

Sheila Nuur Ditrir
Dokumentasi

Soal-1

Query:

```
SELECT
    b.KODEBRG,
    b.NAMABRG,
    COALESCE(SUM(j.JUMLAH), 0) AS JML_JUAL
FROM
    BARANG b
LEFT JOIN
    JUAL j ON b.KODEBRG = j.KODEBRG
GROUP BY
    b.KODEBRG,
    b.NAMABRG;
```

Penjelasan query:

- **LEFT JOIN** digunakan untuk memastikan bahwa semua barang dari tabel **BARANG** ditampilkan, meskipun tidak ada penjualan yang tercatat di tabel **JUAL**.
- **COALESCE(SUM(j.JUMLAH), 0)** digunakan untuk menggantikan **NULL** dengan **0** jika tidak ada penjualan untuk barang tersebut.
- **GROUP BY b.KODEBRG, b.NAMABRG** digunakan untuk mengelompokkan hasil berdasarkan kode barang dan nama barang.

Soal-2

Query:

```
SELECT
    KodeBarang,
    StokAkhir
FROM
    HistoriStokBarang
WHERE Tanggal = (
    SELECT
        MAX(Tanggal)
    FROM
        HistoriStokBarang AS HSB
    WHERE
        HSB.KodeBarang = HistoriStokBarang.KodeBarang)
ORDER BY
    KodeBarang;
```

Penjelasan query:

- **Subquery (MAX(Tanggal)):** Mengambil tanggal terakhir untuk setiap barang (**KodeBarang**) dari tabel **HistoriStokBarang**.
- **WHERE Clause:** Memastikan bahwa hanya data dengan tanggal terbaru untuk masing-masing barang yang diambil.
- **ORDER BY KodeBarang:** Mengurutkan hasil berdasarkan **KodeBarang** untuk kejelasan.

Soal-3

Code:

```
function generateFibonacci(n) {  
    // Inisialisasi array hasil dan variabel a, b  
    let result = [];  
    let a = 1, b = 1;  
  
    // Jika n lebih dari 0, masukkan angka pertama  
    if (n > 0) {  
        result.push(a);  
    }  
  
    // Jika n lebih dari 1, masukkan angka kedua  
    if (n > 1) {  
        result.push(b);  
    }  
  
    // Menghasilkan deret Fibonacci sampai n angka  
    for (let i = 2; i < n; i++) {  
        let next = a + b; // Hitung angka berikutnya  
        result.push(next); // Tambahkan ke array hasil  
        a = b; // Update a menjadi b  
        b = next; // Update b menjadi angka berikutnya  
    }  
  
    // Tampilkan hasil  
    console.log(result.join(' '));  
}  
  
generateFibonacci(3); // Output: 1 1 2  
generateFibonacci(5); // Output: 1 1 2 3 5  
generateFibonacci(12); // Output: 1 1 2 3 5 8 13 21 34 55 89 144
```

```
generateFibonacci(25); // Output: 1 1 2 3 5 8 13 21 34 55 89 144 233 377  
610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025
```

Pseudocode:

1. Input:

n (jumlah angka yang akan dihasilkan)

2. Inisialisasi:

- Buat array result kosong
- Set a dan b sebagai 1 (dua angka pertama dalam deret Fibonacci)

3. Iterasi dari 1 hingga n:

- Tambahkan a ke dalam array result
- Hitung angka berikutnya dengan menjumlahkan a dan b
- Update a menjadi b dan b menjadi angka berikutnya

4. Output:

Tampilkan isi array result sebagai deret Fibonacci

Soal-4

Code:

```
function printStars(n) {  
    // Loop untuk setiap baris  
    for (let i = 1; i <= n; i++) {  
        // Hitung jumlah bintang pada baris ke-i  
        let stars = '';  
        for (let j = 1; j <= 2 * i - 1; j++) {  
            stars += '*';  
        }  
        // Cetak baris bintang  
        console.log(stars);  
    }  
}  
  
printStars(4); // Output untuk input 4  
printStars(3); // Output untuk input 3
```

Pseudocode:

1. Input:

n (jumlah baris yang akan dihasilkan)

2. Inisialisasi:

Buat variabel stars sebagai string kosong

Set i sebagai 1 (untuk baris saat ini)

3. Iterasi dari 1 hingga n:

- Buat string line kosong
- Hitung jumlah bintang pada baris ke-i sebagai $2 * i - 1$
- Iterasi dari 1 hingga $(2 * i - 1)$:
 - Tambahkan bintang ke dalam string line
- Tambahkan string line ke dalam variabel stars
- Tambahkan baris baru ke dalam stars
- Update i untuk iterasi berikutnya

4. Output:

Tampilkan isi variabel stars sebagai pola bintang

soal-5

Code:

```
function formatNumber(input1, input2) {  
    // Ubah input1 menjadi string  
    let numberString = input1.toString();  
  
    // Hitung panjang string dari input1  
    let length = numberString.length;  
  
    // Hitung jumlah nol yang perlu ditambahkan di depan  
    let zerosToAdd = input2 - length;  
  
    // Jika jumlah nol yang perlu ditambahkan lebih besar dari 0  
    if (zerosToAdd > 0) {  
        // Tambahkan nol di depan input1 hingga mencapai panjang yang  
        // diinginkan  
        numberString = '0'.repeat(zerosToAdd) + numberString;  
    }  
  
    // Kembalikan hasil yang sudah diformat  
    return numberString;  
}
```

```
}  
  
console.log(formatNumber(5, 4)); // Output: 0005  
console.log(formatNumber(87, 7)); // Output: 0000087
```

Pseudocode:

1. Input:

- Input1 (angka yang akan diformat)
- Input2 (jumlah total digit yang diinginkan)

2. Inisialisasi:

- Set formattedNumber sebagai string kosong

3. Hitung panjang angka Input1:

- length = Panjang string dari Input1

4. Hitung jumlah nol yang perlu ditambahkan di depan:

- zerosToAdd = Input2 - length

5. Jika zerosToAdd lebih besar dari 0:

- Tambahkan zerosToAdd angka nol di depan Input1
- Set formattedNumber menjadi string dengan zerosToAdd nol di depan Input1
- Format angka Input1 dengan nol di depan hingga panjang total sesuai dengan Input2

6. Jika zerosToAdd kurang dari atau sama dengan 0:

- Set formattedNumber menjadi Input1 (karena panjangnya sudah cukup)

7. Output:

- Tampilkan formattedNumber