

## Jurnal Modul 14

### Penjelasan refactoring jurnal

#### File 1: Observer.java

##### Before:

Hanya interface tanpa komentar.

##### After (Refactor):

- Menambahkan JavaDoc comment di atas interface, menjelaskan fungsinya.
- Tetap mempertahankan method `update()` karena sudah sesuai.

#### File 2: Subject.java

##### Before:

Interface tanpa komentar.

##### After (Refactor):

- Menambahkan **JavaDoc comment** untuk menjelaskan fungsi interface Subject.
- Parameter di method `attach()` dan `detach()` disesuaikan penamaannya menjadi `observer` (dari `o` → lebih deskriptif).

#### File 3: ConcreteSubject.java

##### Before:

- Tidak ada komentar.
- Penamaan variabel `observers`.
- Tidak ada JavaDoc di class dan method.

##### After (Refactor):

- Ganti nama `observers` jadi `observerList` → lebih deskriptif.
- Tambahkan JavaDoc comment di class dan method.
- Tambah inline comment untuk menjelaskan variabel.
- Rapihan whitespace antar method.

#### File 4: ConcreteObserver.java

##### Before:

- Tidak ada komentar.
- Variabel name terlalu umum.

##### After (Refactor):

- Ganti nama name → observerName → lebih jelas.
- Tambahkan JavaDoc comment di class, constructor, dan variabel.
- Rapikan whitespace.

#### File 5: ProgramObserver.java (Main Program)

##### Before:

- Tidak ada komentar.
- Penamaan observer kurang jelas.
- Tidak ada penjelasan tiap blok kode.





##### After (Refactor):

- Tambahkan JavaDoc di class main.
- Tambah comment penjelasan tiap baris penting.
- Rapikan whitespace antar blok logis.
- Ganti nama variabel observer dari observer1/observer2 ke observerA/observerB → lebih readable.

##### Hasil setelah di refactor:

```
Observer A menerima pesan: Halo Observer!  
Observer B menerima pesan: Halo Observer!  
Observer A menerima pesan: Pesan kedua  
PS C:\DEVELOPMENT\Java Script>
```

Checklist Refactoring: 

Faktor	Status
A. Naming Convention	 Sudah sesuai camelCase untuk variable/property, PascalCase untuk class/interface
B. White Space & Indentation	 Rapi 4 spasi per indent, jarak antar blok kode logis
C. Variable/Attribute Declarations	 Semua deklarasi jelas, di awal blok method/constructor
D. Comments	 Sudah ditambahkan penjelasan class, method, dan baris penting

## Tugas Pendahuluan Modul 14

File: matematikaLibraries.js

Perubahan yang dilakukan:

### 1. Naming Convention:

- Semua nama function dibuat lebih deskriptif dan mengikuti **camelCase**, sesuai standar JavaScript.
  - FPB → findFPB
  - KPK → findKPK
  - Turunan → deriveEquation
  - Integral → integrateEquation
- Variable di dalam function juga menggunakan **camelCase**.

### 2. White space & Indentation:

- Menggunakan **2 spasi** konsisten untuk indentasi.
- Memberi jarak antar function agar lebih mudah dibaca.

### 3. Variable/Attribute Declarations:

- Mengganti let menjadi const jika nilai tidak berubah setelah deklarasi, sesuai best practice JavaScript.
- Contoh: let temp → const temp

### 4. Comments:

- Menambahkan **documentation comments** (**/\*\* ... \*/**) pada setiap method.
- Memberi keterangan jelas tentang parameter dan return value.

File: main.js

Perubahan yang dilakukan:

### 1. Naming Convention:

- Tidak perlu banyak diubah, karena variable dan function sudah dipanggil dari library refactored.

## 2. White space & Indentation:

- Konsisten **2 spasi** untuk indentasi.
- Setiap blok console.log diberi **jarak antar statement** agar lebih rapi.

## 3. Comments:

- Menambahkan komentar di setiap bagian pemanggilan function untuk menjelaskan tujuannya.

File: persamaan.json

Perubahan yang dilakukan:

- Tidak perlu refactor, karena format JSON sudah sesuai:
  - Menggunakan **2 spasi indentasi**
  - Key menggunakan **lowerCamelCase**
  - Nilai berupa array numerik

Hasil setelah di refactor:

```
PS C:\DEVELOPMENT\Java Script\KPL_Tsaqif Hisyam Saputra_
FPB(60, 45) = 15
KPK(12, 8) = 24
Turunan dari [1, 4, -12, 9] = 3x3 + 8x2 - 12x
Integral dari [4, 6, -12, 9] = x4 + 2x3 - 6x2 + 9x1 + C
```

Kesimpulan Refactoring

Faktor	Refactored?	Keterangan
A. Naming Convention	✓	Function, variable, dan property pakai camelCase & deskriptif
B. White space & Indentation	✓	Konsisten 2 spasi, jarak antar blok rapi
C. Variable/Attribute Declarations	✓	Gunakan <code>const</code> jika nilai tetap, <code>let</code> bila perlu
D. Comments	✓	Documentation comment di tiap function & inline comment di main