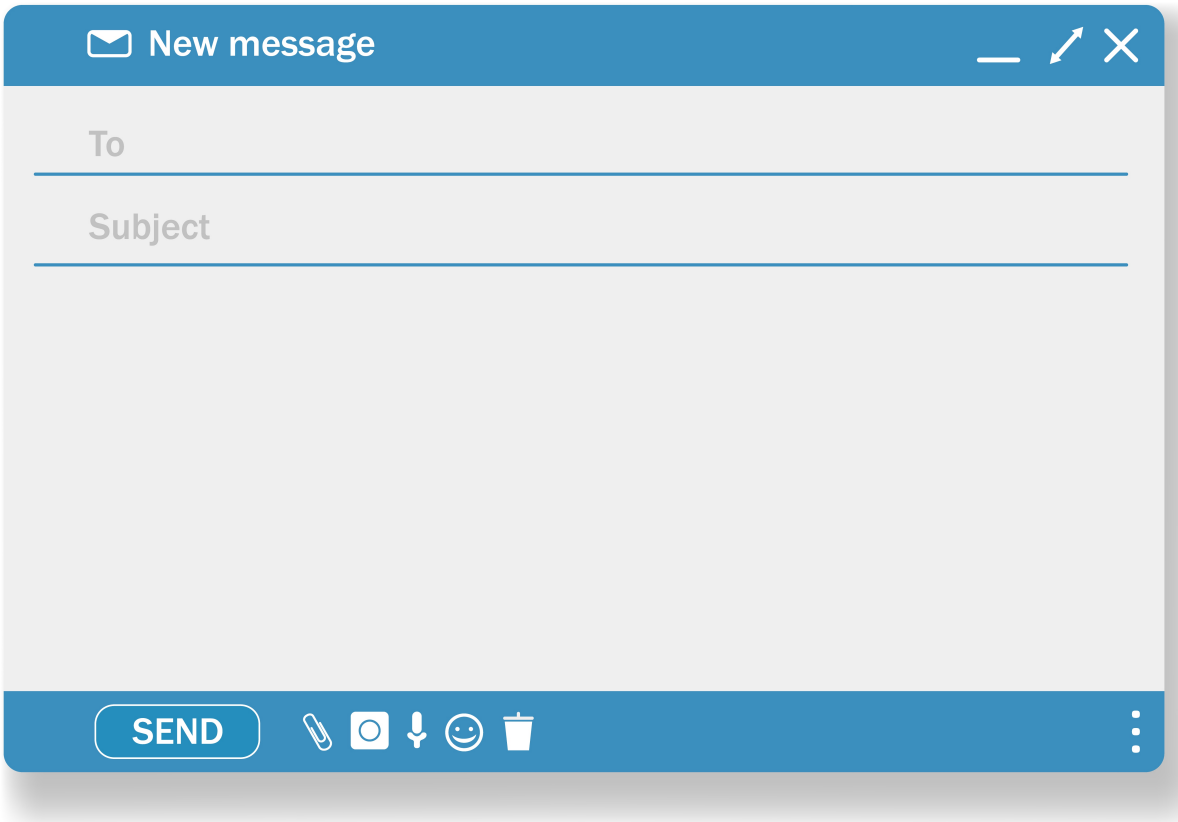
JTI - Modul Praktikum Pembelajaran Mesin

Dasar Teori

Apa itu fitur?

Fitur merupakan representasi dari sebuah data (*raw data*) yang berbentuk nilai-nilai dalam angka. Fitur merupakan atribut, karakteristik, atau penciri dari data tersebut yang menunjukkan sebuah keunikan dari sebuah data. Secara alamiah, fitur didapatkan dari data yang telah tersedia. Selain itu, terkadang, fitur juga memiliki hubungan dengan algoritma atau model pembelajaran mesin yang digunakan. Dengan kata lain, **beberapa model hanya cocok dengan beberapa jenis fitur, begitu juga sebaliknya**. Pemilihan fitur yang tepat, dapat mempermudah model dalam melakukan pembelajaran, sehingga dapat mencapai tujuan dengan lebih baik. Untuk mendapatkan fitur yang sesuai dengan data, model, dan tujuan pembelajaran mesin, dapat dilakukan proses perekayasaan fitur atau lebih sering dikenal dengan istilah ***feature engineering***.

Untuk lebih memahami tentang konsep fitur, mari kita gunakan data pesan email. Perhatikan ilustrasi pada Gambar 2.1. Pada konteks pesan email, data yang digunakan keseluruhan pesan email itu sendiri. Didalam sebuah pesan email pasti terdapat alamat pengirim, alamat tujuan, subyek pesan email, dan isi email. Lalu, mana yang dapat kita jadikan sebagai fitur dalam data pesan email? Beberapa hal yang mungkin revelan untuk menjadi fitur pesan email adalah panjang isi email, kata atau karakter khusus didalam email, nama pengirim, alamat penerima, dan sebagainya. Hal ini menjadikan jumlah fitur didalam sebuah data bisa jadi sangat banyak.



Gambar 2.1. Ilustrasi Email (Image by Upl56 on Freepik)

Jumlah fitur yang kita gunakan juga sangat penting. Jika fitur yang digunakan terlalu sedikit, maka model pembelajaran mesin tidak dapat belajar dengan baik. Jika fitur yang digunakan terlalu banyak, proses pembelajaran akan berlangsung lebih lama dan ada kemungkinan model menjadi lebih sulit untuk menemukan pola-pola unik didalam fitur-fitur tersebut. Kedua hal tersebut dikenal dengan istilah ***underfitting*** dan ***overfitting***.

Ekstraksi Fitur

Setelah mengetahui konsep fitur pada pembelajaran mesin, selanjutnya kita akan membahas bagaimana cara mendapatkan fitur dari sebuah data. Proses ini dikenal dengan nama ekstraksi fitur (***feature extraction***).

Ekstraksi fitur adalah **proses pengurangan (reduksi) dimensi** dari sebuah data mentah menjadi nilai-nilai atau variabel-variabel yang mudah dikelola oleh perangkat komputasi. Pada proses ini juga dilakukan pemilihan dan pengkombinasian nilai-nilai dari data sehingga dapat digunakan sebagai fitur atau representasi dari data tersebut. Untuk memudahkan pemahaman terkait dengan dimensi dan fitur, perhatikan data penumpang kapal Titanic pada Gambar 2.2.

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Gambar 2.2 Data Penumpang Kapal Titanic
(Sumber: https://www.kaggle.com/datasets/yasserh/titanic-dataset)

Dimensi pada data penumpang kapal Titanic merupakan semua kolom pada data mulai dari **PassengerId** hingga **Embarked**. Sehingga, berdasarkan hal tersebut, kita dapat mengatakan bahwa data tersebut terdiri dari 12 dimensi. Lalu, apabila kita memunculkan sebuah pertanyaan, "***Bagaimana caranya jika kita ingin mengetahui pengumpang yang selamat dari tragedi Titanic?***". Maka kita perlu menganalisis kolom mana saya yang mungkin dapat mencirikan bahwa penumpang tersebut selamat atau tidak. Bisa jadi, tidak semua 12 dimensi data kita gunakan menjadi fitur untuk dapat menjawab pertanyaan tersebut.

Mari coba kita analisa bersama-sama.

Informasi terkait dengan apakah penumpang selamat atau tidak, dapat dilihat dari kolom "**Survived**". Nilai "0" untuk tidak selamat dan "1" untuk penumpang yang selamat. Kolom "Survived" akan menjadi acuan kita dalam mencari ciri-ciri penumpang yang selamat. Selanjutnya kita perlu melakukan analisis dimensi mana yang dapat menggambarkan bahwa penumpang tersebut selamat ataupun tidak. *Let's do this!*

- **PassengerId** --> "PassengerId" merupakan kolom yang digunakan untuk mengetahui urutan penumpang. Kesemuanya nilainya unik untuk setiap penumpang, tidak unik untuk membedakan penumpang yang salamat atau tidak. *So*, kita tidak dapat menggunakan PessengerId sebagai fitur.
- **Pclass** --> "Pclass" merupakan informasi terkait dengan kelas penumpang didalam dek kapal Titanic. Nilainya terdiri dari 1,2,3. Bisa jadi kelas penumpang menggambarkan lokasi pada kapal, dek bawah, dek tengah, ataupun dek atas. Potensi dek bawah untuk tidak selamat pada tragedi ini cukup besar secara logika. Sedangkan dek atas sebaliknya. Sehingga, ada kemungkinan "Pclass" dapat menjadi nilai yang unik.

- **Name** --> "Name" merupakan informasi tekait dengan nama penumpang. Seluruhnya unik untuk setiap penumpang, sama seperti "PassengerId". Akan menjadi sulit apabila nilai ini kita gunakan untuk menjadi sebuah penciri penumpang selamat ataupun tidak. Kita tidak akan menggunakannya.
- "**Sex**" --> "Sex" merupakan informasi terkait dengan jenis kelamin penumpang. Nilai ini memiliki kemungkinan untuk menjadi sebuah fitur. Hal ini dikarenakan, pada kondisi darurat, terdapat kemungkinan jenis kelamin tertentu memiliki prioritas untuk diselamatkan terlebih dahulu. Kita dapat melihat pada film Titatic, anak-anak dan wanita merupakan prioritas untuk diselamatkan. Oleh karena itu, kita dapat menggunakan "Sex" sebagai fitur.
- "**Age**" --> Seperti nama kolomnya, "Age" merupakan informasi terkait dengan usia penumpang. Nilai ini juga memiliki kemungkinan untuk dijadikan sebuah fitur dengan alasan yang sama dengan "Sex", yaitu prioritas.
- "**SibSp**" --> "SibSp" merupakan informasi terkait dengan jumlah saudara (*siblings*) dan pasangan ("*spouses*"). Pada tragedi Titanic, keadaan menjadi sangat tidak terkendali, ada kemungkinan setiap orang berusaha menyelamatkan dirinya sendiri. Sehingga, jumlah saudara ataupun pasangan mungkin tidak revelan untuk menggambarkan orang tersebut selamat atau tidak.
- "**Parch**" --> "Parch" merupakan informasi terkait dengan jumlah anak ataupun orang tua yang ikut dalam perjalanan Titanic. Dengan alasan yang sama dengan "SibSp", cukup sulit untuk menjadikan data ini sebagai acuan untuk mengetahui apakah penumpang selamat atau tidak.
- "**Ticket**" --> "Ticket" merupakan informasi nomor tiket untuk setiap penumpang. Nomor ini unik untuk setiap tiket. Sehingga sulit untuk menjadinya penciri "Survived" sama halnya seperti "PassengerID".
- "**Fare**" --> "Fare" merupakan informasi harga tiket yang dibayarkan oleh penumpang. Setiap orang mungkin berbeda. Akan tetapi, tiket untuk setiap kelas pasti tidak jauh berbeda. Sehingga, kita dapat mengelompokkan "Fare" menjadi 3 kelompok dengan rentang harga tertentu. Hal ini dikarenakan "Fare" kemungkinan memiliki korelasi dengan "Pclass". Oleh karena itu, "Fare" memiliki kemungkinan untuk menjadi fitur.
- "**Cabin**" --> "Cabin" merupakan informasi lokasi kabin penumpang. Jika kita perhatikan pada Gambar 2.2, terdapat beberapa data dengan nilai "NaN" pada "Cabin". Hal ini menandakan hilangnya informasi "Cabin" pada data tersebut, atau, bisa jadi penumpang tersebut memang memegang tiket non-kabin. Kita dapat menjadikan ini sebagai fitur dengan syarat. Syaratnya adalah, kita harus mengenaralisir nilai kabin dan non kabin.
- "**Embarked**" --> "Embarked" adalah informasi tentang dimana penumpang akan turun. Dikarenakan targedi Titanic terjadi sebelum Titanic bersandar disalah satu lokasi embarkasi, maka, informasi ini menjadi tidak relevan untuk menjadi acuan apakah penumpang selamat atau tidak.

Berdasarkan analisa yang telah kita lakukan, maka kita mendapatkan,

- "Pclass"
- "Sex"
- "Age"
- "Fare"
- "Cabin"

sebagai fitur yang akan kita gunakan untuk memprediksi apakah penumpang selamat atau tidak. Kita dapat melihat, dimensi yang digunakan mengecil. Tidak semua informasi dari data Titanic kita gunakan untuk memprediksi keselamatan penumpang. Inilah yang dimaksud dengan konsep reduksi dimensi pada proses ekstraksi fitur. Selain itu, berdasarkan analisa yang kita lakukan, kita juga melakukan pemilihan nilai-nilai yang dapat menggambarkan kondisi keselamatan penumpang. Proses inilah yang disebut proses pemiliah nilai atau fitur.

Pertanyaan selanjutnya adalah, apakah fitur yang kita pilih sudah tepat? Jawabannya bisa jadi "ya" bisa jadi "tidak". Diperlukan pengukuran lebih lanjut terkait dengan hal ini. Kita akan mempelajarinya sepanjang perjalanan kita mempelajari materi pembelajaran mesin. *So, keep learning*!

Ekstraksi Fitur Berdasarkan Jenis Data

Kita telah mengetahui konsep ekstraksi fitur menggunakan contoh data Titanic. Data tersebut merupakan data tabular yang termasuk kedalam jenis data terstruktur. Seluruh fitur didapakan dari nilai-nilai pada setiap kolom dengan sangat mudah. Kita dapat langsung mengetahui maksud dan tujuan dari nilai fitur tersebut.

> *Lalu bagaimana dengan fitur pada data semi-terstruktur dan tidak terstruktur?*

Ekstraksi Fitur Data Semi-Terstruktur

Seperti yang telah dijelaskan pada modul sebelumnya, data semi-terstruktur adalah data yang tidak mengikuti pola terstruktur seperti pada tabel ataupun basis data relational, akan tetapi masih memiliki penanda berupa tags ataupun keys yang menyimpan untuk membedakan informasi satu dengan yang lain serta dapat memiliki hirarki. Berdasarkan definisi tersebut, kita tetap dapat mengekstraksi fitur dari data semi-terstruktur berdasarkan tags atau keys yang dimiliki oleh data tersebut. Beberapa contoh data semi terstrutkur yang poluler adalah data yang disimpan dalam bentuk Extensible Markup Language (XML) dan Javascript Object Notation (JSON).

Perhatikan contoh dokumen JSON dibawah ini,

```
[
    "mahasiswa": {
        "nim": "52101000175",
        "nama": "Alex Gorgon",
        "jenis_kelamin": "L",
        "dosen_wali": "Raiden Kolomogorov",
        "ipk": 3.5,
        "perkuliahan": [
            "Dasar Pemrograman",
            "Pemrograman Berbasis Objek",
            "Statistika",
            "Pembelajaran Mesin"
        ]
    },
    "mahasiswa": {
        "nim": "52101000199",
        "nama": "Kenkei Reinan",
        "jenis_kelamin": "P",
        "dosen_wali": "Raiden Kolomogorov",
        "ipk": 3.75,
        "perkuliahan": [
            "Dasar Pemrograman",
            "Pemrograman Berbasis Objek",
            "Statistika",
            "Pembelajaran Mesin"
        ],
        "prestasi": [
            "Juara 1 GEMASTIK",
            "Puteri Indoensia 2045",
            "Juara 1 Liga Voli Mahasiswa"
        ]
    }
]
```

Dokumen JSON tersebut berisi informasi tentang profil mahasiswa pada sebuah kampus. Berdasarkan dokumen tersebut kita dapat mengambil berbagai macam informasi sesuai dengan keys yang diberikan, seperti "nim" ataupun "nama". Dari dokumen JSON tersebut kita juga dapat dengan jelas mengetahui bahwa perbedaan antara data satu dengan data yang lainnya tidak dibedakan berdasarkan baris ataupun kolom, melainkan menggunakan bentuk *array* ataupun *dictionary*. Strukturnyapun sangat memungkinkan berbeda antara satu data dengan yang lain.

Setelah mengekstaksi nilai-nilai yang diperlukan dari data semi-terstruktur, selanjutnya kita dapat menggunakan teknik pengolahan yang serupa dengan teknik pengolahan pada data terstruktur. Perbedaanya adalah terdapat diperlukan satu tahapan untuk mengambil nilai-nilai dari keys sesuai dengan kebutuhan yang kita perlukan.

Ekstraksi Fitur Data Tidak Terstruktur

Kembali ke definisi data tidak terstruktur, yaitu merupakan data yang tidak mengikuti atau menggunakan pola penyusunan (skema) tertentu, mungkin akan tampak sulit untuk melakukan proses ekstraksi fitur. Tidak ada kolom, baris, ataupun keys pada data ini. Lalu bagimana kita mengetahui fiturnya? Sebelum ini mari kita diskusikan terlebih dulu contoh data-data tidak terstruktur.

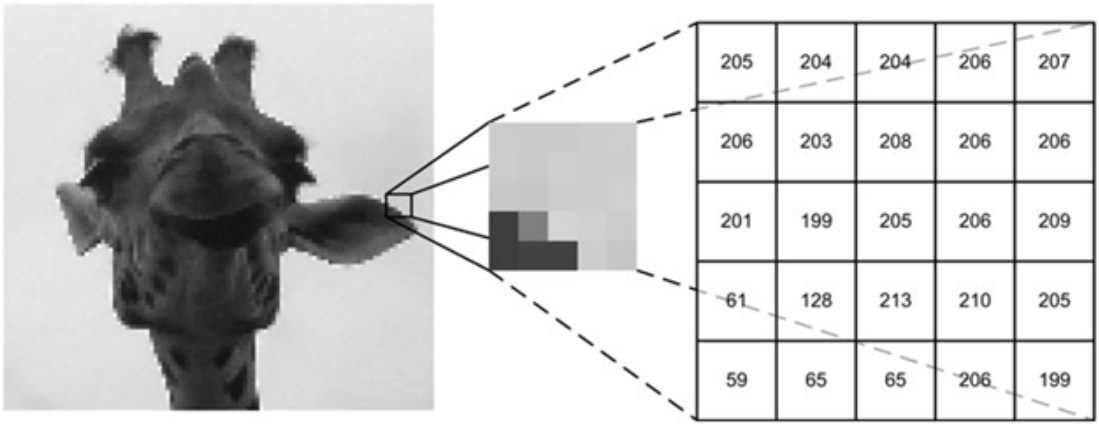
Contoh data tidak terstrutkur diantaranya adalah konten multimedia (suara, gambar, video), dokumen teks, ataupun data sosial media.

- Konten multimedia --> Data suara, gambar, ataupun video tidak disusun berdasarkan struktur tertentu, setiap nilainya bebas untuk mengisi "ruang-ruang" sepanjang-panjang data.
- Dokumen teks --> Dokumen teks seperti PDF ataupun dokumen dari aplikasi pengolah kata mungkin dapat dikatakan semi-terstruktur. Hal ini karenakan mereka memiliki meta-data yang terstruktur untuk menyimpan informasi terkait dokumen tersebut. Akan tetapi, konten didalam dokumen itu sendiri tidak terstruktur. Setiap orang dapat menulis apa saja disana dengan struktur dan rincian yang berbeda-beda.
- Data media sosial --> Sama halnya dengan dokumen teks, sosial media juga dapat dilihat dari sudut pandang semi-terstrukur atau tidak. Semi-terstruktur karena data seperti postingan di media sosial pasti memiliki metadata seperti kapan waktu posting, siapa yang memposting, hashtag yang digunakan, jumlah like, dan sebagainya. Akan tetapi konten yang diposting merupakan data tidak terstruktur. Untuk mengetahui maksud dari konten tersebut, misalnya dalam konteks sentimen, maka kita harus melakukan pengolahan lanjutan untuk mengetahuinya.

Lalu, jika tidak memiliki struktur, bagimana kita mendapatkan fiturnya? Ingat kembali konsep fitur.

> Fitur adalah nilai-nilai unik yang dapat merepresentasikan sebuah data.

Meskipun tidak memiliki struktur, kita hanya perlu mendapatkan nilai-nilai untuk dari data tidak terstruktur. Sebagai contoh, perhatikan Gambar 2.3.



Gambar 2.3 Fitur Pada Gambar

(Sumber: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/)

Sebuah gambar, sebetulnya merupakan kumpulan dari piksel kecil yang menyusun gambar tersebut. Setiap piksel akan memiliki nilainya masing-masing. Untuk mendapatkan fitur dari gambar, kita hanya perlu memperhatikan nilai-nilai piksel dari gambar. Fitur bisa didapatkan dari jumlah nilai piksel tertentu, kedekatan antar nilai, dan sebagainya. Fokusnya adalah, nilai-nilai unik dalam sebuah data tidak terstruktur tetap mungkin untuk ditemukan. Namun untuk memenumukannya, terkadang kita memerlukan proses tambahan yang disebut proses pra-pengolahan data.

Pra Pengolahan Data

Pra pengolahan data merupakan proses yang dilakukan sebelum proses ekstraksi fitur. Proses ini merupakan tahap persiapan agar data yang digunakan pada tahap ekstraksi fitur memiliki standar yang sama. Beberapa hal yang dapat dilakukan pada tahap ini adalah,

- Data imputation --> Mengisi nilai data yang kosong.
- Normalisasi --> Menyekalakan nilai variabel pada rentang tertentu.
- Standarisasi --> Menyekalakan nilai variabel dengan membuat data memiliki nilai $\mu = 0$ dan $\sigma = 1$.
- Resizing --> Menyesuaikan ukuran dimensi (biasanya pada citra).
- Quantization --> Disebut juga sebagai *binning*, yaitu mengelompokkan data berdasarkan rentang tertentu.
- Encoding --> Mengkodekan data kedalam kode nilai tertentu. Teknik ini biasanya digunakan untuk data-data ordinal.
- Dan masih banyak lagi.

Pada modul praktikum ini, kita akan fokus kepada data imputation, normalisasi, dan standarisasi.

Data Imputation

Data imputation merupakan teknik untuk mengganti nilai yang hilang didalam sebuah data. Masih ingat dengan kolom "Cabin" pada data Titanic? Agar data tersebut dapat diolah, kita harus mengganti data tersebut dengan nilai yang lain. "NaN" berarti "Not a Number". Mungkin terlihat seperti kurang relevan dikarenakan "Cabin" berisi nilai karakter (string), akan tetapi "NaN" juga dapat diartikan tidak ada nilai sama sekali disana.

Untuk mengatasi masalah tersebut, terdapat beberapa teknik yang dapat kita lakukan, diantaranya adalah,

- Menggantinya dengan nilai mean.
- Menggantinya dengan nilai median.
- Menggantinya dengan nilai modus.
- Menghapus data yang menggandung nilai "NaN".

Akan tetapi perlu diingat kembali terkait dengan konsep distribusi data. Teknik imputasi yang digunakan dapat disesuaikan dengan pola distribusi data yang dimiliki.

Normalisasi

Normalisasi adalah salah satu teknik penyekalaan fitur untuk menghasilkan nilai fitur yang berada dalam rentang baru, yaitu antara 0 dan 1. Persamaan berikut digunakan pada teknik normalisasi.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

′ adalah skala variabel dalam rentang yang baru, adalah nilai data yang akan dinormalisasi, adalah nilai minimal dari variabel, dan adalah nilai terbesar dari variabel.

Sebagai contoh, pada sebuah data diketahui nilai minimum adalah -10 dan nilai maksimumnya adalah 30. Jika data yang akan dinormalisasi adalah 18.8. Maka perhitungan normalisasinya adalah sebagai berikut,

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

$$x' = \frac{18.8 - (-10)}{30 - (-10)}$$

$$x' = \frac{28.8}{40} = 0.72$$

Standarisasi

Model pembelajaran mesin dapat bekerja lebih baik ketika dilatih pada data yang terstandarisasi. Standarisasi melibatkan penyekalaan ulang pada distribusi nilai, sehingga didapatkan nilai tengah (mean) dari data yang diamati adalah 0 dan nilai standar deviasinya adalah 1. Seperti halnya pada normalisasi, standardisasi berguna untuk model pembelajaran mesin saat data yang digunakan memiliki rentang skala yang jauh berbeda. Persamaan berikut digunakan untuk mendapatkan nilai fitur yang terstandarisasi.

$$x" = \frac{x - \mu}{\sigma}$$

Dimana adalah nilai rata-rata dari fitur dan adalah nilai simpangan baku dari fitur.

Sebagai contoh, jika diketahui nilai rata-rata dari sekumpulan data adalah 10.0. Kemudian simpangan bakunya adalah 5.0. Jika sebuah data memiliki nilai 20.7. Maka dengan menggunakan nilai-nilai tersebut, kita dapat menghitung nilai standarisasinya, yaitu,

$$x" = \frac{x - \mu}{\sigma}$$

$$x" = \frac{20.7 - 10}{5} = 2.14$$

Encoding

Dalam pembelajaran mesin terdapat 2 jenis data yang paling banyak digunakan yaitu tipe data kategorikal dan tipe data numerikal. Data numerik melibatkan fitur yang terdiri dari angka, seperti bilangan bulat atau bilangan desimal, sedangkan tipe data kategorikal adalah atribut yang diperlakukan sebagai simbol berbeda atau hanya nama. Data kategorikal digunakan untuk data yang tidak dapat dihitung secara kuantitatif, sehingga tidak dapat menerima operasi matematika seperti penjumlahan atau perkalian. Namun demikian, nilai-nilainya dapat dibedakan antara satu dengan lainnya. Contoh data kategorikal adalah,

- golongan darah manusia (A, AB, B, O).
- konversi nilai huruf pada mata kuliah (A, B+, B, C+, C, D, dan E).
- tingkatan juara pada ajang perlombaan (juara 1, juara 2, juara 3).

Variabel numerik dapat diubah menjadi variabel ordinal dengan membagi rentang variabel numerik menjadi beberapa bin dan menetapkan nilai ke setiap bin. Misalnya, variabel numerik antara 1 sampai 20 dapat dibagi menjadi variabel ordinal dengan 4 label dengan hubungan ordinal: 1-5, 6-10, 11-15, 16-20. Proses Ini disebut diskritisasi. Oleh karena itu, dapat didefinisikan bahwa,

- **Variabel Nominal** --> Variabel terdiri dari sekumpulan nilai diskrit terbatas tanpa hubungan antar nilai atau tingkatan secara alamiah. Contohnya adalah data nama kota seperti Jakarta, Bandung, Bali.
- **Variabel Ordinal** --> Variabel terdiri dari sekumpulan nilai diskrit yang terbatas dengan urutan peringkat antar nilai. Contohnya variabel ordinal adalah ketika terdapat urutan rendang, sedang, tinggi.

Beberapa implementasi algoritma pembelajaran mesin mengharuskan semua data harus menjadi data numerikal. Dalam artian bahwa data kategorikal harus diubah ke dalam bentuk numerikal. Ada tiga pendekatan umum yang dapat digunakan untuk melakukan konversi variabel ordinal dan variabel kategorikal menjadi nilai numerikal, yaitu,

- Ordinal encoding
- One-hot encoding
- Dummy variable encoding

Ordinal Encoding

Pada metode ordinal encoding, setiap kategori yang unik diberi nilai integer. Misalnya, "merah" adalah 1, "hijau" adalah 2, dan "biru" adalah 3. Biasanya nilai integer yang digunakan berawal dai nilai 0. Ordinal encoding lebih cocok untuk data bertipe variabel nominal dimana tidak terdapat hubungan atau urutan antar variabel.

Sebagai contoh, terdapat data dengan tipe kategorikal seperti yang tertera pada tabel dibawah ini. Data tersebut akan diubah menjadi data dalam bentuk numerik. Setiap data akan dirupakan kedalam bentuk angka secara berurutan.

Kampus Vokasi di Indonesia
Politeknik Negeri Malang
Politeknik Negeri Batam
Politeknik Elektronika Negeri Surabaya
Politeknik Negeri Semarang

Setelah diurutkan maka urutan pertama akan diberi nilai 0, 1, 2, dan seterusnya, sehingga didapatkan hasil pada Tabel dibawah ini.

Kampus Vokasi di Indonesia	Nilai Ordinal
Politeknik Negeri Malang	0
Politeknik Negeri Batam	1
Politeknik Elektronika Negeri Surabaya	2
Politeknik Negeri Semarang	3

One-Hot Encoding

Metode one-hot encoding merepresentasikan nilai kategorikal menggunakan 1 fitur biner untuk setiap nilai yang memungkinkan. Untuk data kategorikal (variabel nominal) dimana tidak ada hubungan urutan peringkat antar nilai penggunaaan ordinal encoding tidak memberikan performa yang bagus pada model pembelajaran mesin. Memaksakan hubungan urutan (ordinal) melalui ordinal encoding memungkinkan model untuk berasumsi bahwa terdapat urutan antar kategori sehingga mengakibatkan kinerja yang buruk atau hasil yang tidak diharapkan. Dalam hal ini, one-hot encoding dapat diterapkan terhadap data yang memiliki tipe ordinal. Tabel dibawah ini menunjukkan contoh one-hot encoding.

Kampus Vokasi di Indonesia	0	1	2	3
Politeknik Negeri Malang	1	0	0	0
Politeknik Negeri Batam	0	1	0	0
Politeknik Elektronika Negeri Surabaya	0	0	1	0
Politeknik Negeri Semarang	0	0	0	1

Langkah pertama yang dilakukan adalah melakukan pengurutan data. Setelah diurutkan selanjutnya nilai biner akan ditambahkan kepada setiap kategori. One-hot encoding akan menambah kolom fitur sesuai dengan nama politeknik yang ada di data. Nilai 1 menunjukkan bahwa pada baris tersebut terdapat data politeknik tersebut sedangkan nilai 0 menunjukkan sebaliknya. Sebagai contoh, Politeknik Negeri Malang akan direpresentasikan dengan [1, 0, 0, 0] dengan “1” untuk nilai biner pertama, kemudian Politeknik Elektronika Negeri Surabaya direpresentasikan dengan [0, 1, 0, 0], dan seterusnya.

Dummy Variable Coding

One-hot encoding membuat satu variabel biner untuk setiap kategori. Terdapat redundansi dalam one-hot encoding. Contoh jika [1, 0, 0] mewakili “Politeknik Negeri Malang” dan [0, 1, 0] mewakili “Politeknik Elektronika Negeri Surabaya”. Maka, pada dummy variabel encoding setiap nilai fitur akan diwakili oleh dua nilai biner, [0, 0]. Tabel dibawah merupakan contoh penerapan dummy variable encoding.

Kampus Vokasi di Indonesia	0	1
Politeknik Negeri Malang	0	0
Politeknik Negeri Batam	0	1
Politeknik Elektronika Negeri Surabaya	1	0
Politeknik Negeri Semarang	1	1

Strategi Pembuatan Data Latih, Validasi, dan Uji

Random Split dan Stratified Split

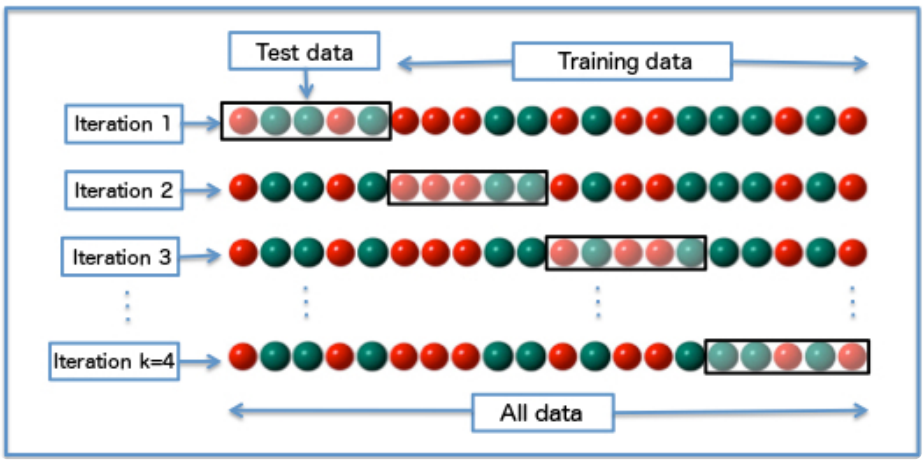
Pemisahan antar jenis data yaitu data latih, data validasi, dan data uji untuk kebutuhan pembuatan model pembelajaran mesin sudah kita bahas pada modul sebelumnya. Pada modul yang lalu, kita telah mengenal *random split* untuk melakukan hal tersebut. Akan tetapi, terdapat teknik lain yang dapat kita gunakan untuk melakukan *splitting* data, yaitu *stratified split* atau *stratified random split*. Sedikit berbeda dengan *random split*, dimana kita langsung memilih secara acak data yang akan kita gunakan sebagai data latih, validasi, maupun uji, ***pada stratified split, teknik ini mengenal strata pada prosesnya***.

Apa maksud dari strata (*stratified*)? Strata dalam konteks ini adalah kelompok, atau kita dapat menganologikannya sebagai label. *Stratified split* akan membagi data kedalam porsi latih, validasi, dan uji sesuai dengan proporsi setiap label atau kelas. Hal ini menyebabkan jumlah tiap label akan memiliki rasio yang sama. Pada ilmu statistika, teknik ini juga masuk dalam teknik sampling. Gambar 2.4 merupakan ilustrasi dari *stratified sampling* yang digunakan pada ilmu statistika.

Gambar 2.4 Ilustrasi Stratfied Sampling
(Sumber: https://www.kaggle.com/code/jardelnascimento/stratified-sampling)

Cross Validation

Setelah kita memahami bagaimana cara melakukan spliting data training, validasi, dan testing, pada bagian ini kita akan belajar terkait dengan teknik lain dalam melakukan *splitting* data, yaitu *cross validation*. Apa itu *cross validation*? *cross validation* adalah teknik pada machine leanring untuk mengevaluasi model dengan cara melakukan evaluasi berganda (*multiple evaluation*). Evaluasi berganda dilakukan dengan cara membagi data menjadi data latih dan uji, yang diistilahkan sebagai ***fold*** sedemiakan sehingga setiap data pernah menjadi data latih ataupun data uji. Hasil pengujian pada model ini adalah rata-rata hasil untuk setiap *fold*. Besaran *spliting* data akan ditentukan oleh nilai *k*-nya. Sebagai contoh, jika kita menggunakan nilai $k = 4$, maka data akan dibagi menjadi 4 bagian. Salah satu dari bagian tersebut akan menjadi data validasi. Kemudian, untuk setiap iterasi, data validasi akan diganti sesuai dengan jumlah *fold*. Untuk memudahkan pemahaman, perhatikan Gambar 2.5.



Gambar 2.5 Ilustrasi Cross Validation
(sumber: https://upload.wikimedia.org/wikipedia/commons/1/1c/K-fold_cross_validation_EN.jpg)