaikan oleh Arthur Samuel.

Arthur Samuel merupakan seorang ilmuwan komputer yang memelopori kecerdasan buatan pada tahun 1959. Dia merupakan orang pertama yang memopulerkan *term*“machine learning”. Menurutnya, machine learning adalah suatu cabang ilmu yang memberi komputer kemampuan untuk belajar tanpa diprogram secara eksplisit. Bagaimana maksudnya?

Mari kita mundur sejenak ke masa sebelum machine learning ditemukan. Seperti dikemukakan oleh Moroney[2], prinsip atau paradigma pemrograman sejak permulaan era komputasi direpresentasikan dalam diagram berikut.



Pada pemrograman tradisional aturan dan data merupakan sebuah input bagi komputer.  Secara eksplisit, aturan diekspresikan dalam bahasa pemrograman. Tambahan masukan berupa data kemudian akan menghasilkan solusi sebagai keluaran. Paradigma pemrograman seperti pada diagram di atas sering disebut sebagai pemrograman tradisional.

Pemrograman tradisional memiliki keterbatasan karena ia *rigid*dengan sekumpulan aturan “if” dan “else” untuk memproses data atau menyesuaikan dengan masukan. Sebagai contoh, kita ingin membuat sebuah program untuk melakukan klasifikasi pelanggan. Kita bisa menggunakan parameter “umur” sebagai data untuk membedakan kategori satu dengan lainnya.

Perhatikan contoh penggunaan if berikut.

|  |
| --- |
| if umur <= 17:  kategori = “Bukan prioritas”  else:  kategori = “Prioritas” |

Pada contoh di atas, kita membuat algoritma program tradisional menggunakan python sebagai bahasa pemrogramannya. Tujuannya untuk melakukan klasifikasi jika umur kurang dari atau sama dengan 17 maka termasuk kategori bukan prioritas. Lalu, bagaimana jika ada penambahan parameter seperti “pendapatan”? Kondisinya akan berubah menjadi seperti berikut.

|  |
| --- |
| if umur <= 17:  if pendapatan <= 3000000:  kategori = “Bukan prioritas”  else:  Kategori = “Prioritas”  else:  kategori = “Prioritas” |

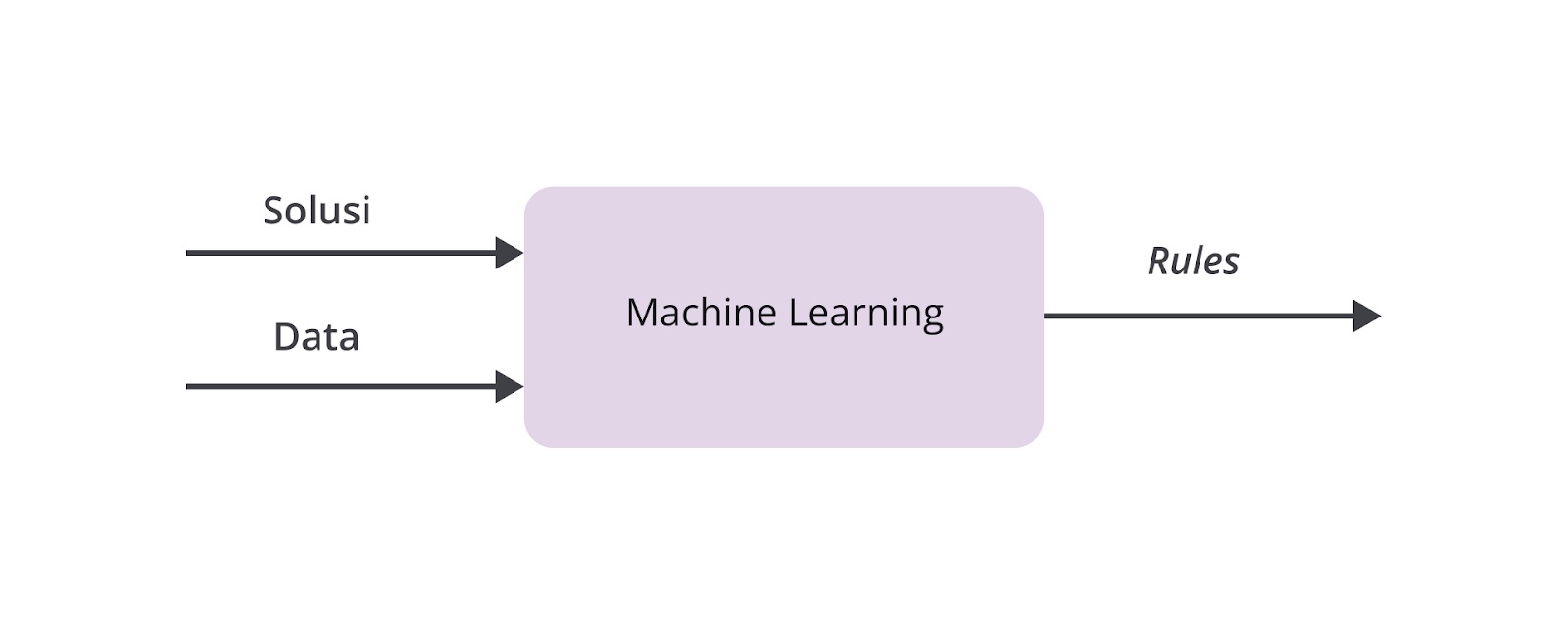
Mudah, bukan? Kita hanya perlu memberikan aturan dan data agar menghasilkan jawaban yang diinginkan. Namun, bayangkan jika kita ingin membuat klasifikasi pelanggan tersebut menjadi lebih detail. Bayangkan kita memiliki parameter sebagai berikut.

|  |  |  |
| --- | --- | --- |
| Parameter | Bukan Prioritas | Prioritas |
| Pengeluaran | 0-1500000 | >1500000 |
| Member | Bukan | Ya |
| Jarak Rumah | >20 KM | <= 20 KM |
| Pekerjaan | Tidak Bekerja | Bekerja |
| Jumlah beli produk | < 10 | >= 10 |

Berdasarkan tabel di atas, tentu kita harus membuat kondisi if lebih banyak lagi. Bayangkan lebih jauh lagi jika kita memiliki sekitar 100 kondisi if else dengan berbagai parameter untuk menentukan segmentasi pelanggan. Tentu kondisi tersebut akan membuat programmer kewalahan. Dengan kondisi yang sangat banyak, kita akan menemui masalah. Hal ini membuat kita menyadari bahwa pemrograman tradisional memiliki keterbatasan dalam menyelesaikan masalah. Perhatikan salah satu contoh jika kita memiliki banyak kondisi if else.



Pada pemrograman tradisional kita merepresentasikan masalah menjadi aturan dalam bahasa pemrograman. Kini ketika hal itu tidak lagi memungkinkan, kita perlu mengubah alur berpikir kita dengan cara yang berbeda. Paradigma baru pemrograman dengan machine learning adalah kita memiliki banyak sekali data dan label bagi data tersebut. Kita juga telah mengetahui keterkaitan antara data dengan label sebagai suatu solusi. Paradigma machine learning dapat direpresentasikan dalam diagram berikut.



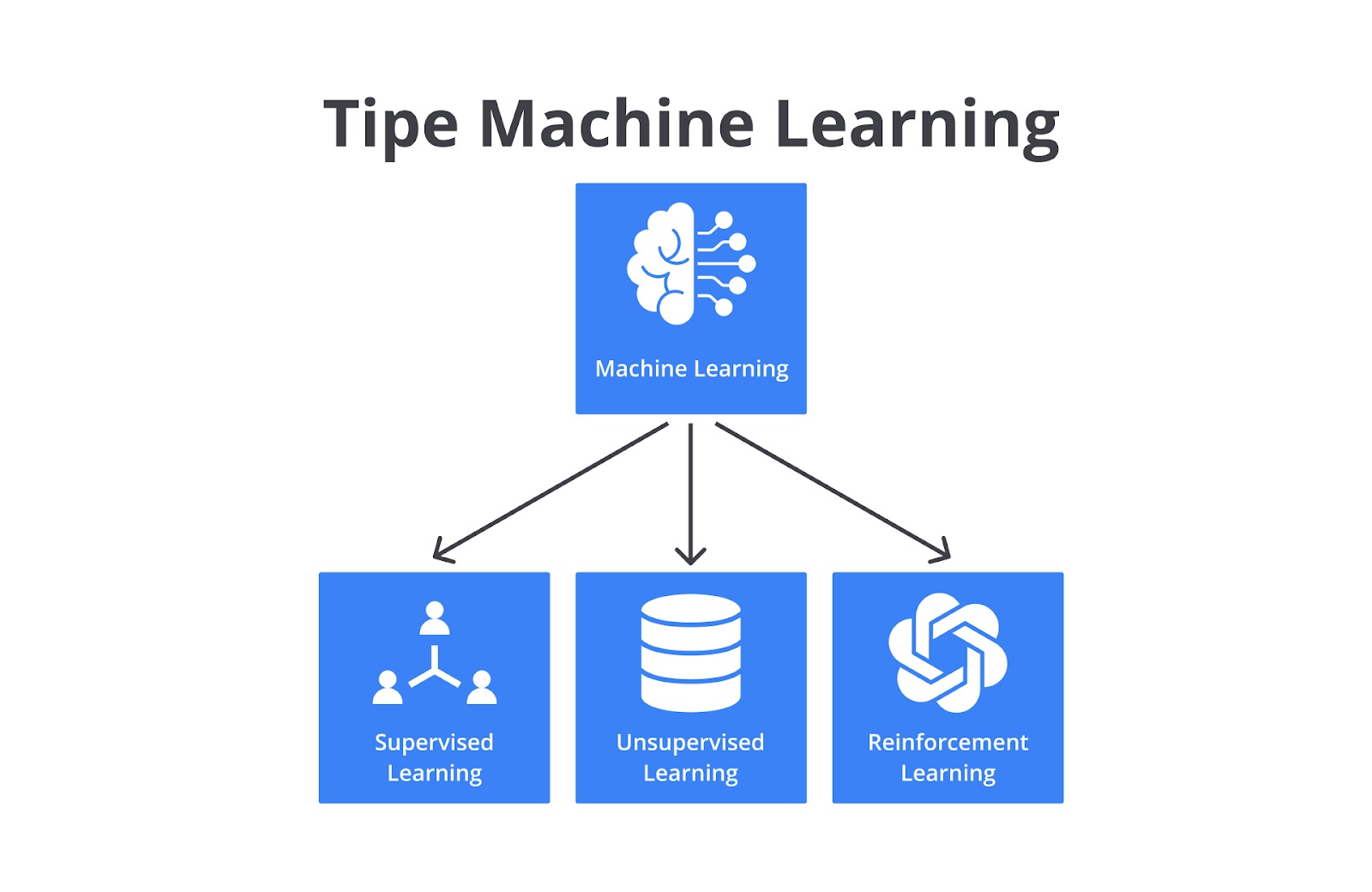
Ketika perusahaan memiliki banyak sekali data dan mengharuskan kita untuk mempelajari data tersebut tentu akan memakan waktu yang cukup lama atau bahkan jika data tersebut berisikan hal yang tabu bagi kita tentu akan membuat pusing tujuh keliling. Dengan menggunakan machine learning kita tidak perlu lagi mempelajari keseluruhan data dan membuat aturan secara ekspilisit. Dengan menggunakan machine learning kita hanya perlu menentukan algoritma sehingga machine learning dapat menentukan aturannya sendiri. Bagaimana caranya?

Algoritma machine learning dapat mencari pola tertentu dari sekumpulan data. Ia menentukan karakteristik data sehingga dapat menyimpulkan sebuah aturan. Perhatikan tanda panah di sebelah kanan diagram paradigma machine learning di atas. Paradigma machine learning menghasilkan aturan atau *rules*sebagai keluarannya.

Selanjutnya, aturan ini dapat digunakan untuk melakukan identifikasi dan prediksi bagi data baru yang relevan dengan model machine learning yang kita miliki. Dengan menerapkan machine learning, kita tidak perlu lagi menuliskan seluruh kondisi yang kompleks tersebut. Sangat efisien dan menarik, ‘kan?

**Tipe-Tipe Machine Learning**

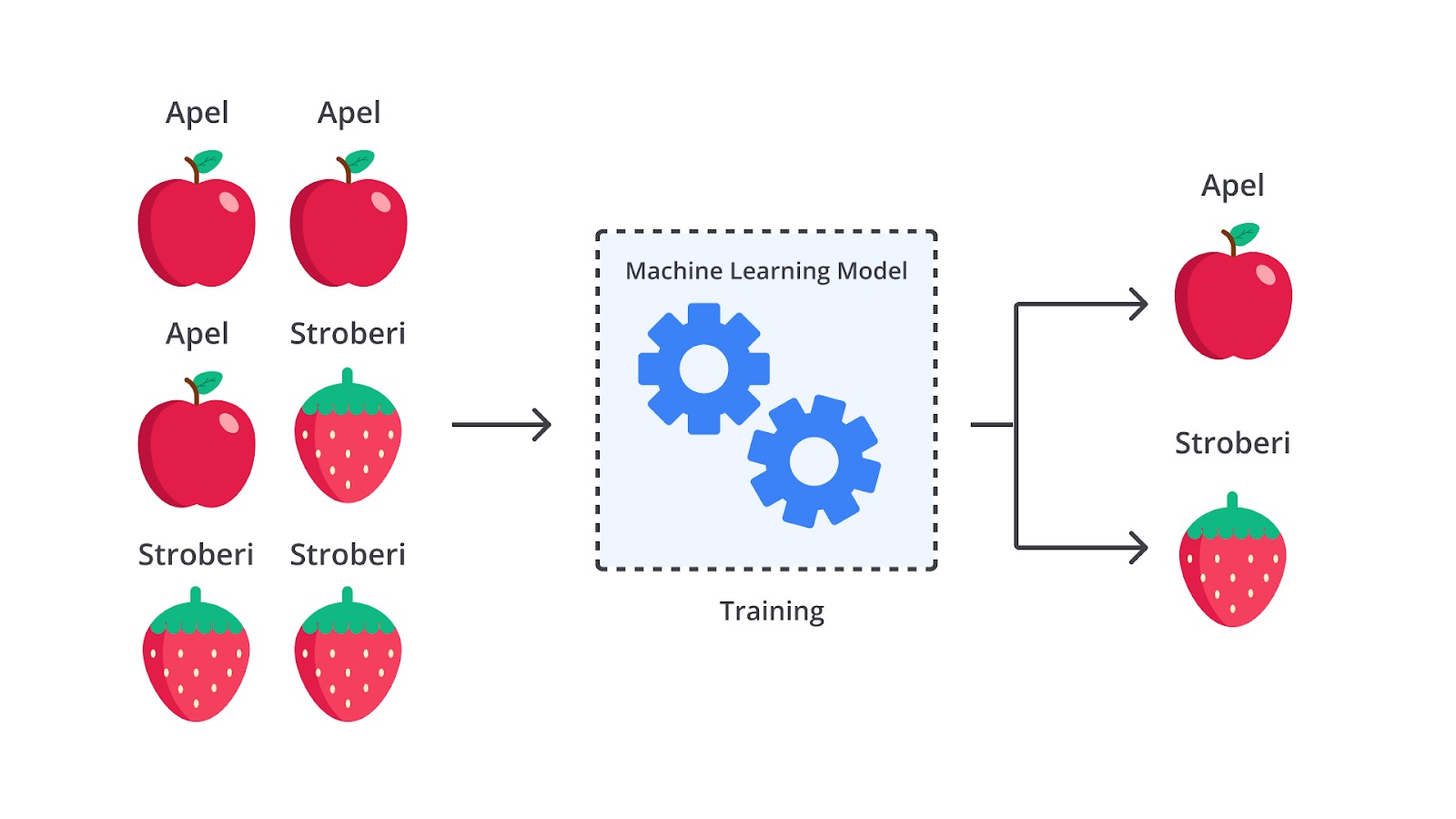
Machine learning dibagi menjadi beberapa kategori. Pada modul ini, kita akan mempelajari tiga kategori yang paling sering digunakan, yaitu supervised learning, unsupervised learning, dan reinforcement learning.



Supervised learning dan unsupervised learning adalah dua kategori yang mungkin familier bagi Anda. Namun, pertanyaannya adalah berdasarkan apakah pembagian kategori tersebut? Benar, pembagian kategori tersebut berdasarkan karakteristik data dan jenis supervisi yang didapatkan oleh program selama pelatihan. Apa maksudnya dan apa bedanya dengan reinforcement learning?

**Supervised Learning**

Supervised learning adalah kategori machine learning yang dalam proses pembelajarannya menggunakan data yang memiliki label atau jawaban. Ketika data dimasukkan ke dalam model machine learning, model akan melakukan perhitungan komputasi terhadap input yang diberikan berdasarkan algoritma yang digunakan. Proses perhitungan komputasi tersebut akan dilakukan secara berulang kali hingga model dapat mengenali karakteristik masing-masing input dan dapat memberikan prediksi label atau jawaban yang sesuai dengan data yang diberikan (data aktual).



Perhatikan gambar di atas, bayangkan Anda memiliki kumpulan data buah-buahan yang terdiri dari apel dan stroberi beserta labelnya. Anda menginginkan model machine learning dapat mempelajari karakteristik kedua buah tersebut berdasarkan data dan label yang sudah ada. Kemudian, setelah model machine learning mempelajari karakteristik dari kedua buah tersebut, Anda mendapatkan sebuah model yang dapat melakukan klasifikasi terhadap buah apel dan stroberi. Akhirnya, saat Anda memberikan gambar dengan karakteristik serupa dengan apel maka model machine learning dapat mengenali gambar tersebut sebagai buah apel.

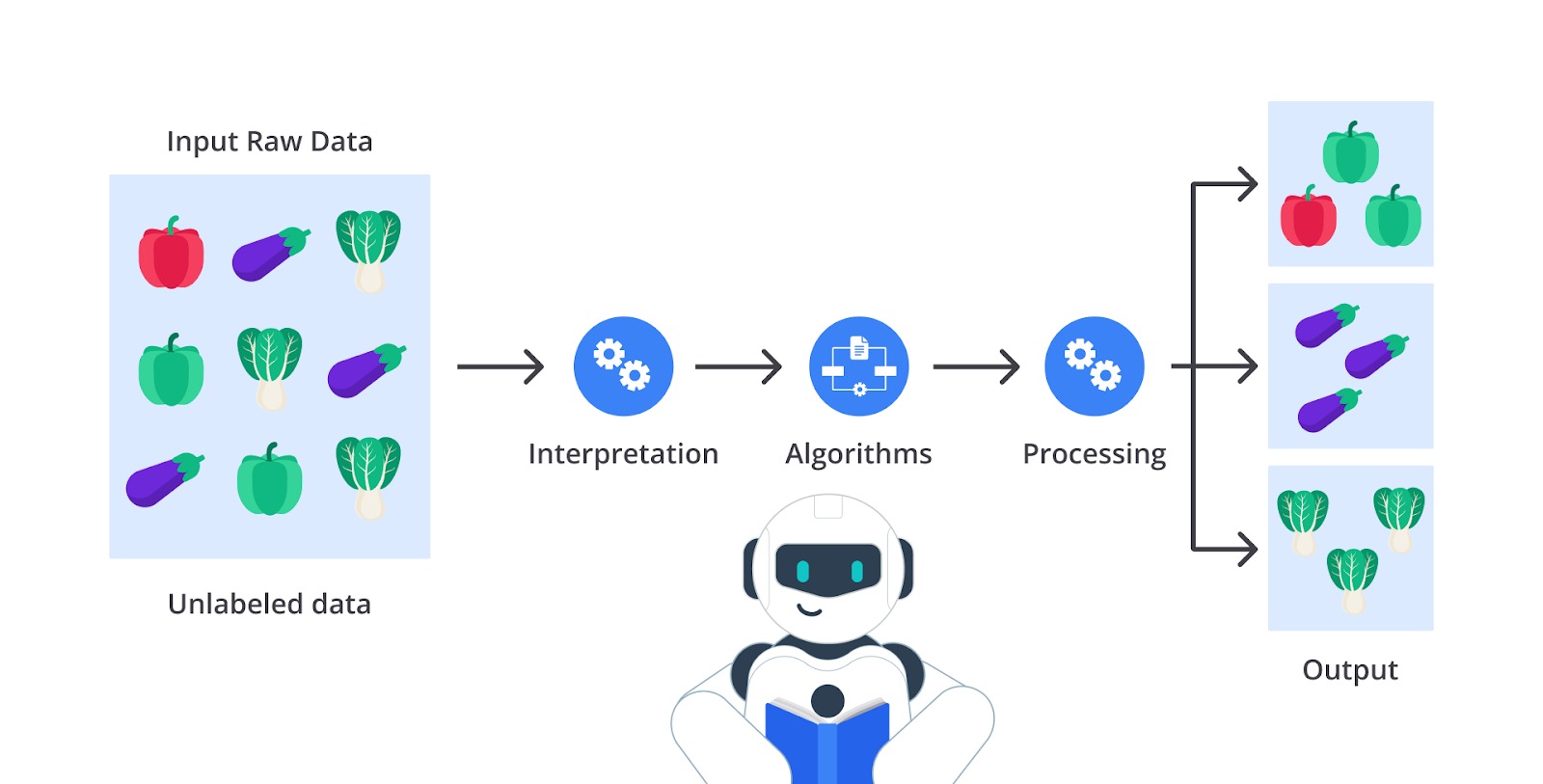
Cerita di atas merupakan salah satu contoh penerapan dari supervised learning. Menarik, bukan? Lebih lanjut Andrew Ng, seorang ilmuwan AI dari Stanford University menyatakan gagasan mengenai supervised learning. Menurutnya, supervised learning merupakan tipe machine learning yang hingga saat ini paling banyak digunakan pada berbagai bidang di industri.

Supervised learning digunakan pada berbagai bidang dikarenakan penggunaannya yang jelas, dengan data beserta jawaban atau label yang sudah dimiliki maka memungkinkan model machine learning mengerjakan tugas yang sudah ditentukan.

Selain itu, supervised learning mudah dipahami dan performa akurasinya pun mudah diukur untuk dievaluasi. Supervised learning dapat membantu permasalahan yang sangat beragam. Dua buah pilar pada supervised learning yaitu *classification*dan *regression* merupakan contoh metode yang paling sering digunakan. Seperti kisah sebelumnya, model tersebut dapat menjalankan tugas untuk melakukan klasifikasi pada data buah yang baru. Selain contoh yang telah disebutkan, supervised learning dapat mengerjakan tugas seperti mengklasifikasikan spam pada pesan, memprediksi harga rumah, dan lain sebagainya.

**Unsupervised Learning**

Berbeda dengan supervised learning, pada unsupervised learning data yang digunakan pada proses pembelajarannya tidak memiliki jawaban atau label. Sehingga, unsupervised learning melakukan proses belajar sendiri untuk melabeli atau mengelompokkan data.



Perhatikan gambar di atas secara saksama, Anda mungkin sudah dapat menduga bahwa dataset yang digunakan tidak memiliki label bahkan masih berupa sayuran yang tercampur. Betul, kemampuan metode ini dapat menemukan kesamaan dan perbedaan pada dataset sehingga membuatnya ideal untuk analisis data eksplorasi, strategi penjualan silang, segmentasi pelanggan, dan pengenalan gambar dan pola seperti contoh di atas.

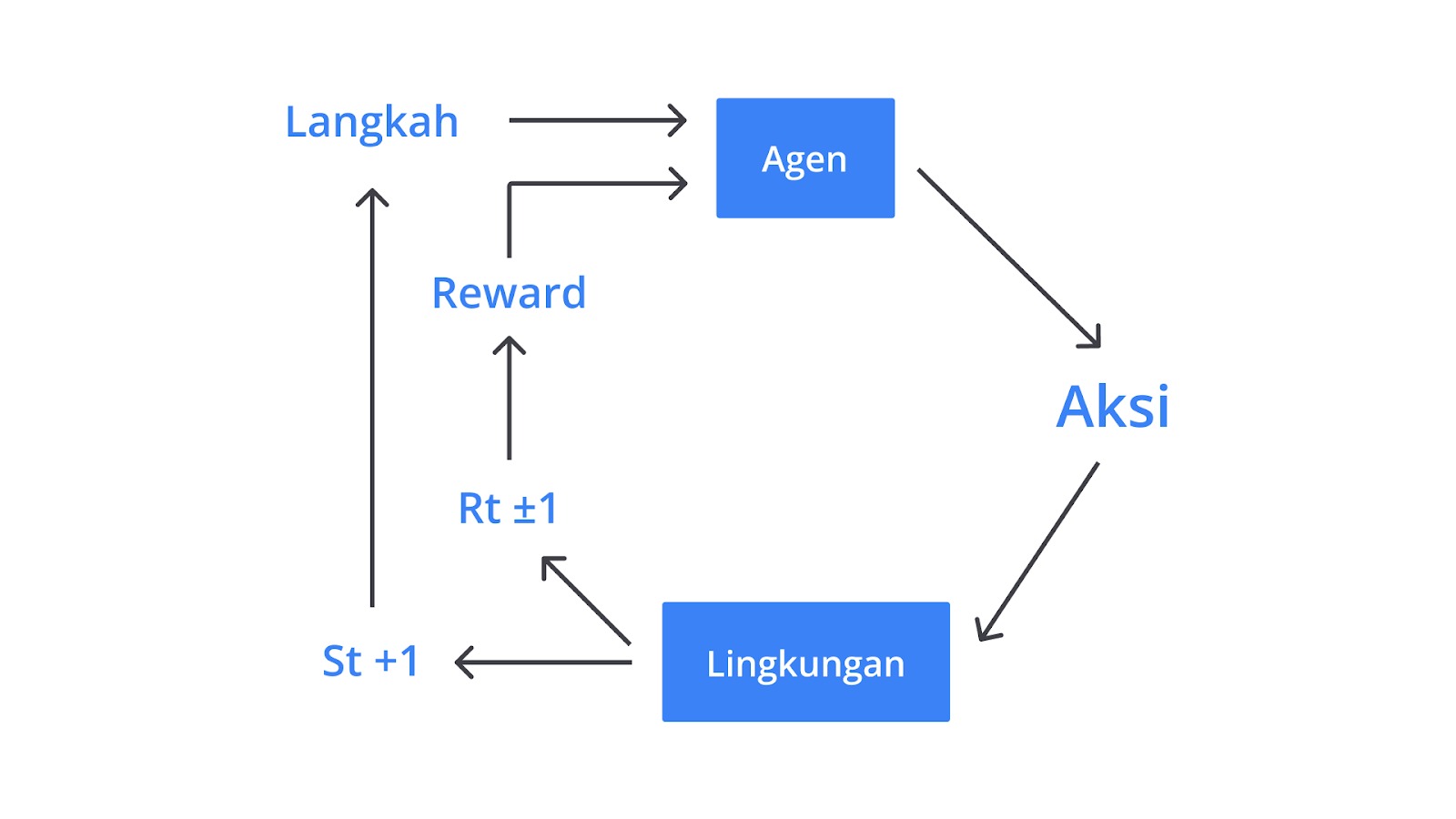
Mungkin rasa ingin tahu mengenai bagaimana unsupervised learning dapat mengeksplorasi data secara mandiri tumbuh di benak Anda. Unsupervised learning akan melakukan proses interpretasi yaitu menemukan sebuah pola berdasarkan karakteristik dataset yang ada, tahapan ini dibantu oleh algoritma yang dapat melakukan perhitungan komputasi untuk membantu menemukan pola atau struktur yang ada di dalamnya.

Akhirnya, model unsupervised learning dapat melakukan processing sehingga dapat mengelompokkan dataset dari yang semula masih tercampur menjadi beberapa kelompok dengan karakteristik yang serupa.

Dengan kata lain, berdasarkan cerita di atas unsupervised learning dapat dilihat sebagai robot/mesin yang berusaha belajar menjawab pertanyaan secara mandiri tanpa ada jawaban yang disediakan manusia.

**Reinforcement Learning**

Reinforcement learning adalah teknik yang mempelajari bagaimana membuat keputusan terbaik secara berurutan untuk memaksimalkan ukuran sukses kehidupan nyata. Entitas pembuat keputusan atau yang biasa disebut *agent* belajar melalui proses *trial*dan *error.*Bagaimana maksudnya? Perhatikan gambar berikut.



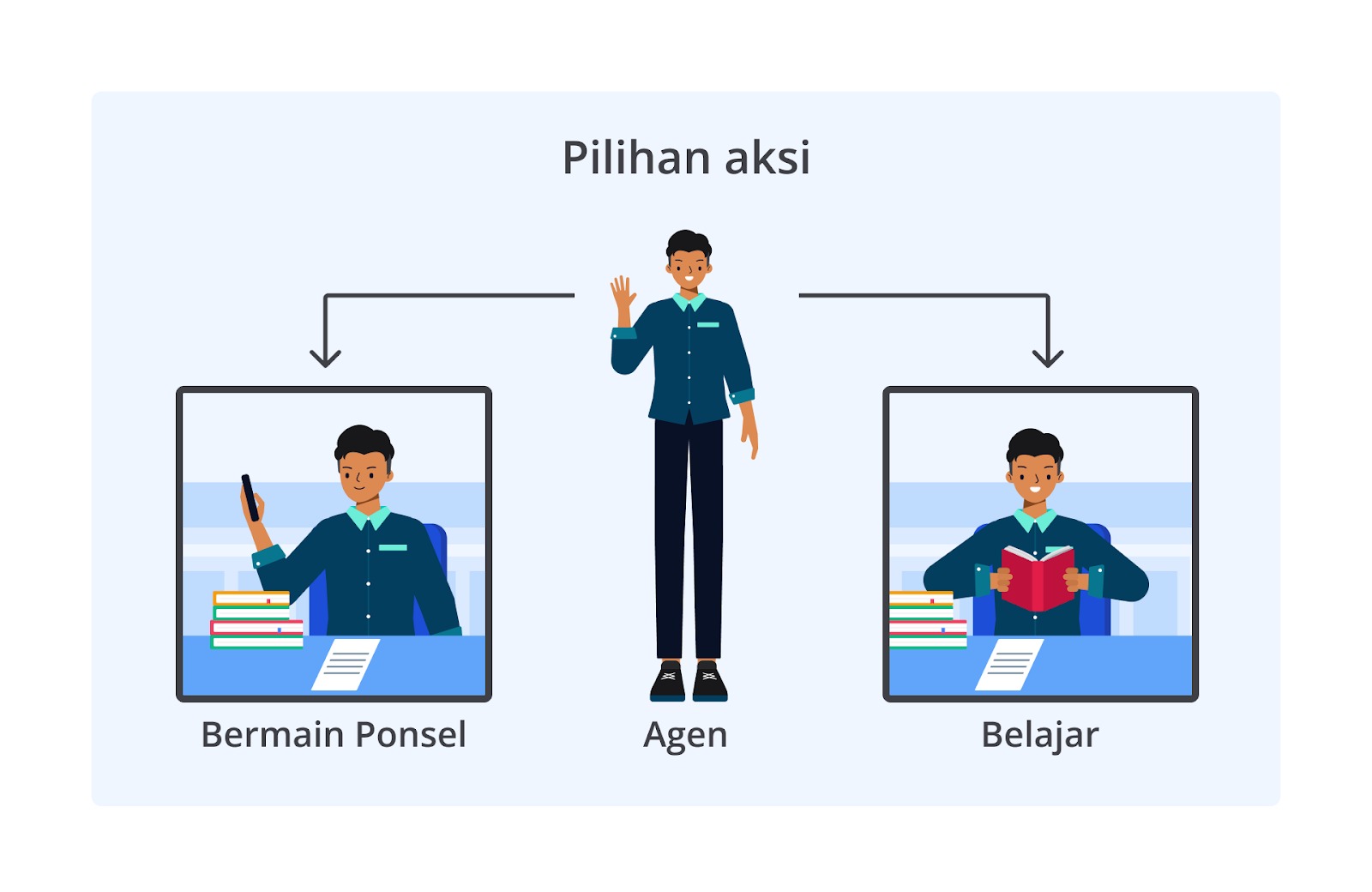
Gambar di atas merupakan cara kerja reinforcement learning. Pada Reinforcement Learning sebuah perangkat lunak yang disebut *agent* membuat pengamatan terhadap lingkungan dan melakukan aksi-aksi yang sesuai dari pengamatan yang dilakukan.

Tujuan dari *agent* adalah untuk melakukan tindakan sehingga menghasilkan keputusan terbaik, secara berurutan, untuk memaksimalkan ukuran sukses kehidupan nyata. Dengan kata lain, *agent* memiliki tujuan untuk melakukan tindakan yang menghasilkan *reward*/hadiah. *Reward* akan diberikan jika agent berhasil melakukan tindakan yang telah ditentukan oleh developer dengan rumus Rt+1. Sebaliknya, ketika agent melakukan kesalahan maka akan diberi sanksi atau pengurangan point (St+1).

Tujuan dari *agent* adalah untuk melakukan tindakan-tindakan yang terus menambah reward/hadiah. Analoginya seperti teorema hadiah dan hukuman. Kita sebagai manusia cenderung melakukan hal yang membuat kita senang (hadiah) dan menghindari kegiatan yang membuat kita menderita (hukuman). Demikian halnya dengan agen pada RL yang berusaha untuk mendapatkan banyak hadiah dan meminimalisasi hukuman.

Agar kalian dapat memahami konsep reinforcement learning lebih jelas, mari kita simak ilustrasi cerita berikut.

Kai merupakan seorang mahasiswa yang belum memiliki informasi terkait peraturan yang ada di kampusnya dan ia berperan sebagai *agent* pada kasus kali ini. Kai ingin mempelajari peraturan kelas sebagai lingkungan barunya. Kelas tersebut memiliki peraturan untuk tidak menggunakan handphone ketika jam pelajaran berlangsung. Dengan adanya peraturan tersebut, Kai harus belajar untuk menaati peraturan tersebut sehingga ia harus belajar fokus di kelas dan tidak boleh menggunakan handphone.



Karena Kai belum mengetahui apa yang harus dilakukannya di kelas, ia harus membuat *policy* di awal secara acak. *Policy* adalah panduan untuk agent melakukan aksi yang sesuai dengan lingkungannya. Nah, berdasarkan policy yang dibuat, agent akan melakukan aksi sesuai dengan policy tersebut. Setelah aksi dikerjakan, agent akan mendapatkan hadiah atau penalti.

Pada aksi pertama, agent memilih menggunakan handphone sehingga mendapatkan penalti yaitu pengurangan poin.

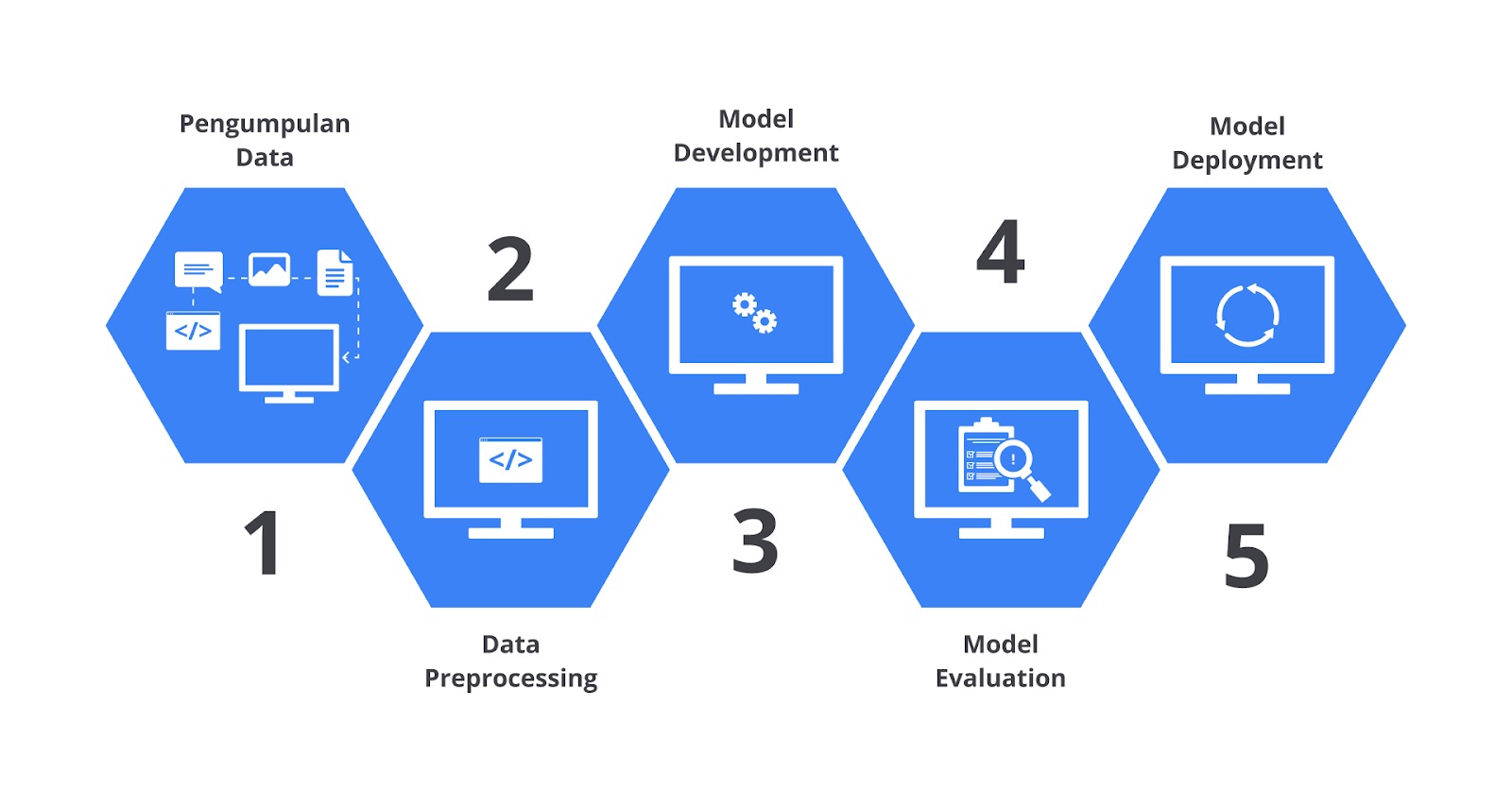


Karena Kai mendapatkan penalti, ia sebagai agent akan mengingatnya dan tentu menghindari penalti tersebut di masa depan. Setelah mendapat penalti, agent akan memperbarui *policy*-nya hingga ia bisa mendapatkan hadiah/poin sebanyak-banyaknya.

Nah, sekarang Anda sudah mengetahui tipe-tipe machine learning. Setelah ini, kita akan mempelajari tahapan-tahapan yang perlu dilalui sebelum kita membuat project machine learning dengan berbagai tipe.

**Machine Learning Workflow**

Pada modul ini kita akan membahas beberapa tahapan machine learning workflow mulai dari pengumpulan data, data *preprocessing*, model *development*, model *evaluation* hingga model *deployment*.



Machine learning workflow memiliki tahapan iteratif yang berarti prosesnya bisa berulang sesuai dengan kebutuhan. Anda dapat mengevaluasi ulang proses yang Anda jalankan dan kembali ke langkah sebelumnya ketika dibutuhkan. Machine learning worfklow umumnya memiliki lima tahapan yang berkesinambungan. Mari kita uraikan bersama tahapan umum pada proses pembuatan machine learning bersama-sama.

**Pengumpulan Data**

Proses pengumpulan data dapat dilakukan dengan mengambil dataset dari sumber yang sudah disediakan, seperti dari *kaggle*, *zenodo*, UCI, dan lain sebagainya. Dari sumber-sumber tersebut, Anda dapat memilih, mengunduh, dan menggunakan dataset yang sesuai dengan kebutuhan Anda. Proses ini relatif mudah. Tantangannya adalah memilih dataset yang tepat untuk model Anda.

Namun, jika Anda adalah seorang Machine Learning Engineer pada sebuah perusahaan yang bertugas untuk membangun model ML untuk keperluan perusahaan, tentu proses pengumpulan datanya tidak semudah mengunduh dataset yang sudah jadi. Anda perlu mengumpulkan dan mengekstrak sendiri data dari berbagai sumber, seperti dari database, file, data sensor, dan sumber lainnya.

Pada tahap ini, Anda juga perlu berurusan dengan berbagai jenis tipe data mulai dari structured data (seperti excel file atau database SQL), hingga unstructured data (seperti text file, email, video, audio, gambar, data sensor, dan lainnya). Seorang analis yang bekerja pada sebuah perusahaan yang melakukan riset dan penasehatan global menyatakan bahwa lebih dari 80% data yang digunakan saat ini adalah unstructured data.

Namun, pada tahap ini dataset yang Anda miliki masih mentah. Agar dapat digunakan dengan maksimal pada pembangunan machine learning, dataset harus dikelola sedemikian rupa sesuai dengan tujuan pembangunan model machine learning. Tahapan tersebut disebut preprocessing data.

**Data Preprocessing**

Data preprocessing adalah tahapan pengolahan data lebih lanjut sehingga menjadi lebih siap dalam pengembangan model machine learning. Dengan kata lain, proses ini mengubah fitur-fitur data ke dalam bentuk yang mudah diinterpretasikan dan diproses oleh algoritma machine learning. Data preprocessing memiliki beberapa bagian, yaitu data cleaning, data transformation, dan data integration.

Beberapa hal yang bisa dilakukan dalam proses data cleaning antara lain

* penanganan missing value,
* data yang tidak konsisten,
* duplikasi data,
* ketidakseimbangan data,
* dan lain sebagainya.

Selain itu, ada juga beberapa hal yang bisa dilakukan untuk proses transformasi data seperti

* scaling atau mengubah skala data agar sesuai dengan skala tertentu,
* standarisasi,
* normalisasi,
* mengonversi data menjadi variabel kategori,
* dan sebagainya.

Lalu, bagaimana dengan data integration? Berikut beberapa hal yang bisa dilakukan pada tahap data integration.

* Menggabungkan dataset.
* Menghilangkan fitur yang duplikat.
* Menyamakan format; dan lain sebagainya.

**Model Development**

Setelah kita memiliki data yang sudah siap digunakan, mari kita mulai proses model development. Pada konteks machine learning, model development bisa berarti dua hal: pemilihan *learning method* atau algoritma ML; dan pemilihan *hyperparameter* terbaik untuk metode machine learning yang dipilih.

Ada seorang peneliti bernama K.P Murphy menuliskan sebuah kalimat,

*“When we have a variety of models of different complexity (e.g., linear or logistic regression models with different degree polynomials, or KNN classifiers with different values of K), how should we pick the right one?”*[17]*.*

Berangkat dari pertanyaan tersebut, menentukan model yang sesuai dengan data yang kita miliki merupakan tahapan yang penting dalam machine learning workflow.

Jie Ding, *et al* dalam tulisannya “*Model Selection Techniques -An Overview*” [5] menyatakan bahwa tidak ada model yang cocok secara universal untuk data dan tujuan apa pun. Pilihan model atau metode yang tidak tepat dapat menyebabkan kesimpulan yang menyesatkan atau performa prediksi yang mengecewakan. Sebagai contoh, saat memiliki kasus klasifikasi biner, kita perlu mempertimbangkan model terbaik untuk data kita, apakah logistic regression atau SVM classifier.

Setelah kita menentukan metode yang cocok untuk data yang ada, kita perlu mengubah hyperparameter untuk mendapatkan performa terbaik dari model. Hyperparameter di sini merupakan variabel yang digunakan untuk mengontrol proses pelatihan model, contohnya seperti *epochs*, *optimizer*, dan lain sebagainya.

Dengan mengubah nilai hyperparameter saat kita menjalankan algoritma ML akan memberikan hasil atau performa model yang berbeda. Proses menemukan performa terbaik model dengan pengaturan hyperparameter yang berbeda ini juga disebut model development.

**Model Evaluation**

Setelah mengutak-atik model dengan hyperparameter yang berbeda, akhirnya Anda mendapatkan model yang kinerjanya cukup baik. Langkah selanjutnya adalah mengevaluasi model akhir pada data uji.

Sederhananya, langkah evaluasi model dapat dijabarkan sebagai berikut.

* Memprediksi label pada data uji.
* Menghitung jumlah dari hasil prediksi berdasarkan data uji. Pada tahap ini menghitung keseluruhan kondisi yang ada, seperti jumlah dari kondisi prediksi yang memiliki status benar ataupun salah.
* Membandingkan hasil prediksi dengan data label yang kita miliki. Dari data perbandingan ini, kita dapat menghitung akurasi atau performa model.

Pada prinsipnya proses model evaluation adalah menilai kinerja model ML pada data baru, yaitu data yang belum pernah “dilihat” oleh model sebelumnya. Evaluasi model bertujuan untuk membuat estimasi generalisasi error dari model yang dipilih, yaitu seberapa baik kinerja model tersebut pada data baru.

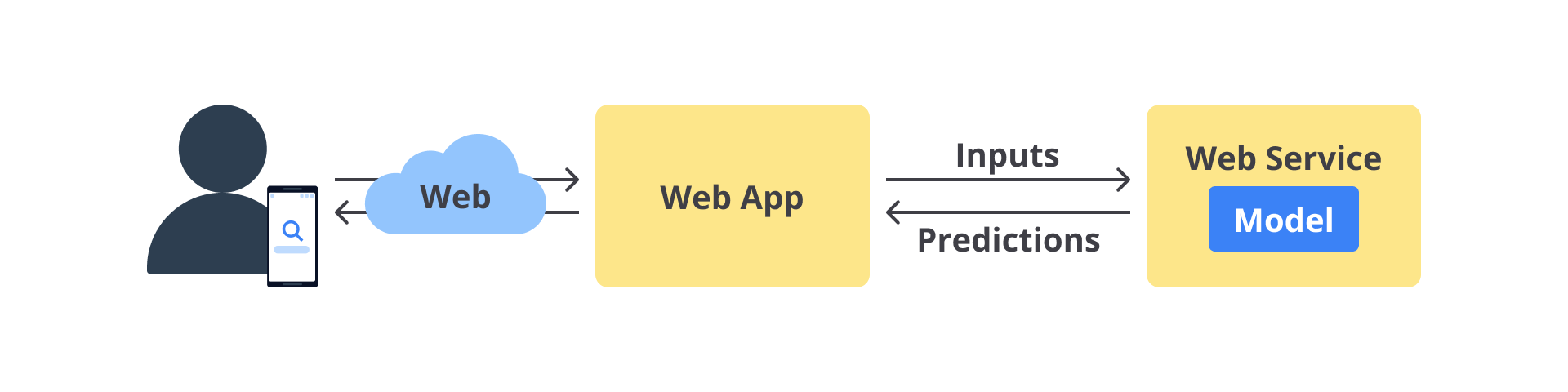
Idealnya, model machine learning yang baik adalah model yang tidak hanya bekerja dengan baik pada data training, tetapi juga pada data baru. Oleh karena itu, sebelum mengirimkan model ke tahap produksi, Anda harus cukup yakin bahwa performa model akan tetap baik dan tidak menurun saat dihadapkan dengan data baru.

**Model Deployment**

*Bonjour*! Sampai sini kita telah belajar mulai dari pengumpulan data, data preprocessing, model development, hingga model evaluation. Bagaimana perjalanan Anda, menyenangkan, bukan? Selamat, kita telah sampai pada tahapan terakhir machine learning workflow yaitu model deployment.

Setelah model dievaluasi, model siap untuk dipakai pada tahap produksi yang biasanya disebut model deployment. Caranya adalah dengan menyimpan model yang telah dilatih dari tahap preprocessing hingga pipeline prediksi. Selanjutnya, jangan lupa untuk membawa seluruh tahapan pada data preprocessing untuk digunakan pada tahapan prediksi. Simpan seluruh tahapan yang dilakukan pada data preprocessing pada sebuah fungsi bernama *predict* yang berguna untuk memproses data baru. Kemudian deploy model tersebut ke sebuah platform seperti web, mobile, IoT dan lain sebagainya. *Last but not least*, kita dapat membuat prediksi dengan memanggil fungsi **predict()** yang sebelumnya telah dibuat. Terdengar sangat mudah, bukan?

Agar penjelasan di atas lebih terbayang mari kita simak gambar berikut. Geron [6] memberikan contoh ilustrasi model deployment seperti tampak dalam gambar berikut.



Untuk lebih memahami gambar di atas mari kita analogikan dengan sebuah studi kasus regresi. Sebuah model regresi untuk menentukan harga rumah akan digunakan dalam situs web. Pengguna akan mengetik beberapa data tentang lokasi yang diinginkan dan mengeklik tombol “Prediksi Harga”. Proses ini akan mengirimkan query yang berisi data ke server, kemudian meneruskannya ke aplikasi web Anda. Terakhir, kode akan memanggil fungsi predict() untuk memberikan hasil prediksi pada Anda.

Bagaimana semakin dalam makin seru, ‘kan?

Dalam pengembangannya, machine learning melibatkan cukup banyak proses dan infrastruktur untuk membangun modelnya. Memahami algoritma machine learning memang penting, tetapi memahami keseluruhan proses machine learning hingga ke tahap produksi juga tak kalah pentingnya. Pada tahap selanjutnya, kita akan mempelajari bagaimana cara merawat dan memperbaharui model agar tetap memiliki kualitas yang baik.

**Model Maintenance**

Umumnya sebuah model yang di-deploy kinerjanya akan turun seiring waktu. Penurunan tersebut terjadi karena model akan terus menemui lebih banyak data baru seiring waktu sehingga menyebabkan akurasi model menurun.

Sebagai contoh dalam kehidupan sesungguhnya, bayangkan kita memiliki sebuah model machine learning yang dapat memprediksi harga sebuah rumah berdasarkan beberapa fitur seperti jumlah kamar, luas tanah dan lain sebagainya. Model machine learning ini dilatih dengan data penjualan pada tahun 2010 dan telah melakukan prediksi dengan baik selama satu tahun kedepan. Namun, seiring berjalannya waktu model ini mengalami penurunan performa sehingga akan menghasilkan prediksi yang buruk pada tahun-tahun berikutnya. Penurunan performa ini dapat diakibatkan oleh banyak faktor, seperti kebiasaan pelanggan, inflasi, atau faktor alami seperti bencana alam. Penurunan performa akan diketahui ketika kita melakukan evaluasi sehingga kita harus melakukan optimasi pada model tersebut.

Untuk mengatasi masalah ini, ada 2 teknik dasar dalam menjaga agar model selalu bisa belajar dengan data-data baru. Dua teknik tersebut yaitu manual retraining dan continuous learning.

**Manual Retraining**

Teknik pertama adalah melakukan ulang proses pelatihan model dari awal sehingga data-data baru yang ditemui di tahap produksi akan digabung dengan data lama. Selanjutnya, model dilatih ulang dari awal menggunakan data lama dan data baru.

Salah satu kekurangan dari proses ini adalah dari sisi waktu dan efektivitas. Proses ini membutuhkan effort yang lebih untuk dilakukan mulai dari pengumpulan data, data preprocessing, penggabungan data lama dengan data baru, hingga proses training. Bayangkan ketika kita harus melatih ulang model dalam jangka waktu mingguan atau bahkan harian, tentunya proses tersebut akan memakan waktu yang lama, bukan?

Proses manual retraining ini memang membutuhkan waktu yang lama dan effort yang lebih banyak. Namun, manual retraining memungkinan kita menemukan model-model baru atau atribut-atribut baru yang menghasilkan performa lebih baik di masa kini bahkan di masa yang akan datang.

**Continuous Learning**

Teknik kedua untuk menjaga model kita up-to-date adalah continuous learning yang menggunakan sistem terotomasi dalam pelatihan ulang model. Berikut alur dari continuous learning.

1. Menyimpan data-data baru yang ditemui pada tahap produksi. Contohnya, ketika sistem mendapatkan harga emas naik, data harga tersebut akan disimpan di database.
2. Ketika data-data baru yang dikumpulkan cukup, lakukan pengujian akurasi dari model terhadap data baru.
3. Jika akurasi model menurun seiring waktu, gunakan data baru, atau kombinasi data lama dan baru untuk melatih dan men-deploy ulang model.

Sesuai namanya, 3 proses di atas dapat terotomasi sehingga kita tidak perlu melakukannya secara manual.

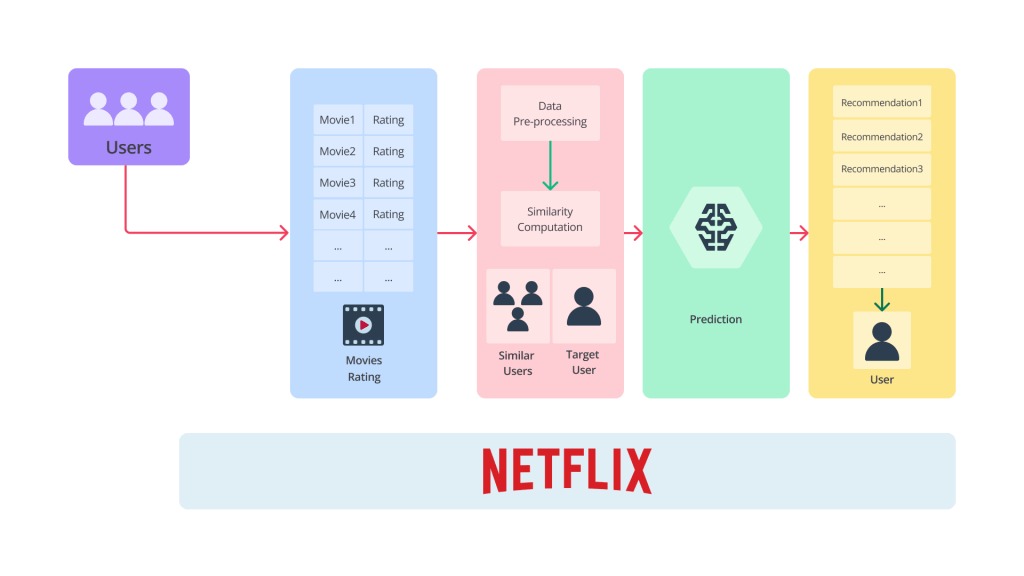
Dengan merawat dan melakukan pembaruan pada model yang telah di-deploy, ini membuat tugas yang dilakukan oleh machine learning tetap konsisten dan tepat sesuai dengan kondisi dan situasi di lapangan.

Setelah melewati perjalanan yang cukup panjang ini, bagaimana perasaan Anda? Semakin penasaran kan terkait pengembangan machine learning? Agar semakin paham terkait materi ini selanjutnya kita akan mempelajari contoh penerapan machine learning di industri.

**Machine Learning di Industri**

Pada era digital saat ini, banyak sekali perusahaan yang menerapkan machine learning untuk membantu menyelesaikan pekerjaannya. Salah satu contoh penerapan AI adalah sistem rekomendasi yang dikembangkan oleh Netflix. Netflix adalah layanan streaming berbasis langganan yang memungkinkan pengguna menonton acara TV dan film di perangkat yang terhubung ke Internet.

Netflix menggunakan machine learning untuk merekomendasikan film kepada pengguna mereka melalui sistem rekomendasi yang mereka bangun. Sistem ini berperan penting dalam meningkatkan pengalaman pengguna dengan menyediakan rekomendasi yang disesuaikan dengan preferensi masing-masing pengguna. Berikut adalah gambaran dari [sistem rekomendasi yang dikembangkan Netflix](https://research.netflix.com/research-area/recommendations" \t "_blank).



Rekomendasi sistem yang dibangun oleh Netflix berjalan sesuai dengan tugasnya, hal ini dibuktikan oleh meningkatnya waktu streaming setelah menerapkan rekomendasi sistem tersebut. Menurut Todd Yellin sebagai wakil presiden dari divisi product [mengatakan](https://mobilesyrup.com/2017/08/22/80-percent-netflix-shows-discovered-recommendation/" \t "_blank) bahwa lebih dari 80% pengikut memercayai dan mengikuti hasil rekomendasi yang diberikan oleh sistem rekomendasi Netflix. Di sisi lain, Netflix percaya bahwa mereka dapat kehilangan hingga [1 miliar dollar](https://businessinsider.com/netflix-recommendation-engine-worth-1-billion-per-year-2016-6) setiap tahunnya jika tidak menggunakan sistem rekomendasi tersebut.

Hal tersebut menjadi sangat masuk akal dikarenakan setiap kali pengguna menggunakan Netflix, rekomendasi sistem tersebut akan memberikan bantuan untuk mencari film kepada pengguna dengan usaha yang ringan. Sebagai pengguna, kita tidak perlu lagi melakukan *scrolling* dengan waktu yang lama, berdasarkan perhitungan yang dilakukan oleh Netflix mereka menyatakan bahwa butuh waktu kurang lebih 90 detik untuk mendapatkan perhatian pengguna untuk memilih film. Hal tersebut yang membuat pengguna terus berlangganan dan Netflix dapat menghindari kerugian karena dampak dari ditinggalkan oleh pelanggan.

Apakah Anda penasaran ingin mengetahui cara rekomendasi Netflix ini bekerja? Mari kita gali sedikit lebih dalam agar lebih memahami salah satu penerapan machine learning yang sudah ada di industri saat ini.

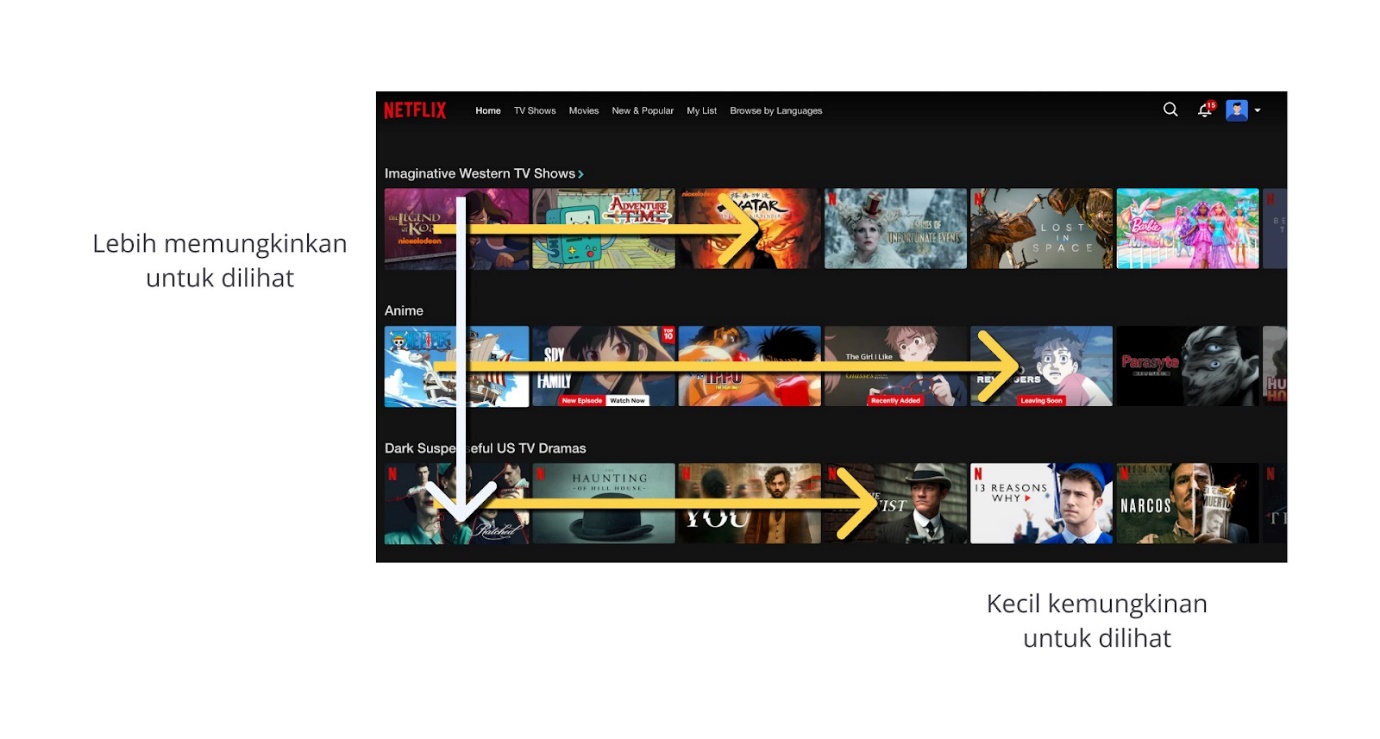
Netflix menggunakan beberapa data yang diambil dari masing-masing pengguna sehingga dapat memperbesar kemungkinan pengguna tersebut menonton film karena sesuai dengan karakteristiknya. Faktor yang diambil dan digunakan pada sistem rekomendasi Netflix yaitu

* Rating yang diberikan,
* Riwayat pencarian atau menonton,
* Kesamaan dengan pengguna yang lainnya,
* Informasi terkait film yang ditonton,
* Jumlah waktu menonton dalam sehari,
* Gawai yang digunakan,
* Waktu regional ketika pengguna menonton (siang/sore/malam).

Setelah data yang dibutuhkan untuk menghasilkan rekomendasi dikumpulkan, selanjutnya data tersebut akan digunakan untuk menjadi *input* yang akan diproses oleh algoritma. Hasil dari proses itulah yang nantinya akan digunakan oleh Netflix. Lalu, muncul di benak kita semua “bagaimana Netflix mengolah data tersebut serta algoritma apa yang digunakannya?” Mari kita cari tahu bersama algoritma apa yang digunakan Netflix sehingga dapat menghasilkan output yang sangat baik sebelumnya.

Pada tulisannya, Netflix menyebutkan bahwa mereka menggunakan beberapa macam algoritma untuk menghasilkan rekomendasi sistem, tetapi tidak menjelaskan arsitekturnya secara detail. Beberapa algoritma yang digunakan Netflix, antara lain *personalised video ranking*(PVR), *top-N Video ranker*, *trending now ranker*, *continue watching ranker*, dan *video-video similarity ranker*. Jika Anda ingin mengetahui lebih detail dari masing-masing algoritma tersebut, [pelajari di sini, ya](https://dl.acm.org/doi/10.1145/2843948" \t "_blank).

Setelah algoritma menjalankan tugasnya, Netflix akan memberikan hasil rekomendasinya (*output*) berupa urutan film yang cocok dengan karakteristik penggunanya. Netflix memberikan hasil rekomendasinya dengan sistem peringkat berbasis dua tingkat yaitu pada kolom (rekomendasi terbaik dari kiri) dan baris (rekomendasi terbaik dari atas)



Mengapa Netflix menggunakan baris dan kolom pada hasil rekomendasinya? Hal tersebut dijelaskan bahwa ada keuntungan dari dua buah sudut pandang yang berbeda yaitu pengguna dan perusahaan. Sebagai pengguna, itu membuat rekomendasi yang disajikan lebih koheren karena memiliki hubungan yang erat pada baris atau kolom yang sama. Selain itu, sebagai perusahaan akan mudah mendapatkan feedback ketika pengguna terindikasi tidak menyukai urutannya dengan cara melakukan *scroll* dan mengabaikan rekomendasinya.

Berdasarkan sudut pandang perusahaan tersebut, Netflix melakukan model maintenance berdasarkan feedback dari pengguna agar dapat memberikan rekomendasi yang relevan setiap waktunya. Netflix akan melakukan pelatihan ulang model untuk meningkatkan akurasi dari prediksi yang diberikan berdasarkan film yang paling sering ditonton oleh pengguna.

Bagaimana contoh salah satu penerapan machine learning di industri ini? Sangat mengagumkan, bukan? Bayangkan jika kalian membuat sebuah sistem machine learning yang berguna bagi kehidupan banyak orang, tentu sangat menyenangkan. Namun, sebelum kalian membayangkan dan berimajinasi terlalu jauh terkait implementasi machine learning, perlu diketahui bahwa saat ini machine learning masih memiliki batasan. Batasan tersebut akan kita bahas di materi berikutnya. Yuk, kita lanjutkan pembahasannya.

**Batasan Machine Learning**

Anda tentu sudah mempelajari berbagai macam pengetahuan tentang machine learning mulai dari pengenalan, tipe-tipe machine learning, machine learning workflow, model maintenance, hingga contoh penerapannya.

Mari kita mundur sejenak pada penerapan machine learning. Anda telah mempelajari salah satu contoh penerapan machine learning yaitu sistem rekomendasi Netflix. Sistem tersebut dapat memberikan rekomendasi dalam waktu yang sangat cepat. Bayangkan jika kita sebagai manusia harus melakukan perhitungan untuk menentukan rekomendasi terhadap pengguna yang berbeda-beda, sangat memakan waktu dan tenaga, bukan?

Penerapan machine learning tersebut menunjukkan hal yang bisa dilakukan oleh machine learning saat ini. Sayangnya, kita hanya melihat serangkaian kisah sukses dan jarang sekali membaca kisah kegagalan machine learning.

Kegagalan tersebut disebabkan oleh ekspektasi pengguna yang terlalu tinggi terhadap machine learning dan meminta programmer untuk membuat hal-hal yang tidak dapat dilakukan oleh machine learning saat ini. Tujuan dari materi ini untuk mengasah intuisi Anda tentang apa yang mungkin dapat dibangun berdasarkan permasalahan di sekitar atau bahkan perusahaan.

Salah satu batasan machine learning saat ini adalah rasa empati yang tidak dimilikinya. Bayangkan ketika Anda akan membalas sebuah email yang berisi tentang keluhan seorang pelanggan yang akan berulang-tahun. Email balasan tersebut akan berisi tentang rasa prihatin karena keluhan yang dirasakan oleh pelanggan beserta ucapan selamat karena berulang tahun.

Akan mudah rasanya jika kita membalas pesan tersebut, tetapi hal tersebut menjadi rumit bagi machine learning karena mesin tidak memiliki empati dan akan membalas sesuai dengan data yang telah dipelajari sebelumnya.

Batasan tersebut merupakan salah satu contoh hal yang tidak bisa dilakukan oleh machine learning saat ini. Kami harap dengan memberikan salah satu contoh batasan pada machine learning, akan dapat membantu mengasah intuisi Anda tentang apa yang dapat dan tidak dapat dilakukan oleh machine learning.

Dengan mempelajari batasan pada machine learning, Anda dapat membayangkan beberapa hal di perusahaan atau kehidupan sehari-hari yang mungkin layak dan tidak layak untuk dieksplorasi lebih lanjut terhadap pengembangan machine learning. Setelahnya, dengan mempelajari materi ini, Anda dapat meningkatkan peluang memilih proyek yang layak dan sesuai dengan permasalahan yang saat ini dihadapi.

Sampai di sini, bagaimana menurut Anda? Batasan yang ada pada machine learning saat ini perlu kita pelajari agar tidak salah menentukan proyek dan terlalu jauh berimajinasi yang akhirnya akan menyulitkan kita pada proses pengembangan machine learning.

Selamat, Anda telah menyelesaikan modul Pengantar Machine Learning ini! Semoga ilmu ini membuat cakrawala pemahaman Anda seputar dunia ML semakin luas. Agar lebih semangat, kita akan bersama-sama mempelajari hal yang lebih seru di modul berikutnya. *Yap*, deep learning! Sampai jumpa dan tetap semangat, ya.

**[Story] Harapan Menjadi Kenyataan**

Saat ini, Kai sudah memiliki pengetahuan terkait AI dan data yang mana keduanya saling memiliki hubungan. Namun, Kai ingat bahwa AI yang ia pelajari sebelumnya sangatlah luas. Akhirnya, ia mencoba untuk membuat AI dimulai dari machine learning untuk mengatasi masalah dalam kehidupan sehari-harinya, yaitu mengklasifikasi data konsumen tokonya.



Dengan harapan yang sangat besar terhadap AI, Kai mulai mempelajari salah satu bidang keilmuan AI ini lebih jauh. Ia menyadari bahwa permasalahan yang ingin diselesaikan saat ini dapat dilakukan oleh machine learning.

Permasalahan yang muncul saat ini, Kai hanya mengetahui machine learning sebagai salah satu bidang keilmuan pada AI. Ia tidak mengetahui secara dalam terkait apa itu machine learning dan proses pengembangannya. Selain itu, beberapa hal yang ia baca terkait machine learning memiliki istilah yang cukup asing.

Kai merasa kebingungan harus melangkah dari mana untuk mempelajari machine learning hingga dapat membangun sebuah proyek machine learning. Namun, karena didorong oleh harapannya yang ingin mempermudah tugasnya,  ia tak berhenti untuk mencari tahu segala informasi terkait machine learning.

Setelah berkutat dan belajar sekian lama, ia akhirnya menemukan sebuah bahan bacaan yang dapat membawanya mengarungi machine learning lebih dalam. Materi tersebut berbagai macam hal terkait machine learning, mulai dari definisi machine learning hingga cara pembuatan machine learning.

Dengan penuh semangat, Kai mulai mempelajari satu per satu mulai dari definisi, tipe-tipe machine learning, proses pengembangan, maintenance, hingga penerapannya. Setelah berbagai informasi dan pelajaran terkait machine learning ia pelajari, akhirnya Kai dapat lebih yakin bahwa machine learning dapat membantu untuk menangani permasalahan yang ada di tokonya itu.

Akhirnya, Kai membangun sebuah sistem yang dapat melakukan klasifikasi data konsumen berdasarkan karakteristiknya sehingga tugasnya untuk melakukan hal yang sama setiap harinya berkurang.

Setelah Kai berhasil mengatasi permasalahan untuk melakukan klasifikasi, rasa penasaran yang ia rasakan semakin tinggi. Hal itu disebabkan karena ia memiliki masalah ketika menjaga tokonya pada masa transisi pandemi yang mengharuskan para pengunjung menggunakan masker. Pada kasus ini, Kai merasa lelah untuk mengingatkan dan melihat satu per satu pengunjung yang datang apakah sudah menggunakan masker atau belum.

Bagaimana perjalanan Kai untuk mengatasi permasalahan tersebut? Apakah AI dapat membantu menangani permasalahan dengan gambar? Mari kita simak perjalanan Kai pada modul selanjutnya agar dapat mengenal AI lebih dalam. Sampai jumpa pada modul deep learning. *Adios!*