BAB 2
LANDASAN TEORI

Bab 2 berisi informasi berupa landasan teori yang memiliki relasi dengan yang berikutnya digunakan saat melakukan penelitian. Pembahasan pada bab ini meliputi konsep dari *retrofit* atau *retrofitting*, berbagai teori, dan metode penelitian. Landasan teori bertujuan untuk memberikan persiapan bagi pembaca mengenai implementasi algoritma *clustering* dengan *K-Means* dan berikutnya algoritma *classification* dengan *Gaussian Naive Bayes* untuk model prediksi kelayakan *retrofit* mobil ke mobil pintar atau *smartcar*.

2.1 Tinjauan Teori

Untuk memulai proses penelitian, referensi terkait penelitian terdahulu perlu dilakukan. Referensi tersebut bertujuan sebagai komparasi metode, proses, dan hasil berdasarkan penelitian yang telah dilakukan. Pada penelitian ini, referensi topik penelitian perlu memiliki relevansi mengenai metode baik dalam bentuk model prediksi menggunakan *clustering* atau *classification* saja, dan dapat dalam bentuk *hybid algorithm*, sementara pada bagian topik, minimal mengenai pengukuran performa produksi emisi *greenhouse gas* (gCO_2e) yang terdiri dari berbagai senyawa kimia, efisiensi bahan bakar, dan objek berupa transportasi berupa mobil.

Tabel 2.1. *Research Gap* dari penelitian sebelumnya

Artikel Jurnal	Nama Jurnal	Penulis & Tahun	Metode	Temuan
gasoline vehicle fuel consumption in energy and environmental impact based on machine learning and multidi mensional big data	*Energies*	*Yang, Yushan and Gong, Nuoya and Xie, Keying and Liu, Qingfei*, 2022	*Linear Regression, Naive Bayes, Neural Network, Random Forest, LightBGM* (MAE, MAPE, MSE, R^2)	Terdapat 25 *features*, *Random Forest* tampil lebih baik karena menghasilkan MEA 0.630 L/100 km, MAPE of 7.5 Percent, MSE 0.805, R^2 0.776, & 10-fold cross-validation score 0.791
Predicting vehicle fuel consumption based on multi-view deep neural network	*Neuro computing*	*Li, Yawen and Zeng, Isabella Yunfei and Niu, Ziheng and Shi, Jiahao and Wang, Ziyang and Guan, Zeli*, 2022	*Multi View Deep Neural Network, Lasso Regression, Random Forest, Gradient Boosted Decision Tree, Root Mean Square Error*	*MVDNN* menghasilkan RMSE terbaik yaitu 0.993

Tabel 2.2. *Research Gap* dari penelitian sebelumnya

Artikel Jurnal	Nama Jurnal	Penulis & Tahun	Metode	Temuan
Analysis and prediction model of fuel consumption and carbon dioxide emissions of light-duty vehicles	*Applied Sciences*	*Hien, Ngo Le Huy and Kor, Ah-Lian*, 2022	*Time series regression, Linear regression, Univariate polynomial regression, Multiple linear regression, Logarithmic regression, Exponential regression, Multivariate polynomial regression, Convolutional neural network*	*Univariate polynomial regression* menghasilkan akurasi sebesar 98 persen untuk prediksi *single variable*, *Multivariate polynomial & Multiple linear regression* menghasilkan akurasi 75 persen untuk *multi variable*, *Convolutional neural network* menghasilkan akurasi sekitar 70 persen.

2.1.1 Penelitian ke-1

Penelitian ini berjudul "*Predicting Gasoline Vehicle Fuel Consumption in Energy and Environmental Impact Based on Machine Learning and Multidimensional Big Data*" yang ditulis oleh "*Yang, Yushan and Gong, Nuoya and Xie, Keying and Liu, Qingfei*" dan di *publish* pada tahun 2022. Penelitian ini bertujuan untuk melakukan prediksi dengan metode regresi berdasarkan penelitian sebelumnya yang memiliki kekurangan pada jumlah *dataset* yang digunakan dan minim *model algoritma* yang digunakan. Dengan adanya masalah yang terdapat pada penelitian sebelumnya, tim peneliti asal Tiongkok ini melakukan peningkatan pada metode algoritma hingga berjumlah lima algoritma dengan bertujuan untuk melakukan penelitian terkait pernyataan bahwa efisiensi bahan bakar yang dipasarkan pada penjualan transportasi kecil memiliki klaim yang salah.

Pada penelitian ini, diketahui memiliki 25 *feature* yang terdiri dari empat tipe *feature* dan memiliki jumlah 171.089 *sample data* selesai tahap *pre-processing* sebagai *dataset* yang digunakan untuk tahap berikutnya. Tipe pertama berjumlah 10 *feature* dan bertipe *overview* dari kendaraan yang meliputi *User ID (Pemiliki kendaraan), City (Kota asal kendaraan), Time (Waktu update informasi kendaraan (Bulan, Tahun)), Brand Name (Nama merek kendaraan), Series Name (Nama seri model kendaraan), Version Year (Tahun produksi kendaraan), Engine (Kubikasi mesin, Tenaga mesin (ps), konfigurasi mesin), Gearbox (Tipe transmisi), refConsumption (referensi konsumsi bahan bakar (L/100 km)), realConsumption (konsumsi bahan bakar nyata (L/100 km))*. Tipe kedua adalah cara pengendaraan yang berjumlah 7 dan dipisah menjadi 2 bagian yaitu distribusi *Gender (Gender dari pengendara), Age (Umur pengendara)*, kemudian bagian kedua memiliki 5 *feature* yang terdiri dari *Car frequency (Frekuensi berkendara), Fuel economy consciousness (Kesadaran atas efisiensi bahan bakar), Driving skill (keahlian pengendara), Driving speed (Kesadaran atas kecepatan pengendara), Driving habit (Kesadaran atas kebiasaan dalam pengendara)*. Tipe terakhir adalah *environment factors* yang berjumlah 11 *feature* meliputi *Average pressure (rata-rata tekanan (0.1 hPa)), Average temperature (rata-rata temperatur (0.1 C)), Average temperature anomaly (rata-rata temperatur anomali (0.1 C)), Mean relative humidity (rata-rata kelembapan relatif (persen)), Average wind speed (rata-rata kecepatan udara (m/s)), Maximum wind direction (Maximum arah mata angin (Azimuth)), Extreme maximum wind direction (Arah mata angin extreme (Azimuth), Average precipitation (rata-rata presitipasi (mm)), Daily precipitation* $\geq$ *0.1 mm days (presipitasi harian (days)), Sunshine time (waktu cerah (0.1 hours)), & Road Grade (kondisi jalan)*.

Penelitian ini menggunakan lima algoritma untuk proses *modelling* yaitu *Linear regression, Naive Bayes regression, Neural Network regression, Random Forest regression, Light Gradient Boosted Machine*, sementara untuk proses evaluasi menggunakan *Mean Absolute Error(MAE), Mean Absolute Percentage Error (MAPE), Mean Square Error (MSE), &* R^2*, K-fold cross validation* . Setelah dilakukan 10-*cross-validation* berikut adalah hasilnya *Linear regression* 0.558, *Naive Bayes* 0.558, *Neural Network* 0.595, *Random Forest* 0.791, & *LightGBM* 0.722.

2.1.2 Penelitian ke-2

Penelitian ini berjudul "*Predicting vehicle fuel consumption based on multi-view deep neural network*" yang di tulis oleh "*Yang, Yushan and Gong, Nuoya and Xie, Keying and Liu*" dan di *publish* pada tahun 2022. Penelitian ini bertujuan untuk melakukan prediksi konsumsi bahan bakar kendaraan yang memiliki *feature* yang sebanyak 21 dan akan dibagi menjadi tiga tipe. Pertama adalah tipe *driver feature* berjumlah 7 *feature* yang meliputi *age (umur pengendara), gender (gender pengendara), freq(frekuensi berkendara), aware (kesadaraan atas keamanan dalam* *berkendara), skill (keahlian pengendara), speed (kecepatan pengendara (km/h)), & habit (habit pengendara*. Kedua tipe *enviromental feature* berjumlah 11 *feature*, pada tipe ini *feature* meliputi *V1004 (rata-rata tekanan (hPa)), V12001 (rata-rata temperatur (Celcius)), V12201 (rata-rata temperatur anomali (C)), V13003 (rata-rata kelembapan relatif (persen)), V11002 (rata-rata kecepatan udara (m/s)), V11212 (Arah mata angin (Azimuth)), V11043 (Arah mata angin extreme (Azimuth), V13011 (rata-rata presitipasi (mm)), V13011_000 (maximum presipitasi (azimuth)), V14032 (waktu cerah (hours)), & Roadgrade (kondisi jalan)* . Terakhir adalah tipe *vehicle feature* berjumlah 3 *feature* yang meliputi *engine_mali (tenaga kendaraan (Hp)), engine_pailing (Kubikasi mesin (Liter)), & engine_guandao (pipeline)*.

Penelitian ini menggunakan teknik regresi menggunakan algoritma *Lasso algorithm* dengan tujuan untuk mencari *feature importance analysis* untuk setiap *feature sample* kemudian menggunakan *Multi-view deep neural network (MVDNN)* kepada tiga tipe *feature* dan menggunakan *activation function* berupa *Rectified Linear Unit (ReLU)*. Pemisahan dari 21 *feature* menjadi 3 tipe yang berbeda dilakukan dengan tujuan untuk melakukan optimisasi proses dari *Deep Neural Network*. Untuk pengujian tingkat kredibilitas yang hasilkan oleh *Lasso regression* maka dilakukan komparasi *Root Mean Square Error (RMSE)* dengan 80 persen *training* dan 20 persen *testing*. Untuk proses *modelling* sendiri terdapat dua algoritma yang digunakan selain MVDNN yaitu *Random Forest* dan *Gradient Boosted Decision Tree (GBDT)*, ketiga algoritma diketahui memiliki *epochs* sebanyak 200 dan menghasilkan RMSE yang berbeda seperti *Random Forest* 1.579, *GBDT* 1.757, dan *MVDNN* 0.993, sehingga dapat disimpulkan bahwa metode *MVDNN* merupakan hasil yang akurat.

2.1.3 Penelitian ke-3

Penelitian ini berjudul "*Analysis and prediction model of fuel consumption and carbon dioxide emissions of light-duty vehicles*" yang ditulis oleh "*Hien, Ngo Le Huy and Kor, Ah-Lian*" dan di *publish* pada tahun 2022. Penelitian ini bertujuan untuk menganalisa performa berdasarkan efisiensi kendaraan ringan, relasi antar faktor, dan membangun *model* prediksi kendaraan. *Dataset* yang digunakan untuk penelitian ini bersumber dari *database* resmi milik pemerintah *Canada*. Selama proses pembangunan *model* ini, terdapat 8 *features* yang terdiri dari *Engine Size (Kubukasi mesin (Liter)), Cylinders (Silinder mesin), Fuel Consumption in City (Konsumsi dalam kota (L/100 km)), Fuel Consumption in Highway (Konsumsi luar kota (L/100 km)), Total Fuel Consumption (Total konsumsi bahan bakar (L/100 km)),* CO_2 *Emissions (Emisi karbon (g/km)),* CO_2 *Rating (Rating karbon dioksida), Smog Rating (Rating tingkat emisi yang dihasilkan (1-10))*.

Penelitian ini melakukan proses pengujian dengan berbagai teknik yang berbeda yaitu *t-test* untuk melakukan komparasi antara konsumsi dalam kota dan luar kota untuk satu kendaraan yang sama, *analysis of varience*

(ANOVA) untuk komparasi total konsumsi bahan bakar dengan karbon dioksida yang dihasilkan untuk setiap kendaraan dan tipe bahan bakar memiliki perbedaan yang berbeda, melakukan analisa korelasi menggunakan *heatmap* dalam memvisualisasikan relasi antara *feature*, dan *chi-square* untuk melakukan investigasi apakah terdapat perbedaan antara *data* pada tahun 2021 dengan tahun 2017 hingga 2020 dan *Root mean square error (RMSE)* untuk evaluasi *regression model*. Penelitian ini menggunakan algoritma *Time series regression, Linear regression, Univariate Polynomial regression (dari 1 hingga 5 derajat), Multiple Linear regression, Logarithmic regression, Exponential regression, Multivariate Polynomial regression & Convolutional Neural Network (CNN)*.

Penelitian terdahulu yang menjadi acuan penelitian ini, diketahui mengimplementasikan pendekatan prediksi berbasis metode regresi. Sementara itu, penelitian yang dijalankan ini menggunakan pendekatan berbasis klasifikasi. Meskipun memiliki perbedaan pendekatan, ketiga penelitian terdahulu memiliki kemiripan dengan penelitian ini seperti penentuan topik, konten *dataset* berupa informasi berupa spesifikasi mobil, dan *feature* yang diproses selama penelitian seperti emisi gas buang, dan efisiensi konsumsi bahan bakar. Pendekatan prediksi menggunakan *clustering* dan *classification* dalam penelitian ini bertujuan sebagai pelengkap dari penelitian terdahulu.

2.2 Retrofit

Retrofit adalah suatu proses atau implementasi pembaruan teknologi yang tersedia pada era terkini kepada suatu objek yang sebelumnya saat diproduksi tidak tersedia teknologi tersebut. Kata *retrofit* tersedia pada akhir tahun 1940an yang memiliki tujuan untuk melakukan modernisasi suatu objek lawas dengan teknologi yang terbaru [7]. Hal ini dilakukan untuk menjadi sebuah solusi berdasarkan efisiensi tenaga dan waktu. *Retrofit* sendiri juga telah diterapkan pada industri seperti infrastruktur bangunan, salah satu contohnya adalah melakukan pembaruan fungsi bangunan dengan cara mengimplementasikan *modular green roofing* kepada atap bangunan yang memiliki efek seperti pengurangan temperatur tinggi pada area sekitar, dan meningkatkan kualitas udara[8]. Sementara pada contoh lainnya adalah dilakukan konversi teknologi menjadi mobil listrik menggunakan *chassis* lama sebagai pondasi utama dengan tujuan untuk mencapai *cost-effectiveness* dan mengurangi dampak dari emisi *Green House Gas* [9].

2.3 Machine Learning

Machine learning adalah cabang dari *artificial intelligence* yang merevolusi indusitri analisis *data* dengan kemampuan untuk mempelajari pola pada *data* secara mandiri dan memperbarui informasi [10]. *Machine learning* diketahui memiliki teknik yang berbeda sesuai dengan tujuannya. Teknik *clustering* merupakan bagian dari *unsupervised learning* yang memiliki fungsi utama untuk melakukan pengelompokan *data* ke dalam *cluster* berdasarkan kemiripan informasi milik *data* seperti jarak tanpa memerlukan *label*. Berikutnya, teknik *classification* merupakan bagian dari *supervised learning* yang bertujuan untuk prediksi atau mengategorikan *data* ke dalam *class* berdasarkan *data* yang telah memiliki *label*.

2.3.1 Clustering

Clustering adalah teknik *unsupervised learning*, teknik ini bekerja dengan cara mengelompokkan sekumpulan objek atau titik data ke dalam *& group* berdasarkan kesamaan yang dimiliki, sehingga *item* dalam *cluster* yang sama atau lebih mirip satu sama lain dibandingkan dengan *item* dalam *cluster* yang berbeda. Pendekatan ini diimplementasikan untuk mengidentifikasi pola, relasi, dan struktur dalam data tanpa mengetahui *label* sebelumnya [11] .

2.3.2 K-Means

K-Means adalah salah satu algoritma *clustering* yang bertujuan untuk melakukan pengelompokan data berdasarkan kemiripan *feature* yang ada. Algoritma ini bekerja dengan cara melakukan pengulangan kepada *data points* ke *centeroid* terdekat kemudian melakukan pembaruan *data points* ke *centeroid* lagi hingga kondisi *convergence* tercapai. Algoritma ini menggunakan paramater berupa *n_clusters* sebagai jumlah *cluster* yang terdapat, *init* sebagai inisialisasi dari *centeroid*, *tol* sebagai nilai toleransi dari *convergence* (*default value 1e-4*), *random_state* untuk melakukan *random seeding* kepada hasil dari *cluster* (*default value 0*), *max_iter* jumlah pengulangan dari iterasi yang dilakukan dalam sekali jalan (*default value 300*), *verbose* untuk melakukan pemantauan selama proses iterasi progres (*default value 0*), *copy_x* melakukan *copy* kepada *input* sebelum komputasi dilakukan (*default value True*), dan *algorithm* adalah gaya algoritma dari *expectation-maximization* yang digunakan, secara *default* menggunakan *'lloyd'*, terdapat opsi alternatif seperti *'elkan'* bertujuan untuk dapat lebih efisien terhadap *cluster* yang sudah diketahui, tetapi mengonsumsi *memory* yang lebih intens karena melakukan alokasi ektra *array* dari *n_samples, & n_clusters* [12].

2.3.3 Mini Batch K-Means

Mini Batch KMeans adalah variasi dari metode *K-Means*. Metode ini miliki *parameter* yang sama dengan metode *K-Means* seperti *n_cluster* sebagai jumlah *cluster* yang ada, *init* sebagai inisialisasi dari *centeroid*, *max_iter* sebagai jumlah pengulangan dari iterasi yang dilakukan dalam sekali jalan (*default value 100*), *tol* sebagai nilai toleransi dari *convergence* (*default value 1e-4*), dan *random_state* sebagai *random seeder* kepada *cluster* (*default value 0*). Metode ini memiliki *parameter* tambahan dalam melakukan konfigurasi saat menjalankan proses *KMeans*. *Parameter* tersebut bertujuan untuk dapat melakukan proses *KMeans* tetapi dengan konfigurasi *parameter* mengenai performa. Daftar *parameter* tersebut adalah *batch_size* bertujuan untuk mengatur jumlah *data samples* yang diproses dalam setiap iterasi, *max_no_improvement* sebagai *stopping criteria* saat melakukan iterasi, dan *reaassignment_ratio* sebagai *controller* terhadap *centroid* yang dinilai memiliki performa cukup buruk sehingga akan diproses ulang kepada *data sample* secara acak [13].

2.3.4 Elbow method

Metode ini bertujuan untuk menemukan jumlah "K" yang diperlukan untuk melakukan *clustering*. Metode ini bekerja dengan cara kalkulasi *Within Cluster Sum of Square* (WCSS), kriteria yang ditentukan untuk menjadi jumlah poin "K" ditentukan saat melakukan *Within Cluster Sum of Square* terdapat kondisi dimana tidak dapat melakukan improvisasi. Pemilihan jumlah "K" dilakukan dengan cara melakukan visualisasi dan menganalisa yang bentuk sudut siku-siku atau dari terjadi sebuah perubahan hasil yang signifikan antara *range Inertia* antar jumlah *"K"*. Berikut dibawah ini merupakan rumus dari *Within Cluster Sum of Square* (WCSS) :

$$WCSS = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{2.1}$$

Rumus *Within Cluster Sum of Square*
Sumber : [14]

Tabel 2.3. Table *Within Cluster Sum of Square* (CWSS)

Simbol	Deskripsi
K	Jumlah *clusters*
C_i	batas akhir*clusters*
x	*data points* di *clusters* C_i
μ_i	*centeroid* dari *clusters* C_i
$\|x-\mu_i\|^2$	*squared Euclidean distance* antara poin x dengan *centeroid* μ_i

2.3.5 Silhouette score

Metode ini merupakan alternatif untuk melakukan pencarian "K". Metode ini bekerja dengan cara melakukan perbandingan jarak antara "a_i" (jarak *data point* dalam cluster yang sama) dengan "b_i" (jarak *data point* terhadap *neighbour* dari cluster terdekat). Metode ini dapat menghasilkan hasil yang lebih baik dibandingkan metode *Elbow*, tetapi juga memerlukan waktu komputasi yang lebih lama. Metode ini memiliki arti dari *value* "-1" sebagai *cluster* yang salah, *value* "0" sebagai *point* mendekati perbatas antar *cluster*, dan *value* "+1" sebagai *point* sangat cocok terhadap *cluster* terdekat. Untuk memilih jumlah "*K*" sendiri maka akan memiliki hasil dengan "*K*" yang tertinggi berdasarkan iterasi yang sudah dilakukan. Berikut dibawah ini merupakan rumus dari *silhouette score* :

$$S(i) = \frac{b_i - a_i}{max(a_i, b_i)} \tag{2.2}$$

Rumus *Silhouette score*
Sumber : [15]

Tabel 2.4. Table Silhouette score

Simbol	Deskripsi
a_i	Rata-rata jarak antara *data point* dalam *cluster* yang sama
b_i	Rata-rata jarak antara *data point* dengan *neigbour* dari *cluster* terdekat

2.3.6 Davies-Bouldin index

Davies-Bouldin Index / Score adalah Metode untuk melakukan evaluasi terhadap *cluster* yang tersedia dengan cara melakukan pengukuran *ratio* terhadap titik penyebaran antara *centroid* yang tersedia *(internal)*, dan melakukan komparasi antara *cluster (external)* bertujuan untuk mendeteksi *cluster* yang dengan *value* terendah [16]. Berikut adalah rumus dari metode evaluasi ini :

$$DBI = \frac{1}{K}\sum_{i=1}^{K} \max j \neq i \left(\frac{S_i + S_j}{M_{ij}}\right) \tag{2.3}$$

Rumus *Davies-Bouldin Index*
Sumber : [17]

Tabel 2.5. Table Davies-Bouldin Index

Simbol	Deskripsi
$\frac{1}{K}$	Rata-rata dari jumlah *cluster*
$\sum_{i=1}^{K}$	Rata-rata jarak antara *data point* dengan *neigbour* dari *cluster* terdekat
$\max j \neq i$	Untuk setiap *cluster* i, Pemilihan *value* terendah kepada *cluster* j
$\frac{S_i+S_j}{M_{ij}}$	Komparasi penyebaran ukuran S_i dan S_j dengan jarak *centeroid* M_{ij}

2.3.7 Classification

Classification adalah teknik *supervised learning* dalam *machine learning* yang melibatkan *label* atau kategori yang telah ditentukan sebelumnya untuk memutuskan kategori *data* berdasarkan *feature* yang tersedia. Tujuannya adalah untuk melatih *model* menggunakan *dataset* dengan *label*, dimana setiap *data* masukan dikaitkan dengan *output* yang sesuai. Dengan proses ini, *model* dapat mempelajari pola dan relasi antar *feature* yang menginterpretasikan *variable 'x'* dengan target tujuan yaitu *variable 'y'* [18].

A Bayes Theorem

Bayes Theorem adalah metode kalkulasi peluang probabilitas kejadian yang dialami oleh subjek berdasarkan probabilitas kondisi. Probabilitas kondisi bekerja dengan cara melakukan kalkulasi suatu kondisi yang diberikan oleh kondisi lainnya [19].

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} \tag{2.4}$$

Rumus probabilitas kondisi.

Tabel 2.6. Table probabilitas kondisi

Simbol	Deskripsi
$P(A \mid B)$	Peluang dari kondisi A diberikan kondisi dari B
$P(A \cap B)$	Terjadinya dari kondisi A diberikan kondisi B
$P(B)$	Probabilitas kondisi dari B

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{2.5}$$

Rumus Bayes Theorem.

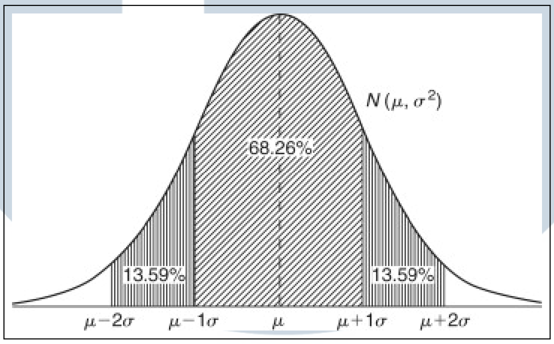
B Gaussian Naïve Bayes

Gaussian naive bayes adalah salah satu tipe *naive bayes* yang melakukan kalkulasi probabilitas berdasarkan distribusi *gaussian*. Distribusi *gaussian* memiliki tiga bagian dan nilai *min-max*, nilai *'0'* memiliki arti sebagai nilai awal, nilai *'-1'* adalah kemungkinan tidak sesuai dengan *feature* tujuan, dan nilai *'1'* adalah kemungkinan sesuai *feature* tujuan. Klasifikasi ini cocok untuk digunakan pada *dataset* yang memiliki *value* bersifat *continuous* karena sifat dari klasifikasi ini

Tabel 2.7. Table Bayes Theorem

Simbol	Deskripsi
$P(A \mid B)$	Peluang dari kondisi A diberikan kondisi dari B
$P(B \mid A)$	Terjadinya dari kondisi B jika diberikan kondisi dari A
$P(A)$	Probabilitas kondisi dari A
$P(B)$	Probabilitas kondisi dari B

adalah melakukan kalkulasi estimasi probabilitas, jika hasil estimasi menghasilkan *'1'* jauh lebih dominan dibandingkan *value '-1'* maka klasifikasi semakin akurat.



Gambar 2.1. *Gaussian distribution & Normal distribution*
sumber: [2]

$$P(x_i \mid y) = \frac{1}{\sqrt{2}\pi\sigma_y^2} e^{-\frac{\left(x_i - \mu_{y,i}\right)^2}{2\sigma_y^2}} \tag{2.6}$$

Rumus Gaussian Naive Bayes.

Tabel 2.8. Table Gaussian Naive Bayes

Simbol	Deskripsi
$P(x_i \mid y)$	Peluang dari X_i diberikan kondisi dari y
$\sqrt{2\pi\sigma_y^2}$	Faktor normalisasi, memastikan total probabilitas adalah 1
e	angka dasar dari logaritma (perkiraan memiliki nilai 2.71828)
σ_y^2	Distribusi varian

2.4 Confusion Matrix

Confusion matrix adalah sebuah metode evaluasi yang bertujuan memvisualisasikan performa algoritma *classification* berdasarkan membandingkan *actual class* dengan hasil *prediction class* [20]. Hasil visualisasi ini terdiri dari *True Positif (TP), True Negative (TN), False Positive (FP), & False Negative (FN).*

Tabel 2.9. Table Confusion Matrix

	Aktual Positif	Aktual Negative
Prediksi Positif	TP	FP
Prediksi Negative	FN	TN

Penjelasan :

1. *True Positive* (TP): Prediksi kasus benar yang benar.
2. *False Positive* (FP): Prediksi kasus benar yang salah.
3. *True Negative* (TN): Prediksi kasus salah yang benar.
4. *False Negative* (FN): Prediksi kasus salah yang salah.

2.4.1 Accuracy

Accuracy adalah salah satu *evaluation metrics* yang mengukur performa *model* berhasil memprediksi *data* dengan benar secara keseluruhan. *Metric* ini bekerja dengan cara menambahkan total dari jumlah *data* yang diuji kemudian dibagi dengan hasil dari *True Positive* (TP) dan *True Negative* (TN) [20]. Berikut adalah rumus dari *accurary* :

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.7}$$

2.4.2 F1-Score, Precision, & Recall

F1-score adalah salah satu *evaluation metrics* yang menghitung *harmonic mean* dari *metrics* lainnya yaitu *precision* dan *recall*. *F1-score* memiliki fungsi untuk memberikan gambaran mengenai performa *model* yang lebih objektif dibandingkan dengan *accuracy*. Jika salah satu *metrics* mendominasi maka *value* dari *F1-score* akan turun [20].

Berikut adalah rumus *Precision*:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{2.8}$$

Berikut adalah rumus *Recall*:

$$\text{Precision} = \frac{TP}{TP+FN} \tag{2.9}$$

Sementara berikut adalah rumus *F1- score*:

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.10}$$

2.4.3 K-Fold cross validation

K-fold cross validation adalah salah satu variasi dari *cross validation* yang bekerja dengan cara membagikan keseluruhan *dataset* secara *random* menjadi *"K" fold* yang memiliki ukuran sama besar. Selanjutnya pengujian akan megalami *iterasi* sebanyak *"K"*, dimana setiap pengulangannya terdapat satu *fold* yang bertindak sebagai *test data* sementara *fold* lainnya menjadi *train data*, hal ini terjadi hingga semua *fold* telah menjadi *test data* sekali [21].