LAPORAN AWAL

LAPORAN PEMROGRAMAN I

LAPORAN KE-1



Disusun Oleh :

Nama :Febiyanto Rizki Qurbandi

NIM :231011450284

Kelas :04TPLP003

TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS PAMULANG

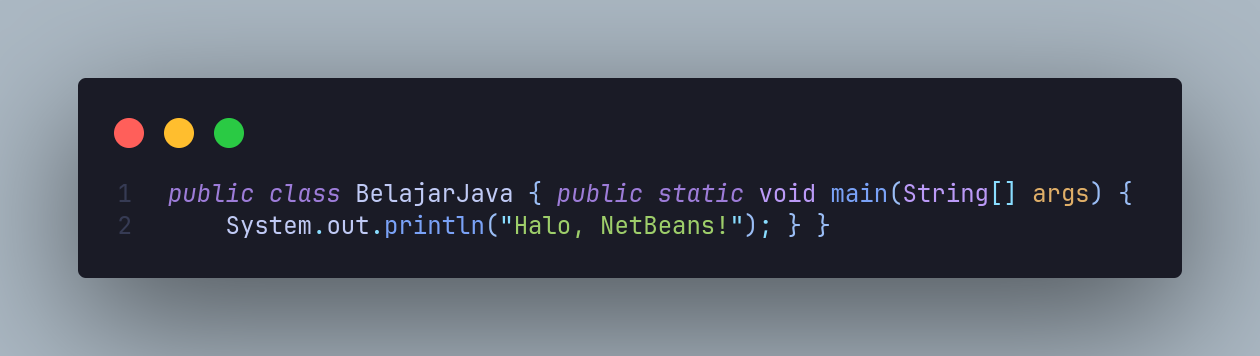
Jl. Surya Kencana No. 1 Pamulang Telp (021)7412566, Fax. (021)7412566

Tangerang Selatan – Banten

Link GITHUB Repo:

https://github.com/NeiaKI/Belajar-Coding/tree/main/Pemrograman-1%20Kuliah-Java

CODE:



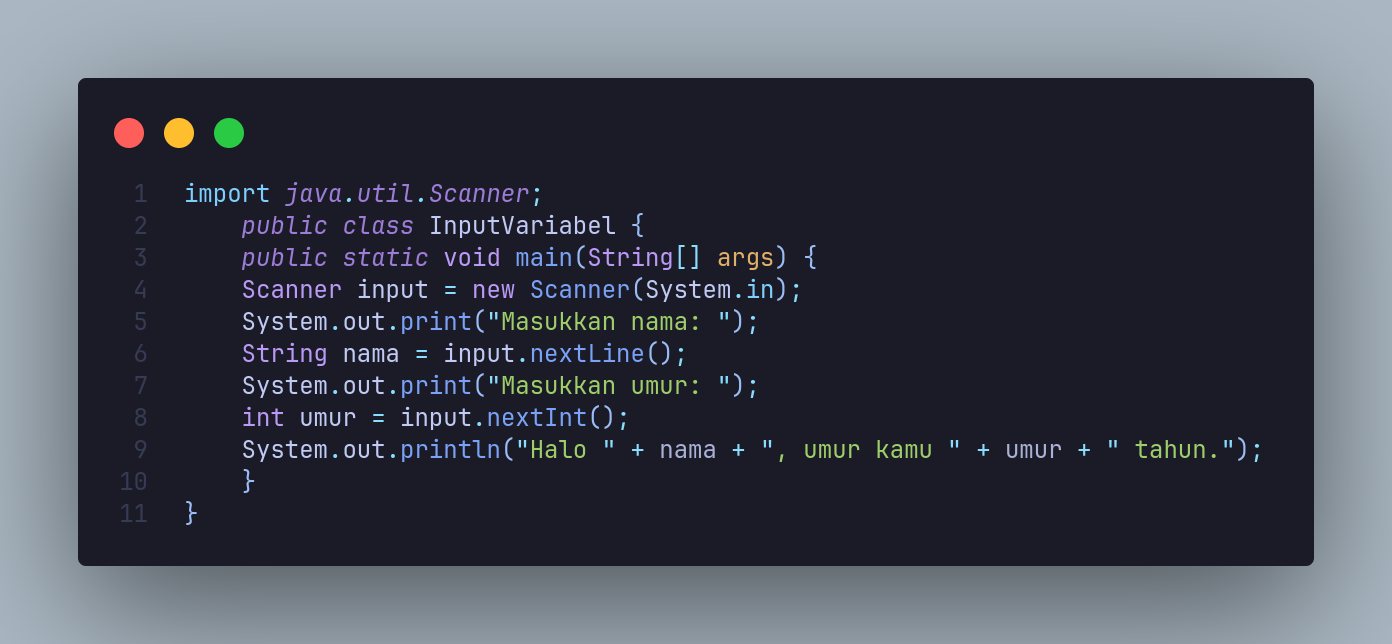
OUTPUT:



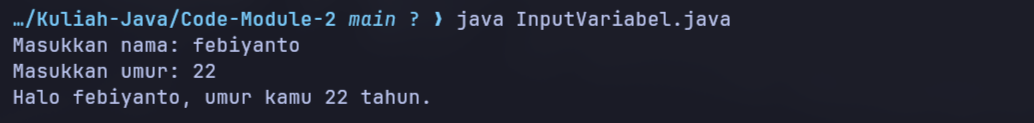
PENJELASAN:

* Definisi Kelas: Kode mendefinisikan sebuah kelas publik bernama BelajarJava yang membungkus program.
* Metode Utama (main): Program memiliki metode main, yang merupakan titik awal (entry point) di mana eksekusi kode akan dimulai.
* Mencetak Output: Program menjalankan satu perintah System.out.println() untuk mencetak teks "Halo, NetBeans!" ke konsol.

CODE;



OUTPUT:



PENJELASAN:

* Import Scanner: Mengimpor kelas Scanner agar program bisa membaca input dari pengguna.
* Definisi Kelas: Kode mendefinisikan sebuah kelas publik bernama InputVariabel.
* Metode Utama (main): Program dimulai dari metode main sebagai titik awal eksekusi.
* Buat Objek Scanner: Membuat objek Scanner bernama input untuk mengambil data dari keyboard (System.in).
* Input Nama: Menampilkan "Masukkan nama: " dan menyimpan input teks (String) dari pengguna ke dalam variabel nama menggunakan input.nextLine().
* Input Umur: Menampilkan "Masukkan umur: " dan menyimpan input angka bulat (int) dari pengguna ke dalam variabel umur menggunakan input.nextInt().
* Mencetak Output: Mencetak sapaan gabungan ke konsol, seperti "Halo Budi, umur kamu 20 tahun.", dengan menggabungkan string dan nilai dari variabel nama dan umur.

CODE:



OUTPUT:



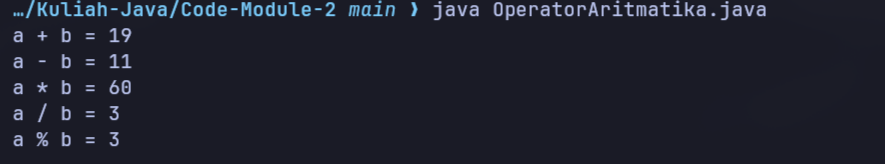
PENJELASAN:

* Definisi Kelas: Kode mendefinisikan kelas publik Main, yang menjadi wadah program.
* Metode Utama (main): Program dimulai dari metode main, yang merupakan titik awal eksekusi.
* Deklarasi Variabel: Lima variabel dibuat dan langsung diisi nilai dengan tipe data berbeda:
* int umur = 21; (angka bulat)
* double ipk = 3.75; (angka desimal)
* char grade = 'A'; (satu karakter)
* boolean aktif = true; (nilai logika benar/salah)
* String nama = "Budi"; (teks)
* Mencetak Output: Program menggunakan System.out.println() untuk mencetak setiap variabel ke konsol.
* Penggabungan String: Operator + digunakan untuk menggabungkan teks label (seperti "Nama : ") dengan nilai dari variabel.

CODE:



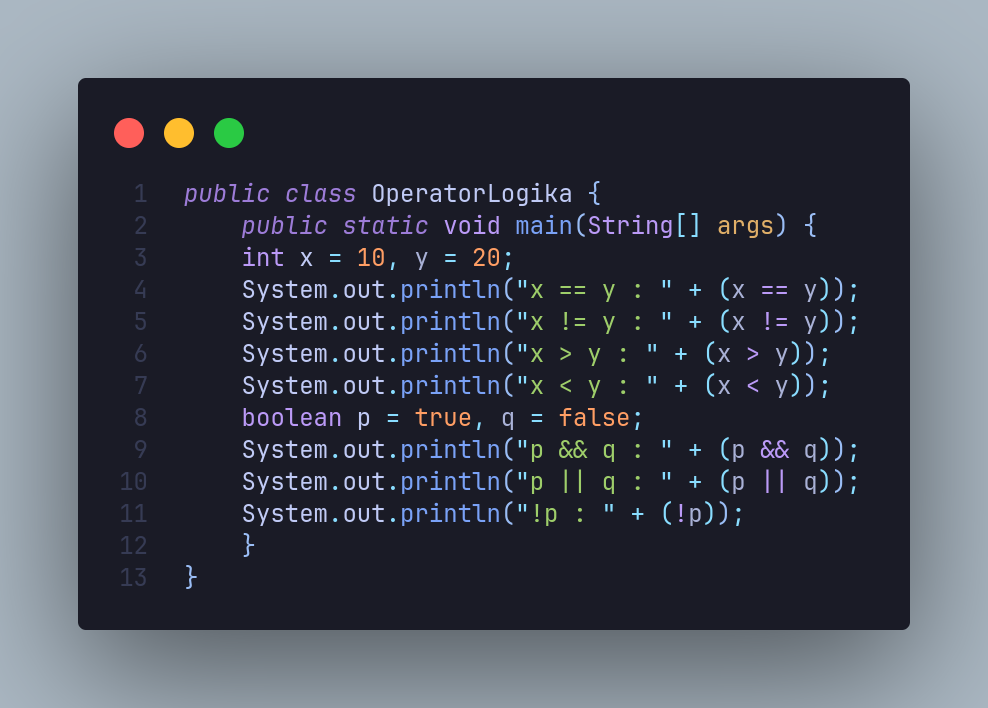
OUTPUT:



Penjelasan:

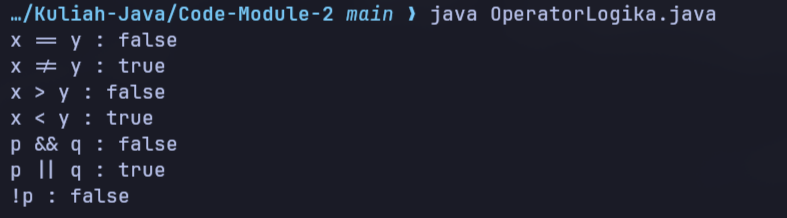
* Definisi Kelas: Kode mendefinisikan sebuah kelas publik bernama OperatorAritmatika.
* Metode Utama (main): Program memiliki metode main sebagai titik awal eksekusi.
* Inisialisasi Variabel: Dua variabel integer (angka bulat) dideklarasikan: a dengan nilai 15 dan b dengan nilai 4.
* Operasi Aritmatika & Output: Program melakukan lima operasi aritmatika dasar menggunakan a dan b, lalu langsung mencetak hasilnya ke konsol:
* Penjumlahan (+): Menghitung a + b (15 + 4) dan mencetak "a + b = 19".
* Pengurangan (-): Menghitung a - b (15 - 4) dan mencetak "a - b = 11".
* Perkalian (\*): Menghitung a \* b (15 \* 4) dan mencetak "a \* b = 60".
* Pembagian (/): Menghitung a / b (15 / 4). Karena keduanya int, hasilnya adalah pembagian bulat, yaitu 3. Mencetak "a / b = 3".
* Modulus (%): Menghitung sisa bagi a % b (15 % 4), yaitu 3. Mencetak "a % b = 3".

CODE:



OUTPUT:

Penjelasan:



* Definisi Kelas: Kode mendefinisikan kelas publik bernama OperatorLogika.
* Metode Utama (main): Program dimulai dari metode main sebagai titik awal eksekusi.
* Variabel Angka: Mendeklarasikan dua variabel integer (angka bulat): x dengan nilai 10 dan y dengan nilai 20.
* Operator Relasi (Perbandingan): Program membandingkan nilai x dan y dan mencetak hasilnya (yang berupa true atau false):
* x == y (10 sama dengan 20) akan mencetak false.
* x != y (10 tidak sama dengan 20) akan mencetak true.
* x > y (10 lebih besar dari 20) akan mencetak false.
* x < y (10 lebih kecil dari 20) akan mencetak true.
* Variabel Logika: Mendeklarasikan dua variabel boolean: p dengan nilai true dan q dengan nilai false.
* Operator Logika: Program melakukan operasi logika menggunakan variabel p dan q:
* p && q (Logika AND: true DAN false) akan mencetak false.
* p || q (Logika OR: true ATAU false) akan mencetak true.
* !p (Logika NOT: kebalikan dari p/true) akan mencetak false.

CODE:



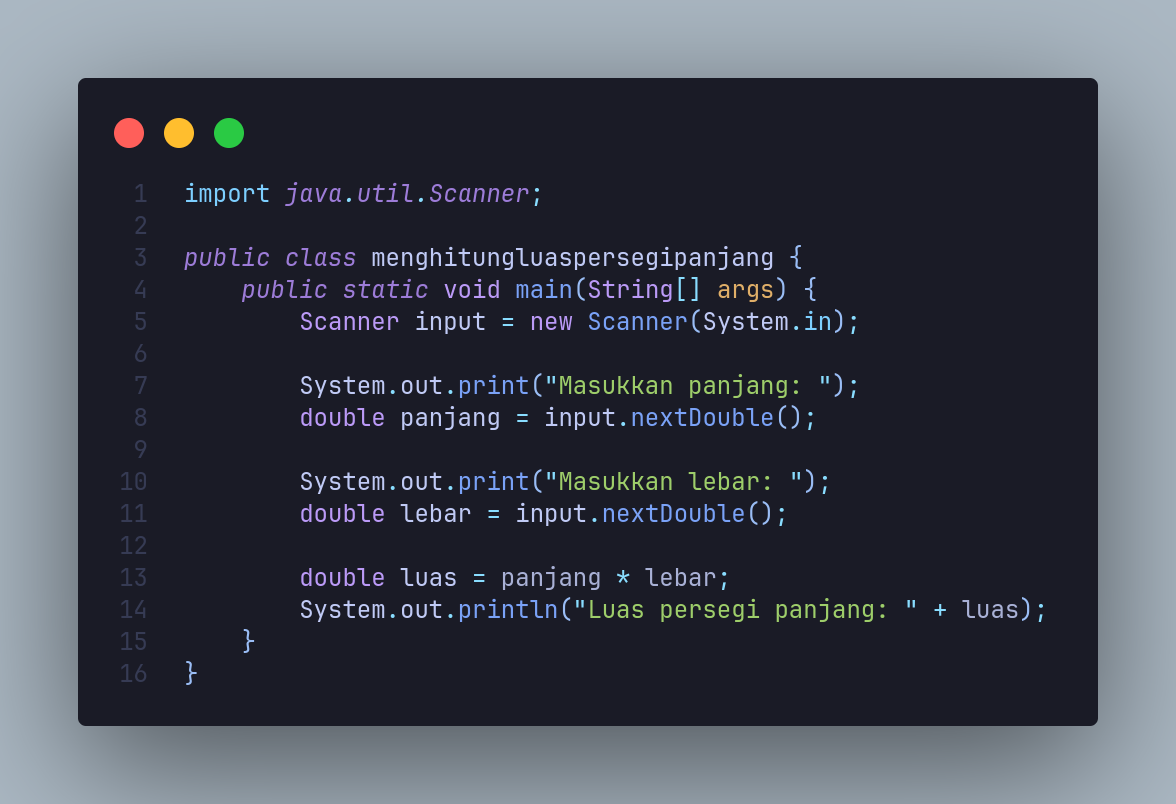
OUTPUT:



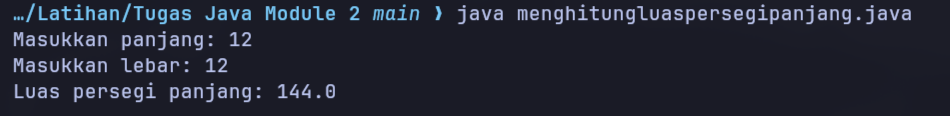
Penjelasan:

* Import Scanner: Baris 1 mengimpor kelas Scanner yang diperlukan untuk membaca input dari pengguna.
* Definisi Kelas: Kode mendefinisikan kelas publik membandingkanduaangka.
* Metode Utama (main): Program dimulai dari metode main.
* Buat Objek Scanner: Membuat objek Scanner bernama input untuk menerima masukan dari keyboard (System.in).
* Input Angka Pertama: Menampilkan teks "Masukkan angka pertama: " dan menunggu pengguna memasukkan angka, yang kemudian disimpan dalam variabel int angka1.
* Input Angka Kedua: Menampilkan teks "Masukkan angka kedua: " dan menyimpan input pengguna berikutnya ke dalam variabel int angka2.
* Struktur if-else if-else: Program menggunakan struktur kontrol untuk membandingkan kedua angka:
* if: Jika angka1 lebih besar dari angka2, cetak "Angka pertama lebih besar dari angka kedua."
* else if: Jika angka1 lebih kecil dari angka2, cetak "Angka kedua lebih besar dari angka pertama."
* else: Jika kedua kondisi di atas salah (artinya angka sama), cetak "Kedua angka sama besar."

CODE:



OUTPUT:



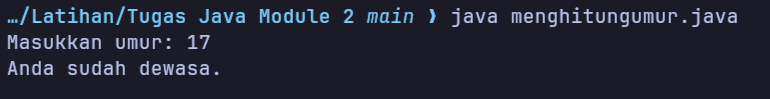
Penjelasan:

* Import Scanner: Mengimpor kelas Scanner untuk memungkinkan program menerima input dari pengguna.
* Definisi Kelas: Kode mendefinisikan sebuah kelas publik bernama menghitungLuasPersegipanjang.
* Metode Utama (main): Program memiliki metode main sebagai titik awal eksekusi.
* Buat Objek Scanner: Membuat objek Scanner bernama input untuk membaca data dari keyboard (System.in).
* Input Panjang: Menampilkan "Masukkan panjang: " dan menyimpan input angka desimal (double) dari pengguna ke dalam variabel panjang.
* Input Lebar: Menampilkan "Masukkan lebar: " dan menyimpan input angka desimal (double) berikutnya ke dalam variabel lebar.
* Hitung Luas: Melakukan perhitungan perkalian panjang \* lebar dan menyimpan hasilnya dalam variabel double bernama luas.
* Cetak Hasil: Menampilkan hasil perhitungan ke konsol dengan format "Luas persegi panjang: [nilai luas]".

CODE:



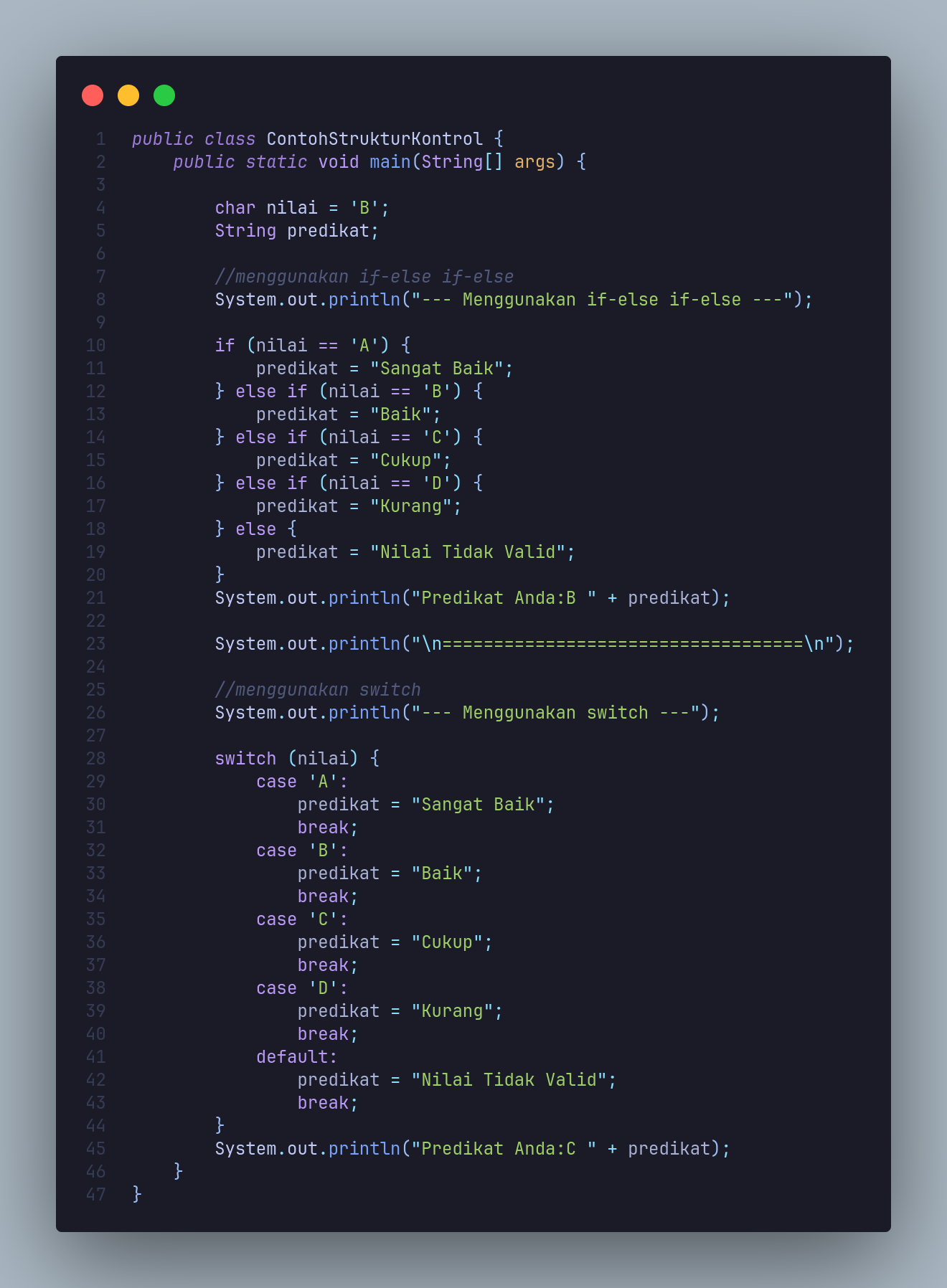
OUTPUT:



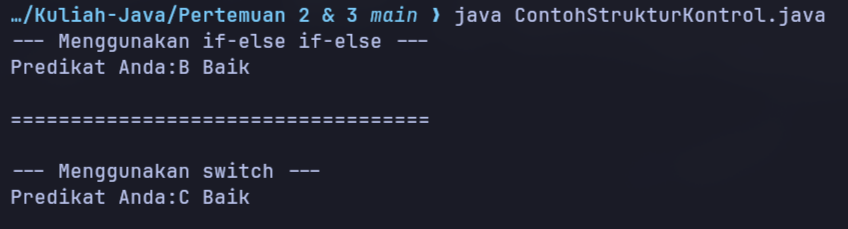
Penjelasam:

* Import Scanner: Mengimpor kelas Scanner untuk membaca input dari pengguna.
* Definisi Kelas: Kode mendefinisikan kelas publik bernama menghitungumur.
* Metode Utama (main): Program dimulai dari metode main.
* Buat Objek Scanner: Membuat objek Scanner bernama input untuk menerima masukan dari keyboard (System.in).
* Input Umur: Menampilkan "Masukkan umur: " dan menunggu pengguna memasukkan angka, yang kemudian disimpan dalam variabel int umur.
* Struktur if-else: Program menggunakan struktur kontrol untuk mengecek nilai umur:
* if: Jika umur lebih besar dari atau sama dengan 17 (>= 17), cetak "Anda sudah dewasa."
* else: Jika kondisi di atas tidak terpenuhi (artinya umur di bawah 17), cetak "Anda belum dewasa."

CODE:



OUTPUT:



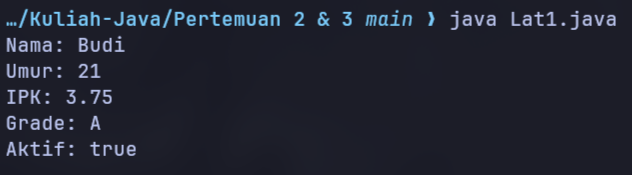
Penjelasan:

* Definisi Kelas & Metode: Kode mendefinisikan kelas ContohStrukturKontrol dengan metode main sebagai titik awal eksekusi.
* Inisialisasi Variabel: Mendeklarasikan variabel char nilai dan memberinya nilai 'B'. Sebuah variabel String predikat juga disiapkan untuk menampung hasil.
* Demo 1: Struktur if-else if-else:
* Program pertama-tama menggunakan rangkaian if-else if untuk menentukan predikat berdasarkan nilai.
* Karena nilai adalah 'B', kondisi pada baris 12 (else if (nilai == 'B')) terpenuhi.
* Variabel predikat diatur menjadi "Baik".
* Program mencetak hasil dari predikat ("Predikat Anda:B Baik").
* Demo 2: Struktur switch:
* Setelah mencetak pemisah, program mengulangi logika yang sama menggunakan switch.
* switch(nilai) mengevaluasi variabel nilai (yaitu 'B').
* Program menemukan kecocokan pada case 'B': (baris 31).
* Variabel predikat kembali diatur menjadi "Baik", dan break menghentikan switch.
* Program mencetak hasil dari predikat sekali lagi ("Predikat Anda:C Baik").

CODE:



OUTPUT:



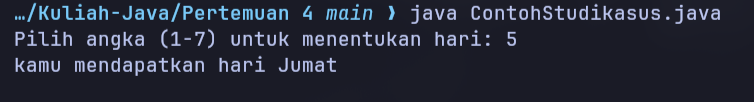
Penjelasan:

* Definisi Kelas: Kode mendefinisikan sebuah kelas publik bernama Lat1 sebagai wadah program.
* Metode Utama (main): Program memiliki metode main yang merupakan titik awal eksekusi.
* Deklarasi Variabel: Lima variabel dibuat dan diinisialisasi (diisi nilai) dengan tipe data yang berbeda:
* int umur = 21; (angka bulat)
* double ipk = 3.75; (angka desimal/pecahan)
* char grade = 'A'; (satu karakter)
* boolean aktif = true; (nilai logika benar/salah)
* String nama = "Budi"; (teks)
* Mencetak Output: Program menggunakan System.out.println() sebanyak lima kali untuk mencetak nilai dari setiap variabel ke konsol, masing-masing didahului oleh label (seperti "Nama: ", "Umur: ", dll.).

CODE:



OUTPUT:



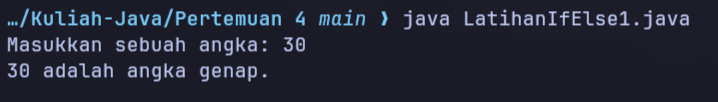
Penjelasan:

* Import Scanner: Mengimpor kelas Scanner agar program dapat membaca input dari pengguna.
* Definisi Kelas & Metode: Mendefinisikan kelas ContohStudikasus dan metode main sebagai titik awal eksekusi.
* Buat Objek Scanner: Membuat objek Scanner bernama input untuk membaca data dari keyboard (System.in).
* Input Angka: Meminta pengguna memasukkan angka (1-7) untuk menentukan hari dan menyimpannya dalam variabel int hari.
* Inisialisasi Variabel: Menyiapkan variabel String namaHari dengan nilai awal kosong.
* Struktur switch: Menggunakan switch untuk mengevaluasi nilai variabel hari:
* case 1 sampai 7: Jika hari bernilai 1, namaHari diisi "Senin"; jika 2, diisi "Selasa", dan seterusnya hingga "Minggu" untuk case 7. Perintah break digunakan untuk keluar dari switch setelah case yang cocok ditemukan.
* default: Jika hari tidak cocok dengan case 1-7, program mencetak "Angka tidak valid." dan menggunakan return untuk menghentikan eksekusi metode main lebih awal.
* Cetak Hasil: Jika switch menemukan case yang cocok (bukan default), program akan mencetak "kamu mendapatkan hari " diikuti dengan nama hari yang tersimpan di namaHari.

CODE: Pertemuan 4



OUTPUT:



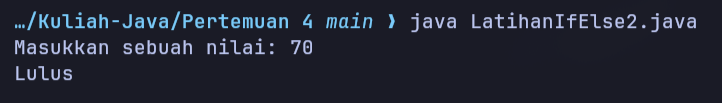
Penjelasan:

* Import Scanner: Mengimpor kelas Scanner agar program dapat menerima input dari pengguna.
* Definisi Kelas & Metode: Mendefinisikan kelas LatihanIfElse1 dan metode main sebagai titik awal eksekusi.
* Input Pengguna: Membuat objek Scanner (input) untuk membaca dari keyboard, lalu meminta pengguna memasukkan angka yang disimpan ke variabel int angka.
* Logika if-else (Ganjil/Genap): Program mengecek kondisi angka:
* if (angka % 2 == 0): Menggunakan operator modulus (%) untuk mengecek sisa bagi. Jika sisa bagi angka dengan 2 adalah 0, cetak bahwa angka tersebut "genap".
* else: Jika sisa baginya bukan 0 (kondisi if salah), cetak bahwa angka tersebut "ganjil".

CODE: Pertemuan 4



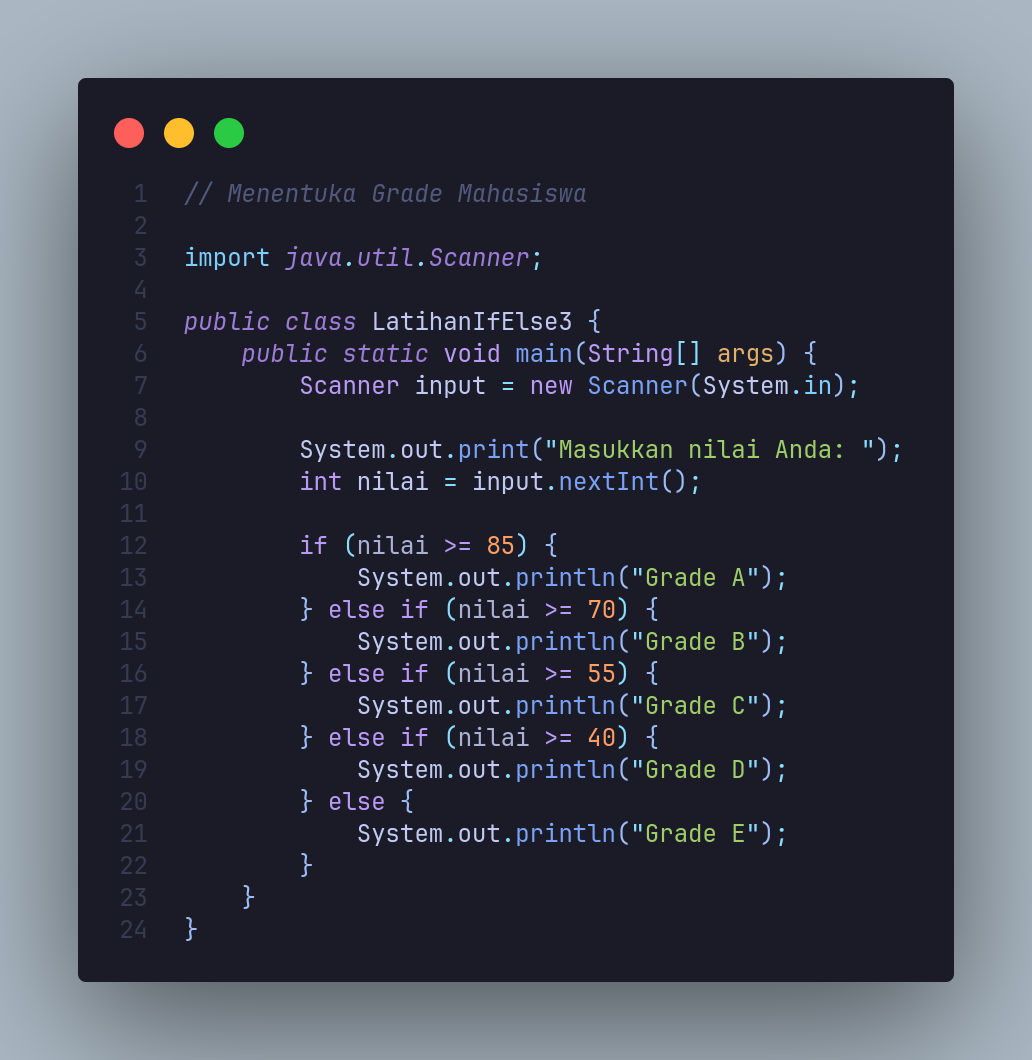
OUTPUT:



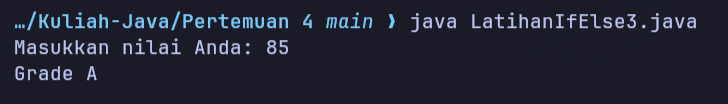
Penjelasan:

* Import Scanner: Mengimpor kelas Scanner agar program dapat menerima input dari pengguna.
* Definisi Kelas & Metode: Mendefinisikan kelas LatihanIfElse2 dan metode main sebagai titik awal eksekusi.
* Input Pengguna: Membuat objek Scanner (input), lalu meminta pengguna memasukkan nilai yang disimpan ke variabel int nilai.
* Logika if-else (Kelulusan): Program mengecek kondisi nilai:
* if (nilai >= 70): Jika nilai lebih besar dari atau sama dengan 70, cetak "Lulus".
* else: Jika kondisi di atas tidak terpenuhi (artinya nilai di bawah 70), cetak "Tidak Lulus".

CODE: Pertemuan 4



OUTPUT:



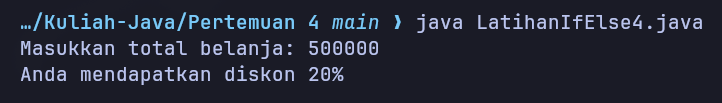
Penjelasan:

* Import Scanner: Mengimpor kelas Scanner agar program dapat menerima input dari pengguna.
* Definisi Kelas & Metode: Mendefinisikan kelas LatihanIfElse3 dan metode main sebagai titik awal eksekusi.
* Input Pengguna: Membuat objek Scanner (input), lalu meminta pengguna memasukkan nilai yang disimpan ke variabel int nilai.
* Struktur if-else if-else (Grading): Program menggunakan struktur if-else if bertingkat untuk menentukan grade:
* Jika nilai >= 85, cetak "Grade A".
* Jika tidak (berarti di bawah 85), tapi nilai >= 70, cetak "Grade B".
* Jika tidak (berarti di bawah 70), tapi nilai >= 55, cetak "Grade C".
* Jika tidak (berarti di bawah 55), tapi nilai >= 40, cetak "Grade D".
* else: Jika semua kondisi di atas salah (berarti nilai < 40), cetak "Grade E".

CODE: Pertemuan 4



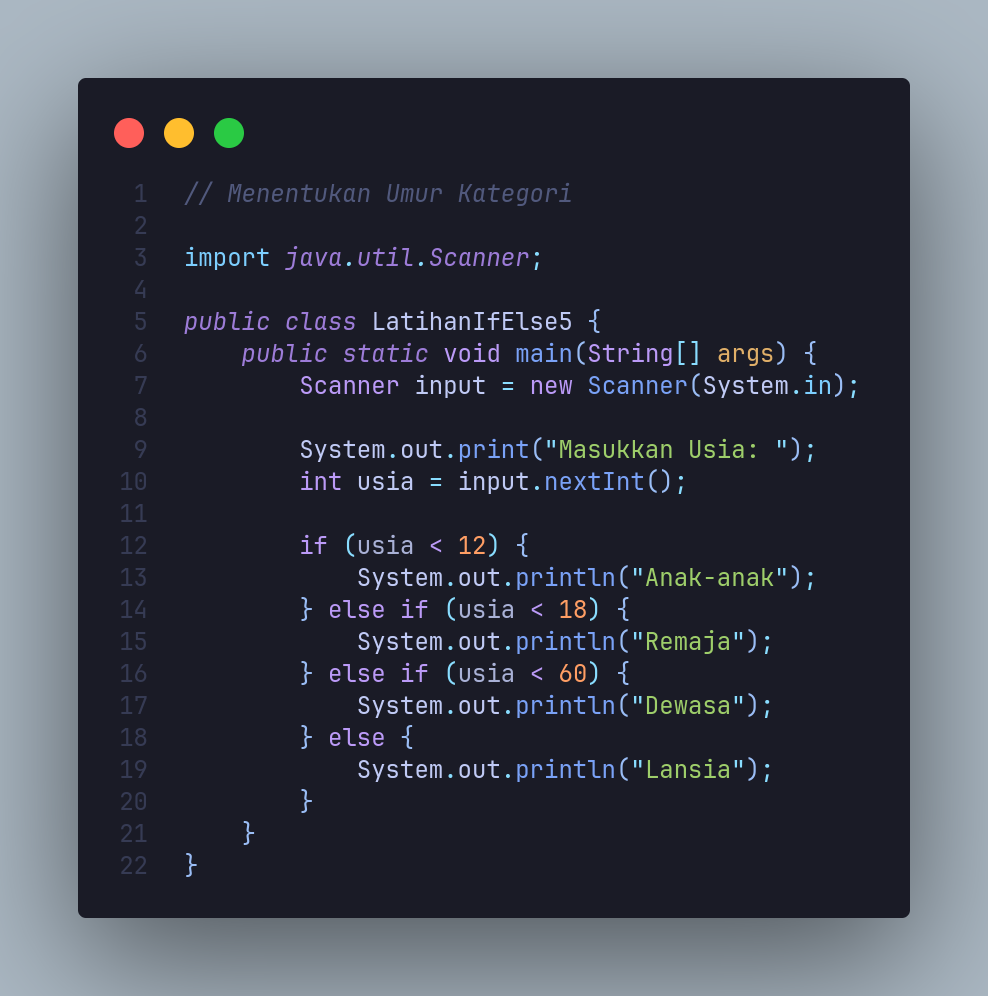
OUTPUT:



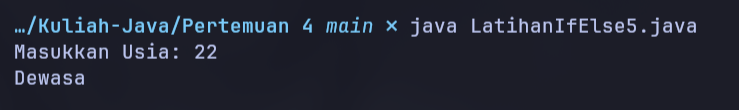
Penjelasan:

* Import Scanner: Mengimpor kelas Scanner agar program dapat menerima input dari pengguna.
* Definisi Kelas & Metode: Mendefinisikan kelas LatihanIfElse4 dan metode main sebagai titik awal eksekusi.
* Input Pengguna: Membuat objek Scanner (input), lalu meminta pengguna memasukkan "total belanja" dan menyimpannya ke variabel int total.
* Struktur if-else if-else (Diskon): Program mengecek nilai total untuk menentukan diskon:
* Jika total >= 500000, cetak "Anda mendapatkan diskon 20%".
* Jika tidak (berarti di bawah 500000), tapi total >= 200000, cetak "Anda mendapatkan diskon 10%".
* else: Jika semua kondisi di atas salah (berarti total < 200000), cetak "Tidak ada diskon".

CODE: Pertemuan 4



OUTPUT:



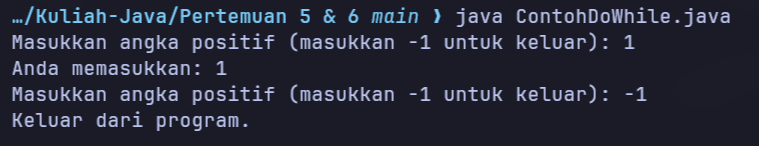
Penjelasan:

* Import Scanner: Mengimpor kelas Scanner agar program dapat menerima input dari pengguna.
* Definisi Kelas & Metode: Mendefinisikan kelas LatihanIfElse5 dan metode main sebagai titik awal eksekusi.
* Input Pengguna: Membuat objek Scanner (input), lalu meminta pengguna memasukkan "Usia" dan menyimpannya ke variabel int usia.
* Struktur if-else if-else (Kategori Usia): Program mengecek nilai usia secara bertingkat:
* Jika usia < 12, cetak "Anak-anak".
* Jika tidak (berarti >= 12), tapi usia < 18, cetak "Remaja".
* Jika tidak (berarti >= 18), tapi usia < 60, cetak "Dewasa".
* else: Jika semua kondisi di atas salah (berarti usia >= 60), cetak "Lansia".

CODE: Pertemuan 5 & 6



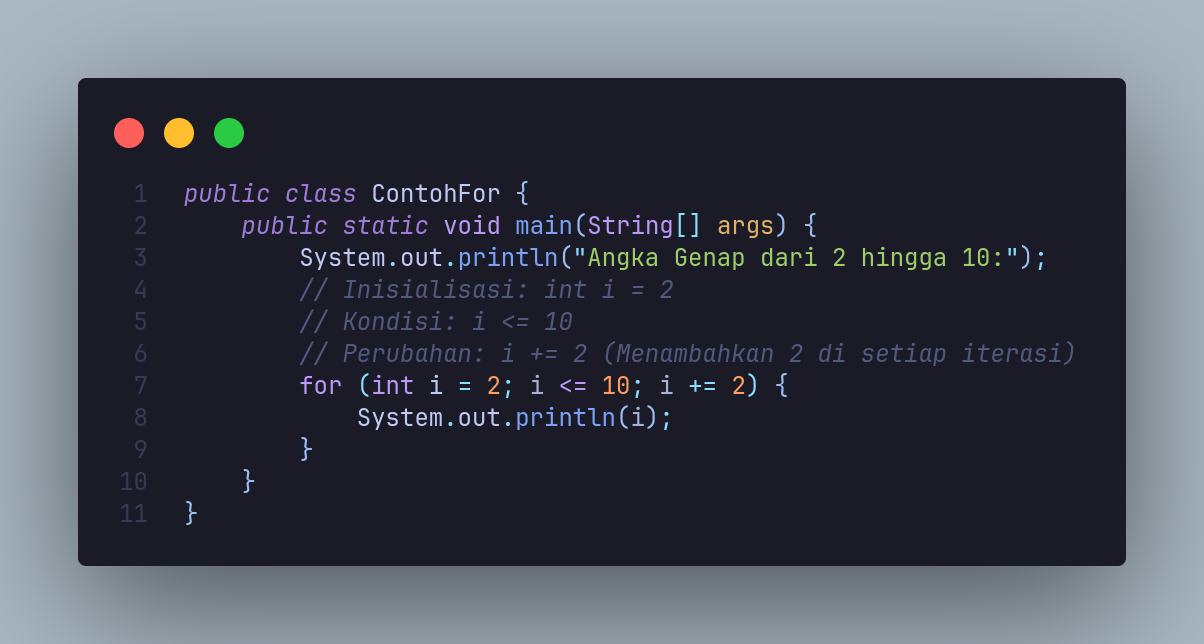
OUTPUT:



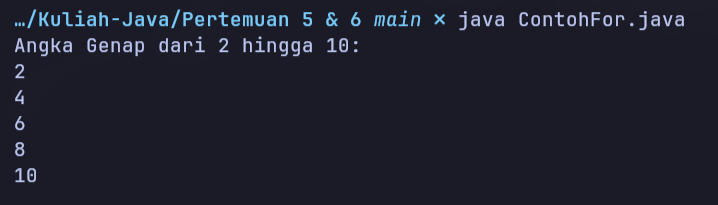
Penjelasan:

* Import Scanner: Mengimpor kelas Scanner untuk membaca input dari pengguna.
* Definisi Kelas & Metode: Mendefinisikan kelas ContohDoWhile dan metode main sebagai titik awal eksekusi.
* Objek Scanner & Variabel: Membuat objek Scanner (scanner) dan mendeklarasikan variabel int input.
* do-while Loop: Memulai loop do-while. Blok do (baris 9-20) akan dieksekusi setidaknya satu kali sebelum kondisi while diperiksa.
* Proses Input: Di dalam loop, program meminta pengguna memasukkan angka (positif atau -1) dan menyimpannya ke variabel input.
* Pengecekan Kondisi (if-else):
* Jika input > 0, cetak angka yang dimasukkan.
* Jika input == -1, cetak pesan keluar.
* Jika lainnya (misal 0 atau negatif lain), cetak pesan error.
* Kondisi while: Setelah blok do selesai, program memeriksa kondisi while (input != -1). Jika input bukan -1, loop akan diulang kembali dari blok do. Jika input adalah -1, loop berhenti.

CODE: Pertemuan 5 & 6



OUTPUT:



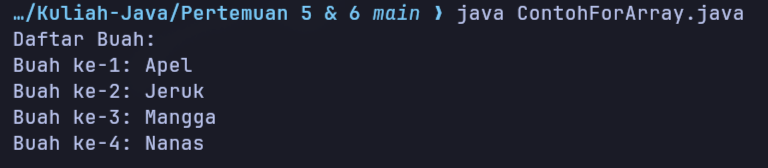
Penjelasan:

* Definisi Kelas & Metode: Mendefinisikan kelas ContohFor dan metode main sebagai titik awal eksekusi.
* Cetak Judul: Program mencetak teks "Angka Genap dari 2 hingga 10:" ke konsol.
* for Loop (Perulangan): Program menggunakan for loop untuk mencetak angka genap.
* Inisialisasi: Memulai loop dengan variabel int i diatur ke 2.
* Kondisi: Loop akan terus berjalan selama i kurang dari atau sama dengan 10 (i <= 10).
* Perubahan (Increment): Setelah setiap iterasi, nilai i ditambah 2 (i += 2).
* Aksi Loop: Di dalam setiap iterasi loop, program mencetak nilai i saat ini. Ini akan mencetak 2, 4, 6, 8, dan 10, masing-masing di baris baru.

CODE: Pertemuan 5 & 6



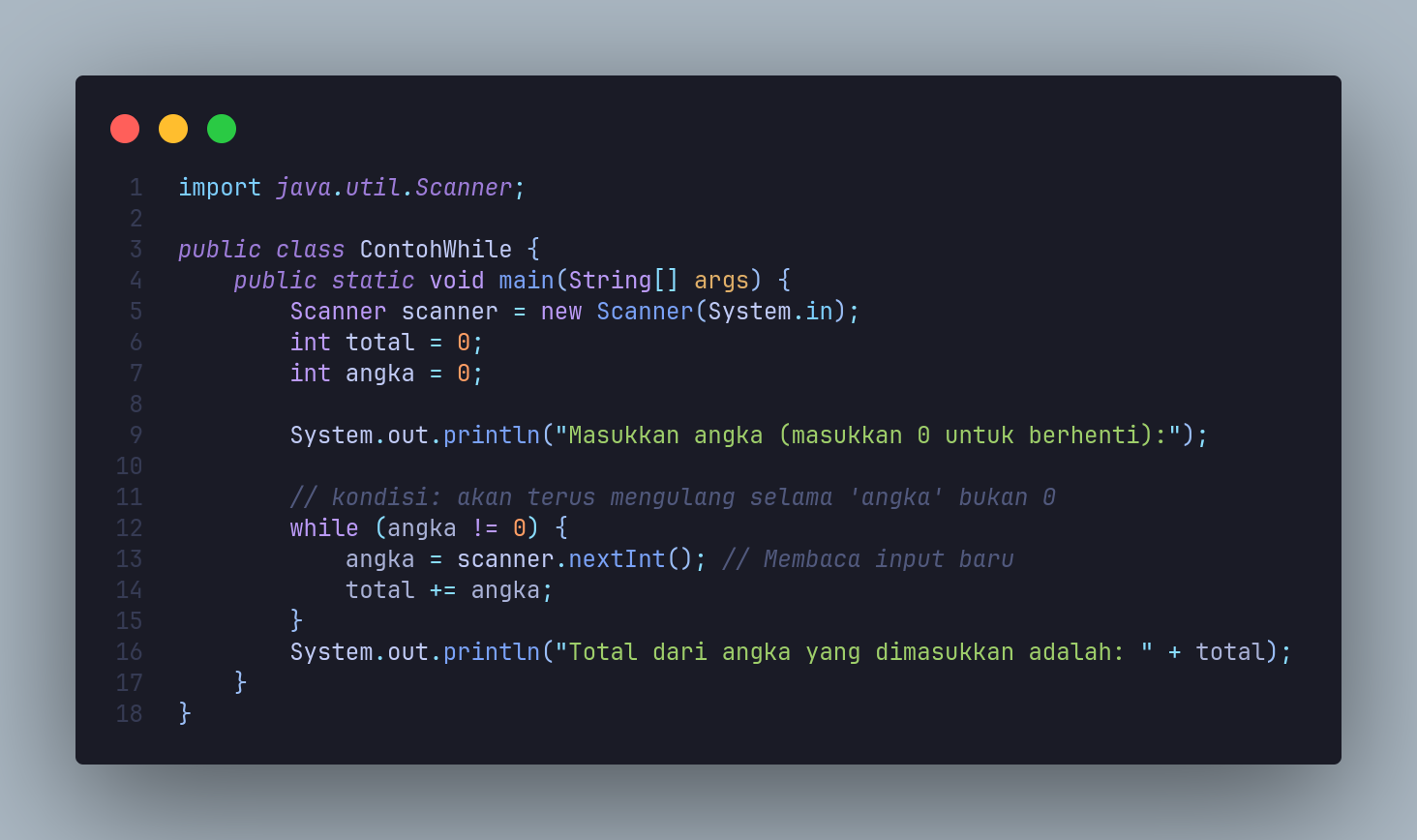
OUTPUT:



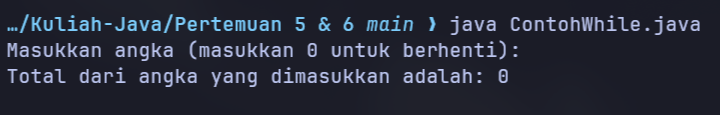
Penjelasan:

* Definisi Kelas & Metode: Mendefinisikan kelas ContohForArray dan metode main sebagai titik awal eksekusi.
* Inisialisasi Array: Membuat sebuah array String bernama buah yang berisi empat elemen: "Apel", "Jeruk", "Mangga", dan "Nanas".
* Cetak Judul: Mencetak teks "Daftar Buah:" ke konsol.
* for Loop (Iterasi Array): Program menggunakan for loop untuk mengakses setiap elemen array berdasarkan indeksnya.
* Loop dimulai dengan int i = 0.
* Berlanjut selama i kurang dari panjang array (buah.length, yaitu 4).
* i bertambah satu (i++) setiap selesai iterasi.
* Aksi Loop: Di dalam loop, program mencetak nomor urut (menggunakan i + 1 agar mulai dari 1) dan nama buah yang sesuai dengan indeks i (misal, "Buah ke-1: Apel", "Buah ke-2: Jeruk", dst.).

CODE : Pertemuan 5 & 6



OUTPUT:



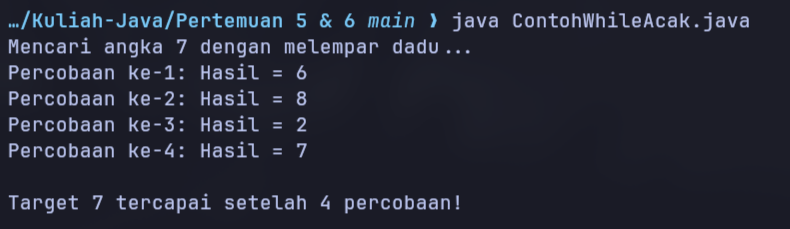
Penjelasan:

* Import & Setup: Mengimpor kelas Scanner, mendefinisikan kelas ContohWhile, dan membuat objek Scanner (scanner).
* Inisialisasi Variabel: Mendeklarasikan dua variabel integer, total dan angka, keduanya diberi nilai awal 0.
* Prompt Pengguna: Mencetak instruksi ke konsol: "Masukkan angka (masukkan 0 untuk berhenti):".
* Pengecekan while Loop: Program mengecek kondisi perulangan while (angka != 0).
* Eksekusi Loop: Karena angka bernilai 0 saat pengecekan pertama, kondisi (0 != 0) adalah false. Akibatnya, blok kode di dalam loop (baris 13-14) tidak pernah dijalankan.
* Cetak Hasil: Program langsung melompat ke baris 16 dan mencetak nilai total saat ini, sehingga outputnya adalah "Total dari angka yang dimasukkan adalah: 0".

CODE: Pertemuan 5 & 6



OUTPUT:



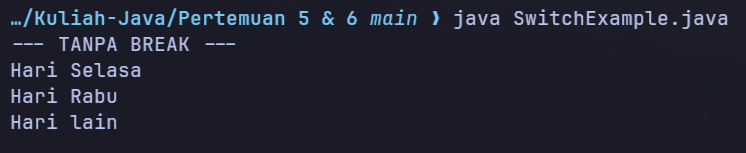
Penjelasan:

* Import Random: Mengimpor kelas Random yang diperlukan untuk menghasilkan angka acak.
* Definisi Kelas & Metode: Mendefinisikan kelas ContohWhileAcak dan metode main.
* Inisialisasi:
* Membuat objek Random bernama rand.
* Menetapkan int target ke nilai 7.
* Menginisialisasi int percobaan dan int hasilDadu ke 0.
* while Loop: Memulai loop yang akan terus berjalan selama hasilDadu tidak sama dengan target (yaitu, bukan 7).
* Isi Loop: Di dalam setiap iterasi loop:
* hasilDadu diisi dengan angka acak baru antara 1 dan 10 (menggunakan rand.nextInt(10) + 1).
* percobaan ditambah 1 (percobaan++).
* Hasil percobaan saat ini dicetak ke konsol.
* Hasil Akhir: Setelah loop berhenti (ketika hasilDadu akhirnya bernilai 7), program mencetak pesan akhir yang menunjukkan berapa banyak percobaan yang diperlukan untuk mencapai target 7.

CODE: Pertemuan 5 & 6



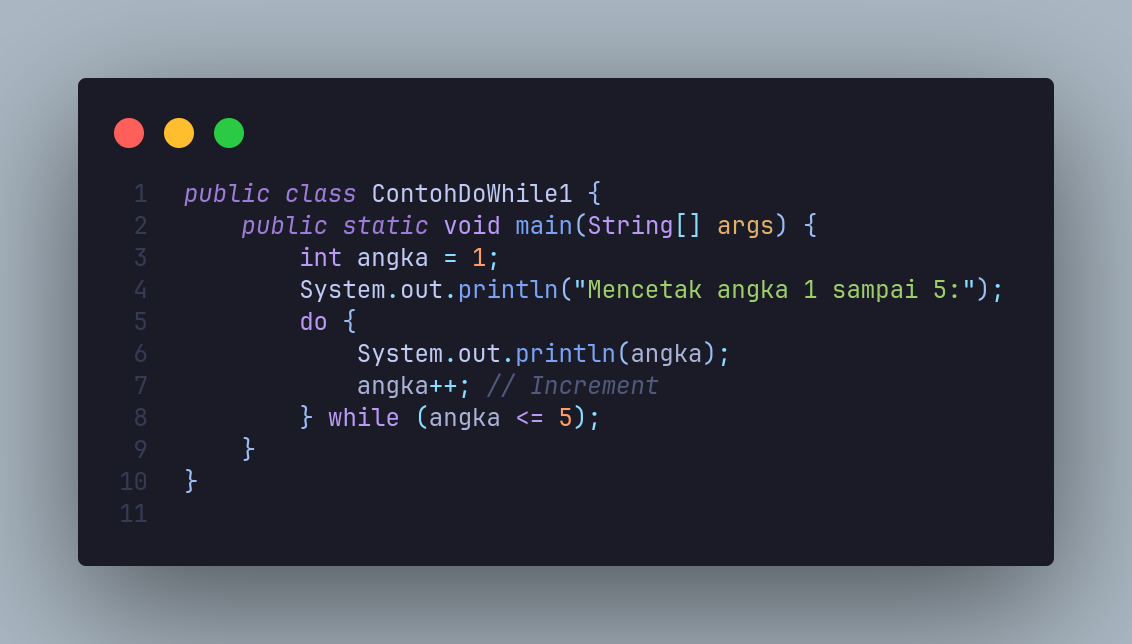
OUTPUT:



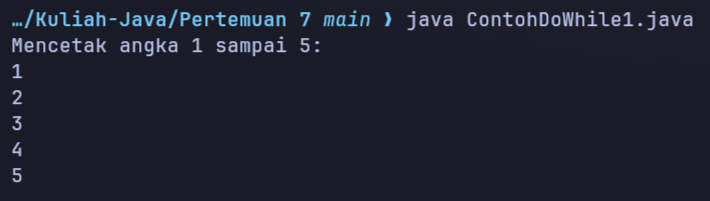
Penjelasan:

* Definisi Kelas & Metode: Mendefinisikan kelas SwitchExample dan metode main sebagai titik awal eksekusi.
* Inisialisasi Variabel: Mendeklarasikan variabel int hari dan memberinya nilai 2.
* Struktur switch: Memulai switch untuk mengevaluasi nilai hari.
* Pencocokan case: Program menemukan kecocokan pada case 2:.
* Eksekusi & "Fall-through":
* Program mencetak "Hari Selasa" (baris 10).
* Karena tidak ada break, eksekusi berlanjut ("fall-through") ke case 3:, lalu mencetak "Hari Rabu" (baris 12).
* Eksekusi berlanjut lagi ke default:, dan mencetak "Hari lain" (baris 14).
* Output Akhir: Hasilnya, tiga baris akan tercetak (Selasa, Rabu, dan Hari lain) karena tidak digunakannya break untuk menghentikan switch setelah case 2 ditemukan.

CODE: Pertemuan 7



OUTPUT:



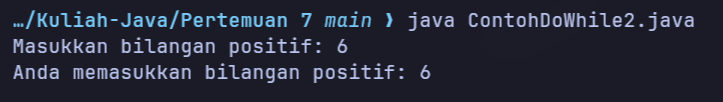
Penjelasan:

* Definisi Kelas & Metode: Mendefinisikan kelas ContohDoWhile1 dan metode main sebagai titik awal eksekusi.
* Inisialisasi Variabel: Mendeklarasikan variabel int angka dan memberinya nilai awal 1.
* Cetak Judul: Mencetak teks "Mencetak angka 1 sampai 5:" ke konsol.
* do-while Loop: Memulai loop do-while. Blok do (baris 6-7) dijamin berjalan setidaknya satu kali.
* Isi Loop:
* System.out.println(angka);: Mencetak nilai angka saat ini.
* angka++;: Menambah nilai angka sebesar 1 (increment).
* Kondisi while: Setelah blok do dieksekusi, program memeriksa kondisi while (angka <= 5). Jika true, loop diulang. Jika false, loop berhenti.
* Hasil Eksekusi: Program akan mencetak angka 1, 2, 3, 4, dan 5, masing-masing di baris baru.

CODE: Pertemuan 7



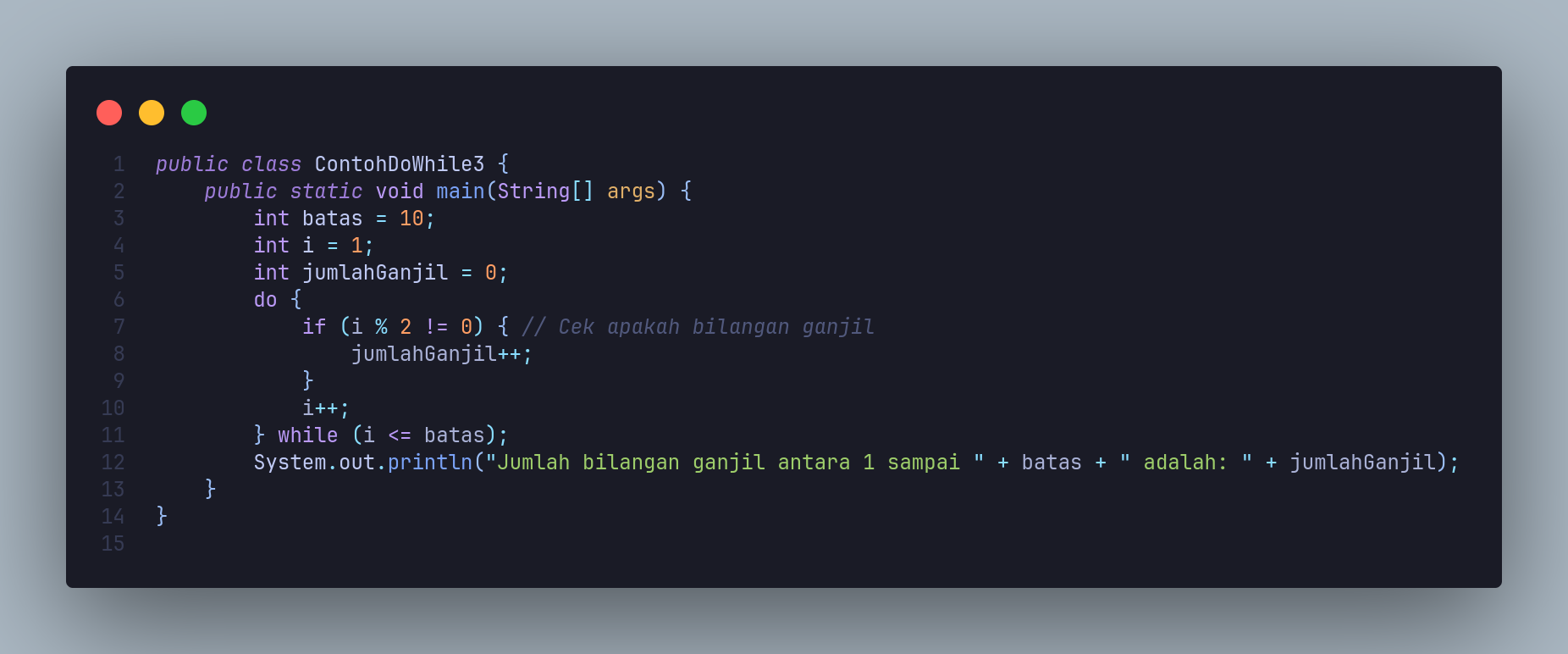
OUTPUT:



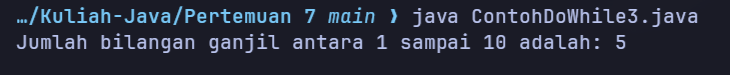
Penjelasan:

* Import Scanner: Mengimpor kelas Scanner untuk membaca input dari pengguna.
* Definisi Kelas & Metode: Mendefinisikan kelas ContohDoWhile2 dan metode main.
* Setup: Membuat objek Scanner (scanner) dan mendeklarasikan variabel int bilangan.
* do-while Loop (Validasi Input): Memulai loop do-while, yang menjamin blok kode di dalamnya berjalan setidaknya satu kali.
* Isi Loop:
* Meminta pengguna memasukkan "bilangan positif".
* Membaca input angka dari pengguna dan menyimpannya ke bilangan.
* Jika bilangan 0 atau negatif (<= 0), cetak pesan error "Bilangan harus positif. Coba lagi.".
* Kondisi while: Loop akan terus diulang (kembali ke do) selama kondisi (bilangan <= 0) benar.
* Hasil: Jika pengguna memasukkan angka positif (kondisi while salah), loop berhenti, dan program mencetak pesan konfirmasi "Anda memasukkan bilangan positif: [angka]".
* Tutup Scanner: Menutup objek scanner untuk melepas sumber daya.

CODE: Pertemuan 7



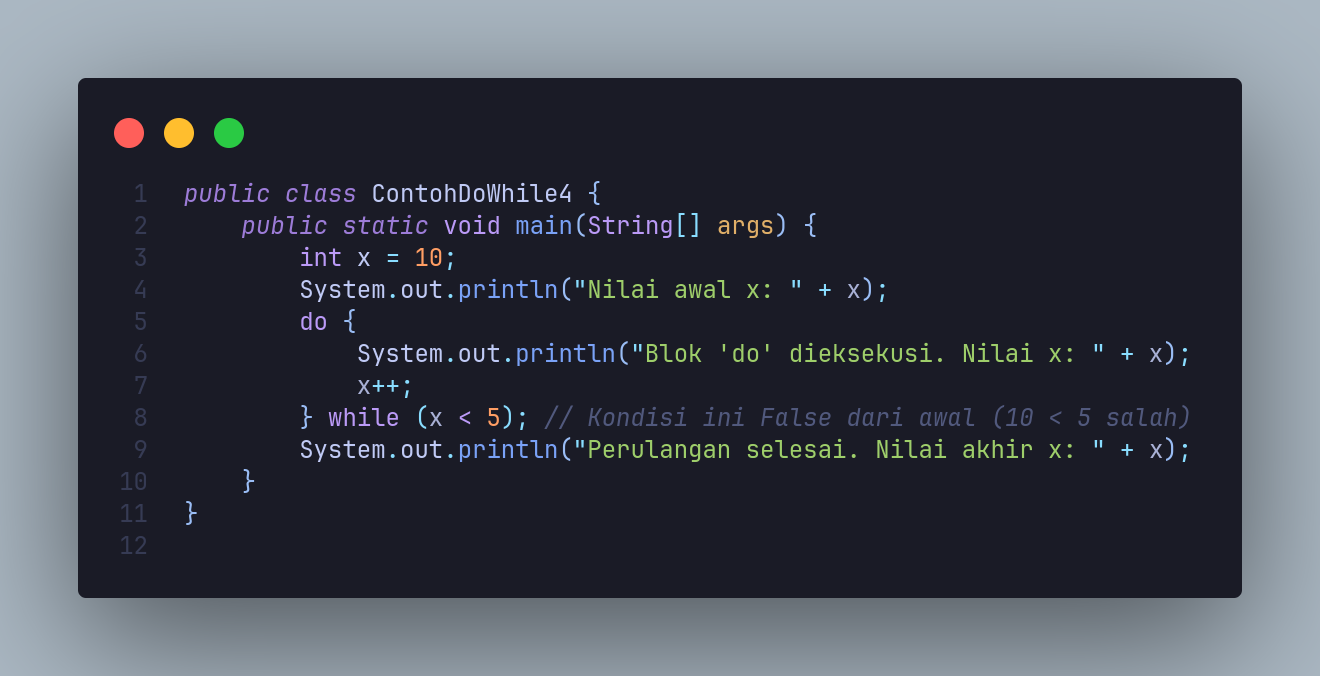
OUTPUT:



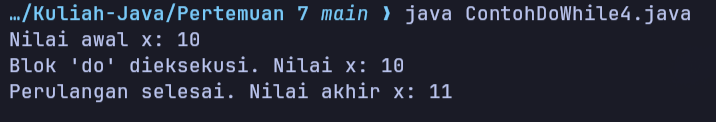
Penjelasan:

* Definisi Kelas & Metode: Mendefinisikan kelas ContohDoWhile3 dan metode main sebagai titik awal eksekusi.
* Inisialisasi Variabel: Menyiapkan tiga variabel integer: batas = 10, i = 1 (sebagai counter awal), dan jumlahGanjil = 0 (untuk menyimpan hasil hitungan).
* do-while Loop: Memulai loop do-while. Blok do (baris 7-10) akan dieksekusi terlebih dahulu.
* Logika Ganjil: Di dalam loop, program mengecek apakah i adalah bilangan ganjil menggunakan if (i % 2 != 0).
* Penghitungan: Jika i ganjil, jumlahGanjil ditambah satu (++). Setelah itu, i ditambah satu (i++) untuk melanjutkan ke angka berikutnya.
* Kondisi while: Loop akan terus diulang selama i masih kurang dari atau sama dengan batas (i <= 10).
* Cetak Hasil: Setelah loop selesai (ketika i mencapai 11), program mencetak nilai akhir jumlahGanjil, yaitu "5".

CODE: Pertemuan 7



OUTPUT:



Penjelasan:

* Definisi Kelas & Metode: Mendefinisikan kelas ContohDoWhile4 dan metode main.
* Inisialisasi Variabel: Mendeklarasikan int x dan memberinya nilai 10.
* Cetak Nilai Awal: Mencetak "Nilai awal x: 10".
* Eksekusi Blok do: Blok do (baris 6-7) dijamin berjalan satu kali meskipun kondisinya salah.
* Mencetak "Blok 'do' dieksekusi. Nilai x: 10".
* Nilai x ditambah satu (x++), sehingga x menjadi 11.
* Pengecekan while: Program memeriksa kondisi while (x < 5).
* Kondisi Salah: Karena x adalah 11, kondisi (11 < 5) bernilai false.
* Loop Berhenti: Loop langsung berhenti setelah satu kali eksekusi.
* Cetak Akhir: Program mencetak "Perulangan selesai. Nilai akhir x: 11".

CODE: Pertemuan 7



OUTPUT:

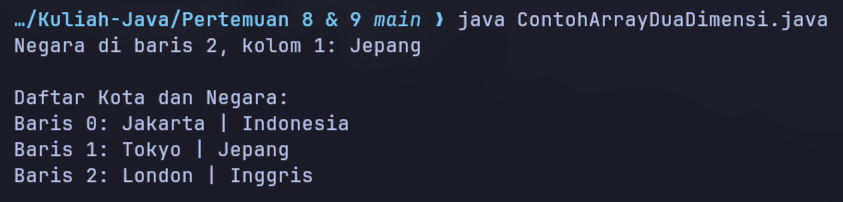


Penjelasan:

* Setup: Mengimpor kelas Scanner, mendefinisikan kelas ContohDoWhile5, dan membuat objek Scanner untuk input pengguna.
* do-while Loop: Memulai loop do-while yang menjamin menu akan ditampilkan setidaknya satu kali.
* Tampilkan Menu: Di dalam blok do, program mencetak menu aplikasi dengan tiga pilihan (1: Tampilkan Data, 2: Tambah Data, 0: Keluar).
* Input Pengguna: Program meminta pengguna memasukkan pilihan dan menyimpannya ke variabel int pilihan.
* Struktur switch: Menggunakan switch untuk mengevaluasi pilihan pengguna:
* case 1: Mencetak "Anda memilih: Tampilkan Data."
* case 2: Mencetak "Anda memilih: Tambah Data."
* case 0: Mencetak "Terima kasih. Aplikasi ditutup."
* default: Mencetak "Pilihan tidak valid." jika inputnya angka lain.
* Kondisi while: Setelah blok do selesai, program memeriksa kondisi while (pilihan != 0). Jika pilihan bukan 0, loop akan diulang (menu ditampilkan lagi).
* Keluar dari Loop: Jika pengguna memasukkan 0, kondisi while menjadi false, loop berhenti, dan scanner ditutup.

CODE: Pertemuan 8 & 9

OUTPUT:



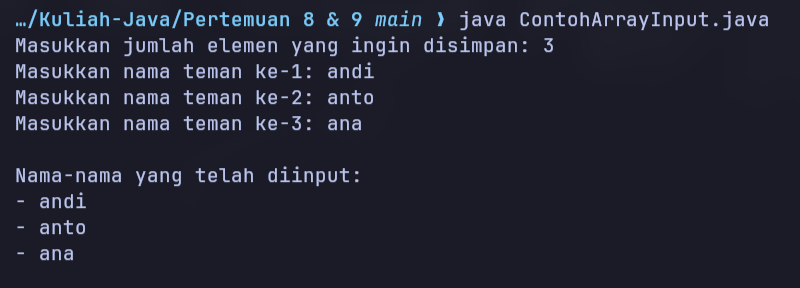
Penjelasan:

* Definisi Kelas & Metode: Mendefinisikan kelas ContohArrayDuaDimensi dan metode main sebagai titik awal eksekusi.
* Inisialisasi Array 2D: Membuat sebuah array dua dimensi (array di dalam array) String bernama koordinat yang berisi 3 baris, di mana setiap baris menyimpan pasangan kota dan negara.
* Akses Elemen Langsung: Mencetak satu elemen spesifik dari array, yaitu koordinat[1][1]. Ini mengakses elemen pada baris indeks 1 ("Tokyo", "Jepang") dan kolom indeks 1 ("Jepang").
* Iterasi dengan Nested Loop: Menggunakan dua loop for yang bersarang (luar dengan i untuk baris, dalam dengan j untuk kolom) untuk mengunjungi setiap elemen array.
* Cetak Berformat: Loop luar mencetak "Baris [i]: ". Loop dalam mencetak setiap elemen (kota dan negara) dan menambahkan pemisah " | " di antaranya. println() di akhir loop luar digunakan untuk pindah ke baris baru.

CODE: Pertemuan 8 & 9



OUTPUT:



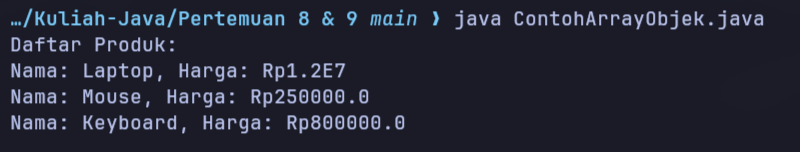
Penjelasan:

* Import & Setup: Mengimpor kelas Scanner, mendefinisikan kelas ContohArrayInput, dan membuat objek Scanner (scanner) untuk input.
* Tentukan Ukuran Array: Meminta pengguna memasukkan "jumlah elemen" dan menyimpan angka tersebut ke dalam variabel int ukuran.
* Inisialisasi Array: Membuat array String bernama namaTeman dengan besar sesuai nilai ukuran yang baru saja dimasukkan.
* Loop Input (Mengisi Array): Menggunakan for loop yang berjalan sebanyak ukuran kali (dari i=0 sampai ukuran-1) untuk meminta pengguna memasukkan nama, yang kemudian disimpan di namaTeman[i].
* Loop Output (Membaca Array): Menggunakan loop "for-each" untuk menelusuri setiap elemen (nama) di dalam array namaTeman dan mencetaknya ke konsol.
* Tutup Scanner: Menutup objek scanner untuk melepas sumber daya setelah selesai digunakan.

CODE: Pertemuan 8 & 9



OUTPUT:



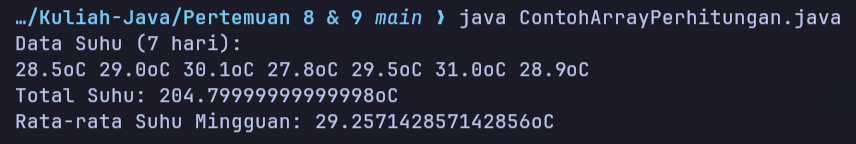
Penjelasan:

* Definisi Kelas Produk: Membuat blueprint (cetakan) bernama Produk yang memiliki dua atribut, yaitu String nama dan double harga.
* Constructor Produk: Kelas Produk memiliki constructor (baris 5-8) yang berfungsi untuk menerima nilai awal nama dan harga ketika sebuah objek Produk baru dibuat.
* Definisi Kelas Utama: Program utama berjalan di dalam kelas ContohArrayObjek dan metode main.
* Inisialisasi Array Objek: Membuat sebuah array bernama daftarProduk yang disiapkan untuk menampung 3 referensi ke objek Produk.
* Mengisi Array: Mengisi array pada indeks 0, 1, dan 2 dengan membuat tiga objek Produk baru (Laptop, Mouse, Keyboard) menggunakan constructor yang tadi telah didefinisikan.
* Iterasi dan Output: Menggunakan loop "for-each" untuk menelusuri setiap objek (p) di dalam daftarProduk, lalu mencetak atribut nama dan harga dari masing-masing objek tersebut.

CODE: Pertemuan 8 & 9



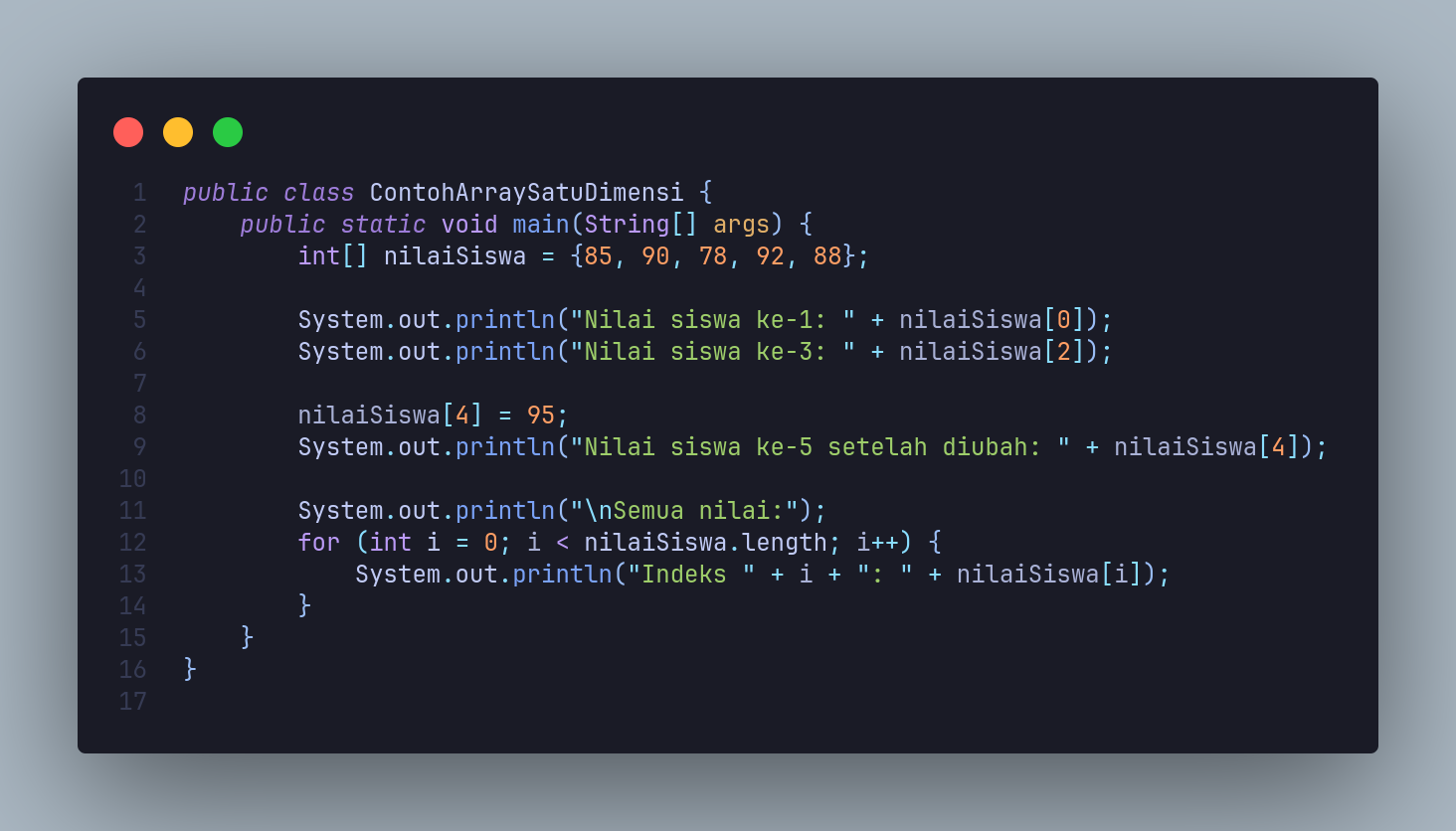
OUTPUT:



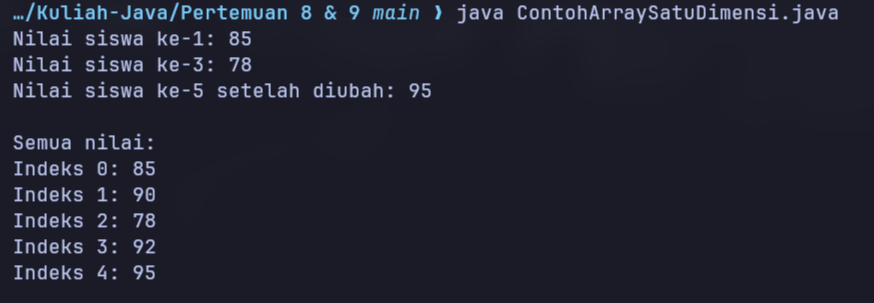
Penjelasan:

* Definisi Kelas & Metode: Mendefinisikan kelas ContohArrayPerhitungan dan metode main.
* Inisialisasi Array: Membuat array double bernama suhuMingguan yang berisi 7 data suhu (28.5, 29.0, dst.).
* Inisialisasi Total: Membuat variabel double totalSuhu dan memberinya nilai awal 0.0 untuk menampung penjumlahan.
* Loop Penjumlahan (For-Each): Menggunakan loop "for-each" untuk menelusuri setiap suhu dalam array suhuMingguan, lalu menambahkannya ke totalSuhu.
* Hitung Rata-rata: Menghitung rataRata dengan membagi totalSuhu dengan jumlah elemen dalam array (suhuMingguan.length).
* Cetak Detail Data: Mencetak jumlah data hari, lalu menggunakan loop "for-each" lagi untuk mencetak setiap nilai suhu individu.
* Cetak Hasil Akhir: Mencetak nilai akhir dari totalSuhu dan rataRata suhu mingguan.

CODE: Pertemuan 8 & 9



OUTPUT:



Penjelasan:

* Definisi Kelas & Metode: Mendefinisikan kelas ContohArraySatuDimensi dan metode main.
* Inisialisasi Array: Membuat array integer nilaiSiswa dengan 5 nilai awal: {85, 90, 78, 92, 88}. (Indeks 0 sampai 4).
* Akses Elemen: Mencetak nilai pada indeks 0 ("Nilai siswa ke-1: 85") dan indeks 2 ("Nilai siswa ke-3: 78").
* Modifikasi Elemen: Mengubah nilai pada indeks 4 (yang awalnya 88) menjadi 95.
* Akses Elemen (Setelah Ubah): Mencetak nilai baru pada indeks 4, yaitu "Nilai siswa ke-5 setelah diubah: 95".
* Iterasi Array: Menggunakan for loop (dengan i dari 0 sampai 4) untuk mencetak semua nilai dalam array satu per satu, beserta nomor indeksnya.