**PEMROGRAMAN MOBILE**

**“Pengantar Bahasa Pemrograman Dart - Bagian 3”**

Dosen Pengampu: Ade Ismail, S.Kom., MTI.



Oleh:

YUNIKA PUTERI DWI ANTIKA

2241760048 / SIB-3E

**PROGRAM STUDI SISTEM INFORMASI BISNIS**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2024**

**Praktikum 1: Eksperimen Tipe Data List**

|  |  |
| --- | --- |
| **Langkah** | **Jawaban/Deskripsi** |
| 1 | Menambahkan fungsi main kedalam kode |
| 2 | Meksekusi (Run) kode pada Langkah 1     * Kode di atas membuat sebuah daftar angka “[1, 2, 3]” dan memeriksa beberapa kondisi, seperti panjang daftar dan nilai elemen pada posisi tertentu. Setelah itu, kode mencetak panjang daftar dan nilai elemen di posisi ke-1. Kemudian, nilai di posisi ke-1 diubah menjadi “1”, dan hasilnya dicetak lagi. |
| 3 | Mengubah kode pada langkah 1 menjadi variabel final yang mempunyai index = 5 dengan default value = null. Serta, isi dari elemen index ke-1 dan ke-2 dengan nama dan NIM.    **Hasil**     * Pada kode ini, sebuah list dengan panjang 5 dibuat dan diisi dengan null sebagai nilai default. Kemudian, elemen pada indeks ke-1 diisi dengan nama lengkap, dan elemen pada indeks ke-2 diisi dengan NIM. Kode ini menggunakan assert untuk memastikan bahwa panjang list adalah 5 dan elemen pada indeks ke-1 serta ke-2 tidak null. Setelah itu, panjang list, nama, dan NIM dicetak. Kode ini berhasil menampilkan panjang list dan nilai pada indeks yang diisi tanpa masalah karena List<dynamic>.filled memungkinkan kita mengubah nilai-nilai di dalam list. |

**Praktikum 2: Eksperimen Tipe Data Set**

|  |  |
| --- | --- |
| **Langkah** | **Jawaban/Deskripsi** |
| 1 | Menambahkan fungsi main kedalam kode |
| 2 | Meksekusi (Run) kode pada Langkah 1     * Outputnya yaitu menampilkan kelima unsur halogen tersebut di dalam kurung kurawal `{}`. Kode ini membuat sebuah kumpulan (set) yang berisi lima unsur halogen yaitu “fluorine”, “chlorine”, “bromine”, “iodine”, dan “astatine”. Kumpulan (set) adalah jenis data yang menyimpan elemen-elemen unik tanpa urutan tertentu. Dan karena set tidak memiliki urutan tetap, elemen-elemen bisa muncul dalam urutan yang berbeda setiap kali program dijalankan. |
| 3 | Menambahkan elemen nama dan NIM pada kedua variabel Set tersebut dengan dua fungsi berbeda yaitu .add() dan .addAll(). Dan menghapus variabel Map.    **Hasil**     * Outpunya yaitu kedua “Set” tersebut menampilkan nama dan NIM tanpa urutan yang tetap. Pada kode ini, membuat dua variabel “Set”, yaitu “names1” dan “names2”, yang bisa menampung data tanpa duplikasi. Untuk menambahkan data, menggunakan dua cara berbeda yaitu menggunakan “.add()” untuk menambah satu elemen, dan “.addAll()” untuk menambah beberapa elemen sekaligus. Keduanya digunakan untuk memasukkan nama dan NIM ke dalam “Set”. |

**Praktikum 3: Eksperimen Tipe Data Maps**

|  |  |
| --- | --- |
| **Langkah** | **Jawaban/Deskripsi** |
| 1 | Menambahkan fungsi main kedalam kode |
| 2 | Meksekusi (Run) kode pada Langkah 1     * Outputnya yaitu mencetak seluruh isi dari kedua Map tersebut. Kode ini mendefinisikan dua Map di Dart, yaitu “gifts” dan “nobleGases”. Sebuah Map adalah koleksi pasangan key-value, di mana setiap key memiliki value yang sesuai. * gifts memiliki tiga pasangan yaitu “first” berasosiasi dengan “partridge”, “second” dengan “turtledoves”, dan “fifth” dengan angka 1. Di sini, key-nya adalah tipe String dan valuenya bisa berupa String atau int. * nobleGases juga memiliki tiga pasangan yaitu “2” berasosiasi dengan “helium”, “10” dengan “neon”, dan “18” dengan angka “2”. Di sini, key-nya adalah tipe int, dan valuenya bisa berupa String atau int. |
| 3 | Menambahkan elemen nama dan NIM pada tiap variabel kode (gifts, nobleGases, mhs1, dan mhs2).    **Hasil**     * Outputnya yaiut mencetak nilai dari masing-masing map. Pada kode Menambahkan dua map tambahan yaitu “mhs1” dan “mhs2”. Map “mhs1” merupakan map yang menyimpan informasi nama dan NIM, menggunakan “String” sebagai key. Sedangkan “mhs2” menyimpan informasi yang sama, tetapi menggunakan “int” sebagai key. |

**Praktikum 4: Eksperimen Tipe Data List: Spread dan Control-flow Operators**

|  |  |
| --- | --- |
| **Langkah** | **Jawaban/Deskripsi** |
| 1 | Menambahkan fungsi main kedalam kode |
| 2 | Meksekusi (Run) kode pada Langkah 1     * Outpunya yaitu mencetak isi dari kedua list dan juga menampilkan panjang dari “list2”, yang memiliki 4 elemen karena terdiri dari angka “0” dan elemen-elemen dari “list1”. Pada ini, terdapat dua list yaitu “list1” dan “list2”. “list1” berisi tiga elemen, yaitu “[1, 2, 3]”. “list2” didefinisikan dengan operator penyebaran “...”, yang digunakan untuk menyalin elemen dari “list1” ke dalam “list2”. Sebelum elemen “list1”, ada tambahan angka “0”, sehingga “list2” menjadi “[0, 1, 2, 3]”. |
| 3 | Menambahkan variabel list berisi NIM menggunakan Spread Operators.    **Hasil**     * Pada ini, “list1” berisi elemen “[1, 2, null]”, di mana “null” merupakan nilai kosong. Dengan menggunakan operator penyebaran “...?”, menyebarkan elemen dari “list1” ke dalam “list3”, di mana “null” diabaikan. Hasilnya adalah “list3” yang berisi “[0, 1, 2, null]”, dan panjangnya adalah 4. Selanjutnya, variabel “nim” berisi NIM sebagai list dengan satu elemen. Dalam “kombinasiList”, operator penyebaran digunakan lagi untuk menggabungkan elemen dari “list1”, “nim”, dan angka “0” menjadi satu list, yang mencakup semua elemen tanpa menghilangkan null. |
| 4 | Menambahkan kode program   1. Variabel promoActive ketika true     **Hasil**     1. Variabel promoActive ketika true     **Hasil**     * Pada kode ini, jika “promoActive” diset ke “true”, maka elemen “Outlet” akan ditambahkan ke dalam list “nav”, sehingga hasilnya adalah “['Home', 'Furniture', 'Plants', 'Outlet']”. Tetapi, jika “promoActive” diset ke “false”, elemen “Outlet” tidak akan ditambahkan, dan list “nav” hanya akan berisi “['Home', 'Furniture', 'Plants']”. Jadi, elemen “Outlet” hanya muncul di list jika kondisi “promoActive” terpenuhi (true). |
| 5 | Menambahkan kode program   1. Jika pengguna adalah Manager     **Hasil**     1. Jika pengguna adalah Staff     **Hasil**     * Jika variabel “login” diatur ke “Manager”, maka elemen “Inventory” akan ditambahkan ke dalam list “nav2”, menjadikannya “['Home', 'Furniture', 'Plants', 'Inventory']”. Tetapi, jika “login” diatur ke “Staff” atau nilai lain yang bukan “Manager”, elemen “Inventory” tidak akan ditambahkan, sehingga list “nav2” hanya berisi “['Home', 'Furniture', 'Plants']”. Jadi, hanya pengguna dengan status “Manager” yang akan melihat menu “Inventory” dalam navigasi. |
| 6 | Menambahkan kode program    **Hasil**     * Pada kode ini, membuat daftar “listOfStrings” yang berisi string yang dihasilkan dari setiap elemen dalam daftar “listOfInts”, seperti “#1”, “#2”, dan “#3”. Kode ini menggunakan **Collection For** untuk memudahkan pembuatan daftar baru dengan elemen yang dihasilkan dari iterasi melalui daftar lain. Dengan menggunakan **Collection For**, bisa dengan cepat dan efisien membuat daftar yang kompleks tanpa perlu menulis banyak kode tambahan. Sehingga, sangat berguna untuk memproses dan mengubah data dari koleksi lain, seperti mengonversi angka menjadi string atau menambahkan elemen berdasarkan kondisi tertentu. |

**Praktikum 5: Eksperimen Tipe Data Records**

|  |  |
| --- | --- |
| **Langkah** | **Jawaban/Deskripsi** |
| 1 | Menambahkan fungsi main kedalam kode |
| 2 | Meksekusi (Run) kode pada Langkah 1     * Pada kode ini, record adalah sebuah "tuple" (rekaman) yang berisi beberapa nilai dengan berbagai tipe data, termasuk label (a, b). Kita dapat mengakses elemen-elemen dalam tuple menggunakan urutan indeks atau label. Tuple ini memungkinkan untuk menggabungkan beberapa nilai berbeda dalam satu variabel tanpa memerlukan struktur data yang kompleks seperti class. |
| 3 | Menambahkan kode diluar scope void main()    **Hasil**     * Kode ini menukar dua nilai di dalam list. Sebelum pertukaran, list berisi dua angka “[100, 200]”. Fungsi “tukar()” mengambil dua elemen dalam list, kemudian menukarnya menggunakan variabel sementara “temp”. Setelah proses penukaran selesai, hasil baru ditampilkan sebagai “[200, 100]”. |
| 4 | Menambahkan kode didalam scope void main()    **Hasil**     * Kode ini mendeklarasikan variabel mahasiswa sebagai record yang terdiri dari dua field yaitu satu “String” untuk nama dan satu “int” untuk NIM. Setelah inisialisasi, variabel diisi dengan nama dan NIM, kemudian dicetak menggunakan “print()”. |
| 5 | Menambahkan kode diluar scope void main()    **Hasil**     * Kode ini membuat record “mahasiswa2” dengan empat field yaitu dua field tanpa nama (akses dengan “$1” dan “$2”) dan dua field bernama (“a” dan “b”). Field “$1” berisi nama saya, dan field a berisi NIM saya. Setiap field dicetak menggunakan print”()”. |

**Tugas Praktikum**

|  |  |
| --- | --- |
| **Soal** | **Jawaban/Deskripsi** |
| 2 | Jelaskan yang dimaksud Functions dalam bahasa Dart!  Jawaban:  Functions dalam bahasa Dart adalah blok kode yang dapat dipanggil untuk melakukan tugas tertentu. Functions membantu memecah kode menjadi bagian-bagian yang lebih kecil dan dapat digunakan kembali. Fungsi dapat menerima input (parameter) dan mengembalikan output (nilai). |
| 3 | Jelaskan jenis-jenis parameter di Functions beserta contoh sintaksnya!  Jawaban:  Dart memiliki beberapa jenis parameter dalam functions:   1. **Positional Parameters:** Parameter yang harus diberikan sesuai urutannya. Contoh:     **Hasil**     1. **Named Parameters:** Parameter yang dapat disebutkan dengan nama saat dipanggil. Contoh:     **Hasil**   1. **Optional Parameters:** Parameter yang tidak harus diisi saat pemanggilan. Contoh:     **Hasil** |
| 4 | Jelaskan maksud Functions sebagai first-class objects beserta contoh sintaknya!  Jawaban:  Dalam Dart, functions adalah first-class objects, artinya fungsi dapat disimpan dalam variabel, dipassing sebagai argumen, dan dikembalikan dari fungsi lain. Contoh:    **Hasil** |
| 5 | Apa itu Anonymous Functions? Jelaskan dan berikan contohnya!  Jawaban:  Anonymous functions (atau lambda expressions) adalah fungsi tanpa nama yang didefinisikan langsung di tempat penggunaannya. Contoh:    **Hasil** |
| 6 | Jelaskan perbedaan Lexical scope dan Lexical closures! Berikan contohnya!  Jawaban:   1. **Lexical Scope:** mengacu pada cara di mana variabel diakses berdasarkan lokasi mereka didefinisikan dalam kode. Artinya, variabel yang didefinisikan dalam satu scope (blok kode) hanya dapat diakses dalam scope tersebut dan sub-scope yang ada di dalamnya.     **Hasil**     1. Lexical Closures: Adalah fungsi yang "menutup" scope variabel yang ada di luar fungsi tersebut. Contoh:     **Hasil** |
| 7 | Jelaskan dengan contoh cara membuat return multiple value di Functions!  Jawaban:  Dart mendukung pengembalian beberapa nilai dengan menggunakan tuples (record) atau objek. Contoh dengan tuple:    **Hasil** |