
Pengenalan CSV



Pengenalan CSV

- CSV singkatan dari comma separated value, yaitu format data tabular yang biasanya menggunakan comma sebagai pemisah tiap data kolom nya
- CSV merupakan format sederhana dari format Spreadsheet seperti Microsoft Excel
- Salah satu keuntungan menggunakan format CSV adalah, karena berbasis text, sehingga mudah untuk dibaca dan dimodifikasi dengan editor apapun



Karakter Pemisah

- Walaupun secara default pemisah pada file CSV adalah , (koma), tapi tidak menjadi keharusan juga
- Beberapa format CSV kadang menggunakan pemisah ; (titik koma), karakter tab, atau yang lainnya



Contoh File CSV

Eko, Kurniawan, Khannedy, 100

Budi, , Nugraha, 90

Joko, Morro, Anwar, 90

Rudi, , Awaludin, 85



CSV Header

- Pada beberapa kasus, kadang pembuat CSV menyisipkan baris pertama sebagai Header, untuk mempermudah yang membukanya mengerti isi dari file CSV



Contoh File CSV dengan Header

Nama Depan, Nama Tengah, Nama Belakang, Nilai Ujian
Eko, Kurniawan, Khannedy, 100
Budi, , Nugraha, 90
Joko, Morro, Anwar, 90
Rudi, , Awaludin, 85

Pengenalan Apache Commons CSV



Pengenalan Apache Commons CSV

- Java tidak memiliki fitur bawaannya untuk mengolah format file CSV, oleh karena itu kita perlu library untuk mengolah data file CSV
- Apache Commons CSV adalah salah satu library yang populer untuk mengolah data file dengan format CSV
- Apache Commons CSV adalah library yang opensource, sehingga kita bisa gunakan secara gratis
- <https://commons.apache.org/proper/commons-csv/>

Membuat Project



Membuat Project

- <https://start.spring.io/>



Menambah Dependency

```
</dependency>  
<dependency>  
    <groupId>org.apache.commons</groupId>  
    <artifactId>commons-csv</artifactId>  
    <version>1.10.0</version>  
</dependency>  
<dependency>
```

Membuat CSV



Membuat CSV

- Untuk membuat CSV, kita bisa menggunakan class CSVPrinter yang terdapat di library Commons CSV
- Saat membuat CSVPrinter, kita perlu tentukan output tujuan dari CSV
- Di class CSVPrinter, terdapat method printRecord() yang bisa kita gunakan untuk menambah data ke CSV
- <https://commons.apache.org/proper/commons-csv/apidocs/org/apache/commons/csv/CSVPrinter.html>



Kode : Membuat CSV

```
@Test
void createCSV() throws IOException {
    StringWriter writer = new StringWriter();

    CSVPrinter printer = new CSVPrinter(writer, CSVFormat.DEFAULT);
    printer.printRecord("Eko", "Khannedy", 100);
    printer.printRecord("Budi", "Nugraha", 95);
    printer.flush();

    System.out.println(writer.getBuffer().toString());
}
```

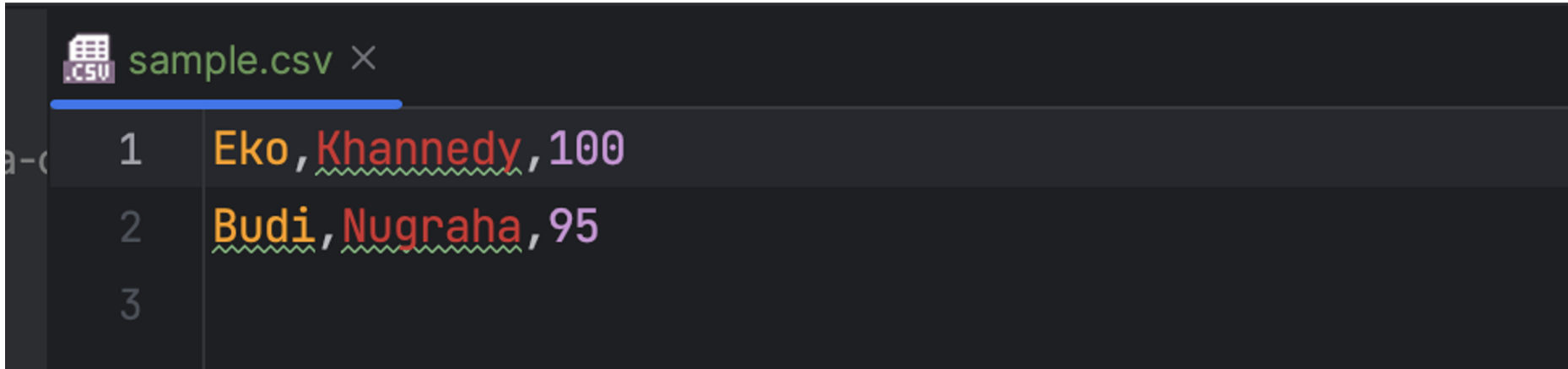
Membaca CSV



Membaca CSV

- Untuk membaca CSV, kita bisa menggunakan class CSVParser
- CSVParser adalah turunan dari Iterable, sehingga otomatis kita bisa melakukan iterasi datanya menggunakan perulangan foreach
- <https://commons.apache.org/proper/commons-csv/apidocs/org/apache/commons/csv/CSVParser.html>
- Tiap perulangan, kita bisa ambil datanya dalam bentuk CSVRecord

Kode : File CSV



The image shows a code editor window with a dark theme. The title bar at the top reads 'sample.csv' with a small icon of a document and a close button. The editor contains two lines of CSV data. The first line is '1,Eko,Khannedy,100' and the second line is '2,Budi,Nugraha,95'. The text is color-coded: '1' and '2' are grey, 'Eko' and 'Budi' are yellow, 'Khannedy' and 'Nugraha' are red, and '100' and '95' are purple. Wavy green lines are under the names 'Khannedy' and 'Nugraha'. A blue horizontal bar is positioned above the first line of data.

1	Eko	Khannedy	100
2	Budi	Nugraha	95



Kode : Membaca CSV

```
@Test
void readCSV() throws IOException {
    Path path = Path.of("sample.csv");
    Reader reader = Files.newBufferedReader(path);

    CSVParser parser = new CSVParser(reader, CSVFormat.DEFAULT);
    for (CSVRecord record : parser) {
        System.out.println("First Name : " + record.get(0));
        System.out.println("Last Name : " + record.get(1));
        System.out.println("Value : " + record.get(2));
    }
}
```

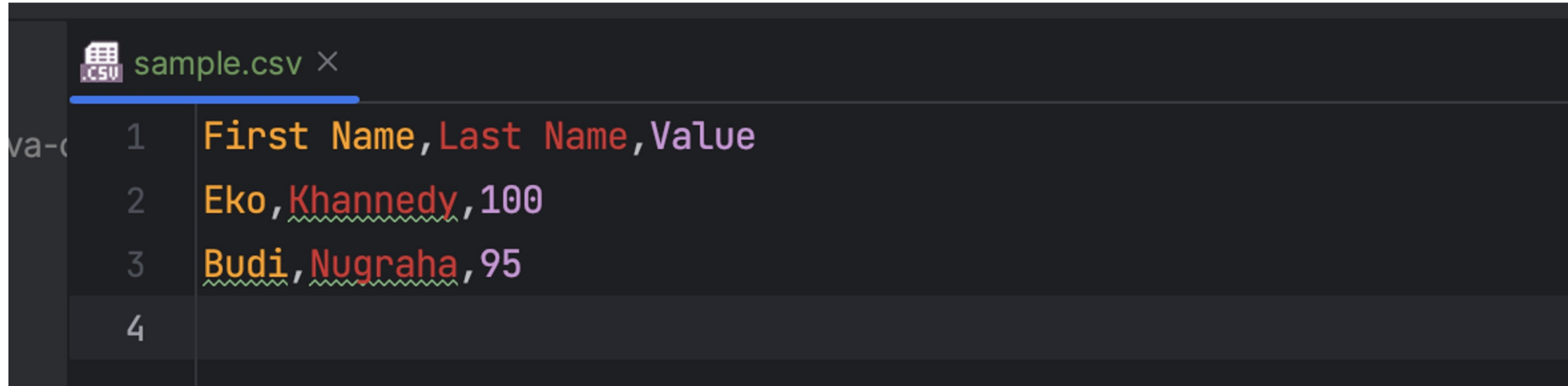
CSV Header



CSV Header

- Seperti yang sempat dibahas di awal, kadang-kadang, saat kita membuat file CSV, biasanya kita menambahkan baris pertama sebagai Header
- Saat menggunakan Commons CSV, kita harus memberi tahu CSVFormat jika baris pertama adalah kolom, jadi kita bisa menggunakan method `setHeader()` untuk memberitahu bahwa baris pertama adalah header
- Keuntungan menggunakan header, kita bisa menggunakan nama header untuk mendapatkan nilai dari tiap kolom di CSV, jadi tidak perlu menggunakan nomor index lagi

Kode : File CSV



```
sample.csv x
1 First Name,Last Name,Value
2 Eko,Khannedy,100
3 Budi,Nugraha,95
4
```

The image shows a code editor window with a tab labeled 'sample.csv'. The editor contains a CSV file with three rows of data. The first row is the header: 'First Name,Last Name,Value'. The second row is 'Eko,Khannedy,100' and the third row is 'Budi,Nugraha,95'. The text is color-coded: 'First Name' is orange, 'Last Name' is red, 'Value' is purple, 'Eko' is orange, 'Khannedy' is red, '100' is purple, 'Budi' is orange, 'Nugraha' is red, and '95' is purple. The names 'Khannedy' and 'Nugraha' are underlined with a green wavy line. The editor has a dark background and a light blue line for the first row.

First Name	Last Name	Value
Eko	Khannedy	100
Budi	Nugraha	95



Kode : Membaca CSV dengan Header

```
@Test
void readCSVWithHeader() throws IOException {
    Path path = Path.of("sample.csv");
    Reader reader = Files.newBufferedReader(path);

    CSVFormat format = CSVFormat.DEFAULT.builder().setHeader().build();
    CSVParser parser = new CSVParser(reader, format);
    for (CSVRecord record : parser) {
        System.out.println("First Name : " + record.get("First Name"));
        System.out.println("Last Name : " + record.get("Last Name"));
        System.out.println("Value : " + record.get("Value"));
    }
}
```



Menulis CSV Header

- CSVFormat juga bisa digunakan untuk menulis CSV dengan Header
- Kita cukup sebutkan saja nama-nama header dengan method `setHeader()`



Kode : Menulis CSV dengan Header

```
@Test
void createCSVWithHeader() throws IOException {
    StringWriter writer = new StringWriter();

    CSVFormat format = CSVFormat.DEFAULT.builder()
        .setHeader("First Name", "Last Name", "Value").build();
    CSVPrinter printer = new CSVPrinter(writer, format);
    printer.printRecord("Eko", "Khannedy", 100);
    printer.printRecord("Budi", "Nugraha", 95);
    printer.flush();

    System.out.println(writer.getBuffer().toString());
}
```

CSV Format



CSV Format

- Secara default, format file CSV itu dipisahkan menggunakan , (koma)
- Tapi kadang ada beberapa format CSV lain yang mungkin menggunakan pemisah dengan karakter tab, titik koma, atau yang lainnya
- Bahkan beberapa Spreadsheet editor, memiliki aturan tertentu untuk membuat CSV file
- Untungnya, Commons CSV mendukung beberapa format Spreadsheet editor
- Kita bisa lihat di class CSVFormat, terdapat banyak sekali format yang didukung, selain DEFAULT
- <https://commons.apache.org/proper/commons-csv/apidocs/index.html>



Kode : CSV Format TDF

```
@Test
void createCSVWithFormat() throws IOException {
    StringWriter writer = new StringWriter();

    CSVFormat format = CSVFormat.TDF.builder()
        .setHeader("First Name", "Last Name", "Value").build();
    CSVPrinter printer = new CSVPrinter(writer, format);
    printer.printRecord("Eko", "Khannedy", 100);
    printer.printRecord("Budi", "Nugraha", 95);
    printer.flush();

    System.out.println(writer.getBuffer().toString());
}
```