1

Pengenalan

> "People worry that computers will get too smart and take over the world, but the real problem is that they're too stupid and they've already taken over the world."
>
> Pedro Domingos

Penulis yakin istilah *machine learning* atau *deep learning* sudah tidak asing di telinga pembaca. *Machine learning* dan *deep learning* adalah salah satu materi kuliah pada jurusan Teknik Informatika atau Ilmu Komputer. Selain mengenal kedua istilah tersebut di kuliah, pembaca mungkin mengenal istilah tersebut karena digunakan untuk pemasaran (*marketing*). Sebagai permulaan, *machine learning* dan *deep learning* bukanlah kedua hal yang sangat berbeda[1]. Perlu diingat, *deep learning* adalah bagian dari *machine learning*. *Machine learning* sudah diaplikasikan pada banyak hal, baik untuk klasifikasi gambar, mobil tanpa pengemudi, klasifikasi berita, dsb. Bab ini menjelaskan konsep paling dasar dan utama *machine learning*.

1.1 Kecerdasan Buatan

Pada bagian pembukaan (*kata pengantar*) telah dijelaskan bahwa kami menganggap kamu sudah memiliki pengetahuan dasar tentang *artificial intelligence* (kecerdasan buatan), kami akan memberikan sedikit ikhtisar apa hubungan kecerdasan buatan dan pembelajaran mesin. Saat pertama kali kamu mendengar istilah "kecerdasan buatan", mungkin kamu akan terpikir robot yang memiliki raga fisik. Tetapi, kecerdasan buatan tidak hanya terbatas pada sesuatu yang memiliki raga fisik. Raga fisik berguna untuk interaksi

[1] Walau istilah *deep learning* belakangan ini lebih populer.

yang ramah bagi manusia. Tidak mesti memiliki raga fisik, kecerdasan buatan sesungguhnya adalah program[2] yang memiliki bentuk matematis (instruksi); kita sebut sebagai **agen**. Berbeda dengan program biasa yang menghasilkan aksi berdasarkan instruksi, tujuan kecerdasan buatan adalah menciptakan program yang mampu mem-program (*output* program adalah sebuah program). Secara teori, program adalah ***automaton***[3] yang menjalankan suatu instruksi. Sama halnya dengan program pada umumnya, agen kecerdasan buatan juga menjalankan suatu instruksi. Yang menjadikanya beda dengan program biasa adalah **kemampuan untuk belajar**[4].

Pada bidang keilmuan kecerdasan buatan, kita ingin menciptakan agen yang mampu melakukan pekerjaan yang membutuhkan kecerdasan manusia. Perhatikan, disini disebut kecerdasan manusia; hewan pun cerdas, tapi kecerdasan manusia dan hewan berbeda; yang kita ingin aproksimasi adalah kecerdasan manusia. Akan tetapi, kecerdasan manusia susah didefinisikan karena memiliki banyak aspek misalnya nalar (logika), kemampuan berbahasa, seni, dsb. Karena kecerdasan manusia memiliki banyak dimensi, kita dapat mencoba menyelesaikan masalah pada sub bidang lebih kecil (*divide and conquer*). Sampai saat ini pun, peneliti belum juga mengetahui secara pasti apa yang membuat manusia cerdas, apa itu sesungguhnya cerdas, dan bagaimana manusia dapat menjadi cerdas. Dengan demikian, keilmuan kecerdasan buatan adalah interdisiplin, memuat: psikologis, linguistik, ilmu komputer, biologi, dsb. Bila kamu bertanya apakah program deterministik dapat disebut *kecerdasan buatan*, jawabannya "iya", *to some extent* (sampai pada level tertentu) karena memenuhi dimensi *acting rationally* (dijelaskan pada subbab 1.2).

Permasalahan utama bidang kecerdasan buatan terdiri dari (dari klasik sampai lebih modern)[5], yaitu:

1. ***Planning***. Diberikan *start state* dan *goal state*, agen harus merencanakan sekuens aksi untuk merubah *start state* menjadi *goal state*. Contoh permasalahan *planning* adalah merencanakan rute perjalanan dari kota A ke kota B. Bisa jadi, saat merencanakan sekuens aksi, ada kendala (*constraints*) yang harus dioptimisasi.

2. ***Representasi pengetahuan***, yaitu merepresentasikan pengetahuan dalam bentuk formal. Dengan representasi formal tersebut, kita dapat melakukan inferensi dengan operasi logika berbentuk simbolik, misal logika preposisi, logika orde pertama (*first-order logic*), teori Fuzzy, *abductive reasoning*, ontologi, maupun jaringan semantik (*semantic web*) [3].

[2] Secara sederhana, program adalah kumpulan atau sekuens instruksi.
[3] Kami sarankan untuk membaca buku [2] untuk materi automata.
[4] Perlu diperhatikan, definisi ini adalah pandangan modern.
[5] Silahkan merujuk Association for the Advancement of Artificial Intelligence (AAAI).

3. ***Machine learning***, yaitu teknik untuk melakukan inferensi terhadap data dengan pendekatan matematis. Inti *machine learning* adalah untuk membuat model (matematis) yang merefleksikan pola-pola data (seiring kamu membaca buku ini, kamu akan lebih mengerti). Ini adalah bahasan utama buku ini.

4. ***Multi-agent system***, yaitu sistem yang memiliki banyak agen berinteraksi satu sama lain untuk menyelesaikan permasalahan. Agen satu mengerjakan suatu hal tertentu, kemudian bekerja bersama untuk menyelesaikan masalah yang lebih besar (tidak dapat diselesaikan sendiri).

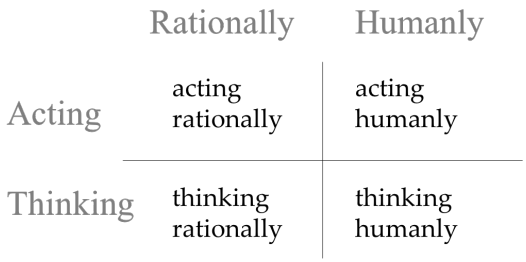
5. Dan lain sebagainya, silahkan mengacu pada topik konferensi *Association for the Advancement of Artificial Intelligence* (AAAI) .

Perhatikan, sub keilmuan representasi pengetahuan dan *machine learning* sama-sama melakukan inferensi, tetapi pada representasi yang berbeda. Inferensi pada bidang keilmuan representasi pengetahuan mencakup tentang bagaimana cara (langkah dan proses) mendapatkan sebuah keputusan, diberikan premis. Pada *machine learning*, inferensi yang dimaksud lebih menitikberatkan ranah hubungan variabel. Misalnya, *apakah penjualan akan meningkat apabila kita meningkatkan biaya marketing.* Bila kamu ingat dengan mata pelajaran matematika SMA (logika preposisi), kamu sadar bahwa membuat sistem cerdas menggunakan representasi pengetahuan simbolik itu susah. Kita harus mendefinisikan *term*, aturan logika, dsb. Belum lagi kita harus mendefinisikan aturan-aturan secara manual. Representasi pengetahuan secara tradisional dianggap relatif kurang *scalable*, khususnya apabila kita bekerja dengan data yang besar. Sementara itu, *machine learning* berada pada daerah representasi data/ilmu/pengetahuan dalam bentuk matematis karena keilmuan *machine learning* diturunkan dari matematika dan statistika.

Pada masa sekarang, kita dianugrahi dengan data yang banyak (bahkan tidak terbatas), teknik *machine learning* menjadi intuitif untuk melakukan inferensi pada data yang besar. Hal ini yang menyebabkan *machine learning* menjadi populer karena konstruksi model inferensi dapat dilakukan secara otomatis. *Machine learning* ibarat sebuah "alat", sama seperti rumus matematika. Bagaimana cara menggunakannya tergantung pada domain permasalahan. Dengan demikian, kamu harus paham betul bahwa memahami teknik-teknik *machine learning* saja tidak cukup. Kamu juga harus mengetahui domain aplikasi yang bersesuaian karena pemanfaatan teknik-teknik *machine learning* dapat berbeda pada domain yang berbeda. Sedikit cerita, sub keilmuan *data science* mempelajari banyak domain, misalnya data pada domain sosial, ekonomi, bahasa, maupun visual. Seiring kamu membaca buku ini, kami harap kamu semakin mengerti hal ini.

1.2 Intelligent Agent

Agen cerdas memiliki empat kategori berdasarkan kombinasi dimensi cara inferensi (*reasoning*) dan tipe kelakuan (*behaviour*) [4, 5]. Kategori agen dapat dilihat pada Gambar 1.1 dengan penjelasan sebagai berikut:



Gambar 1.1. Dimensi kecerdasan

1. ***Acting Humanly***. Pada dimensi ini, agen mampu bertingkah dan berinteraksi layaknya seperti manusia. Contoh terkenal untuk hal ini adalah *turing test*. Tujuan dari *turing test* adalah untuk mengevaluasi apakah suatu sistem mampu "menipu" manusia. Disediakan seorang juri, kemudian juri berinteraksi dengan sesuatu di balik layar. Sesuatu di balik layar ini bisa jadi manusia atau program. Program dianggap mampu bertingkah (berinteraksi) seperti layaknya manusia apabila juri tidak dapat membedakan ia sedang berkomunikasi dengan manusia atau program.

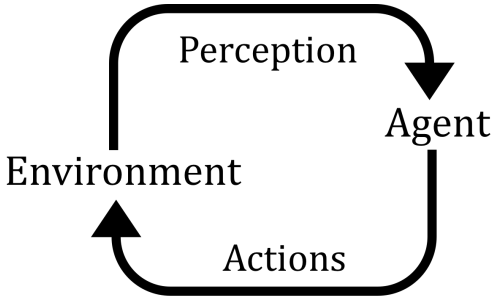
2. ***Acting Rationally***. Pada dimensi ini, agen mampu bertingkah dengan optimal. Tindakan optimal belum tentu menyerupai tindakan manusia, karena tindakan manusia belum tentu optimal. Misalnya, agen yang mampu memiliki rute terpendek dari suatu kota A ke kota B untuk mengoptimalkan penggunaan sumber daya. Sebagai manusia, bisa saja kita mencari jalan sesuka hati.

3. ***Thinking Humanly***. Pada dimensi ini, agen mampu berpikir seperti manusia dalam segi kognitif (e.g. mampu mengerti apa itu kesedihan atau kesenangan). Dapat dibilang, meniru bagaima proses berpikir di otak terjadi (pemodelan otak).

4. ***Thinking Rationally***. Pada dimensi ini, agen mampu berpikir secara rasional. Sederhananya sesuai dengan konsep logika matematika. *Thinking Humanly* lebih cenderung pada pemodelan kognitif secara umum, sementara dimensi *thinking rationally* cenderung pada pemodelan proses berpikir dengan prinsip optimisasi (apa yang harus dilakukan agar hasil

optimal).

Perlu dicatat, "*acting*" berarti agen mampu melakukan aksi. Sementara "*thinking*" adalah pemodelan proses. Untuk mewujudkan interaksi manusia-komputer seperti manusia-manusia, tentunya kita ingin *intelligent agent* bisa mewujudkan dimensi *acting humanly* dan *thinking humanly*. Sayangnya, manusia tidak konsisten [6]. Sampai saat ini, konsep kecerdasan buatan adalah meniru manusia; apabila manusia tidak konsisten, peneliti susah untuk memodelkan cara berpikir/tingkah laku manusia. Dengan hal itu, saat ini kita paling mungkin menciptakan agen yang mempunyai dimensi *acting rationally*.



Gambar 1.2. *Agent* vs *environment* [7]

Perhatikan Gambar 1.2! Agen mengumpulkan informasi dari lingkungannya, kemudian memberikan respon berupa aksi. Lingkungan (*environment*) yang dimaksud bisa jadi macam-macam, misal: rumah, papan catur, agen lain, dsb. Kita ingin agen melakukan aksi yang benar. Tentu saja kita perlu mendefinisikan secara detail, teliti, tepat (*precise*), apa arti "aksi yang benar". Dengan demikian, lebih baik apabila kita mengukur kinerja agen, menggunakan ukuran kinerja (*performance measure*). Misalnya untuk robot pembersih rumah, *performance measure*-nya adalah seberapa persen debu yang dapat ia bersihkan. *Performance measure*, secara matematis dikenal sebagai fungsi utilitas (*utility function*), yaitu fungsi apa yang harus dimaksimalkan/diminimalkan oleh agen tersebut. Setiap tindakan yang dilakukan agen rasional harus mengoptimalkan nilai *performance measure* atau *utility function*. Pada buku ini, istilah *performance measure* dan *utility function* merujuk pada hal yang sama[6].

[6] Buku ini berfokus pada *performance measure* secara matematis, buku lain belum tentu berfokus pada hal yang sama.

1.3 Konsep Belajar

Bayangkan kamu berada di suatu negara asing! Kamu tidak tahu norma yang ada di negara tersebut. Apa yang kamu lakukan agar bisa menjadi orang "normal" di negara tersebut? Tentunya kamu harus **belajar**! Kamu mengamati bagaimana orang bertingkah laku di negara tersebut dan perlahan-lahan mengerti norma yang berlaku. Belajar adalah usaha memperoleh kepandaian atau ilmu; berlatih; berubah tingkah laku atau tanggapan yang disebabkan oleh pengalaman[7]. Pembelajaran adalah proses, cara, perbuatan atau menjadikan orang atau makhluk hidup belajar[1]. Akan tetapi, pada *machine learning*, yang menjadi siswa bukanlah makhluk hidup, tapi mesin.

Definsi sebelumnya mungkin sedikit "abstrak", kita harus mengkonversi definisi tersebut sebagai definsi operasional (bentuk komputasi). Secara operasional, belajar adalah perubahan tingkah laku berdasarkan pengalaman (*event*/data) untuk menjadi lebih baik (mengoptimisasi parameter terhadap ***performance measure/utility function***).

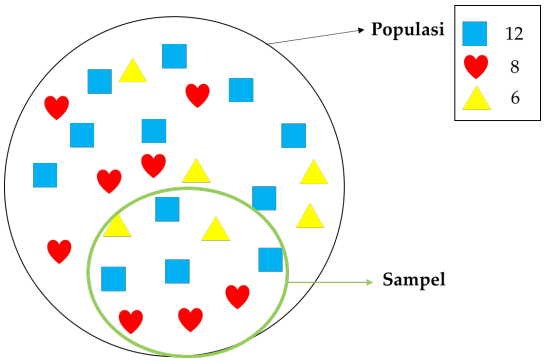
1.4 Statistical Learning Theory

Pada masa sekarang ini data bertebaran sangat banyak dimana-mana. Pemrosesan data secara manual tentu adalah hal yang kurang bijaksana. Beberapa pemrosesan data yang dilakukan seperti kategorisasi (kategorisasi teks berita), peringkasan dokumen, ekstraksi informasi (mencari subjek, objek, dan relasi diatara keduanya pada teks), rekomendasi produk berdasarkan catatan transaksi, dll [7]. Tujuan *machine learning* minimal ada dua: **memprediksi masa depan** (*unobserved event*); dan/atau **memperoleh ilmu pengetahuan** (*knowledge discovery/discovering unknown structure*). Kedua hal ini berkaitan sangat erat. Sebagai contoh, manusia tahu bahwa cara menggunakan pensil dan pulpen sama, walaupun saat kita belum pernah menggunakan pulpen (penulis berasumsi kamu belajar menulis menggunakan pensil). Memprediksi masa depan berarti kita tahu bahwa pulpen adalah alat tulis. *Knowledge discovery* berarti kita tahu bahwa cara menggunakan pulpen dan pensil itu sama, walaupun belum pernah menggunakan pulpen sebelumnya[8].

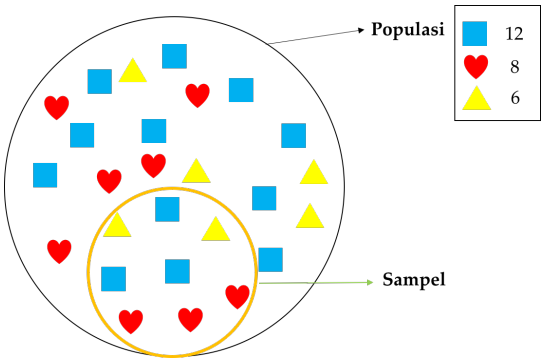
Untuk mencapai tujuan tersebut, kita menggunakan data (sampel), kemudian membuat model untuk menggeneralisasi "aturan" atau "pola" data sehingga kita dapat menggunakannya untuk mendapatkan informasi/membuat keputusan [8, 9]. *Statistical learning theory* (yang diaplikasikan pada *machine learning*) adalah teknik untuk memprediksi masa depan dan/atau menyimpulkan/mendapatkan pengetahuan dari data **secara rasional dan non-paranormal**. Hal ini sesuai dengan konsep *intelligent agent*, yaitu bertingkah berdasarkan lingkungan. Dalam hal ini, yang bertindak sebagai lingkungan

[7] KBBI Web, diakses pada 10 Oktober 2016
[8] Baca *zero-shot learning*

adalah data. *Performance measure*-nya adalah seberapa akurat prediksi agen tersebut atau seberapa mirip "pola" data yang ditemukan terhadap data asli. Disebut *statistical* karena basis pembelajarannya memanfaatkan banyak teori statistik untuk melakukan inferensi (misal memprediksi *unobserved event*)[9]



Gambar 1.3. Ilustrasi makanan pesta 1



Gambar 1.4. Ilustrasi makanan pesta 2

Perhatikan Gambar 1.3 (permasalahan yang disederhanakan). Misalkan kamu diundang ke suatu pesta. Pada pesta tersebut ada 3 jenis kue yang disajikan. Kamu ingin mengetahui berapa rasio kue yang disajikan dibandingkan masing-masing jenisnya (seluruh populasi). Tetapi, karena susah untuk menganalisis seluruh data atau keseluruhan data tidak tersedia, kamu mengambil beberapa sampel. Dari sampel tersebut, kamu mendapati bahwa ada 4 buah kue segi empat, 3 buah kue hati dan 2 buah kue segitiga. Lalu kamu

[9] Selain itu, *machine learning* juga banyak memanfaatkan teori aljabar linear.

menyimpulkan (model) bahwa perbandingan kuenya adalah 4:3:2 (segiempat:hati:segitiga). Perbandingan tersebut hampir menyerupai kenyataan seluruh kue yaitu 4:2.67:2. Tentu saja kondisi ini terlalu ideal.

Perhatikan Gambar 1.4, temanmu Ari datang juga ke pesta yang sama dan ingin melakukan hal yang sama (rasio kue). Kemudian ia mengambil beberapa sampel kue. Dari sampel tersebut ia mendapati bahwa ada 3 buah segiempat, 3 buah hati dan 2 buah segitiga, sehingga perbandingannya adalah 3:3:2. Tentunya hal ini sangat melenceng dari populasi.

Dari dua sampel yang berbeda, kita menyimpulkan, menginferensi (*infer*) atau mengeneralisasi dengan berbeda. Kesimpulan yang kita buat berdasarkan sampel tersebut, kita anggap merefleksikan populasi, kemudian kita menganggap populasi memiliki aturan/pola seperti kesimpulan yang telah kita ciptakan [10]. Baik pada statistika maupun *statistical machine learning*, pemilihan sampel (selanjutnya disebut ***training data***) adalah hal yang sangat penting. Apabila *training data* tidak mampu merepresentasikan populasi, maka model yang dihasilkan pembelajaran (*training*) tidak bagus. Untuk itu, biasanya terdapat juga ***development data*** dan ***test data***. Mesin dilatih menggunakan *training data*, kemudian diuji kinerjanya menggunakan *development data*[10] dan *test data*. Seiring dengan membaca buku ini, konsep *training data*, *development data*, dan *test data* akan menjadi lebih jelas.

Seperti halnya contoh sederhana ini, persoalan *machine learning* sesungguhnya menyerupai persoalan *statistical inference* [10]. Kita berusaha mencari tahu populasi dengan cara menyelidiki fitur (*features* atau sifat-sifat) yang dimiliki sampel. Kemudian, menginferensi aksi yang harus dilakukan terhadap *unobserved data* berdasarkan kecocokan fitur-fitur *unobserved data* dengan model/aturan yang sudah ada.

Dari sisi metode pembelajaran, algoritma *machine learning* dapat dikategorikan sebagai: *supervised learning* (subbab 1.6), *semi-supervised learning* (subbab 1.8), *unsupervised learning* (subbab 1.9), dan *reinforcement learning*. Masing-masing metode akan dibahas pada subbab berikutnya (kecuali *reinforcement learning* diluar cakupan buku ini).

1.5 Training, Development, Testing Set

Terdapat dua istilah penting dalam pembangunan model *machine learning* yaitu: ***training*** dan ***testing***. *Training* adalah proses membangun model dan *testing* adalah proses menguji kinerja model pembelajaran. *Dataset* adalah kumpulan data (sampel dalam statistik). Sampel ini adalah data yang kita gunakan untuk membuat model maupun mengevaluasi model *machine learning*. Umumnya, *dataset* dibagi menjadi tiga jenis yang tidak beririsan (satu sampel pada himpunan tertentu tidak muncul pada himpunan lainnya):

[10] Pada umumnya bertindak sebagai *stopping criterion* saat proses *training*.

1. ***Training set*** adalah himpunan data yang digunakan untuk melatih atau membangun model. Pada buku ini, istilah *training data(set)* mengacu pada *training set*.
2. ***Development set*** atau ***validation set*** adalah himpunan data yang digunakan untuk mengoptimisasi saat melatih model. Model dilatih menggunakan *training set* dan pada umumnya kinerja **saat latihan** diuji dengan *development set*. Hal ini berguna untuk generalisasi (agar model mampu mengenali pola secara generik). Pada buku ini, istilah *development/validation data(set)* mengacu pada hal yang sama.
3. ***Testing set*** adalah himpunan data yang digunakan untuk menguji model setelah **proses latihan selesai**. Pada buku ini, istilah *testing data(set)* atau *test set* mengacu pada *testing set*. Perlu kami tekankan, *testing set* adalah *unseen data*. Artinya, model dan manusia tidak boleh melihat sampel ini saat proses latihan. Banyak orang yang tergoda untuk melihat *testing set* saat proses latihan walaupun itu adalah tingkah laku yang buruk karena menyebabkan *bias*.

Satu sampel pada himpunan data kita sebut sebagai ***data point*** atau instans (***instance***) yang merepresentasikan suatu kejadian statistik (*event*). Perlu diingat, *training*, *development*, dan *testing data* diambil (*sampled*) dari distribusi yang sama dan memiliki karakteristik yang sama (*independently and identically distributed*). Distribusi pada masing-masing dataset ini juga sebaiknya seimbang (*balanced*) dan memuat seluruh kasus. Misal, sebuah dataset *binary classification* sebaiknya memuat 50% kasus positif dan 50% kasus negatif.

Pada umumnya, rasio pembagian *dataset* adalah (80% : 10% : 10%) atau (90% : 5% : 5%) (*training*:*development*:*testing*). *Development set* pada umumnya bisa tidak digunakan apabila *dataset* berukuran kecil (hanya dibagi menjadi *training* dan *testing set* saja). Dalam kasus ini, pembagian *dataset* menjadi *training* dan *testing* set pada umumnya memiliki rasio (90% : 10%), (80% : 20%), (70% : 30%), atau (50% : 50%). Pada kasus ini, kinerja saat *training* diuji menggunakan *training set*.

Saat tidak menggunakan *development set* (hanya ada *training* dan *testing set*), kita juga memiliki opsi untuk mengevaluasi model dengan metode *K-cross-validation*[11]. Artinya, kita membagi *training dataset* menjadi K bagian. Kita menggunakan $K - 1$ bagian untuk *training*, kemudian menguji kinerja model saat latihan (*validation*) menggunakan satu bagian. Hal ini diulangi sebanyak K kali dimana sebuah bagian data digunakan sebagai *testing set* sebanyak sekali (bergilir). Hal ini akan dijelaskan lebih lanjut pada bab-bab selanjutnya.

[11] `https://www.openml.org/a/estimation-procedures/1`

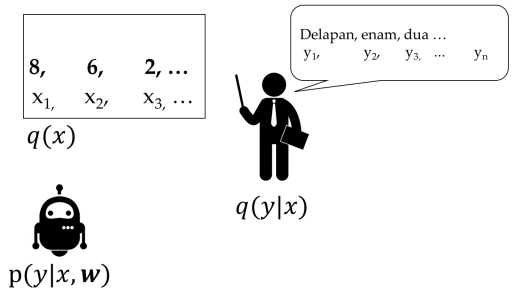
1.6 Supervised Learning

Jika diterjemahkan secara literal, *supervised learning* adalah pembelajaran terarah/terawasi. Artinya, pada pembelajaran ini, ada guru yang mengajar (mengarahkan) dan siswa yang diajar. Kita disini berperan sebagai guru, kemudian mesin berperan sebagai siswa. Perhatikan Gambar 1.5 sebagai ilustrasi! Pada Gambar 1.5, seorang guru menuliskan angka di papan "8, 6, 2" sebagai contoh untuk siswanya, kemudian gurunya memberikan cara membaca yang benar untuk masing-masing angka. Contoh angka melambangkan ***input***, kemudian cara membaca melambangkan ***desired output***. Pasangan ***input-desired output*** ini disebut sebagai *instance* (untuk kasus supervised learning).

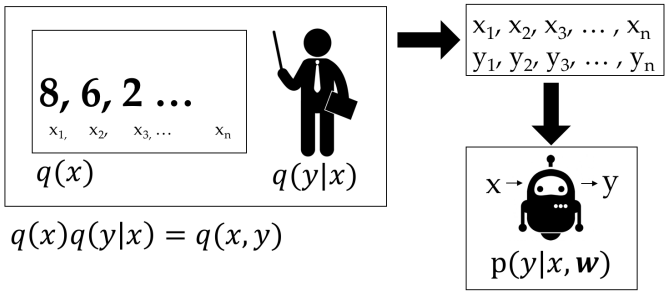
Gambar 1.5. *Supervised learning*

Perhatikan Gambar 1.6 dan Gambar 1.7, x adalah kejadian (*event – random variable*), untuk *event* tertentu dapat dinotasikan sebagai $\{x_1, x_2, x_3, \cdots, x_N\}$. x dapat berupa vektor, teks, gambar, dan lain sebagainya (perhatikan konteks pembahasan buku). Demi pembahasan yang cukup generik, pada bab ini kita membicarakan x yang merepresentasikan *event*, *data point*, atau *instance*. Seorang guru sudah mempunyai jawaban yang benar untuk masing-masing contoh dengan suatu fungsi distribusi probabilitas kondisional (***conditional probability density function***) $q(y \mid x)$ baca: *function q for y given x*, melambangkan hasil yang benar/diharapkan untuk suatu event. Siswa (mesin) **mempelajari tiap pasang pasangan *input-desired output (training data)*** dengan mengoptimalkan *conditional probability density function* $p(y \mid x, \mathbf{w})$, dimana y adalah target (*output*), x adalah input dan vektor $\mathbf{w}$ adalah ***learning parameters***. Proses belajar ini, yaitu mengoptimalkan $\mathbf{w}$ disebut sebagai training. Semakin kamu membaca buku ini, konsep ini akan menjadi semakin jelas.

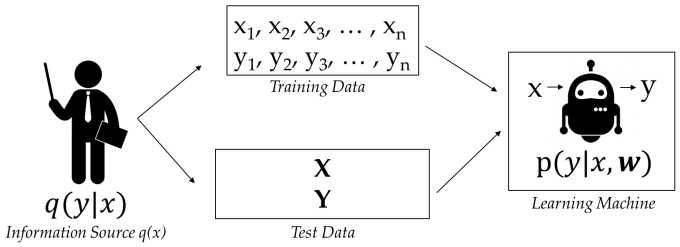
Perhatikan Gambar 1.8! model memiliki panah ke *training data* dan *test data*, artinya model hasil *training* sangat bergantung pada **data dan guru**.



Gambar 1.6. *Supervised learning - mathematical explanation*



Gambar 1.7. *Supervised learning - mathematical explanation* 2



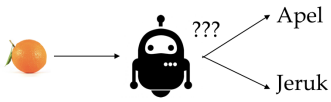
Gambar 1.8. *Supervised learning framework*

Model yang dihasilkan *training* (hasil pembelajaran kemampuan siswa) untuk data yang sama bisa berbeda untuk guru yang berbeda[12].

Tujuan *supervised learning*, secara umum untuk melakukan klasifikasi (*classification*). Misalkan mengklasifikasikan gambar buah (apa nama buah pada gambar), diilsutrasikan pada Gambar 1.9. Apabila hanya ada dua kategori, disebut ***binary classification***. Sedangkan bila terdapat lebih dari dua

[12] Penulis rasa hal ini sangat intuitif berhubung hal serupa terjadi pada manusia.

kategori, disebut ***multi-label classification***. Ada tipe klasifikasi lain disebut *soft classification* yaitu klasifikasi menggunakan probabilitas (seperti pada *fuzzy logic*) misalkan suatu berita memuat 30% olah raga dan 70% politik.



Gambar 1.9. Ilustrasi klasifikasi buah

$$p(y \mid x, \mathbf{w}) \tag{1.1}$$

Pemahaman *supervised learning* adalah mengingat persamaan 1.1. Ada tiga hal penting pada *supervised learning* yaitu *input*, *desired output*, dan *learning parameters*. Perlu ditekankan *learning parameters* berjumlah lebih dari satu, dan sering direpresentasikan dengan vektor (*bold*). Berdasarkan model yang dibuat, kita dapat melakukan klasifikasi (misal simbol yang ditulis di papan adalah angka berapa). Secara konseptual, klasifikasi didefinisikan sebagai persamaan 1.2 yaitu memilih label (kelas/kategori y) paling optimal dari sekumpulan label C, diberikan (*given*) suatu instans data tertentu.

$$\hat{y_i} = \arg\max_{y_i \in C} p(y_i \mid x_i, \mathbf{w}) \tag{1.2}$$

1.7 Regresi

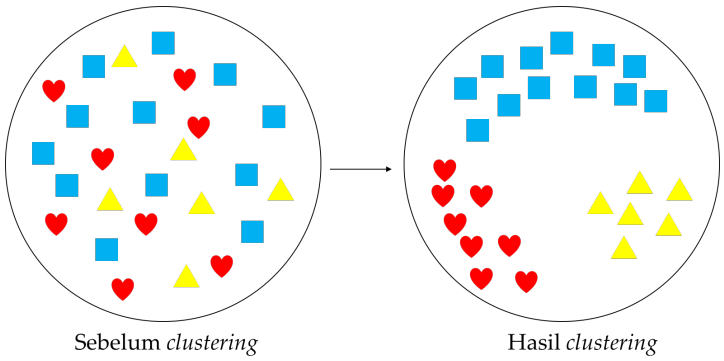
Pada persoalan regresi, kita ingin memprediksi *output* berupa bilangan kontinu. Misalnya pada regresi suatu fungsi polinomial, kita ingin mencari tahu fungsi $f(x)$ diberikan data $\{(x_1, y_1), \cdots, (x_N, y_N)\}$. Setelah itu, kita gunakan fungsi aproksimasi untuk mencari tahu nilai y_{N+1} dari data baru x_{N+1}. Perbedaan regresi dan klasifikasi adalah pada tipe *output*. Untuk regresi, tipe *output* adalah nilai kontinu; sementara tipe *output* pada persoalan klasifikasi adalah suatu objek pada himpunan (i.e., memilih opsi pada himpunan jawaban). Tetapi, kita dapat mengkonversi fungsi regresi menjadi fungsi klasifikasi (dijelaskan pada bab 5).

1.8 Semi-supervised Learning

Semi-supervised learning mirip dengan *supervised learning*, bedanya pada proses pelabelan data. Pada *supervised learning*, ada "guru" yang harus membuat "kunci jawaban" *input-output*. Sedangkan pada *semi-supervised learning*

tidak ada "kunci jawaban" eksplisit yang harus dibuat guru. Kunci jawaban ini dapat diperoleh secara otomatis (misal dari hasil *clustering*). Pada kategori pembelajaran ini, umumnya kita hanya memiliki sedikit data. Kita kemudian menciptakan data tambahan baik menggunakan *supervised* ataupun *unsupervised learning*, kemudian membuat model belajar dari data tambahan tersebut.

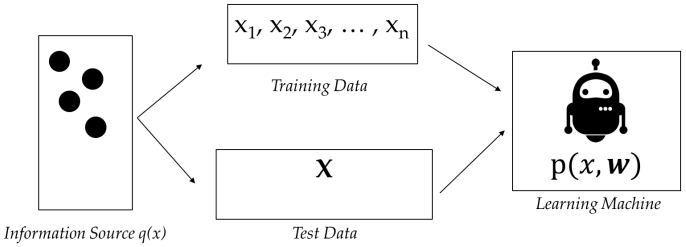
1.9 Unsupervised Learning



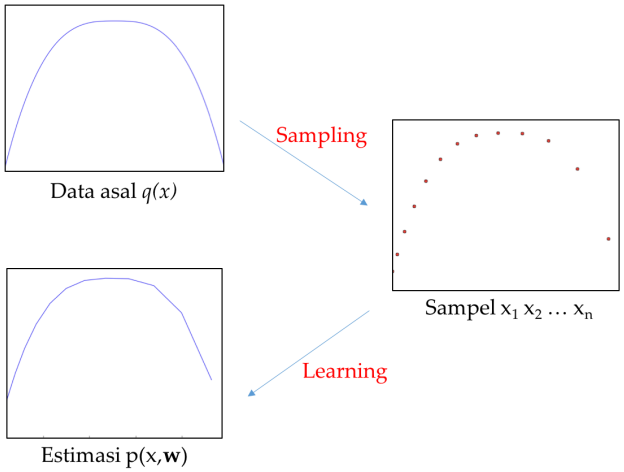
Gambar 1.10. Ilustrasi *clustering*

Jika pada *supervised learning* ada guru yang mengajar, maka pada *unsupervised learning* tidak ada guru yang mengajar. Contoh permasalahan unsupervised learning adalah *clustering*. Mengingat contoh kue sebelumnya, kita ingin mengelompokkan kue-kue yang sama, diilustrasikan oleh Gambar 1.10. Yang kamu lakukan adalah membuat kelompok-kelompok berdasarkan karakteristik kue, misal kelompok kue biru, kelompok kue kuning, atau kelompok kue merah. Teknik-teknik mengelompokkan ini akan dibahas pada bab-bab berikutnya. Contoh algoritma *unsupervised learning* sederhana adalah *K-means* (bab 4).

Perhatikan Gambar 1.11 dan Gambar 1.12! Berbeda dengan supervised learning yang memiliki *desired output*, pada *unsupervised learning* tidak ada *desired output* (jelas, tidak ada gurunya, tidak ada yang memberi contoh). Kita ingin mencari tahu distribusi asli data $q(x)$, berdasarkan beberapa sampel data. *Learning* dilakukan dengan mengoptimalkan $p(x \mid \mathbf{w})$ yang mengoptimasi parameter $\mathbf{w}$. Perbedaan antara estimasi dan fungsi asli disebut sebagai ***generalization loss*** (atau ***loss*** saja – dijelaskan pada bab 5). Kunci pemahaman *unsupervised learning* adalah mengingat persamaan 1.3, yaitu ada *input* dan parameter.



Gambar 1.11. *Unsupervised learning framework*



Gambar 1.12. *Generalization error of unsupervised learning*

$$p(x \mid \mathbf{w}) \tag{1.3}$$

Perlu kami tekankan, *unsupervised learning* $\neq$ *clustering*! *Clustering* adalah salah satu bentuk *unsupervised learning*; yaitu salah satu hasil inferensi persamaan 1.3. *Unsupervised learning* adalah mencari sifat-sifat (*properties*) data. Kita ingin aproksimasi $p(x \mid \mathbf{w})$ semirip mungkin dengan $q(x)$, dimana $q(x)$ adalah distribusi data yang asli. Dataset di-sampel dari distribusi $q(x)$, kemudian kita ingin mencari tahu $q(x)$ tersebut.

1.10 Proses Belajar

Seperti yang sudah dijelaskan pada subbab sebelumnya, pada *supervised* maupun *unsupervised learning*, kita ingin mengestimasi sesuatu dengan teknik *machine learning*. Kinerja *learning machine* berubah-ubah sesuai dengan parameter $\mathbf{w}$ (parameter pembelajaran). Kinerja *learning machine* diukur oleh

fungsi tujuan (*utility function/performance measure*), yaitu mengoptimalkan nilai fungsi tertentu; misalnya meminimalkan nilai *error*, atau meminimalkan *loss* (dijelaskan kemudian). Secara intuitif, *learning machine* sama seperti saat manusia belajar. Kita awalnya membuat banyak kesalahan, tetapi kita mengetahui/diberi tahu mana yang benar. Untuk itu kita menyesuaikan diri secara perlahan agar menjadi benar (iteratif). Inilah yang juga dilakukan *learning machine*, yaitu mengubah-ubah parameter $\mathbf{w}$ untuk mengoptimalkan suatu fungsi tujuan[13].

Secara bahasa lebih matematis, kami beri contoh *supervised learning*. Kita mempunyai distribusi klasifikasi asli $q(y \mid x)$. Dari distribusi tersebut, kita diberikan beberapa sampel pasangan *input-output* $\{z_1, z_2, z_3, \ldots, z_n\}; z_i = (x_i, y_i)$. Kita membuat *learning machine* $p(y \mid x, \mathbf{w})$. Awalnya diberi (x_1, y_1), *learning machine* mengestimasi fungsi asli dengan mengoptimalkan parameter $\mathbf{w}$ sesuai dengan data yang ada. Seiring berjalannya waktu, ia diberikan data observasi lainnya, sehingga *learning machine* menyesuaikan dirinya (konvergen) terhadap observasi yang baru $(x_2, y_2), (x_3, y_3), \ldots$. Semakin lama, kita jadi makin percaya bahwa *learning machine* semakin optimal (mampu memprediksi fungsi aslinya). Apabila kita diberikan data sejumlah tak hingga, kita harap aproksimasi kita sama persis dengan distribusi aslinya.

1.11 Tips

Jujur, pengarang sendiri belum menguasai bidang ini secara penuh, tetapi berdasarkan pengalaman pribadi (+ membaca) dan beberapa rekan; ada beberapa materi wajib yang harus dipahami untuk mengerti bidang *machine learning*. Sederhananya, kamu harus menguasai banyak teori matematika dan probabilitas agar dapat mengerti *machine learning* sampai tulang dan jeroannya. Kami tidak menyebutkan bahwa mengerti *machine learning* secara intuitif (atau belajar dengan pendekatan deskriptif) itu buruk, tetapi untuk mengerti sampai dalam memang perlu mengerti matematika (menurut pengalaman kami). Disarankan untuk belajar materi berikut:

1. Matematika Diskrit dan Teori Bilangan
2. Aljabar Linier dan Geometri (vektor, matriks, skalar, dekomposisi, transformasi, tensor, dsb)
3. Kalkulus (diferensial dan integral)
4. Teori Optimasi (*Lagrange multiplier, Convex Iptimization, Gradient Descent, Integer Linear Problem,* dsb)
5. Probabilitas dan Statistika (probabilitas, *probability densities*, *hypothesis testing*, *inter-rater agreement*, Bayesian, *statistical mechanics*)
6. Teori Fuzzy

[13] Saat membaca ini, kamu mungkin akan menganggap bahwa teknik *machine learning* adalah fungsi-parametrik. Sebenarnya, ada juga algoritma *machine learning* non-parametrik.

Mungkin kamu sudah tahu, tetapi penulis ingin mengingatkan ada dua buku yang sangat terkenal ("kitab") sebagai materi belalajar *machine learning* dan *deep learning*:

1. Patten Recognition and Machine Learning, oleh Christopher M. Bishop [8]
2. Deep Learning, oleh Ian Goodfellow, Yoshua Bengio, dan Aaron Courville [11]

Apabila pembaca memiliki kesempatan, penulis sarankan untuk membaca kedua buku tersebut.

1.12 Contoh Aplikasi

Sebenarnya, aplikasi pemanfaatan *machine learning* sudah terasa dalam kehidupan sehari-hari. Contoh mudahnya adalah produk-produk Google, misalnya google translate (machine translation, handwritten recognition, speech recognition, Alpha Go). Berikut adalah beberapa artikel menarik:

1. `techcrunch google AI beats go world champion`
2. `http://www-formal.stanford.edu/jmc/whatisai/node3.html`
3. `https://www.google.com/selfdrivingcar/`
4. `http://www.osnews.com/story/26838/Palm_I_m_ready_to_wallow_now/page2/`

Soal Latihan

1.1. Aplikasi

(a) Carilah contoh-contoh penerapan *machine learning* pada kehidupan sehari-hari selain yang telah disebutkan!
(b) Mengapa mereka menggunakan teknik *machine learning* untuk menyelesaikan permasalahan tersebut?
(c) Apakah tidak ada opsi teknik lainnya? Jelaskan bila ada!
(d) Apa kelebihan dan kekurangan teknik *machine learning* daripada teknik lainnya (yang kamu jelaskan pada soal (c))?

1.2. Kecerdasan

Jelaskan tahapan perkembangan kecerdasan manusia berdasarkan kategori usia! Dari hal ini, kamu akan mengerti kenapa beberapa peneliti membuat agen cerdas berdasarkan kategori usia tertentu.