

JURNAL MODUL 14

```
1  const readline = require("readline");
2
3  // Membuat interface untuk input dari user via terminal
4  const rl = readline.createInterface({
5    input: process.stdin,
6    output: process.stdout
7  });
8
9  // Membuat array dari 0 sampai 49
10 const array = Array.from({ length: 50 }, (_, index) => index);
11
12 // Melakukan iterasi terhadap array
13 array.forEach((num) => {
14   let output = `${num}`;
15
16   // Cek apakah habis dibagi 2 dan 3
17   if (num % 2 === 0 && num % 3 === 0) {
18     output += " ##$";
19   } else if (num % 2 === 0) {
20     output += " ##";
21   } else if (num % 3 === 0) {
22     output += " $$";
23   }
24
25   // Menampilkan hasil ke terminal
26   console.log(output);
27 });
28
29 // Menerima input angka dari user
30 rl.question("Masukkan angka (1-10000): ", (nilaiString) => {
31   const nilaiInt = parseInt(nilaiString);
32
33   // Validasi input user
34   if (!isNaN(nilaiInt) && nilaiInt >= 1 && nilaiInt <= 10000) {
35     console.log(`Anda memasukkan angka: ${nilaiInt}`);
36   } else {
37     console.log("Masukkan angka yang valid antara 1 hingga 10000!");
38   }
39
40   // Menutup interface readline
41   rl.close();
42 });
43
```

Kode yang ditulis telah mematuhi konvensi penamaan (naming convention) yang baik.

Penggunaan variabel seperti `rl`, `array`, `num`, `nilaiString`, dan `nilaiInt` sudah mengikuti gaya `camelCase` yang lazim digunakan dalam JavaScript. Selain itu, nama-nama variabel tersebut

cukup representatif dan menggambarkan fungsinya, sehingga memudahkan pembacaan dan pemahaman hanya dari nama variabelnya.

Untuk method atau fungsi bawaan seperti `readline.createInterface`, `Array.from`, `array.forEach`, `rl.question`, `parseInt`, `isNaN`, `console.log`, dan `rl.close()`, semuanya ditulis sesuai standar penulisan yang berlaku di JavaScript dan menggunakan sintaks resmi dari pustaka yang bersangkutan.

Dari aspek tata letak, kode telah menggunakan indentasi sebanyak 4 spasi secara konsisten, yang membuat struktur kode lebih mudah dibaca dan dikelola, baik oleh programmer lain maupun oleh penulis kode itu sendiri di kemudian hari.

Deklarasi variabel seluruhnya menggunakan `const`, yang merupakan praktik terbaik dalam JavaScript modern karena mencegah nilai variabel berubah secara tidak sengaja. Tidak ditemukan penggunaan `var`, yang memang sudah tidak direkomendasikan dalam pengembangan JavaScript masa kini. Selain itu, kode juga telah dilengkapi dengan komentar, yang membantu memberikan konteks tambahan dan meningkatkan keterbacaan program.

Hasil Running :

```
29
30 #$$$
31
32 ##
33 $$
34 ##
35
36 #$$$
37
38 ##
39 $$
40 ##
41
42 #$$$
43
44 ##
45 $$
46 ##
47
48 #$$$
49
Masukkan angka (1-10000): 12
Anda memasukkan angka: 12
```

