

Praktikum Pemrograman Berorientasi Object 1

LAPORAN PRAKTIKUM MATA KULIAH Pemrograman Berorientasi Object 1



Nama: Agung Bahtiar

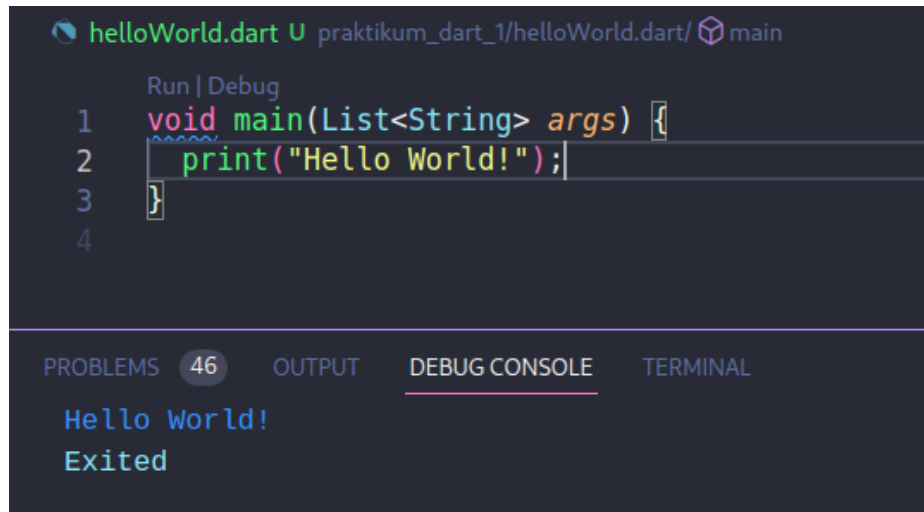
NIM: 362258302093

Kelas : 1A TRPL

**PROGRAM STUDI TEKNOLOGI REKAYASA PERANGKAT LUNAK
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BANYUWANGI
2022-2023**

1. Hello world

Program ini akan menampilkan output hello world yang didapatkan dari fungsi print();

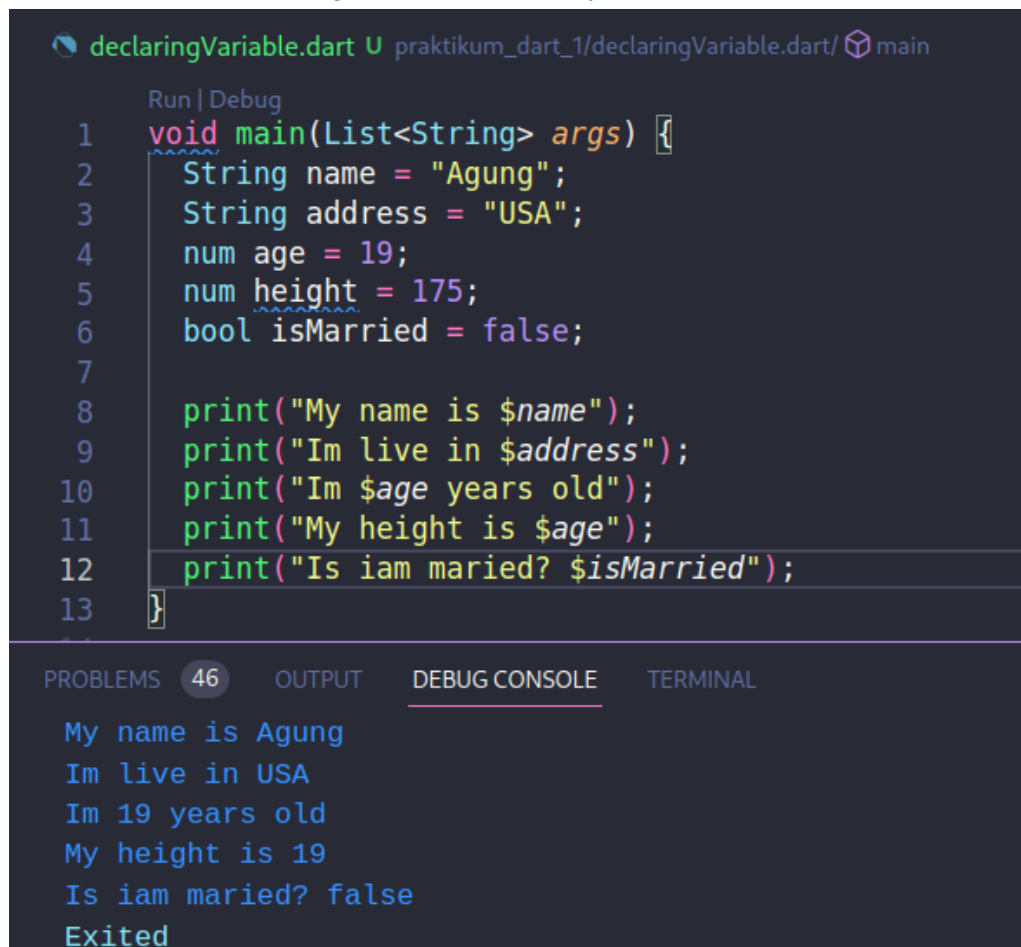


```
helloWorld.dart U praktikum_dart_1/helloWorld.dart/ main
Run | Debug
1 void main(List<String> args) {
2   print("Hello World!");
3 }
4

PROBLEMS 46 OUTPUT DEBUG CONSOLE TERMINAL
Hello World!
Exited
```

2. Declaring Variable

Untuk mendeklarasikan variabel di dart diawali dengan kata kunci tipe data apa yang akan dibuat dan dilanjutkan nama variable lalu isi dari variable, untuk menampilkan variabel bisa menggunakan tanda \$ diikuti dengan nama variabelnya.



```
declaringVariable.dart U praktikum_dart_1/declaringVariable.dart/ main
Run | Debug
1 void main(List<String> args) {
2   String name = "Agung";
3   String address = "USA";
4   num age = 19;
5   num height = 175;
6   bool isMarried = false;
7
8   print("My name is $name");
9   print("Im live in $address");
10  print("Im $age years old");
11  print("My height is $age");
12  print("Is iam married? $isMarried");
13 }

PROBLEMS 46 OUTPUT DEBUG CONSOLE TERMINAL
My name is Agung
Im live in USA
Im 19 years old
My height is 19
Is iam married? false
Exited
```

3. Kalkulasi matematika sederhana

Untuk melakukan kalkulasi sederhana pada dart bisa menggunakan operator aritmatika seperti pada program, + untuk tambah, - untuk kurang, * untuk perkalian dan / untuk pembagian.

```
kalukulasiMatematikaSederhana.dart U praktikum_dart_1/kalukulasiMatematikaSederhana.dart
Run | Debug
1 void main(List<String> args) {
2   int num1 = 10;
3   int num2 = 3;
4
5   int sum = num1 + num2;
6   int diff = num1 - num2;
7   int mul = num1 * num2;
8   double div = num1 / num2;
9
10  print("the sum is $sum");
11  print("The diff is $diff");
12  print("The mul is $mul");
13  print("The div is $div");
14 }
15

PROBLEMS 46 OUTPUT DEBUG CONSOLE TERMINAL
the sum is 13
The diff is 7
The mul is 30
The div is 3.3333333333333335
Exited
```

4. const

Const merupakan variabel yang bersifat immutable tidak dapat diubah lagi atau bernilai tetap, pada code di bawah saat variable pi diubah akan terjadi error.

```
const.dart U praktikum_dart_1/const.dart/ main
Run | Debug
1 void main(List<String> args) {
2   const pi = 3.14;
3   // pi = 4.23; Tidak bisa di ubah akan terjadi error
4   print("Value dari pi adalah $pi");
5 }
6

PROBLEMS 46 OUTPUT DEBUG CONSOLE TERMINAL
Value dari pi adalah 3.14
Exited
```

```
const.dart praktikum_dart_1 1
⊗ Constant variables can't be assigned a value. dart(assignment_to_const) [Ln 3, Col 3] ^
Try removing the assignment, or remove the modifier 'const' from the variable.
```

5. Koversi tipe data

a. String -> integer

Untuk merubah tipe data atau parsing dari string menjadi integer bisa menggunakan fungsi `parse()`;

```
konversiData.dart U praktikum_dart_1/konversiData.dart/ main
Run | Debug
1 void main(List<String> args) {
2   // String -> int
3   String strValue = "5";
4   int intValue = int.parse(strValue);
5   int hasilPenjumlahan = intValue + 10;
6   print("Hasil Penjumlahannya adalah $intValue");
7
8 }
9

PROBLEMS 48 OUTPUT DEBUG CONSOLE TERMINAL
Hasil Penjumlahannya adalah 5
Exited
```

b. Integer -> String

Untuk merubah tipe data dari integer ke string bisa menggunakan fungsi `.toString()`;

```
konversiData.dart U praktikum_dart_1/konversiData.dart/ main
/
8 // Int -> String
9 int oneInInteger = 1;
10 String oneInString = oneInInteger.toString();
11 print("Ini adalah variable $oneInString dalam string");
12 }
13

PROBLEMS 47 OUTPUT DEBUG CONSOLE TERMINAL
Ini adalah variable 1 dalam string
Exited
```

6. User Input

a. Input String

Sebelum Melakukan io dengan dart pastikan untuk sudah mengimport library yang sudah di butuhkan yaitu dart:io, inputan pada dart diterima dengan stdin.readLineSync(); yang bertipe data string.

```
userInput.dart U praktikum_dart_1/userInput.dart/ main
1  import 'dart:io';
2
3  void main(List<String> args) {
4    // Input String
5    print("Enter Your Name: ");
6    String? name = stdin.readLineSync();
7    print("Halo ${name}");
8  }
```

PROBLEMS 50 OUTPUT DEBUG CONSOLE TERMINAL

```
agungb@CatD ~/c/b/praktikum_dart_1 (master)> dart run userInput.dart
Enter Your Name:
Agung
Halo Agung
agungb@CatD ~/c/b/praktikum_dart_1 (master)>
```

b. Input Integer

Karna default tipe data yang diterima input adalah string maka untuk merubahnya ke int harus di parsing terlebih dahulu dengan fungsi parse();

```
userInput.dart U praktikum_dart_1/userInput.dart/ main
1  import 'dart:io';
2
3  void main(List<String> args) {
4    print("Enter Your age");
5    int? age = int.parse(stdin.readLineSync()!);
6    print("Your Age is $age");
7  }
```

PROBLEMS 49 OUTPUT DEBUG CONSOLE TERMINAL

```
agungb@CatD ~/c/b/praktikum_dart_1 (master)> dart run userInput.dart
Enter Your age
19
Your Age is 19
agungb@CatD ~/c/b/praktikum_dart_1 (master)>
```

7. String manipulation

a. Bentuk Penulisan string

String dapat ditulis dalam petik 1 maupun 2, dan untuk menulis string dalam beberapa baris bisa menggunakan petik 2 3x.

```
stringManipulation.dart U praktikum_dart_1/stringManipulation.dart/ main
Run | Debug
1 void main(List<String> args) {
2   String text1 = 'Single line string';
3   String text2 = "Single line string with double quotes";
4   String text3 = """ This is
5     multiline
6     String
7     """;
8   print(text1);
9   print(text2);
10  print(text3);
11 }
```

PROBLEMS 51 OUTPUT DEBUG CONSOLE TERMINAL

Single line string
Single line string with double quotes
This is
multiline
String

Exited

b. Menggabungkan String

Untuk menggabungkan string bisa menggunakan operator + dengan syarat ke 2 nya harus bertipe data string.

```
stringManipulation.dart U praktikum_dart_1/stringManipulation.dart/ main
Run | Debug
1 void main(List<String> args) {}
2 // Menggabungkan String
3 String firstName = "Agung ";
4 String lastName = "Bahtiar";
5 print("Nama lengkap saya adalah" + firstName + lastName);
6 }
7
```

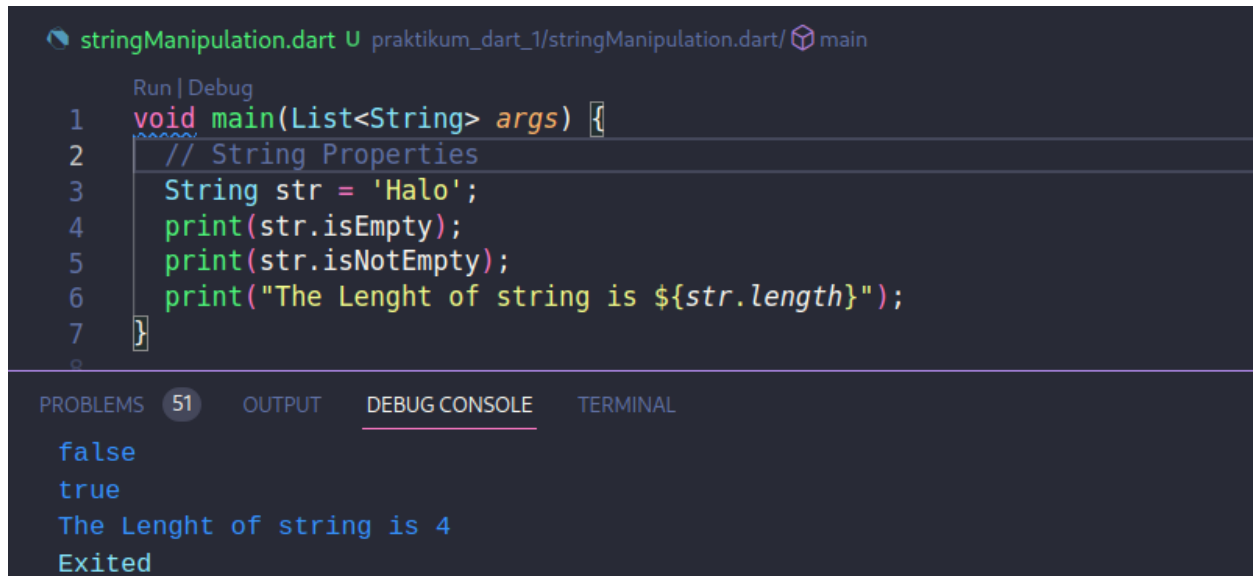
PROBLEMS 52 OUTPUT DEBUG CONSOLE TERMINAL

Nama lengkap saya adalahAgung Bahtiar

Exited

c. String Properties

String mempunyai beberapa fungsi yang bisa digunakan seperti isEmpty untuk mengecek apakah variable string tersebut kosong, isEmpty untuk mengecek apakah tidak kosong dan .length untuk mengecek panjang karakter dari string.



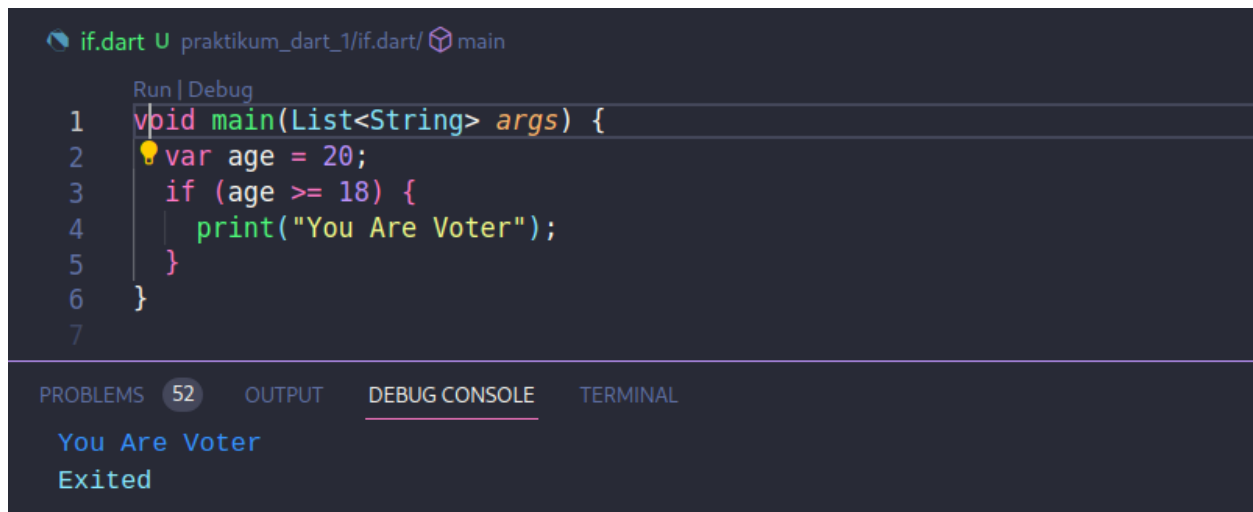
```
stringManipulation.dart U praktikum_dart_1/stringManipulation.dart/ main
Run | Debug
1 void main(List<String> args) {
2   // String Properties
3   String str = 'Halo';
4   print(str.isEmpty);
5   print(str.isNotEmpty);
6   print("The Lenght of string is ${str.length}");
7 }
```

PROBLEMS 51 OUTPUT DEBUG CONSOLE TERMINAL

false
true
The Lenght of string is 4
Exited

8. IF

Pada percabangan akan dilakukan pengecekan sesuai kondisi yang sudah ditentukan jika true maka akan menjalankan baris programnya



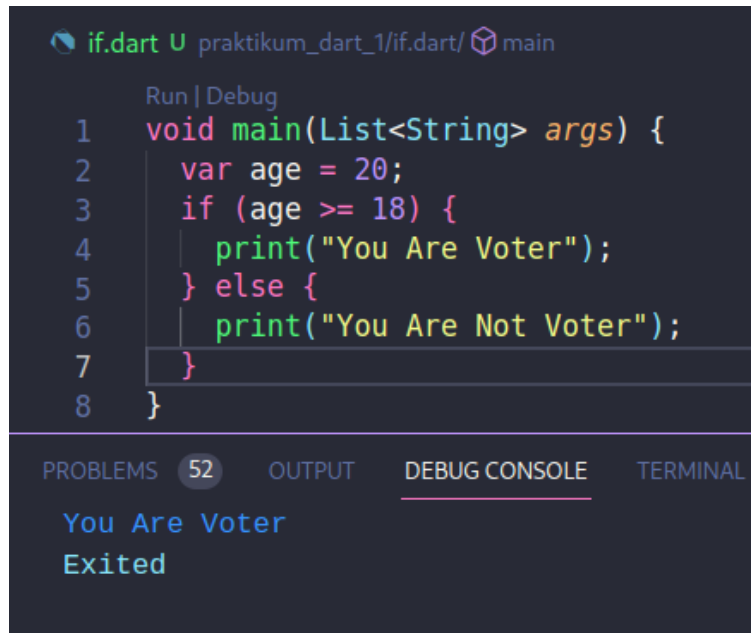
```
if.dart U praktikum_dart_1/if.dart/ main
Run | Debug
1 void main(List<String> args) {
2   var age = 20;
3   if (age >= 18) {
4     print("You Are Voter");
5   }
6 }
7
```

PROBLEMS 52 OUTPUT DEBUG CONSOLE TERMINAL

You Are Voter
Exited

9. IF-ELSE

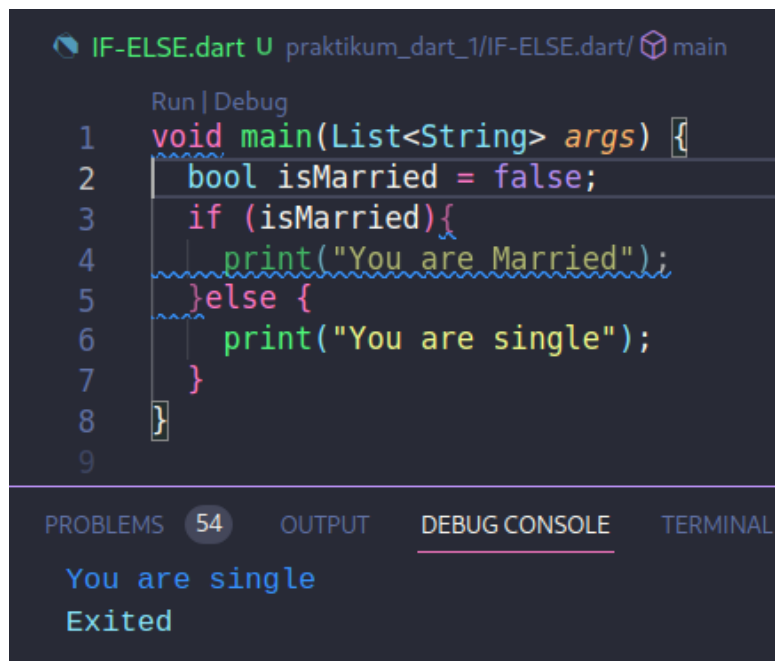
Pada percabangan akan dilakukan pengecekan sesuai kondisi yang sudah ditentukan jika true maka akan menjalankan baris programnya dan jika false akan menjalankan program yang sudah ditentukan di blok else.



```
if.dart U praktikum_dart_1/if.dart/ main
Run | Debug
1 void main(List<String> args) {
2   var age = 20;
3   if (age >= 18) {
4     print("You Are Voter");
5   } else {
6     print("You Are Not Voter");
7   }
8 }
```

PROBLEMS 52 OUTPUT DEBUG CONSOLE TERMINAL

You Are Voter
Exited



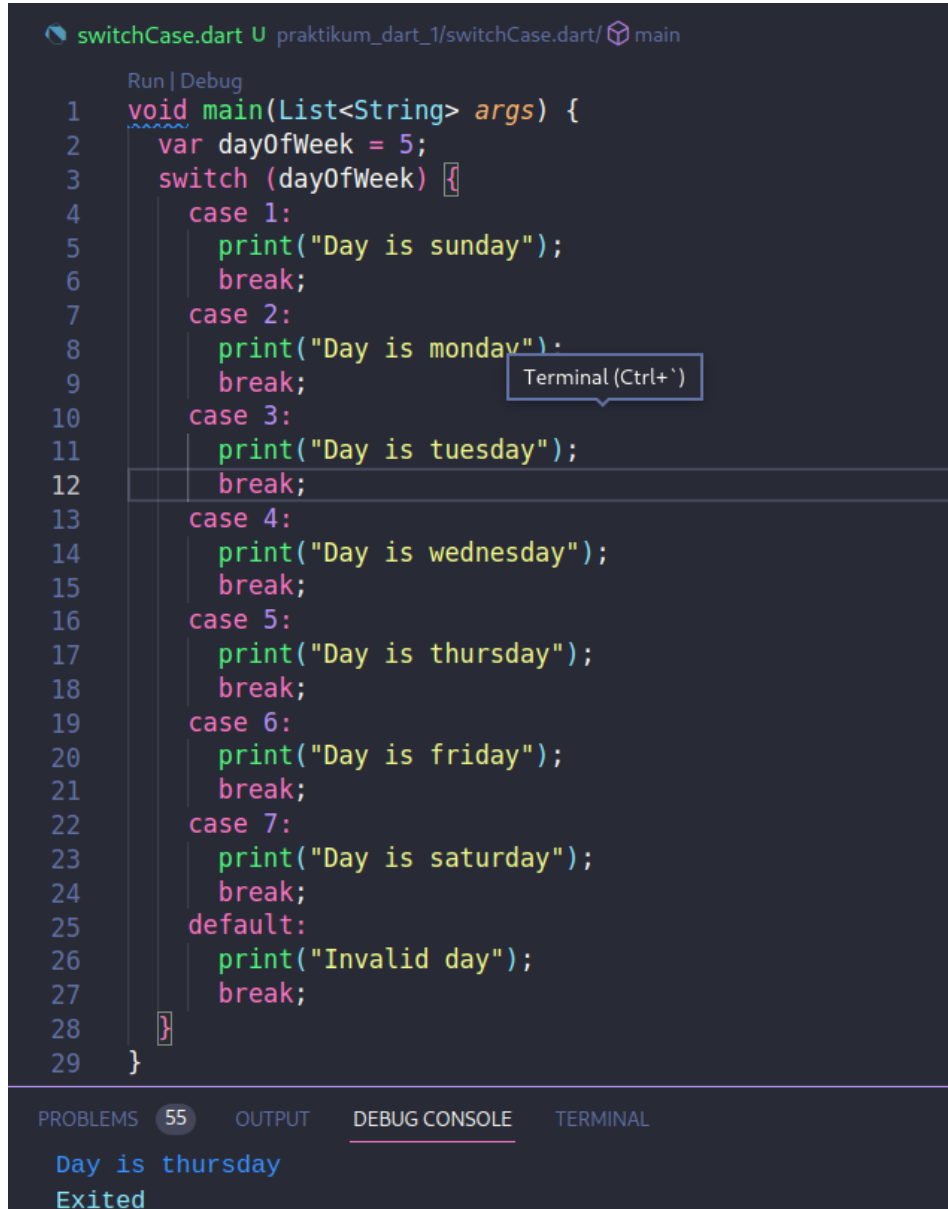
```
IF-ELSE.dart U praktikum_dart_1/IF-ELSE.dart/ main
Run | Debug
1 void main(List<String> args) {
2   bool isMarried = false;
3   if (isMarried){
4     print("You are Married");
5   }else {
6     print("You are single");
7   }
8 }
9
```

PROBLEMS 54 OUTPUT DEBUG CONSOLE TERMINAL

You are single
Exited

10. Switch

Switch sama dengan percabangan if-else, hanya saja penulisannya berbeda, pada switch kondisi akan di cek sesuai dengan case yang sudah ditentukan, jika sudah sesuai maka program dalam blok case tersebut akan dijalankan lalu di break untuk menghentikan flow yang sedang berjalan.



```
switchCase.dart U praktikum_dart_1/switchCase.dart/ main
Run | Debug
1 void main(List<String> args) {
2   var dayOfWeek = 5;
3   switch (dayOfWeek) {
4     case 1:
5       print("Day is sunday");
6       break;
7     case 2:
8       print("Day is monday");
9       break;
10    case 3:
11      print("Day is tuesday");
12      break;
13    case 4:
14      print("Day is wednesday");
15      break;
16    case 5:
17      print("Day is thursday");
18      break;
19    case 6:
20      print("Day is friday");
21      break;
22    case 7:
23      print("Day is saturday");
24      break;
25    default:
26      print("Invalid day");
27      break;
28  }
29 }
```

Terminal (Ctrl+')

PROBLEMS 55 OUTPUT DEBUG CONSOLE TERMINAL

Day is thursday
Exited

11. Ternary Operator

Sama dengan percabangan lainnya ada kondisi yang harus dipenuhi dahulu jika bernilai true maka bagian kiri tanda : yang dijalankan dan sebaliknya.

```
ternaryOperator.dart U praktikum_dart_1/ternaryOperator.dart/ main
Run | Debug
1 void main(List<String> args) {
2   int num1 = 10;
3   int num2 = 15;
4   int max = (num1 > num2) ? num1 : num2;
5   print("The greatest number is $max");
6 }

PROBLEMS 56 OUTPUT DEBUG CONSOLE TERMINAL
The greatest number is 15
Exited
```

12. For-Loop

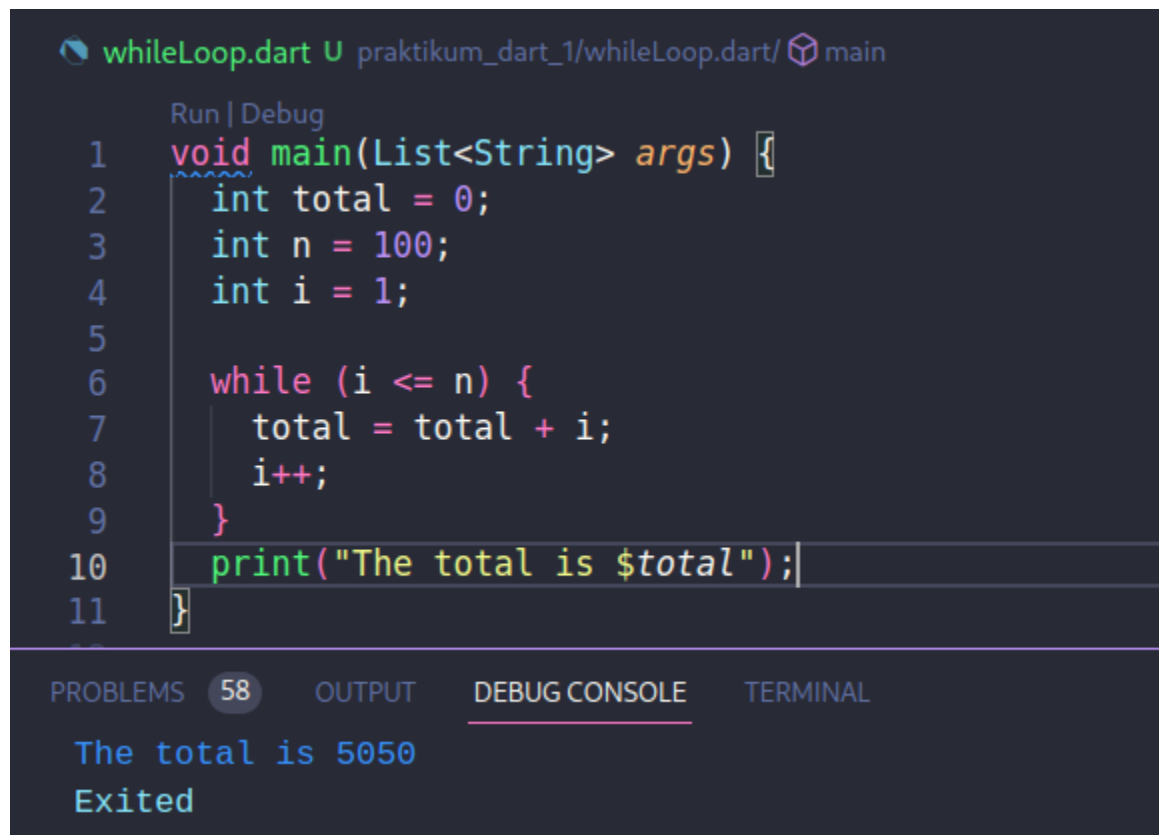
Perulangan for dijalankan dengan inisialisasi sebuah variabel terlebih dahulu yang nantinya akan dilakukan pengecekan kondisi jika sesuai maka baris kode akan dijalankan dan variable awal akan di increment/decrement, proses ini akan berlangsung terus menerus sampai kondisi bernilai false.

```
forLoop.dart U praktikum_dart_1/forLoop.dart/ main
Run | Debug
1 void main(List<String> args) {
2   for (var i = 0; i < 10; i++) {
3     print(i);
4   }
5 }
6

PROBLEMS 57 OUTPUT DEBUG CONSOLE TERMINAL
0
1
2
3
4
5
6
7
8
9
Exited
```

13. While Loop

Saat variable *i* kurang dari variable *n*, isi variabel *total* dengan variable *total* ditambah *i* dan lakukan increment ke variable *i*, saat kondisi bernilai false tampilkan nilai dari variabel *total*.



```
whileLoop.dart U praktikum_dart_1/whileLoop.dart/ main
Run | Debug
1 void main(List<String> args) {
2   int total = 0;
3   int n = 100;
4   int i = 1;
5
6   while (i <= n) {
7     total = total + i;
8     i++;
9   }
10  print("The total is $total");
11 }
```

PROBLEMS 58 OUTPUT DEBUG CONSOLE TERMINAL

The total is 5050
Exited

14. Do While

Pada do while loop, code yang ada pada block do akan dijalankan terlebih dahulu lalu di lanjutkan pengecekan kondisinya pada while yang akan di ulang sampai kondisi tidak terpenuhi lagi.



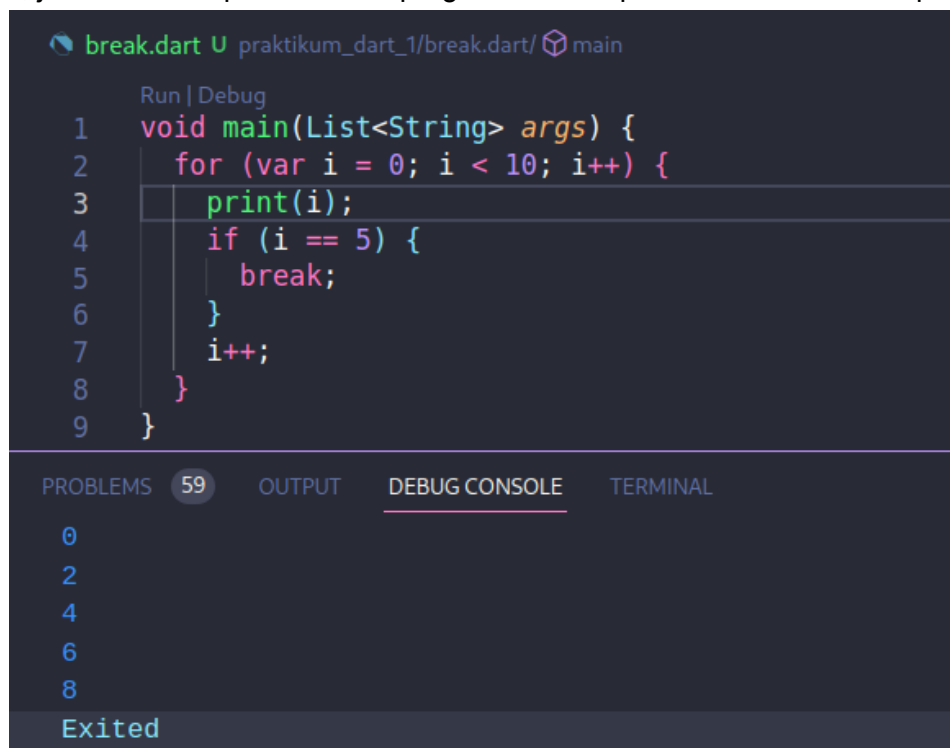
```
doWhile.dart U praktikum_dart_1/doWhile.dart/ main
Run | Debug
1 void main(List<String> args) {
2   int i = 1;
3   do {
4     print(i);
5     i++;
6   } while (i <= 5);
7 }
```

PROBLEMS 59 OUTPUT DEBUG CONSOLE TERMINAL

1
2
3
4
5
Exited

15. Break

For loop akan di lakukan seperti biasa hanya saja setiap perulangan akan dilakukan pengecekan, jika kondisi terpenuhi maka program akan di paksa berhenti dari loop.



```
break.dart U praktikum_dart_1/break.dart/ main
Run | Debug
1 void main(List<String> args) {
2   for (var i = 0; i < 10; i++) {
3     print(i);
4     if (i == 5) {
5       break;
6     }
7     i++;
8   }
9 }
```

PROBLEMS 59 OUTPUT DEBUG CONSOLE TERMINAL

0
2
4
6
8
Exited

16. Continue

For loop akan di lakukan seperti biasa hanya saja setiap perulangan akan dilakukan pengecekan, jika kondisi terpenuhi maka program di bawah continue akan dilewati dan langsung menjalankan perulangan selanjutnya jika ada.



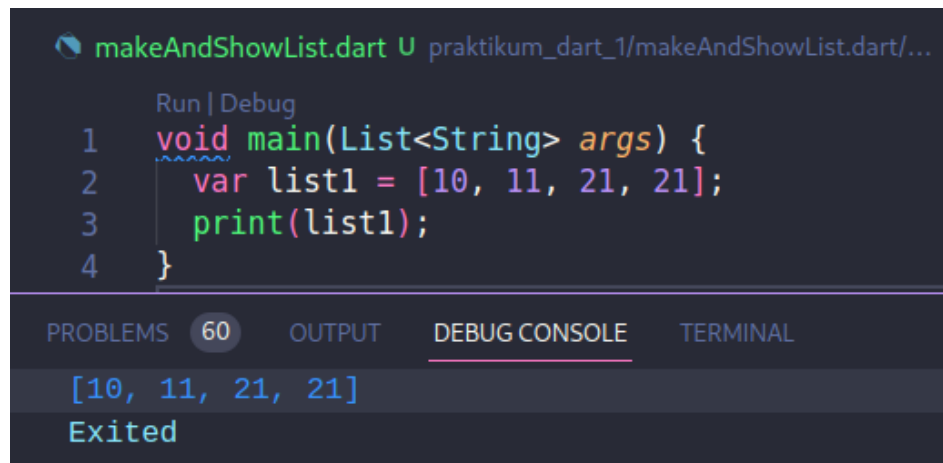
```
continue.dart U praktikum_dart_1/continue.dart/ main

Run | Debug
1 void main(List<String> args) {
2   for (var i = 0; i < 10; i++) {
3     if (i == 5) {
4       continue;
5     }
6     print(i);
7   }
8 }
9

PROBLEMS 59 OUTPUT DEBUG CONSOLE TERMINAL
0
1
2
3
4
6
7
8
9
Exited
```

17. Membuat dan menampilkan list

Untuk membuat list seperti variable hanya saja value yang ada pada list di apit dengan []
Untuk menampilkan bisa menggunakan print() diisi dengan nama listnya.



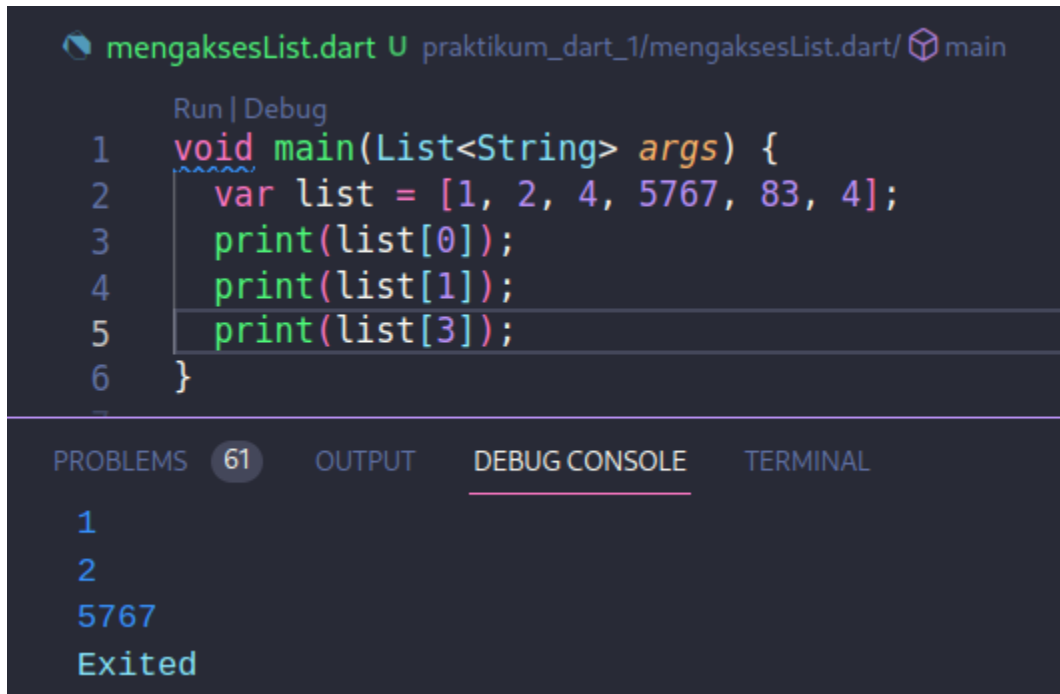
```
makeAndShowList.dart U praktikum_dart_1/makeAndShowList.dart/...

Run | Debug
1 void main(List<String> args) {
2   var list1 = [10, 11, 21, 21];
3   print(list1);
4 }

PROBLEMS 60 OUTPUT DEBUG CONSOLE TERMINAL
[10, 11, 21, 21]
Exited
```

18. Mengakses list

List diakses menggunakan nomor index, yang di mulai dari 0



The screenshot shows an IDE window with a file named `mengaksesList.dart`. The code defines a `main` function that creates a list of integers and prints elements at specific indices. The output console shows the values 1, 2, and 5767, followed by 'Exited'.

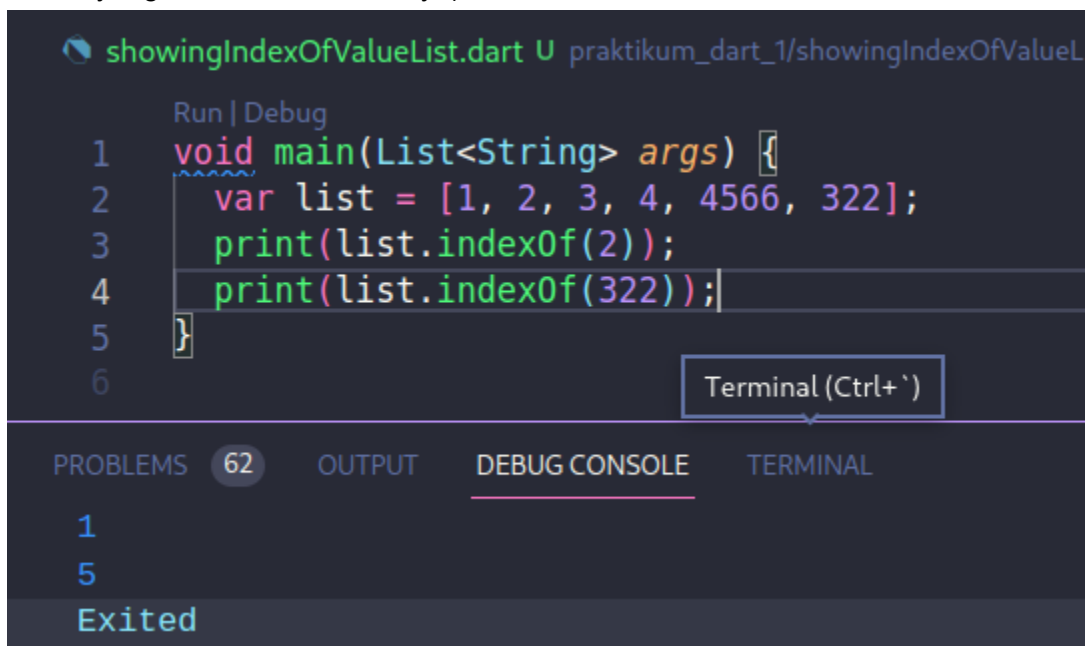
```
Run | Debug
1 void main(List<String> args) {
2   var list = [1, 2, 4, 5767, 83, 4];
3   print(list[0]);
4   print(list[1]);
5   print(list[3]);
6 }
```

PROBLEMS 61 OUTPUT DEBUG CONSOLE TERMINAL

1
2
5767
Exited

19. Menampilkan index sebuah nilai pada list

Untuk menampilkan index pada list bisa menggunakan fungsi `.indexOf`(diisi dengan value yang mau di lihat index nya);



The screenshot shows an IDE window with a file named `showingIndexOfValueList.dart`. The code defines a `main` function that creates a list of integers and prints the indices of the values 2 and 322 using the `indexOf` method. The output console shows the indices 1 and 5, followed by 'Exited'.

```
Run | Debug
1 void main(List<String> args) {
2   var list = [1, 2, 3, 4, 4566, 322];
3   print(list.indexOf(2));
4   print(list.indexOf(322));
5 }
6
```

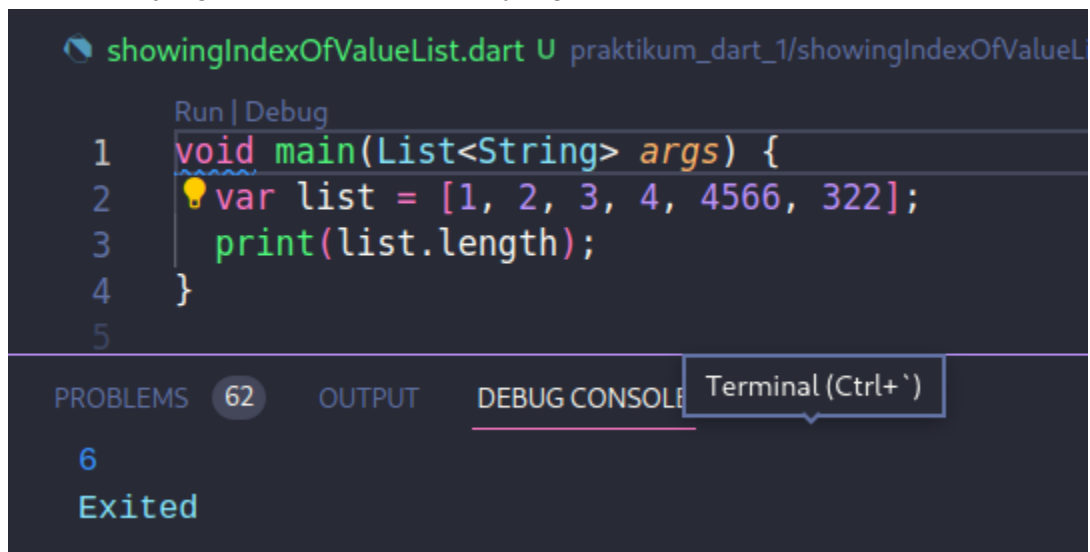
PROBLEMS 62 OUTPUT DEBUG CONSOLE TERMINAL

1
5
Exited

Terminal (Ctrl+ `)

20. Menampilkan panjang sebuah list

Untuk menampilkan panjang dari sebuah list bisa menggunakan fungsi `.length` yang akan menampilkan panjang dari seluruh index list yang ada.



The screenshot shows an IDE window with a file named `showingIndexOfValueList.dart`. The code is as follows:

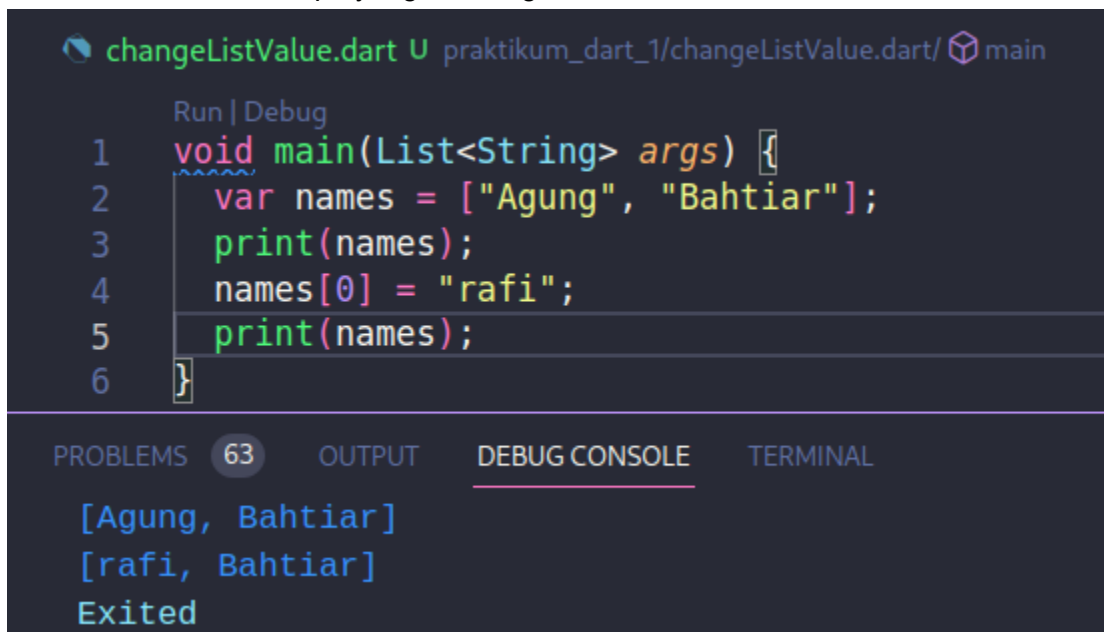
```
1 void main(List<String> args) {  
2   var list = [1, 2, 3, 4, 4566, 322];  
3   print(list.length);  
4 }  
5
```

Below the code editor, the **DEBUG CONSOLE** tab is active, showing the output:

```
6  
Exited
```

21. Mengganti nilai pada list

Untuk mengganti value pada list bisa dilakukan seperti pada variable hanya saja tentukan terlebih dahulu index ke berapa yang akan diganti



The screenshot shows an IDE window with a file named `changeListValue.dart`. The code is as follows:

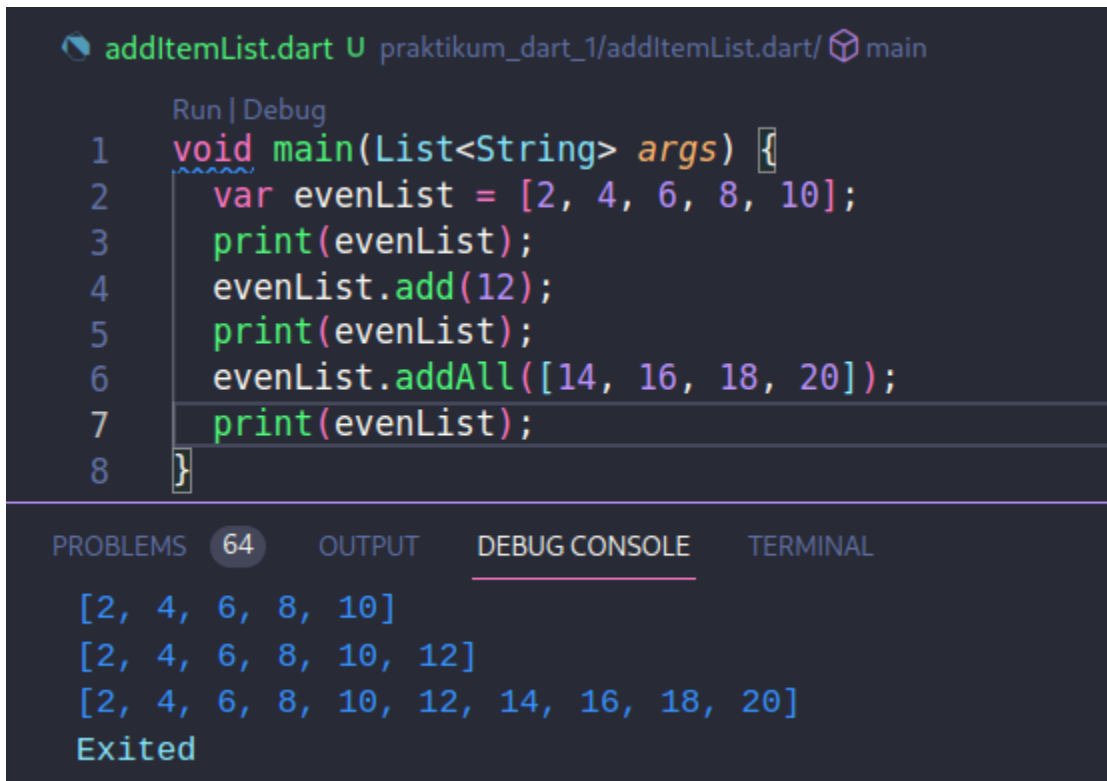
```
1 void main(List<String> args) {  
2   var names = ["Agung", "Bahtiar"];  
3   print(names);  
4   names[0] = "rafi";  
5   print(names);  
6 }
```

Below the code editor, the **DEBUG CONSOLE** tab is active, showing the output:

```
[Agung, Bahtiar]  
[rafi, Bahtiar]  
Exited
```

22. Menambahkan nilai pada list

Untuk menambahkan value baru ke list bisa menggunakan fungsi `.add` diisi dengan value yang akan ditambahkan, dan bisa juga menambahkan list yang sudah ada dengan fungsi `.addAll` diikuti isi dari list yang akan ditambahkan



The screenshot shows an IDE window with a file named `addItemList.dart` and a `main` function. The code defines a list `evenList` with even numbers from 2 to 10. It then adds the number 12 to the list and prints it. Finally, it adds another list `[14, 16, 18, 20]` to `evenList` and prints the updated list.

```
addItemList.dart U praktikum_dart_1/addItemList.dart/ main
Run | Debug
1 void main(List<String> args) {
2   var evenList = [2, 4, 6, 8, 10];
3   print(evenList);
4   evenList.add(12);
5   print(evenList);
6   evenList.addAll([14, 16, 18, 20]);
7   print(evenList);
8 }
```

The output panel shows the following results:

```
[2, 4, 6, 8, 10]
[2, 4, 6, 8, 10, 12]
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
Exited
```


23. Menyisipkan nilai pada list

Untuk menyisipkan nilai pada list menggunakan fungsi `.insert` yang diisi dengan no index beberapa yang akan disisipkan dan diikuti isi dari valuenya

```
listInsert.dart U praktikum_dart_1/listInsert.dart/ main
Run | Debug
1 void main(List<String> args) {
2   var myList = [1, 2, 3, 4, 5, 6, 7, 8, 9];
3   print(myList);
4   myList.insert(5, 51); // Index , Element
5   print(myList);
6 }
7
```

PROBLEMS	65	OUTPUT	DEBUG CONSOLE	TERMINAL
			[1, 2, 3, 4, 5, 6, 7, 8, 9]	
			[1, 2, 3, 4, 5, 51, 6, 7, 8, 9]	
			Exited	

24. Menghapus item dari list

Menghapus item dari list menggunakan fungsi `removeAt()` yang diisi dengan no index value yang akan di hapus

```
listRemoveItem.dart U praktikum_dart_1/listRemoveItem.dart/ main
Run | Debug
1 void main(List<String> args) {
2   var myList = [2, 4, 6, 8, 10];
3   print("List Sebelum di hapus $myList");
4   myList.removeAt(0);
5   print("List Setelah di hapus $myList");
6 }
7
```

PROBLEMS	66	OUTPUT	DEBUG CONSOLE	TERMINAL
			List Sebelum di hapus [2, 4, 6, 8, 10]	
			List Setelah di hapus [4, 6, 8, 10]	
			Exited	

25. Loop dalam list

Menampilkan isi dari list satu persatu bisa menggunakan perulangan bisa namun ada cara yang lebih efektif menggunakan for in, buat variabel untuk menerima element list dan masukkan list yang akan ditampilkan



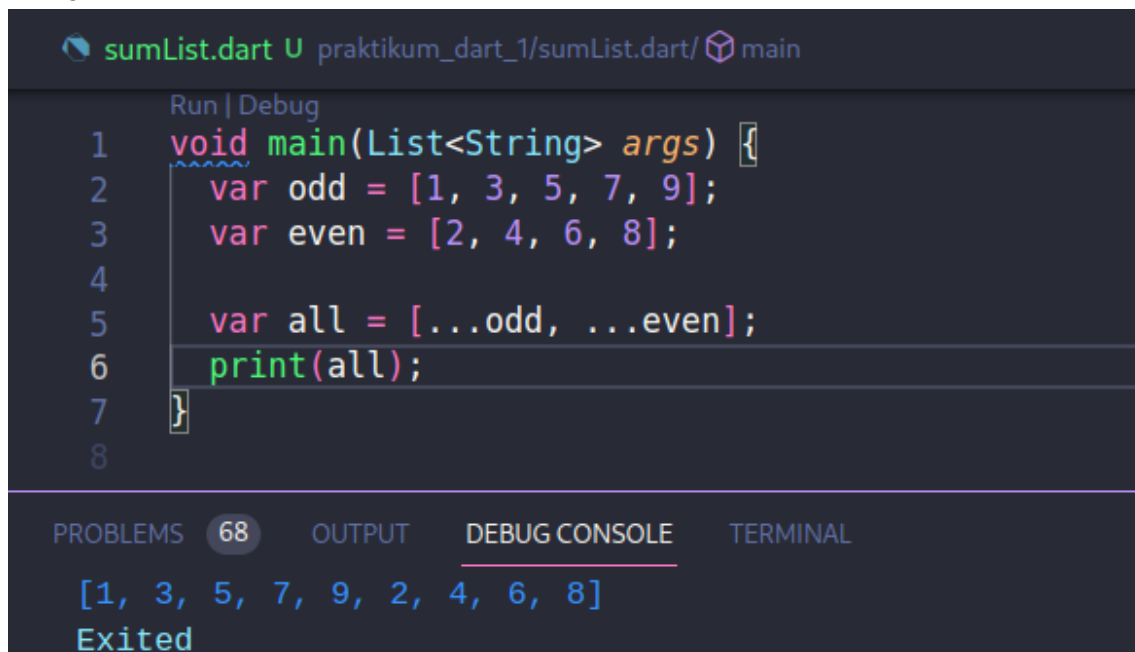
```
loopInList.dart U praktikum_dart_1/loopInList.dart/...  
Run | Debug  
1 void main(List<String> args) {  
2   var friend = ['agung', 'rizal', 'yogi', 'falen']  
3  
4   for (String names in friend) {  
5     print(names);  
6   }  
7 }  
8
```

PROBLEMS 67 OUTPUT DEBUG CONSOLE TERMINAL

agung
rizal
yogi
falen
Exited

26. Menggabungkan list

Menggabungkan 2 list bisa menggunakan ternary operator yaitu ...didepan nama list yang akan di gabungkan



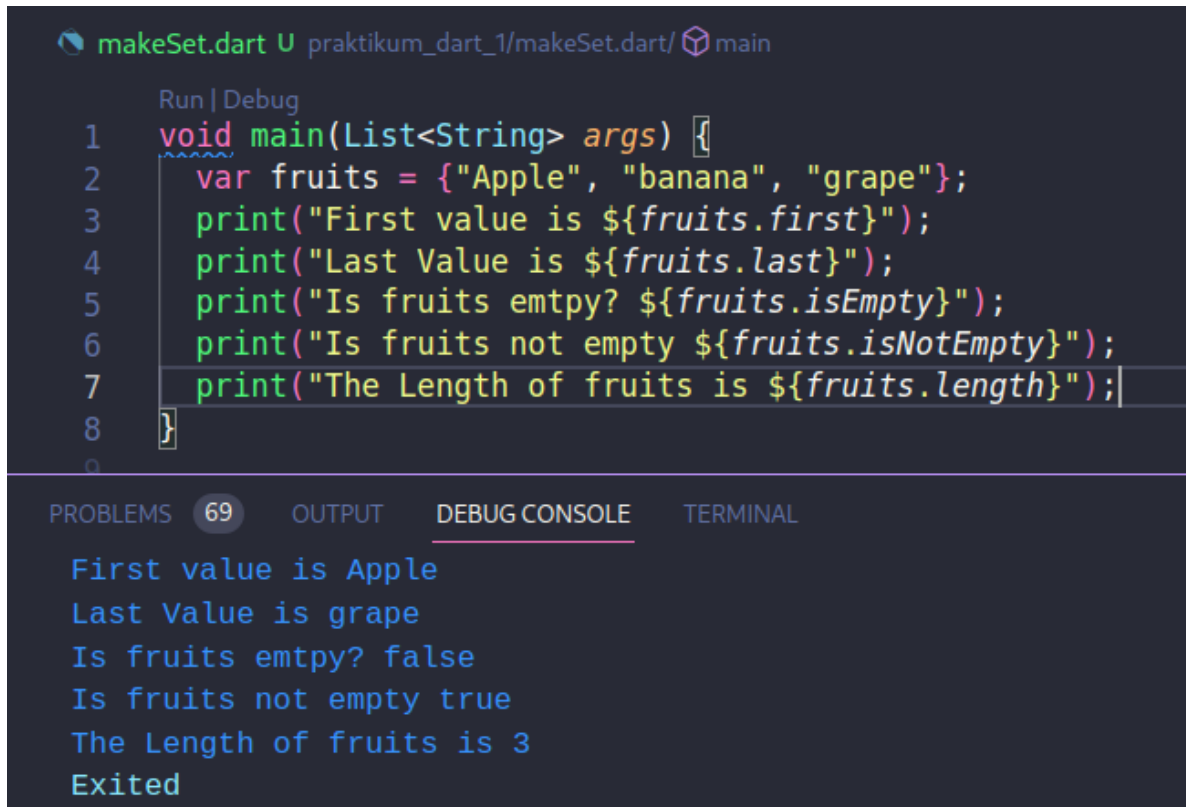
```
sumList.dart U praktikum_dart_1/sumList.dart/ main  
Run | Debug  
1 void main(List<String> args) {  
2   var odd = [1, 3, 5, 7, 9];  
3   var even = [2, 4, 6, 8];  
4  
5   var all = [...odd, ...even];  
6   print(all);  
7 }  
8
```

PROBLEMS 68 OUTPUT DEBUG CONSOLE TERMINAL

[1, 3, 5, 7, 9, 2, 4, 6, 8]
Exited

27. Membuat set

Set sama seperti list hanya saja set tidak bisa diakses menggunakan nomor index seperti list dan set juga tidak bisa menerima value duplikat, set memiliki beberapa fungsi untuk mengaksesnya seperti `.first` yang akan menampilkan isi pertama dari set, `.last` menampilkan isi terakhir, dan ada `isEmpty()` dan `isNotEmpty()` untuk mengecek isi dari set, serta `.length` mengetahui panjang dari set.



The screenshot shows an IDE window with a file named `makeSet.dart` in the `praktikum_dart_1` directory. The code defines a `main` function that creates a set of fruits and prints various properties. The output console shows the results of these operations.

```
makeSet.dart U praktikum_dart_1/makeSet.dart/ main
Run | Debug
1 void main(List<String> args) {
2   var fruits = {"Apple", "banana", "grape"};
3   print("First value is ${fruits.first}");
4   print("Last Value is ${fruits.last}");
5   print("Is fruits empty? ${fruits.isEmpty}");
6   print("Is fruits not empty ${fruits.isNotEmpty}");
7   print("The Length of fruits is ${fruits.length}");
8 }
9

PROBLEMS 69 OUTPUT DEBUG CONSOLE TERMINAL
First value is Apple
Last Value is grape
Is fruits empty? false
Is fruits not empty true
The Length of fruits is 3
Exited
```

28. Menambah atau mengurangi item dalam set

.add untuk menambahkan item dalam set yang akan di letakkan di ujung set, dan untuk menghapus item bisa menggunakan fungsi .remove() yang diisi dengan item apa yang akan di hapus.

```
setManipulation.dart U praktikum_dart_1/setManipulation.dart/ main
Run | Debug
1 void main(List<String> args) {
2   var fruits = {"Apple", "banana", "grape"};
3   print(fruits);
4   fruits.add("Lemon");
5   fruits.add("Guava");
6   print("After adding lemon and guava $fruits");
7
8   fruits.remove("Apple");
9   print("After removing apple $fruits");
10 }
```

PROBLEMS 70 OUTPUT DEBUG CONSOLE TERMINAL

```
{Apple, banana, grape}
After adding lemon and guava {Apple, banana, grape, Lemon, Guava}
After removing apple {banana, grape, Lemon, Guava}
Exited
```

29. Membuat dan menampilkan map

Untuk membuat map dibuat dengan Kata kunci Map <tipe data , tipe data> diikuti dengan nama map yang akan dibuat dan map di bungkus oleh tanda { } pasangan key dan value di batasi oleh tanda : dan antar key dibatasi dengan koma (,)

```
makeAndShowMaps.dart U praktikum_dart_1/makeAndShowMaps.dart/ main
Run | Debug
1 void main(List<String> args) {
2   Map<String, String> countryCapital = {
3     "USA": "Washington D.c",
4     "India": "New Delhi",
5     "China": "Beijing"
6   };
7
8   print(countryCapital);
9 }
10
```

PROBLEMS 71 OUTPUT DEBUG CONSOLE TERMINAL

```
{USA: Washington D.c, India: New Delhi, China: Beijing}
Exited
```

30. Mengakses value dari sebuah key

Mengakses value dari sebuah key cukup dengan menulis `namamap["diikuti dengan key nya"]`;

```
makeAndShowMaps.dart U praktikum_dart_1/makeAndShowMaps.dart/ main
Run | Debug
1 void main(List<String> args) {
2   Map<String, String> countryCapital = {
3     "USA": "Washington D.c",
4     "India": "New Delhi",
5     "China": "Beijing"
6   };
7   print(countryCapital);
8   print(countryCapital["USA"]);
9 }
10
```

PROBLEMS 71 OUTPUT DEBUG CONSOLE TERMINAL

```
{USA: Washington D.c, India: New Delhi, China: Beijing}
Washington D.c
Exited
```

31. Null Safety

Dalam mendeklarasikan variable tanpa isi atau null tidak diperbolehkan kecuali membuat variabelnya menjadi nullable dengan cara, memberi tanda ? setelah tipe data;

```
nullSafety.dart 1, U praktikum_dart_1/nullSafety.dart/ main
Run | Debug
1 void main(List<String> args) {
2   int a = 10; // Non nullable
3   int b = null; // Error
4   int? c; // nullable
5 }
6
```