Paradigma Pemrograman di JavaScript: Functional Programming

Andvantages of Functional programming

- Its pure function, provides confidence of not changing things outside of its scope.
- Its reduces the complexity, need not to worry about how it is doing it, focus will be only on what it is doing.
- Ease of testing, because it does not depend on state of the application and result verification also will be easy.
- It makes the code more readable.
- Functional programming makes code easier to understand.

Functional programming

- Functional Programming treats computation as the evaluation of mathematical functions.
- “a subset of functional programming which treats all functions as deterministic mathematical functions, or pure functions”
- Functional Programming avoids changing-state and mutable data
- Pure functional language: Ex : Common Lisp, Scheme, Clojure, Wolfram Language, Racket, Erlang, OCaml Haskell, F#
- other programming languages support programming in a functional: C++11, Kotlin, Perl, PHP, Python and Scala,Javasript

Functional programming di JavaScript.

```
/// Traditional Imperative Loop:
const numList = [1, 2, 3, 4, 5, 6, 7, 8, 9,
10]
let result = 0;
for (let i = 0; i < numList.length; i++) {
  if (numList[i] % 2 === 0) {
    result += (numList[i] * 10)
  }
}
```

```
//Functional Programming with
higher-order functions:
const result = [1, 2, 3, 4, 5, 6, 7, 8, 9,
10]
        .filter(n => n % 2 === 0)
        .map(a => a * 10)
        .reduce((a, b) => a + b)
```

Javascript Support Function

- Array methods which help to achieve functional programming : find, map, reduce, every, some
- Libraries to support FP: RamdaJS, UnderscoreJS, lodash
- Side Effects:
- Immutability: