

# **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	<p>Menambahkan fungsi main kedalam kode</p> <pre>Pertemuan 4 &gt; list.dart &gt; main Run   Debug 1 void main() { 2   var list = [1, 2, 3]; 3   assert(list.length == 3); 4   assert(list[1] == 2); 5   print(list.length); 6   print(list[1]); 7 8   list[1] = 1; 9   assert(list[1] == 1); 10  print(list[1]); 11 }</pre>
2	<p>Meksekusi (Run) kode pada Langkah 1</p> <pre>Connecting to VM Service at ws://127.0.0.1:52583/C_yVCKKqew=/ws Connected to the VM Service. 3 2 1 Exited.</pre> <p>➤ 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.</p>
3	<p>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.</p> <pre>final list = list&lt;dynamic&gt;.filled(5, null); // Membuat list mutable dengan panjang 5 list[1] = 'Yunika Puteri Dwi Antika'; // Mengisi elemen index ke-1 dengan nama list[2] = '2241760048'; // Mengisi elemen index ke-2 dengan NIM  assert(list.length == 5); // Memastikan panjang list adalah 5 assert(list[1] != null); // Memastikan elemen di index ke-1 sudah diisi assert(list[2] != null); // Memastikan elemen di index ke-2 sudah diisi  print(list.length); // Menampilkan panjang list print(list[1]); // Menampilkan elemen di index ke-1 (Nama) print(list[2]); // Menampilkan elemen di index ke-2 (NIM)</pre> <p><b>Hasil</b></p>

	<pre>Connecting to VM Service at ws://127.0.0.1:53986/rG5tw4A4CPI=/ws Connected to the VM Service. 5 Yunika 2241760048  Exited.</pre> <p>➤ 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 <code>List&lt;dynamic&gt;.filled</code> memungkinkan kita mengubah nilai-nilai di dalam list.</p>
--	--

## Praktikum 2: Eksperimen Tipe Data Set

Langkah	Jawaban/Deskripsi
1	<p>Menambahkan fungsi main kedalam kode</p> <pre>Pertemuan 4 &gt; set.dart &gt; ... Run   Debug 1 void main() { 2   var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine', 'astatine'}; 3   print(halogens); 4 } 5  </pre>
2	<pre>Connecting to VM Service at ws://127.0.0.1:51805/HOqMXE6Cj1M=/ws Connected to the VM Service. {fluorine, chlorine, bromine, iodine, astatine}  Exited.</pre> <p>➤ Outputnya yaitu menampilkan kelima unsur halogen tersebut di dalam kurung kurawal <code>{}</code>. 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</p>

	tidak memiliki urutan tetap, elemen-elemen bisa muncul dalam urutan yang berbeda setiap kali program dijalankan.
3	<p>Menambahkan elemen nama dan NIM pada kedua variabel Set tersebut dengan dua fungsi berbeda yaitu <code>.add()</code> dan <code>.addAll()</code>. Dan menghapus variabel Map.</p> <pre>//LANGKAH 3 // Membuat tiga variabel Set var names1 = &lt;String&gt;{}; Set&lt;String&gt; names2 = {}; // Ini juga bekerja sebagai Set // var names3 = {}; // Creates a map, not a set.  // Menambahkan nama dan NIM menggunakan .add() untuk names1 names1.add('Yunika Puteri Dwi Antika'); names1.add('2241760048');  // Menambahkan nama dan NIM menggunakan .addAll() untuk names2 names2.addAll({'Yunika Puteri Dwi Antika', '2241760048'});  // Mencetak kedua set print('\$names1'); print('\$names2');</pre> <p><b>Hasil</b></p> <pre>Connecting to VM Service at ws://127.0.0.1:52547/kNS_QMKJAKo=/ws Connected to the VM Service. {fluorine, chlorine, bromine, iodine, astatine} 2 {Yunika Puteri Dwi Antika, 2241760048}  Exited.</pre> <p>➤ 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 “<code>.add()</code>” untuk menambah satu elemen, dan “<code>.addAll()</code>” untuk menambah beberapa elemen sekaligus. Keduanya digunakan untuk memasukkan nama dan NIM ke dalam “Set”.</p>

### Praktikum 3: Eksperimen Tipe Data Maps

Langkah	Jawaban/Deskripsi
1	Menambahkan fungsi main kedalam kode

	<div> <div>Pertemuan 4 &gt; maps.dart &gt; main</div> <div>Run   Debug</div> <pre> 1 void main() { 2     var gifts = { 3         // Key:    Value 4         'first': 'partridge', 5         'second': 'turtledoves', 6         'fifth': 1 7     }; 8 9     var nobleGases = { 10        2: 'helium', 11        10: 'neon', 12        18: 2, 13    }; 14 15    print(gifts); 16    print(nobleGases); 17 } 18 </pre> </div>
2	<div> <div>Meksekusi (Run) kode pada Langkah 1</div> <div> <div>Connecting to VM Service at ws://127.0.0.1:53092/vtMBw_eQheU=/ws</div> <div>Connected to the VM Service.</div> <div>{first: partridge, second: turtledoves, fifth: 1}</div> <div>{2: helium, 10: neon, 18: 2}</div> <div>Exited.</div> </div> <div> <p>➤ 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.</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• 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.</li> </ul> </div> </div>
3	<div> <div>Menambahkan elemen nama dan NIM pada tiap variabel kode (gifts, nobleGases, mhs1, dan mhs2).</div> </div>

```
// LANGKAH 3
var mhs1 = Map<String, String>();
gifts['first'] = 'partridge';
gifts['second'] = 'turtledoves';
gifts['fifth'] = 'golden rings';
mhs1['Nama'] = 'Yunika Puteri Dwi Antika';
mhs1['NIM'] = '2241760048';

var mhs2 = Map<int, String>();
nobleGases[2] = 'helium';
nobleGases[10] = 'neon';
nobleGases[18] = 'argon';
mhs2[1] = 'Yunika Puteri Dwi Antika';
mhs2[6] = '2241760048';

print('Gifts: $gifts');
print('NobleGases 2: $nobleGases');
print('Mahasiswa 1: $mhs1');
print('Mahasiswa 2: $mhs2');
```

### Hasil



```
Connecting to VM Service at ws://127.0.0.1:53976/YlP7K9gSL0o=/ws
Connected to the VM Service.
{first: partridge, second: turtledoves, fifth: 1}
{2: helium, 10: neon, 18: 2}
Gifts: {first: partridge, second: turtledoves, fifth: golden rings}
NobleGases 2: {2: helium, 10: neon, 18: argon}
Mahasiswa 1: {Nama: Yunika Puteri Dwi Antika, NIM: 2241760048}
Mahasiswa 2: {1: Yunika Puteri Dwi Antika, 6: 2241760048}

Exited.
```

- Outputnya yaitu 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

	<div> <div>Pertemuan 4 &gt;  spread_controlFlow_operators.dart &gt;  main</div> <div>Run   Debug</div> <pre> 1 void main() { 2     var list1 = [1, 2, 3]; 3     var list2 = [0, ...list1]; 4     print(list1); 5     print(list2); 6     print(list2.length); 7 } 8 </pre> </div>
2	<div> <div>Meksekusi (Run) kode pada Langkah 1</div> <div> <pre> Connecting to VM Service at ws://127.0.0.1:54551/tmmOFkCh_tA=/ws Connected to the VM Service. [1, 2, 3] [0, 1, 2, 3] 4 Exited. </pre> </div> <div> <p>➤ 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]”.</p> </div> </div>
3	<div> <div>Menambahkan variabel list berisi NIM menggunakan Spread Operators.</div> <div> <pre> // LANGKAH 3 var list1 = [1, 2, null]; var list2 = [0, ...list1]; // Menggabungkan angka 0 dengan elemen dari list1 var list3 = [0, ...?list1]; // Menggabungkan angka 0 dengan elemen dari list1, mengabaikan null jika list1 adalah null var nim = ['2241760048']; var kombinasiList = [0, ...?list1, ...nim]; // Menggabungkan angka 0, elemen dari list1, dan NIM print(list1); print(list2); print(list2.length); print(list3.length); print(kombinasiList); </pre> </div> <div> <div>Hasil</div> <div> <pre> Connecting to VM Service at ws://127.0.0.1:55289/ZDuPV9M6hPM=/ws Connected to the VM Service. [1, 2, null] [0, 1, 2, null] 2 4 [0, 1, 2, null, 2241760048] Exited. </pre> </div> <div> <p>➤ 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</p> </div> </div> </div>

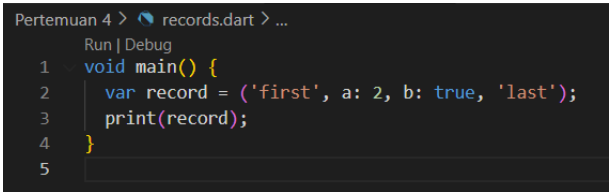
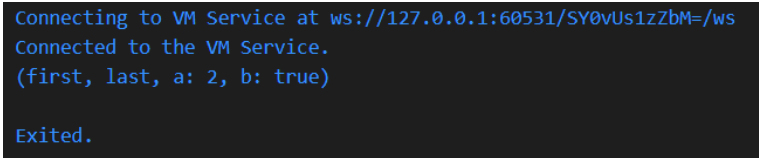
	<p>“null” diabaikan. Hasilnya adalah “list3” yang berisi “[0, 1, 2, null]”, dan panjangnya adalah 4. Selanjutnya, variabel “nim” berisi NIM sebagai list dengan satu elemen. Dalam “kombinasiList”, operator penyebaran digunakan lagi untuk menggabungkan elemen dari “list1”, “nim”, dan angka “0” menjadi satu list, yang mencakup semua elemen tanpa menghilangkan null.</p>
4	<p>Menambahkan kode program</p> <p>a. Variabel promoActive ketika true</p> <pre>// LANGKAH 4 // Variabel promoActive ketika true var promoActive = true; var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet']; print(nav);</pre> <p><b>Hasil</b></p> <pre>Connecting to VM Service at ws://127.0.0.1:55749/ysPZ1wubkzk=/ws Connected to the VM Service. [Home, Furniture, Plants, Outlet]  Exited.</pre> <p>b. Variabel promoActive ketika false</p> <pre>// Variabel promoActive ketika false var promoActive = false; var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet']; print(nav);</pre> <p><b>Hasil</b></p> <pre>Connecting to VM Service at ws://127.0.0.1:55853/4kF9Z2KwxEO=/ws Connected to the VM Service. [Home, Furniture, Plants]  Exited.</pre> <p>➤ 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).</p>
5	<p>Menambahkan kode program</p> <p>c. Jika pengguna adalah Manager</p>



	<pre data-bbox="475 197 1233 313">// LANGKAH 5 // Jika pengguna adalah Manager var login = 'Manager'; var nav2 = ['Home', 'Furniture', 'Plants', if (login == 'Manager') 'Inventory']; print(nav2);</pre> <p data-bbox="475 336 547 369"><b>Hasil</b></p> <pre data-bbox="475 392 1233 544">Connecting to VM Service at ws://127.0.0.1:56198/2LJhifo7y-M=/ws Connected to the VM Service. [Home, Furniture, Plants, Inventory]  Exited.</pre> <p data-bbox="427 566 823 600">d. Jika pengguna adalah Staff</p> <pre data-bbox="475 622 1233 716">// Jika pengguna adalah Staff var login = 'Staff'; var nav2 = ['Home', 'Furniture', 'Plants', if (login == 'Manager') 'Inventory']; print(nav2);</pre> <p data-bbox="475 734 547 768"><b>Hasil</b></p> <pre data-bbox="475 790 1233 943">Connecting to VM Service at ws://127.0.0.1:56333/VcLwoj6wYX0=/ws Connected to the VM Service. [Home, Furniture, Plants]  Exited.</pre> <p data-bbox="427 965 1377 1328">➤ 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.</p>
6	<p data-bbox="379 1361 756 1395">Menambahkan kode program</p> <pre data-bbox="379 1417 1137 1581">//LANGKAH 6 var listOfInts = [1, 2, 3]; var listOfStrings = ['#0', for (var i in listOfInts) '#\$i']; assert(listOfStrings[1] == '#1'); print(listOfStrings);</pre> <p data-bbox="379 1608 451 1641"><b>Hasil</b></p> <pre data-bbox="379 1664 1137 1827">Connecting to VM Service at ws://127.0.0.1:56580/eTkPmd6sfaU=/ws Connected to the VM Service. [#0, #1, #2, #3]  Exited.</pre> <p data-bbox="427 1861 1377 2007">➤ 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 <b>Collection For</b> untuk</p>

	<p>memudahkan pembuatan daftar baru dengan elemen yang dihasilkan dari iterasi melalui daftar lain. Dengan menggunakan <b>Collection For</b>, 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.</p>
--	---

## Praktikum 5: Eksperimen Tipe Data Records

Langkah	Jawaban/Deskripsi
1	<p>Menambahkan fungsi main kedalam kode</p>  <pre>Pertemuan 4 &gt; records.dart &gt; ... Run   Debug 1 void main() { 2   var record = ('first', a: 2, b: true, 'last'); 3   print(record); 4 } 5</pre>
2	<p>Meksekusi (Run) kode pada Langkah 1</p>  <pre>Connecting to VM Service at ws://127.0.0.1:60531/SY0vUs1zZbM=/ws Connected to the VM Service. (first, last, a: 2, b: true) Exited.</pre> <p>➤ 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.</p>
3	Menambahkan kode diluar scope void main()

```
// LANGKAH 3
(int, int) tukar((int, int) record) {
    var (a, b) = record;
    return (b, a); // Tukar posisi a dan b
}

Run | Debug
void main() {
    // Membuat record dengan dua nilai (x, y)
    var record = (100, 200);
    print('Sebelum ditukar: $record'); // Cetak record sebelum ditukar

    // Menggunakan fungsi tukar untuk menukar nilai record
    var recordTukar = tukar(record);
    print('Setelah ditukar: $recordTukar'); // Cetak record setelah ditukar
}
```

### Hasil

```
Connecting to VM Service at ws://127.0.0.1:61372/8Jcww2fT22s=/ws
Connected to the VM Service.
Sebelum ditukar: (100, 200)
Setelah ditukar: (200, 100)

Exited.
```

- 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()

```
// LANGKAH 4
Run | Debug
void main() {
    // Record type annotation in a variable declaration:
    (String, int) mahasiswa;

    // Inisialisasi nama dan NIM
    mahasiswa = ('Yunika Puteri Dwi Antika', 2241760048);

    // Mencetak nilai record mahasiswa
    print(mahasiswa);
}
```

### Hasil

```
Connecting to VM Service at ws://127.0.0.1:61839/MNd4RdMIUp0=/ws
Connected to the VM Service.
(Yunika Puteri Dwi Antika, 2241760048)

Exited.
```

- Kode ini mendeklarasikan variabel mahasiswa sebagai record yang terdiri dari dua field yaitu satu “String” untuk nama dan satu “int”

	<p>untuk NIM. Setelah inisialisasi, variabel diisi dengan nama dan NIM, kemudian dicetak menggunakan “print()”.</p>
5	<p>Menambahkan kode diluar scope void main()</p> <pre>// LANGKAH 5 Run   Debug void main() {     // Membuat record dengan beberapa field     var mahasiswa2 = ('Yunika Puteri Dwi Antika', a: 2241760048, b: true, 'last');      // Mencetak field dari record     print(mahasiswa2.\$1); // Prints 'Yunika Puteri Dwi Antika'     print(mahasiswa2.a); // Prints 2241760048     print(mahasiswa2.b); // Prints true     print(mahasiswa2.\$2); // Prints 'last' }</pre> <p><b>Hasil</b></p> <pre>Connecting to VM Service at ws://127.0.0.1:62051/QqJ8eNCYZbs=/ws Connected to the VM Service. Yunika Puteri Dwi Antika 2241760048 true last  Exited.</pre> <p>➤ 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”()”.</p>

## Tugas Praktikum

Soal	Jawaban/Deskripsi
2	<p>Jelaskan yang dimaksud Functions dalam bahasa Dart!</p> <p><b>Jawaban:</b></p> <p>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).</p>
3	<p>Jelaskan jenis-jenis parameter di Functions beserta contoh sintaksnya!</p> <p><b>Jawaban:</b></p>

Dart memiliki beberapa jenis parameter dalam functions:

- a. **Positional Parameters:** Parameter yang harus diberikan sesuai urutannya. Contoh:

```
// Soal 3a
void greet(String name) {
  print('Hello, $name');
}

Run | Debug
void main() {
  greet('Yunika'); // Memanggil fungsi greet dengan parameter 'Yunika'
}
```

### Hasil

```
Connecting to VM Service at ws://127.0.0.1:62764/heORwPerd4Y=/ws
Connected to the VM Service.
Hello, Yunika

Exited.
```

- b. **Named Parameters:** Parameter yang dapat disebutkan dengan nama saat dipanggil. Contoh:

```
// Soal 3b
void greet((required String name, required int age)) {
  print('Hello, $name, Age: $age');
}

Run | Debug
void main() {
  greet(name: 'Yunika', age: 20); // Memanggil fungsi greet dengan parameter nama dan umur
}
```

### Hasil

```
Connecting to VM Service at ws://127.0.0.1:62943/1z1DGs3e1KA=/ws
Connected to the VM Service.
Hello, Yunika, Age: 20

Exited.
```

- c. **Optional Parameters:** Parameter yang tidak harus diisi saat pemanggilan. Contoh:

```
// Soal 3c
void greet(String name, [int? age]) {
  print('Hello, $name, Age: $age');
}

Run | Debug
void main() {
  greet('Yunika', 20); // Memanggil dengan parameter age
  greet('Yunika');    // Memanggil tanpa parameter age
}
```

### Hasil

		<pre>Connecting to VM Service at ws://127.0.0.1:63082/V0d2cv5-vfo=/ws Connected to the VM Service. Hello, Yunika, Age: 20 Hello, Yunika, Age: null  Exited.</pre>
4	<p>Jelaskan maksud Functions sebagai first-class objects beserta contoh sintaknya!</p> <p><b>Jawaban:</b></p> <p>Dalam Dart, functions adalah first-class objects, artinya fungsi dapat disimpan dalam variabel, dipassing sebagai argumen, dan dikembalikan dari fungsi lain. Contoh:</p> <pre>// Soal 4 Run   Debug void main() {   var add = (int a, int b) =&gt; a + b;   print(add(3, 4)); // Output: 7 }</pre> <p><b>Hasil</b></p> <pre>Connecting to VM Service at ws://127.0.0.1:63174/GemoZjx8Ycg=/ws Connected to the VM Service. 7  Exited.</pre>	
5	<p>Apa itu Anonymous Functions? Jelaskan dan berikan contohnya!</p> <p><b>Jawaban:</b></p> <p>Anonymous functions (atau lambda expressions) adalah fungsi tanpa nama yang didefinisikan langsung di tempat penggunaannya. Contoh:</p> <pre>// Soal 5 Run   Debug void main() {   var list = [1, 2, 3];   var squared = list.map((x) =&gt; x * x);   print(squared.toList()); // Output: [1, 4, 9] }</pre> <p><b>Hasil</b></p> <pre>Connecting to VM Service at ws://127.0.0.1:63247/svFjT9QxUW4=/ws Connected to the VM Service. [1, 4, 9]  Exited.</pre>	
6	Jelaskan perbedaan Lexical scope dan Lexical closures! Berikan contohnya!	

### Jawaban:

- a. **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.

```
// Soal 6a
Run | Debug
void main() {
    var luarVariable = 'luar!';

    void dalamFunction() {
        var dalamVariable = 'dalam!';
        print(luarVariable); // Mengakses variabel dari luar scope
        print(dalamVariable); // Mengakses variabel dari dalam scope
    }

    dalamFunction();

    // print(dalamVariable); // Ini akan menyebabkan error karena dalamVariable tidak dapat diakses di luar dalamFunction
}
```

### Hasil

```
Connecting to VM Service at ws://127.0.0.1:63436/H0r-jni0Qns=/ws
Connected to the VM Service.
luar!
dalam!

Exited.
```

- b. **Lexical Closures:** Adalah fungsi yang "menutup" scope variabel yang ada di luar fungsi tersebut. Contoh:

```
// Soal 6b
Function makeCounter() {
    var count = 0;

    // Mengembalikan fungsi yang mengakses variabel 'count'
    return () {
        count += 1;
        return count;
    };
}

Run | Debug
void main() {
    var counter = makeCounter(); // Membuat counter yang mengingat variabel 'count'

    print(counter()); // Output: 1
    print(counter()); // Output: 2
    print(counter()); // Output: 3
}
```

### Hasil

```
Connecting to VM Service at ws://127.0.0.1:63516/U0wkq8KyXeI=/ws
Connected to the VM Service.
1
2
3

Exited.
```

### Jawaban:

Dart mendukung pengembalian beberapa nilai dengan menggunakan tuples (record) atau objek. Contoh dengan tuple:

```
// Soal 7
(int, String) getPersonInfo() {
  return (24, 'Yunika Puteri Dwi Antika');
}

Run | Debug
void main() {
  var info = getPersonInfo();
  print(
    'Age: ${info.$1}, Name: ${info.$2}'); // Output: Age: 24, Name: Yunika Puteri Dwi Antika
}
```

### Hasil

```
Connecting to VM Service at ws://127.0.0.1:63610/mDk-RASdJM4=/ws
Connected to the VM Service.
Age: 24, Name: Yunika Puteri Dwi Antika

Exited.
```