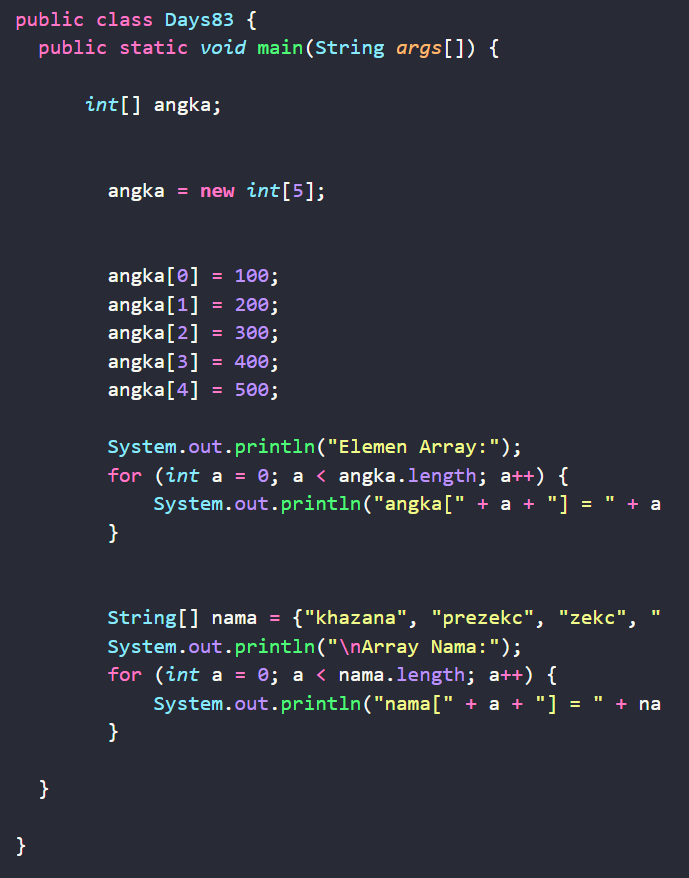
**Rangkuman materi ISC Days 83-99**

1. **Membuat Objek Array satu dimensi**

**Penjelasan:**

1. **Deklarasi Array:**
   * int[] angka;: Baris ini mendeklarasikan sebuah variabel bernama angka yang akan menampung sebuah array bertipe integer. Namun, pada tahap ini, array belum dialokasikan memori.
2. **Inisialisasi Array:**
   * angka = new int[5];: Baris ini mengalokasikan memori untuk array angka dengan ukuran 5 elemen. Artinya, array ini dapat menyimpan 5 bilangan bulat.
3. **Memberikan Nilai ke Elemen Array:**
   * angka[0] = 100;, angka[1] = 200;, dan seterusnya: Baris-baris ini memberikan nilai pada masing-masing elemen array. Indeks dimulai dari 0, jadi angka[0] mengacu pada elemen pertama, angka[1] pada elemen kedua, dan seterusnya.
4. **Mencetak Elemen Array:**
   * Menggunakan perulangan for, kode ini mencetak setiap elemen array angka beserta indeksnya.
5. **Array String:**
   * String[] nama = {"khazana", "prezekc", "zekc", "maco"};: Di sini, kita membuat array bertipe String bernama nama dan langsung menginisialisasinya dengan beberapa nama.
6. **Mencetak Elemen Array String:**
   * Sama seperti sebelumnya, kode ini mencetak setiap elemen array nama beserta indeksnya.

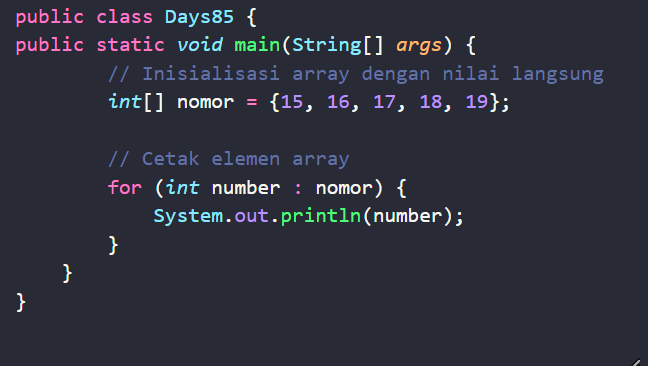
**Kesimpulan:**

Kode di atas menunjukkan cara membuat dan menginisialisasi array satu dimensi dalam Java, baik untuk tipe data numerik (integer) maupun non-numerik (String). Konsep utama yang digunakan adalah:

* Deklarasi: Menentukan tipe data dan nama variabel array.
* Inisialisasi: Memblokir memori untuk menyimpan elemen-elemen array.
* Akses Elemen: Menggunakan indeks untuk mengakses elemen tertentu dalam array.

Konsep Penting Lainnya:

* Panjang Array: angka.length akan memberikan jumlah total elemen dalam array angka.
* Tipe Data: Elemen dalam sebuah array harus memiliki tipe data yang sama.

**85.Mengisi elemen ke Array satu dimensi pada saat di inisialisasi**

**Penjelasan:**

1. **Membuat dan menginisialisasi sebuah array:**
   * int[] nomor = {15, 16, 17, 18, 19};
     + Baris ini membuat sebuah array dengan nama nomor yang berisi bilangan bulat.
     + Array ini langsung diisi dengan nilai-nilai 15, 16, 17, 18, dan 19.
2. **Mencetak semua elemen dalam array:**
   * for (int number : nomor) { ... }
     + Ini adalah bentuk perulangan for-each yang digunakan untuk mengiterasi setiap elemen dalam array nomor.
     + Pada setiap iterasi, nilai elemen saat ini disimpan dalam variabel number.
     + System.out.println(number); mencetak nilai number ke konsol, sehingga kita bisa melihat semua elemen dalam array.

**Penjelasan lebih detail:**

* Array: Adalah kumpulan data dengan tipe yang sama, dalam hal ini adalah bilangan bulat. Array nomor menyimpan lima bilangan bulat.
* Inisialisasi langsung: Kita bisa langsung memberikan nilai awal pada saat membuat array, seperti yang dilakukan pada baris pertama.
* Perulangan for-each: Ini adalah cara yang lebih singkat untuk mengiterasi semua elemen dalam sebuah koleksi (seperti array). Kita tidak perlu mengurus indeks seperti pada perulangan for biasa.
* Mencetak: System.out.println() digunakan untuk menampilkan output ke konsol.

**86. Mencetak elemen pada Array satu dimensi**

A computer code with colorful text

Description automatically generated**Penjelasan:**

1. **Deklarasi dan Inisialisasi Array:**

* int[] array = {10, 20, 30, 40, 50};
* Membuat sebuah array bertipe integer bernama array dan langsung mengisi nilai-nilai awal 10, 20, 30, 40, dan 50.

1. **Perulangan for:**

* for (int i = 0; i < array.length; i++) { ... }
* Ini adalah perulangan for yang digunakan untuk mengakses setiap elemen dalam array.
* i = 0: Inisialisasi variabel i dengan nilai 0, yang akan digunakan sebagai indeks untuk mengakses elemen array.
* i < array.length: Kondisi perulangan, yang akan terus berjalan selama i kurang dari jumlah elemen dalam array.
* i++: Setelah setiap iterasi, nilai i akan bertambah 1, sehingga pada iterasi berikutnya, elemen berikutnya akan diakses.

1. **Mencetak Elemen Array:**

* System.out.println("Elemen ke-" + i + ": " + array[i]);
* Mencetak pesan ke konsol yang menunjukkan indeks elemen saat ini (i) dan nilai elemen tersebut (array[i]).

A computer code with colorful text

Description automatically generated**87. Mengubah Nilai Array satu dimensi pada index Tertentu**

**Penjelasan:**

* + 1. **Deklarasi dan Inisialisasi Array:**
* int[] angka = {2, 25, 06, 20, 02};
  + Membuat sebuah array bertipe integer bernama angka dan langsung mengisi nilai-**nilai awal 2, 25, 6, 20, dan 2.**

1. **Mencetak Array Awal:**

* Menggunakan perulangan for, mencetak semua elemen array sebelum perubahan.

1. **Mengubah Nilai Elemen:**

* angka[0] = 55;
  + Mengakses elemen pertama dalam array (indeks 0) dan mengubah nilainya menjadi55.

1. **Mencetak Array Setelah Perubahan:**

* Menggunakan perulangan for lagi, mencetak semua elemen array setelah perubahan.

A screen shot of a computer program

Description automatically generated**88. soal**

1. **Array yang Ditetapkan: Array berisi angka-angka tetap {10, 20, 30, ..., 100}.**
2. **Input Pengguna: Program meminta pengguna memasukkan nilai untuk dicari.**
3. **Pencarian: Menggunakan loop for untuk memeriksa apakah nilai yang dimasukkan pengguna ada di dalam array.**
4. **Output: Jika ditemukan, akan mencetak "Nilai ditemukan!" dan jika tidak, mencetak "Nilai tidak ditemukan.".**

A screen shot of a computer program

Description automatically generated**89. Soal**

* **Loop Tanpa Batas**:

Menggunakan while (true) untuk meminta input pengguna terus-menerus.

* **Mendeteksi Kata "stop"**:

Jika pengguna mengetikkan "stop" (case-insensitive), program akan keluar dari loop menggunakan break.

* **Validasi Input**:

Menggunakan try-catch untuk menangani error ketika input bukan angka.

* **Pesan untuk Pengguna**:

Menambahkan petunjuk untuk pengguna agar tahu mereka dapat mengetik "stop" untuk keluar.

A computer screen shot of text

Description automatically generated90. soal

* **Input Pengguna**:

Program meminta pengguna untuk memasukkan bilangan genap positif.

Jika angka yang dimasukkan bukan genap atau bukan positif, program akan menampilkan pesan kesalahan.

* **Perhitungan Perkalian**:

Program menggunakan loop for untuk mengalikan angka dari 1 hingga n.

Langkah-langkah perkalian disusun dalam sebuah StringBuilder.

* **Output**:

Program mencetak langkah-langkah perkalian dan hasil akhirnya.

* A computer screen shot of text

  Description automatically generated91. soal
* **mport Scanner**:

Program menggunakan Scanner untuk membaca input dari pengguna.

* **Loop Tanpa Batas**:

while (true) digunakan agar program terus meminta input hingga kondisi tertentu terpenuhi.

* **Pengecekan Genap atau Ganjil**:
* **Bilangan genap**:

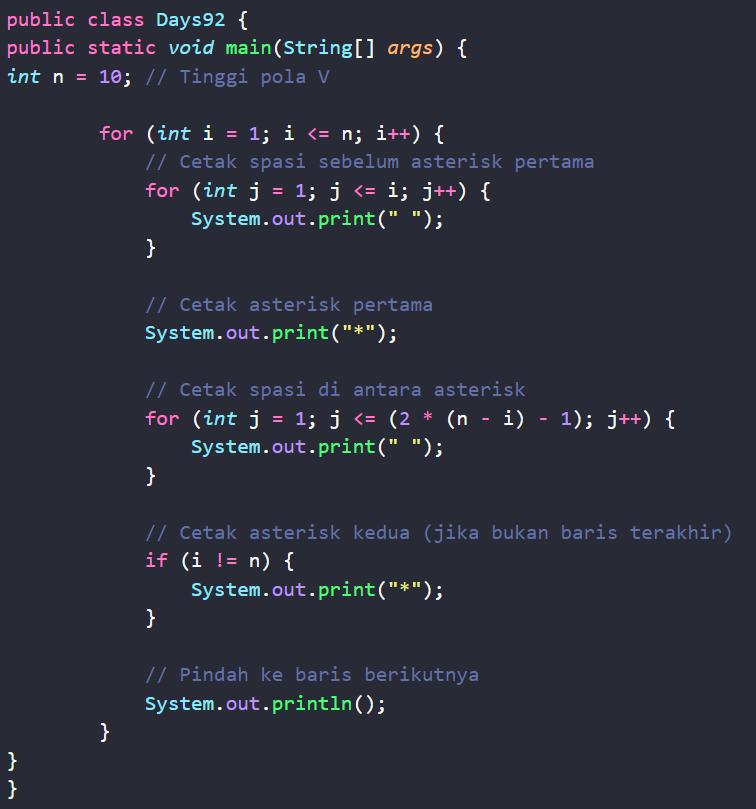
Diperiksa dengan angka % 2 == 0. Jika benar, program mencetak "Genap" dan meminta input lagi.

* **Bilangan ganjil**:

Jika angka tidak memenuhi kondisi genap, program mencetak "You and I, end" dan keluar dari loop menggunakan break.

* **Menutup Scanner**:

scanner.close() dipanggil setelah loop untuk membersihkan sumber daya.

92. Asterisk Pattern : Pola V

Untuk membuat pola berbentuk huruf **V** menggunakan karakter asterisk (\*) di Java, Anda dapat menggunakan loop for untuk mengatur posisi setiap karakter.

**Penjelasan Program:**

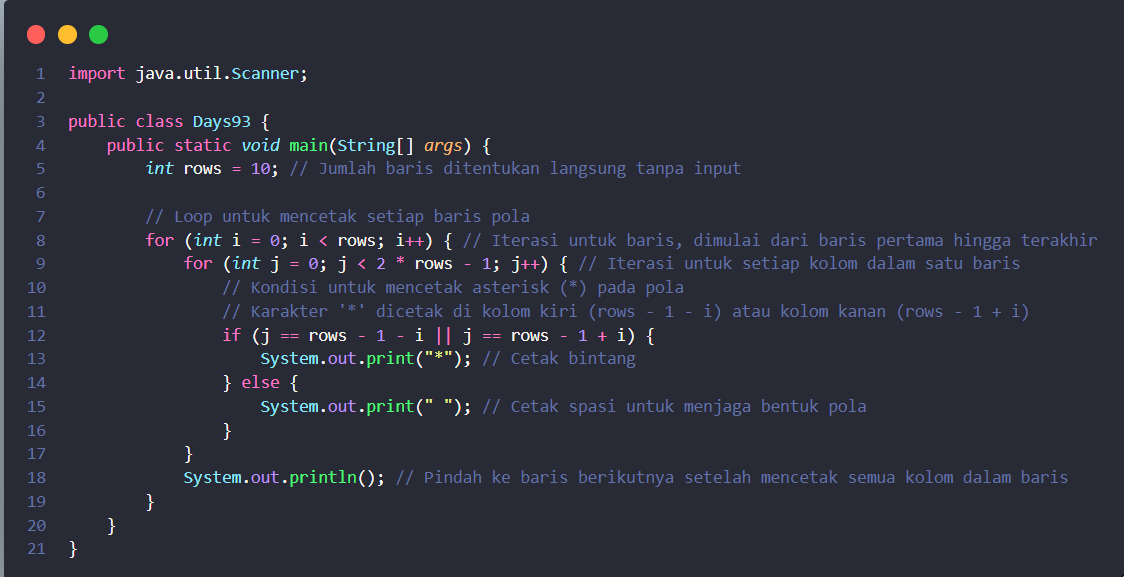
1. **Input Tinggi Pola (n):**  
   n menentukan jumlah baris pada pola V.  
   Misalnya, jika n = 5, pola akan memiliki 5 baris.
2. **Penggunaan Loop:**

**Baris Utama (for i = 1 to n):**  
Loop ini mengontrol setiap baris dalam pola V.

**Spasi Awal:**  
Spasi ditambahkan berdasarkan nomor baris i, sehingga kedua garis miring tampak menyempit ke bawah.

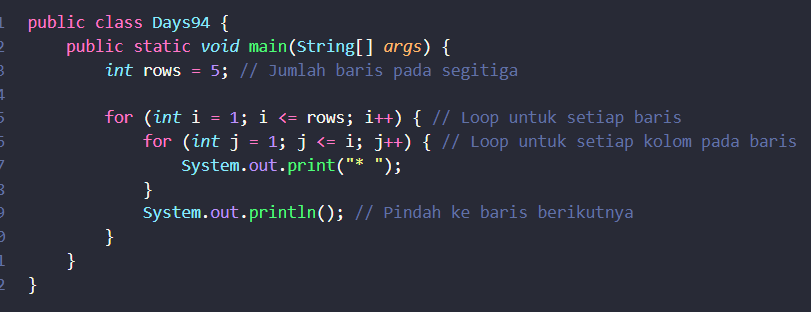
**Asterisk Pertama dan Kedua:**  
Karakter \* dicetak di awal dan akhir setiap baris (kecuali pada baris terakhir, hanya satu \*).

1. **Logika pada Spasi Tengah:**  
   Spasi tengah dihitung berdasarkan formula (2 \* (n - i) - 1) agar jarak di antara dua garis miring berkurang pada setiap baris.

93. pola V terbalik

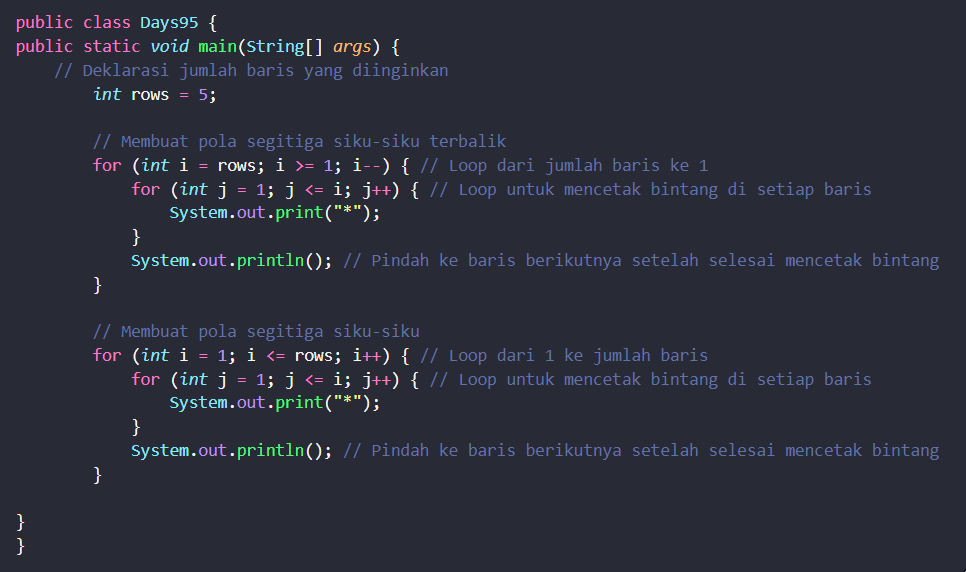
**Penjlesan :**

* + 1. **Variabel rows**:
* Menentukan jumlah baris pola.
* Nilai 5 diatur langsung tanpa menggunakan input dari pengguna.
  + 1. **Loop Baris (for i)**:
* Loop ini mengontrol jumlah baris pada pola.
* Dimulai dari baris pertama (i = 0) hingga baris terakhir (i < rows).
  + 1. **Loop Kolom (for j)**:
* Loop ini mencetak karakter di setiap kolom pada satu baris.
* Jumlah total kolom dalam pola dihitung sebagai 2 \* rows - 1 untuk memastikan pola simetris.
  + 1. **Logika Cetak Asterisk atau Spasi**:
* Kondisi if (j == rows - 1 - i || j == rows - 1 + i):
  + Cetak \* di posisi kiri pola (rows - 1 - i) dan kanan pola (rows - 1 + i).
* Jika tidak memenuhi kondisi, cetak spasi untuk menjaga bentuk pola.

94. Asterisk Pattern : Segitiga siku-siku

**Penjelasan**

* **Outer Loop (for i)**: Mengontrol jumlah baris pada segitiga.
* **Inner Loop (for j)**: Mencetak jumlah asterisk sesuai dengan nomor baris.
* **System.out.println()**: Digunakan untuk pindah ke baris baru setelah mencetak satu baris.

95. segitiga siku dan kebalikan nya

**Penjelasan Komentar:**

* **Deklarasi Variabel rows:** Variabel rows digunakan untuk menentukan jumlah baris pola yang akan dibuat. Anda bisa mengubah nilainya untuk mendapatkan pola yang lebih besar atau lebih kecil.
* **Segitiga Siku-Siku Terbalik:**

for (int i = rows; i >= 1; i--): Loop ini dimulai dari nilai rows (5 dalam hal ini) dan berakhir pada 1.

for (int j = 1; j <= i; j++): Loop dalam ini mencetak bintang sebanyak nilai i pada setiap iterasi.

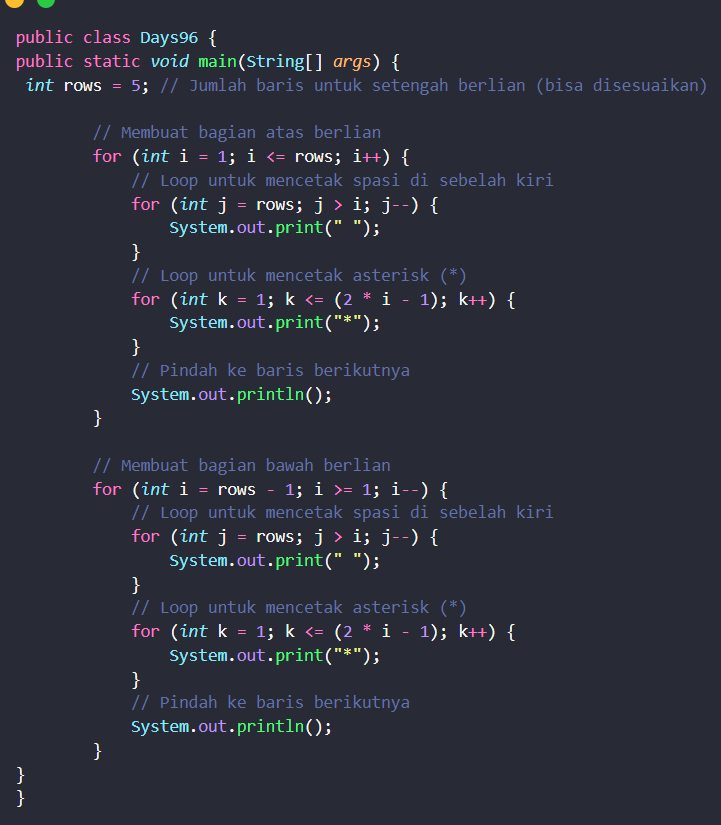
System.out.println();: Digunakan untuk memindahkan ke baris baru setelah mencetak bintang di satu baris.

* **Segitiga Siku-Siku:**

for (int i = 1; i <= rows; i++): Loop ini dimulai dari 1 dan berakhir pada nilai rows.

for (int j = 1; j <= i; j++): Loop dalam ini mencetak bintang sebanyak nilai i pada setiap iterasi.

System.out.println();: Digunakan untuk memindahkan ke baris baru setelah mencetak bintang di satu baris.

96. Asterisk Pattern : Diamond

**Penjelasan Kode:**

1. **Variabel rows:**

Menentukan jumlah baris untuk setengah berlian (5 dalam contoh ini).

1. **Bagian atas berlian:**

Loop i berjalan dari 1 hingga rows.

**Loop spasi:** j mencetak spasi untuk membuat bentuk segitiga ke kanan.

**Loop asterisk:** k mencetak asterisk sesuai pola (2 \* i - 1) di setiap baris.

1. **Bagian bawah berlian:**

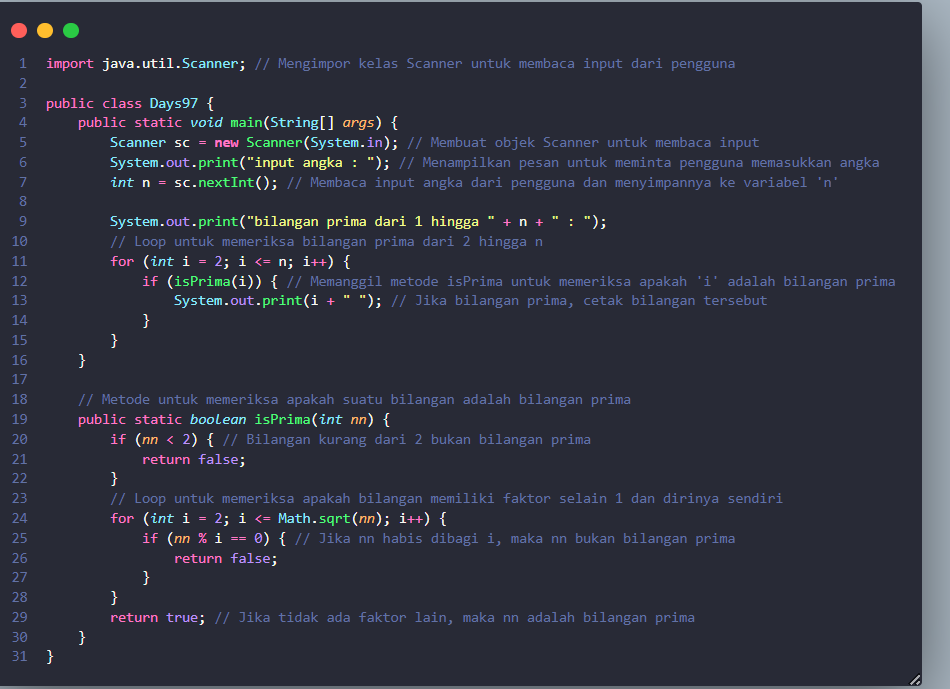
Loop i berjalan mundur dari rows - 1 hingga 1.

**Loop spasi:** Sama seperti bagian atas.

**Loop asterisk:** Sama seperti bagian atas tetapi dalam urutan menurun.

1. **Pindah baris:**

System.out.println(); digunakan untuk berpindah ke baris baru setelah mencetak spasi dan asterisk.

Soal : Mencetak bilangan prima dari 1-n

**Penjelasan:**

1. **Input dari Pengguna**:
   * Program meminta pengguna untuk memasukkan angka n.
   * Input ini digunakan untuk menentukan batas atas bilangan prima yang akan dicetak.
2. **Looping dan Pengecekan Bilangan Prima**:
   * Perulangan dimulai dari angka 2 karena bilangan prima dimulai dari angka tersebut.
   * Setiap angka diperiksa menggunakan metode isPrima.
3. **Metode isPrima**:
   * **Optimisasi**: Alih-alih memeriksa semua angka hingga nn - 1, loop hanya berjalan hingga akar kuadrat dari nn (Math.sqrt(nn)). Hal ini mengurangi jumlah iterasi dan meningkatkan efisiensi.
   * **Logika**: Jika nn habis dibagi oleh bilangan antara 2 hingga √nn, maka nn bukan bilangan prima.
4. **Output**:
   * Semua bilangan prima antara 1 dan n dicetak ke layar, dipisahkan oleh spasi.

A computer screen shot of a program code

Description automatically generated98. Soal : Total penjumlahan bilangan genap dari x ke y

**Penjelasan Kode dengan Komentar:**

1. **Import Scanner**:
   * Baris import java.util.Scanner; digunakan untuk membaca input dari pengguna.
2. **Input dari Pengguna**:
   * Program meminta pengguna untuk memasukkan dua angka, xxx dan yyy, sebagai batas perhitungan.
3. **Logika Pertukaran Nilai**:
   * Jika x>yx > yx>y, nilai-nilai tersebut ditukar menggunakan variabel sementara temp agar xxx menjadi batas bawah dan yyy menjadi batas atas.
4. **Perulangan dan Logika Bilangan Genap**:
   * Loop berjalan dari xxx hingga yyy.
   * Dengan if (i % 2 == 0), dicek apakah angka tersebut genap.
   * Bilangan genap ditambahkan ke variabel total.