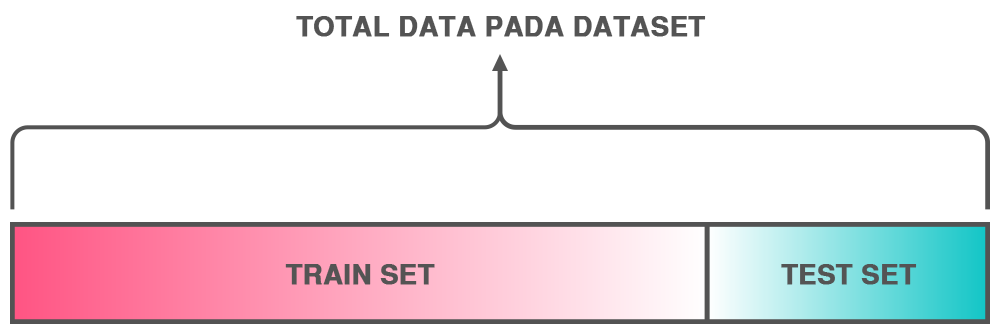
Dataset yang telah dibersihkan dan diproses kemudian siap kita latih dengan machine learning. Satu-satunya cara untuk mengetahui apakah model machine learning kita bagus atau tidak adalah dengan mengujinya pada kasus atau data baru yang belum dikenali oleh model. Kita bisa saja membuat model dan langsung mengujinya pada tahap produksi lalu memonitor kualitasnya. Tapi jika ternyata model yang kita kembangkan bekerja dengan buruk, pelanggan dan klien kita akan komplain. Selain itu, cara ini tentu akan memakan sumber daya dan biaya yang lebih besar.

**Training Set dan Test Set**

Pilihan yang lebih baik adalah dengan membagi dataset menjadi 2 bagian yaitu *data training* dan *data testing.*Berikut adalah gambaran bagaimana total data pada dataset dibagi menjadi dua bagian: *train set dan test set.*



Dengan demikian, kita bisa melakukan pelatihan model pada train set, kemudian mengujinya pada test set --sekumpulan data yang belum dikenali model. Ingat bahwa membandingkan hasil prediksi dengan label sebenarnya dalam test set merupakan proses evaluasi performa model. Dengan menguji model terhadap data testing, kita dapat melihat kesalahan yang dibuat dan memperbaikinya sebelum mulai membawa model kita ke tahap produksi.

Penting untuk kita memilih rasio yang sesuai dalam pembagian dataset. Saat membagi dataset, kita perlu membuat informasi pada kedua bagian tetap berimbang. Kita tidak ingin mengalokasikan terlalu banyak informasi pada data testingagar algoritma ML dapat belajar dengan baik pada data training. Tetapi, jika alokasi data pada data testing terlalu kecil, kita tidak bisa mendapatkan estimasi performa model yang akurat.

Data testing diambil dengan proporsi tertentu. Pada praktiknya, pembagian data training dan data testing yang paling umum adalah 80:20, 70:30, atau 60:40, tergantung dari ukuran atau jumlah data. Namun, untuk dataset berukuran besar, proporsi pembagian 90:10 atau 99:1 juga umum dilakukan. Misal jika ukuran dataset sangat besar berisi lebih dari 1 juta *record*, maka kita dapat mengambil sekitar 10 ribu data saja untuk testing alias sebesar 1% saja.

Pada modul ini, kita akan belajar membagi dataset dengan fungsi train\_test\_split dari library sklearn. Perhatikan contoh kode berikut.

1. from sklearn.model\_selection import train\_test\_split
2. x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=1 )

Dengan fungsi train\_test\_split dari library sklearn, kita membagi array X dan y ke dalam 20% data testing (test\_size=0.2 ). Misal total dataset A yang kita miliki adalah 1000 record, dengan test\_size=0.2, maka data testing kita berjumlah 200 record dan jumlah data training sebesar 800 (80%).

Sebelum proses pemisahan, fungsi train\_test\_split telah mengacak dataset secara internal terlebih dahulu. Jika tidak, data testing hanya akan berisi semua data pada kelas tertentu saja. Misal dataset A kita terdiri dari 5 kelas dengan jumlah masing-masing kelas sebesar 200 record, maka dengan proses *shuffling* sebelum pemisahan, data testing akan memiliki data dari 5 kelas yang ada. Tanpa proses shuffling, seluruh data dari kelas 1 - 4 akan berakhir di set data training, dan data testing hanya berisi data dari kelas 5 saja. Proses shuffling menjaga rasio informasi pada data training dan testing tetap berimbang.

Melalui parameter random\_state, fungsi train\_test\_split menyediakan *random seed*yang tetap untuk internal pseudo-random generatoryang digunakan pada proses *shuffling.*Umumnya, nilai yang digunakan adalah 0, atau 1, atau ada juga yang menggunakan 42. Menentukan parameter random\_state bertujuan untuk dapat memastikan bahwa hasil pembagian dataset konsisten dan memberikan data yang sama setiap kali model dijalankan. Jika tidak ditentukan, maka tiap kali melakukan split, kita akan mendapatkan data train dan tes berbeda, yang juga akan membuat akurasi model ML menjadi berbeda tiap kali di-*run.*

Berikut adalah contoh kode untuk memahami bagaimana penentuan random\_state bekerja pada dataset.

1. from sklearn.model\_selection import train\_test\_split
3. X\_data = range(10)
4. y\_data = range(10)
6. print("random\_state ditentukan")
7. for i in range(3):
8. X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_data, y\_data, test\_size = 0.3, random\_state = 42)
9. print(y\_test)

12. print("random\_state tidak ditentukan")
13. for i in range(3):
14. X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_data, y\_data, test\_size = 0.3, random\_state = None)
15. print(y\_test)

Output:

1. random\_state ditentukan
2. [8, 1, 5]
3. [8, 1, 5]
4. [8, 1, 5]
6. random\_state tidak ditentukan
7. [0, 2, 6]
8. [6, 1, 3]
9. [2, 0, 3]