

Nama : Izhar Mayranda  
Program : Intro To NodeJs

## Summary Sesi 2

- Class

Class adalah cetakan untuk membuat objek yang berisi method ataupun property. Walaupun Javascript bukan bahasa program yang berbasis OOP, tetapi Javascript tetap bisa menerapkan fitur OOP.

Contoh class : `class identitas {  
 constructor(nama) {  
 this.name = nama } }`

- Event Emitter

Event Emitter adalah salah satu class dari package events, event emitter bisa membantu kita membuat program dengan pola event driven (alur suatu kejadian).

Setiap instance dari Event Emitter mempunyai fungsi on dan emit.

- Fungsi on menerima dua parameter yakni nama event dan fungsi listener
- Fungsi emit menerima banyak parameter, parameter pertama adalah nama event yang akan dipanggil, parameter seterusnya adalah fungsi/parameter fungsi listener.

- Buffer

Buffer adalah tempat untuk menyimpan nilai sementara yang digunakan oleh stream hingga data tersebut digunakan.

Contoh membuat buffer:

`const tampung = Buffer.alloc(15)`

fungsi alloc menerima parameter number yang menjadikannya seberapa banyak data diterima dalam bit.

- Stream

Stream adalah teknik untuk menampilkan secara file dengan cara mencicil file tersebut.

Perbedaan Stream dengan readFile() adalah, readFile() membuka file secara keseluruhan baru ditampilkan, tetapi hal ini tidak efektif untuk file yang besar karena membutuhkan waktu yang lama, sedangkan dengan teknik stream kita bisa membuka file yang besar tanpa harus membuka semuanya terlebih dahulu.

Contoh membaca file dengan stream (anggap input.txt dan output.txt terletak di folder yang sama):

```
const bacaStream = fs.createReadStream('./input.txt', utf-8)
bacaStream.on('data', (data) => {
  console.log(data)
})
```

Contoh mengubah file dengan stream dalam hal ini piping stream:

```
let tulisStream = createWriteStream('./output.txt', utf-8)
bacaStream.pipe(tulisStream)
```

Maka isi file input akan berpindah ke file output.

## - File System

File System di javascript membutuhkan modul fs.

### ➤ Membaca File dan menulis file

Membaca file bisa menggunakan function `fs.readFile()`, sedangkan menulis file bisa menggunakan `fs.writeFile()` kedua fungsi tersebut menerima dua parameter yakni nama file, tipe encoding file, dan fungsi.

### ➤ Membuka File

Kita bisa membuka file dengan method `fs.open()`, lalu kita bisa membaca, menulis, mengisi file tersebut sesuai flag yang diberikan. Method ini menerima tiga parameter, yakni namafile, flag, dan fungsi.

### ➤ Membuat file

Membuat file bisa menggunakan method `fs.append(namaFile,isiFile,fungsi)`

### ➤ Menghapus File

`fs.unlink()` adalah method yang digunakan untuk menghapus file, method tersebut menerima nama file dan fungsi untuk parameteranya.

## - Promise

Promise adalah suatu objek yang digunakan untuk membuat program berjalan asynchronous. Parameter dari promise adalah fungsi callback dimana parameteranya adalah fungsi `resolve()` ketika promise terpenuhi dan fungsi `reject()` ketika promise gagal terpenuhi.

Ada juga fungsi `.then()` yang akan melakukan deretan kode ketika resolve sukses dijalankan, dan `.catch()` ketika `reject()` sukses dijalankan.

Contoh:

```
const coba = () => { return new Promise((resolve, reject) => {
    if (true) { resolve("Janji terpenuhi")}
    else {reject("Ingkar janji")}})}
const handlerSukses = (sukses) => { console.log(sukses)}
const handlerGagal = (gagal) => { console.log(gagal)}
coba().then(handlerSukses).catch(handlerGagal)
maka output dari kode diatas adalah 'Janji terpenuhi'
```

## - Argv

Argv adalah fungsi yang menerima nilai dari command prompt

Contoh: `console.log(Number(process.argv[2])+Number(process.argv[3]))`

Dengan menambahkan argumen tambahan di terminal misalnya `node`

`c:\H8\Sesi2\promise.js" 1 2`, maka outputnya adalah 3