

Rangkuman Materi

Pada modul ini kita telah mempelajari konsep asynchronous process pada Dart. Prinsip umumnya adalah beberapa kode atau proses yang bisa berjalan bersamaan. Pada modul ini juga kita mengenal Future dan implementasinya untuk menangani proses asynchronous.

Beberapa hal yang telah kita pelajari, antara lain:

- Dalam *synchronous program*, kode program akan berjalan secara berurutan dari atas ke bawah.
- Dalam *asynchronous program* memungkinkan suatu operasi bisa berjalan sembari menunggu operasi lainnya selesai.
- Future adalah sebuah objek yang mewakili nilai potensial atau kesalahan yang akan tersedia pada waktu mendatang.
- Terdapat tiga kondisi yang memungkinkan saat menerapkan objek Future, antara lain:
 - Uncompleted, ketika operasi *asynchronous* tertunda pada durasi waktu tertentu sehingga operasi tersebut mengembalikan nilai Future.
 - Completed with data, ketika operasi *asynchronous* berhasil berjalan, objek Future akan mengembalikan nilai dengan tipe data tertentu.
 - Completed with error, ketika operasi *asynchronous* berjalan dan mengalami kegagalan, objek Future akan mengembalikan nilai eror.
- Metode `delayed` pada objek Future berfungsi untuk menunda jalannya operasi kode bergantung dengan durasi yang ditentukan.

```
1. var coffee = Future.delayed(Duration(seconds: 3), () {  
2.     return 'Coffee Boba';  
3. });
```

- Objek Future memiliki beberapa fungsi/method untuk menangani state, antara lain:
 - `then()`, fungsi yang menangani kondisi *completed with data*.
 - `catchError()`, fungsi yang menangani kondisi *completed with error*.
 - `whenComplete()`, fungsi yang berjalan saat proses *asynchronous* dan state lainnya berakhir.
- **Keyword** `async` dan `await` mendukung proses *asynchronous program*, sehingga memungkinkan untuk menulis kode *asynchronous* yang terlihat mirip layaknya kode *synchronous*.