

LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK



JUDUL :
FUNDAMENTAL DART

Disusun oleh:
Ajeng Azzahra Novika Ramadhan (21102112)

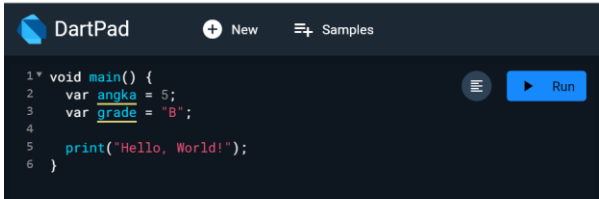
TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
BANYUMAS, JAWA TENGAH
2024

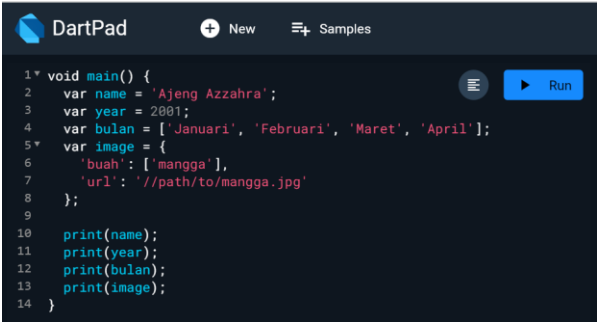
Pembahasan

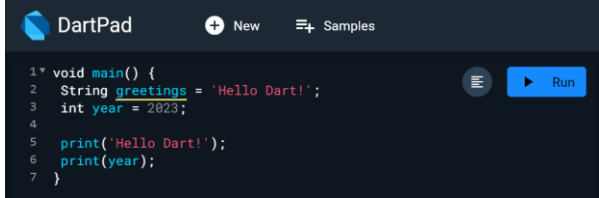
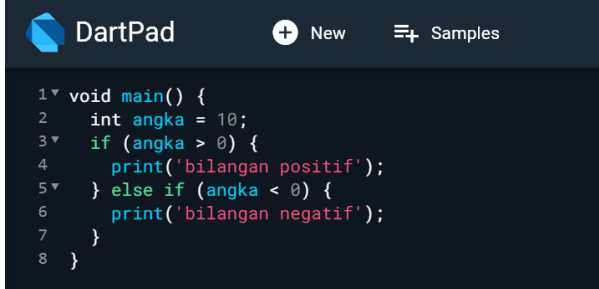
Fundamental Dart

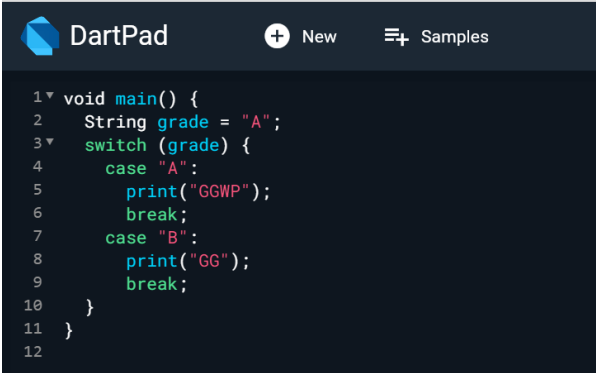
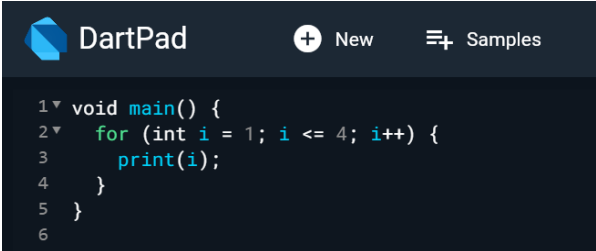
Dalam mata kuliah fundamental dart ini dianggap mendasar karena menjadi dasar berbagai algoritme dan teknik yang digunakan dalam aplikasi ilmu komputer seperti resolusi ketergantungan, penjadwalan tugas, dan jalur pemrosesan data. Ini memberikan metode yang andal untuk mengelola dan memproses dependensi dan sumber daya secara terstruktur dan efisien.

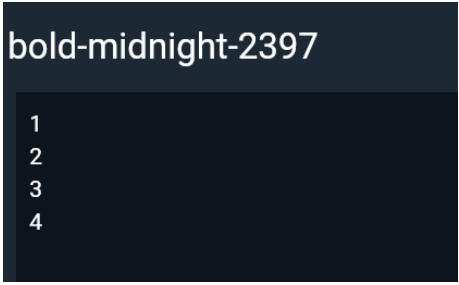
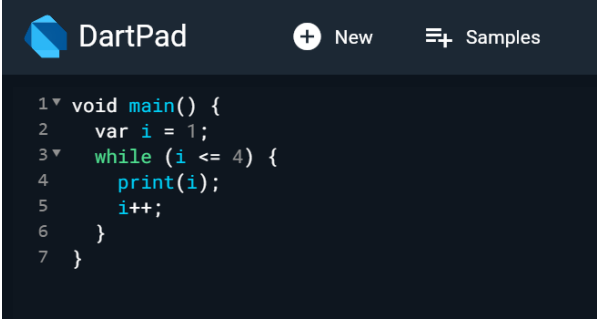
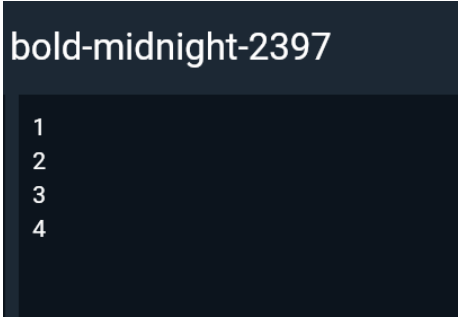
Langkah-Langkah Praktikum

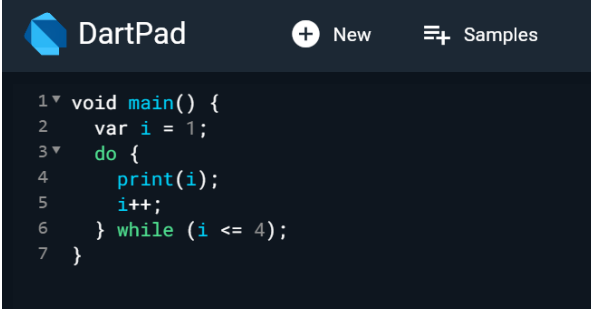

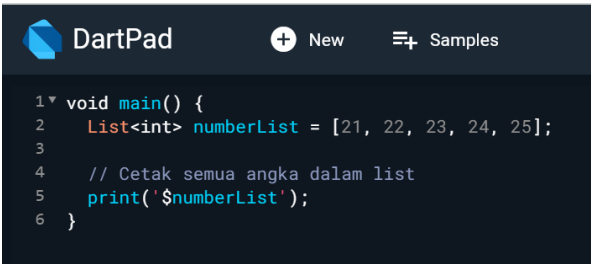
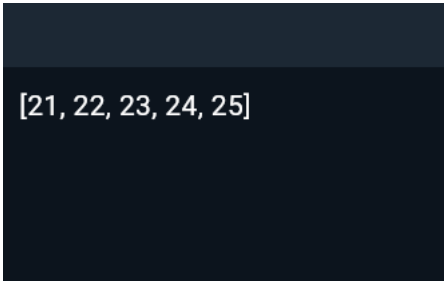
Langkah Praktikum	Pembahasan
<p>Gambar Langkah 1</p>  <pre>1 void main() { 2 var angka = 5; 3 var grade = "B"; 4 5 print("Hello, World!"); 6 }</pre>	<p>Pembahasan dari Gambar Langkah 1 yaitu pada fungsi main ini titik masuk utama (entry point) dari program yang akan dieksekusi ketika program dimulai. Kemudian var angka ini berfungsi untuk mendeklarasikan tipe data yang akan ditentukan secara otomatis berdasarkan nilai yang diberikan. Kemudian var grade ini berfungsi untuk mendeklarasikan tipe data dinamis yang mungkin merepresentasikan sebuah nilai atau grade dalam sebuah sistem penilaian. Yang terakhir yaitu print("Hello, World!"); berfungsi untuk mencetak pesan yang umumnya di</p>

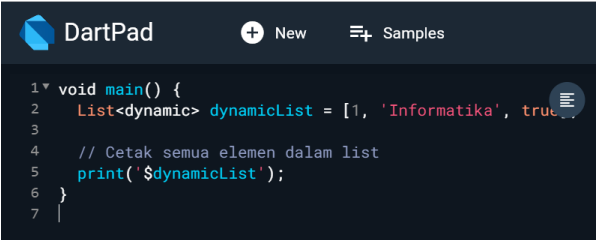
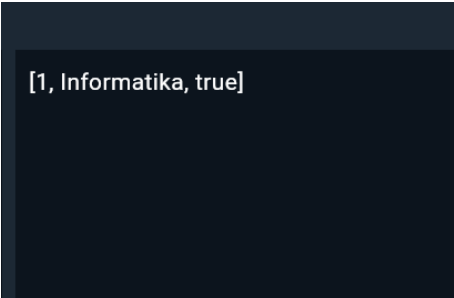
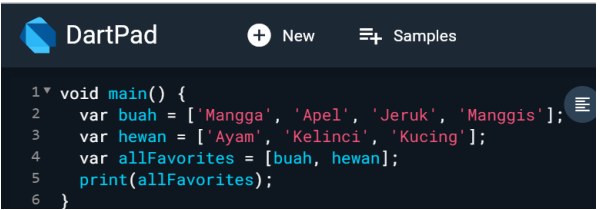
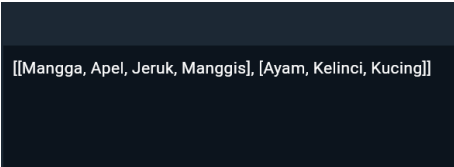
	<p>gunakan sebagai contoh dalam banyak bahasa pemrograman untuk menunjukkan bagaimana mencetak sesuatu ke layer.</p> <div data-bbox="882 459 1353 674"> <p>bold-midnight-2397</p> <p>Hello, World!</p> </div>
<p>Gambar Langkah 2</p> <div data-bbox="255 759 853 1079">  <pre> 1* void main() { 2 var name = 'Ajeng Azzahra'; 3 var year = 2001; 4 var bulan = ['Januari', 'Februari', 'Maret', 'April']; 5* var image = { 6 'buah': ['mangga'], 7 'url': '//path/to/mangga.jpg' 8 }; 9 10 print(name); 11 print(year); 12 print(bulan); 13 print(image); 14 } </pre> </div>	<p>Pembahasan dari Gambar Langkah 2 yaitu yang pertama ini pada variabel name ini string yang menyimpan nama yang di masukkan adalah Ajeng Azzahra. Kemudian langkah yang selanjutnya yaitu variabel year ini integer yang menyimpan tahun kelahiran 2001. Kemudian variabel bulan ini list yang berisi nama bulan. Kemudian variabel image ini map yang memiliki dua kunci : 'buah' dan 'url'. Kunci 'buah' ini memiliki nilai berupa string yaitu 'mangga'. Kunci yang kedua memiliki nilai string yang menunjukkan URL gambar untuk buah tersebut. Selanjutnya fungsi dari print ini untuk mencetak nilai variabel yang akan di panggil.</p>

	<div data-bbox="884 210 1142 255">bold-midnight-2397</div> <div data-bbox="900 277 1283 389"> <p>Ajeng Azzahra 2001 [Januari, Februari, Maret, April] {buah: [mangga], url: //path/to/mangga.jpg}</p> </div>
<div data-bbox="252 501 529 537">Gambar Langkah 3</div> <div data-bbox="252 551 852 748">  <pre> 1 void main() { 2 String greetings = 'Hello Dart!'; 3 int year = 2023; 4 5 print('Hello Dart!'); 6 print(year); 7 } </pre> </div>	<div data-bbox="884 501 1340 1093"> <p>Pembahasan dari Gambar Langkah 3 yaitu pada String greetings ini mendeklarasikan tipe data yang bernilai Hello Dart ini cara untuk menyimpan dan menggunakan string dalam Dart. Sedangkan int year ini di deklarasikan untuk menyimpan dan menggunakan angka dan bulan dalam Dart.</p> </div> <div data-bbox="884 1128 1149 1173">bold-midnight-2397</div> <div data-bbox="900 1209 999 1263"> <p>Hello Dart! 2023</p> </div>
<div data-bbox="252 1415 708 1451">Gambar Langkah 4 (if and else)</div> <div data-bbox="252 1464 852 1751">  <pre> 1 void main() { 2 int angka = 10; 3 if (angka > 0) { 4 print('bilangan positif'); 5 } else if (angka < 0) { 6 print('bilangan negatif'); 7 } 8 } </pre> </div>	<div data-bbox="884 1415 1340 1823"> <p>Pembahasan dari Gambar Langkah 4 yaitu pada program if angka ini di gunakan untuk memeriksa apakah nilai if angka ini lebih besar dari 0, jika benar program tersebut mencetak bilangan positif.</p> </div>

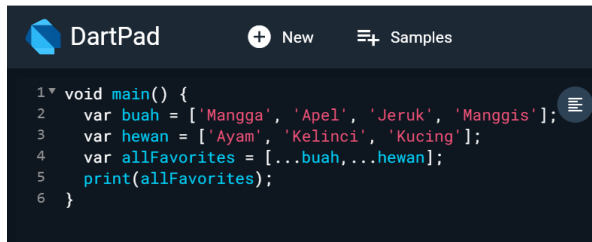
	<div data-bbox="884 210 1337 430"> <p>bold-midnight-2397</p> <p>bilangan positif</p> </div>
<p>Gambar Langkah 5 (switch case)</p> <div data-bbox="255 528 852 900">  <pre> 1 void main() { 2 String grade = "A"; 3 switch (grade) { 4 case "A": 5 print("GGWP"); 6 break; 7 case "B": 8 print("GG"); 9 break; 10 } 11 } 12 </pre> </div>	<p>Pembahasan dari Gambar Langkah 5 yaitu pada program ini menggunakan string grade A yang mana string ini diinisialisasikan dengan nilai A. Jika nilai grade A pada program akan mencetak “GGWP” menggunakan pernyataan print. Dan akan menghasilkan output “GGWP”.</p> <div data-bbox="884 1081 1337 1337"> <p>bold-midnight-2397</p> <p>GGWP</p> </div>
<p>Gambar Langkah 6 (for loops)</p> <div data-bbox="255 1438 852 1688">  <pre> 1 void main() { 2 for (int i = 1; i <= 4; i++) { 3 print(i); 4 } 5 } 6 </pre> </div>	<p>Pembahasan dari Gambar Langkah 6 yaitu pada program ini menggunakan struktur perulangan for yang mana variabel ‘i’ di inisialisasikan dengan nilai 1. Setelah mencetak nilai variabel ‘i’ ditambahkan dengan 1 proses ini berlanjut hingga nilai ‘i’ mencapai 4 karena kondisi i <= 4 akan menjadi salah ketika nilai i</p>

	<p>sama dengan 5.</p> 
<p>Gambar Langkah 7 (while)</p> 	<p>Pembahasan dari Gambar Langkah 7 yaitu pada program ini menggunakan struktur perulangan while yang mana diinisialisasi dengan nilai 1. Selama nilai 'i' kurang dari atau sama dengan 4 program akan mencetak nilai 'i' menggunakan pernyataan print. Proses ini terus berlanjut hingga nilai 'i' mencapai 5 karena saat ini 'i' sama dengan 5 kondisi 'i' <= 4 menjadi salah dan program keluar dari loop.</p> 
<p>Gambar Langkah 8 (do while)</p>	<p>Pembahasan dari Gambar Langkah 8 yaitu pada program ini menggunakan struktur perulangan do-while. Program</p>

 <pre> 1 void main() { 2 var i = 1; 3 do { 4 print(i); 5 i++; 6 } while (i <= 4); 7 } </pre>	<p>variabel i akan mencetak nilai 1 jika kondisi $i \leq 4$ dievaluasi tersebut masih benar maka program akan Kembali ke langkah pertama dan mencetak nilai i kembali. Proses ini terus berlanjut hingga i mencapai 5 pada titik ini kondisi $i \leq 4$ menjadi salah dan program keluar dari loop.</p> 
<p>Gambar Langkah 9</p>  <pre> 1 void main() { 2 List<int> numberList = [21, 22, 23, 24, 25]; 3 4 // Cetak semua angka dalam list 5 print('\$numberList'); 6 } </pre>	<p>Pembahasan dari Gambar Langkah 9 yaitu program ini menggunakan list, numberList ini berisi angka-angka bertipe int. <code>print('\$numberList');</code> berfungsi untuk mengekspresikan nilai variabel dalam string.</p> 
<p>Gambar Langkah 10</p>	<p>Pembahasan dari Gambar Langkah 10 yaitu program ini</p>

 <pre> 1* void main() { 2 List<dynamic> dynamicList = [1, 'Informatika', true]; 3 4 // Cetak semua elemen dalam list 5 print('\$dynamicList'); 6 } 7 </pre>	<p>menggunakan variabel <code>dynamicList</code> yang di deklarasikan sebagai list yang dapat menyimpan berbagai jenis tipe data, seperti integer, string, dan boolean. Dan fungsi print ini akan memanggil <code>dynamicList</code>nya.</p> 
<p>Gambar Langkah 11</p>  <pre> 1* void main() { 2 var buah = ['Mangga', 'Apel', 'Jeruk', 'Manggis']; 3 var hewan = ['Ayam', 'Kelinci', 'Kucing']; 4 var allFavorites = [buah, hewan]; 5 print(allFavorites); 6 } </pre>	<p>Pembahasan dari Gambar Langkah 11 yaitu program ini cara menggunakan list yang pertama variabel <code>buah</code> yang berisikan beberapa nama buah, variabel <code>hewan</code> yaitu list yang berisikan beberapa nama hewan, variabel <code>allFavorites</code> yaitu list yang berisi dua list buah dan hewan kemudian fungsi print <code>allFavorites</code> ini untuk memanggil program yang berhasil di cetak list.</p> 

Gambar Langkah 12



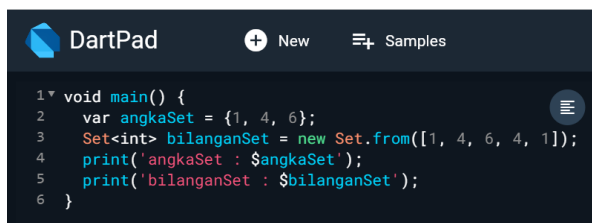
```
1* void main() {  
2  var buah = ['Mangga', 'Apel', 'Jeruk', 'Manggis'];  
3  var hewan = ['Ayam', 'Kelinci', 'Kucing'];  
4  var allFavorites = [...buah, ...hewan];  
5  print(allFavorites);  
6 }
```

Pembahasan dari Gambar Langkah 12 yaitu program ini menggabungkan dua list buah dan hewan dengan menggunakan operator spread. Dalam variabel buah ini berisikan list beberapa nama buah kemudian variabel hewan list yang berisikan beberapa nama hewan. Sedangkan fungsi print allFavorites ini hasil menggabungkan operator spread dengan menggunakan elemen-elemen dari kedua list tersebut akan dimasukkan ke dalam list.



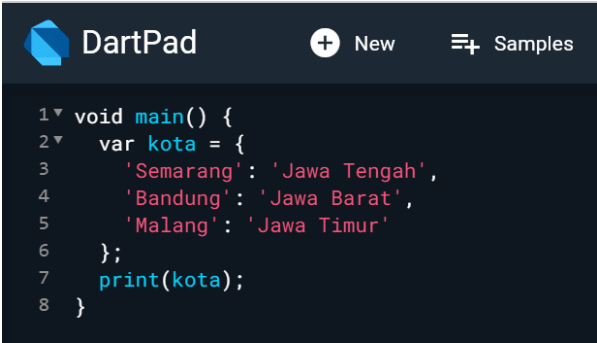
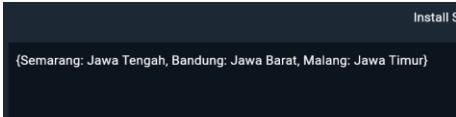
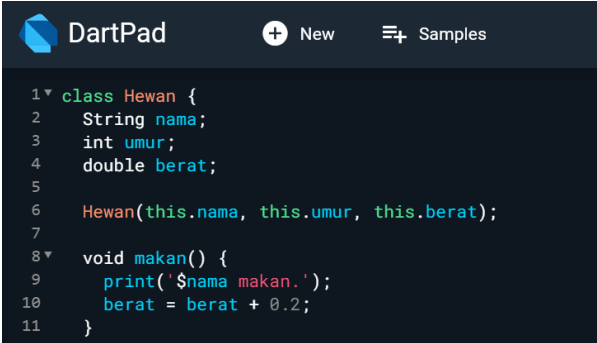
```
[Mangga, Apel, Jeruk, Manggis, Ayam, Kelinci, Kucing]
```

Gambar Langkah 13



```
1* void main() {  
2  var angkaSet = {1, 4, 6};  
3  Set<int> bilanganSet = new Set.from([1, 4, 6, 4, 1]);  
4  print('angkaSet : $angkaSet');  
5  print('bilanganSet : $bilanganSet');  
6 }
```

Pembahasan dari Gambar Langkah 13 yaitu program ini menggunakan tipe data Set, dalam variabel angkaSet di deklarasikan sebagai set dengan beberapa nilai yang di berikan secara langsung. Kemudian dalam variabel bilanganSet dideklarasikan sebagai set yang menggunakan konstruktor Set.from() di mana

	<p>nilai-nilai dari list [1, 4, 6, 4, 1] di salin ke dalam set karena tipe data set tidak mengizinkan elemen duplikat nilai yang sama hanya akan di masukkan sekali.</p> <pre>angkaSet : {1, 4, 6} bilanganSet : {1, 4, 6}</pre>
<p>Gambar Langkah 14</p>  <pre>1 void main() { 2 var kota = { 3 'Semarang': 'Jawa Tengah', 4 'Bandung': 'Jawa Barat', 5 'Malang': 'Jawa Timur' 6 }; 7 print(kota); 8 }</pre>	<p>Pembahasan dari Gambar Langkah 14 yaitu program di atas membuat sebuah map. Pada variabel kota ini sebuah map yang menggunakan string sebagai kunci (nama kota) dan string sebagai nilai (nama provinsi). Kemudian fungsi print (kota) ini nama kota yang akan di panggil di dalam program.</p> 
<p>Gambar Langkah 15</p>  <pre>1 class Hewan { 2 String nama; 3 int umur; 4 double berat; 5 6 Hewan(this.nama, this.umur, this.berat); 7 8 void makan() { 9 print('\$nama makan. '); 10 berat = berat + 0.2; 11 } 12 }</pre>	<p>Pembahasan dari Gambar Langkah 15 yaitu program di atas mendefinisikan sebuah kelas 'Hewan' dengan property 'nama', 'umur', dan 'berat' serta dua metode yaitu 'makan()' dan 'tidur()'.</p>

```

12
13 void tidur() {
14     print('$nama sedang tidur');
15 }
16 }
17
18 void main() {
19     var kucing = Hewan('Ketty', 2, 3.2);
20     kucing.makan();
21     kucing.tidur();
22     print(kucing.berat);
23 }

```

```

Ketty makan.
Ketty sedang tidur
3.4000000000000004

```

Gambar Langkah 16

DartPad

```

1 class Animal {
2     String _name = '';
3     int _age = 0;
4     double _weight = 0;
5
6     Animal(this._name, this._age, this._weight);
7
8     // Setter
9     set name(String value) {
10         _name = value;
11     }
12
13     // Getter
14     double get weight => _weight;
15
16     void eat() {
17         print('$_name is eating.');
```

```

18         _weight = _weight + 0.2;
19     }
20     void sleep() {
21         print('$_name is sleeping.');
```

```

22     }
23     void poop() {
24         print('$_name is pooping.');
```

```

25         _weight = _weight - 0.1;
26     }
27 }
28
29 void main() {
30     var myPet = Animal('Fluffy', 2, 2.5);
31     myPet.eat();
32     myPet.sleep();
33     myPet.poop();
34
35     print('Name: ${myPet._name}');
36     print('Age: ${myPet._age} years');
37     print('Weight: ${myPet._weight}[kg]');
38 }

```

Pembahasan dari Gambar Langkah 16 yaitu program ini mendefinisikan sebuah kelas Animal dengan property_name, _age, dan _weight yang bersifat private. Selain itu terdapat konstruktor untuk menginisialisasi property tersebut serta beberapa metode untuk melakukan Tindakan hewan, seperti eat(), sleep(), dan poop().

```

Fluffy is eating.
Fluffy is sleeping.
Fluffy is pooping.
Name: Fluffy
Age: 2 years
Weight: 2.6[kg]

```

Gambar Langkah 17

Pembahasan dari Gambar Langkah 17 yaitu program ini menggunakan turunan dari

```

bin > meong.dart > main
1 //[[9].Inheritance
2 import 'hewan.dart';
3
4 class Meong extends Hewan {
5   String warnaBulu;
6
7   Meong(
8     String nama,
9     int umur,
10    double berat,
11    this.warnaBulu,
12    ) : super(nama, umur, berat);
13
14   void jalan() {
15     print('$nama berjalan');
16   }
17
18   void warna() {
19     print('$nama berwarna $warnaBulu');
20   }
21 }

```

```

bin > meong.dart > main
Run | Debug
23 void main() {
24   var kucing = Meong('Ketty', 2, 3.2, 'Putih');
25   kucing.jalan();
26   kucing.makan();
27   kucing.warna();
28   print(kucing.berat);
29 }

```

kelas Hewan yang di import dari file hewan.dart. Konstruktor Meong ini memanggil dari kelas Hewan menggunakan sintaks super(nama, umur, berat) dan menginisialisasi property tambahan warnaBulu. Dalam fungsi main() terdapat objek kucing dari kelas Meong yang di buat dengan menggunakan konstruktor yang telah di tentukan.

```

Ketty berjalan
Ketty makan.
Ketty berwarna Putih
Nama: Ketty
Umur: 2 tahun
Berat: 3.2 kg

```

Gambar Langkah 18

```

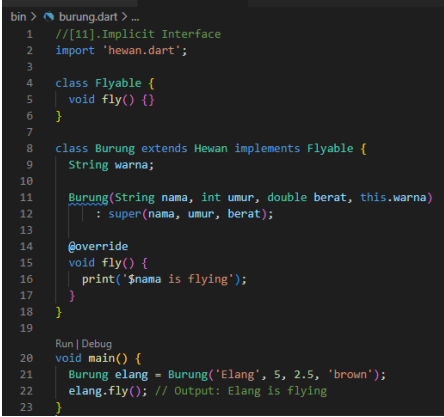
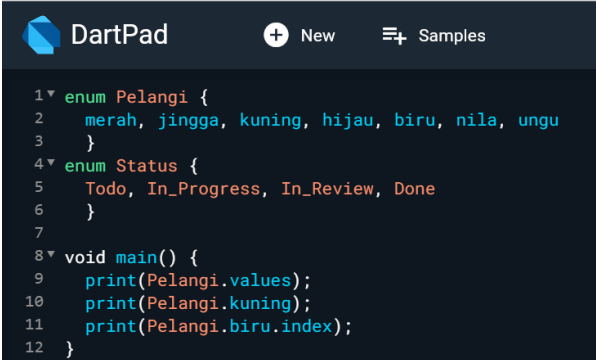
DartPad
+ New  Samples

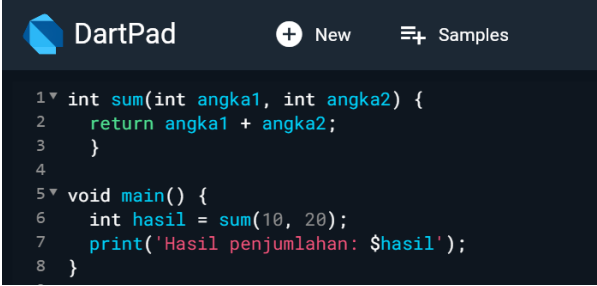

1 // Definisi kelas abstract Hewan
2* abstract class Hewan {
3   String nama;
4   int umur;
5   double berat;
6
7   Hewan(this.nama, this.umur, this.berat);
8
9   // Metode makan dideklarasikan sebagai metode abstract
10  void makan();
11
12* void tidur() {
13   print('$nama sedang tidur');
14 }
15 }

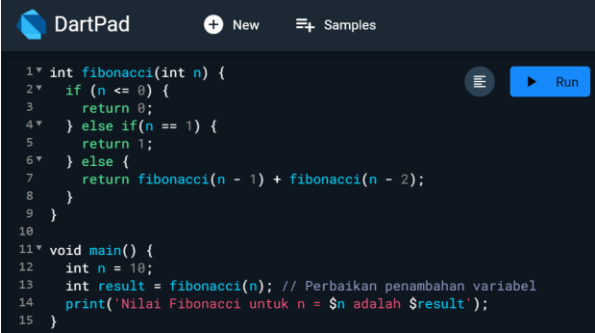
17
18 // Kelas turunan Kucing dari Hewan
19* class Kucing extends Hewan {
20   // Constructor untuk kelas Kucing
21   Kucing(String nama, int umur, double berat) : super(nama, umur, berat);
22
23   // Implementasi metode makan dari kelas Hewan
24* void makan() {
25   print('$nama makan');
26   berat += 0.2;
27 }
28 }
29
30* void main() {
31   // Membuat objek dari kelas Kucing
32   var kucingObjek = Kucing('Melati', 2, 2.0);
33   kucingObjek.makan();
34   kucingObjek.tidur();
35   print('Nama: ${kucingObjek.nama}');
36   print('Umur: ${kucingObjek.umur} tahun');
37   print('Berat: ${kucingObjek.berat} kg');
38 }
39

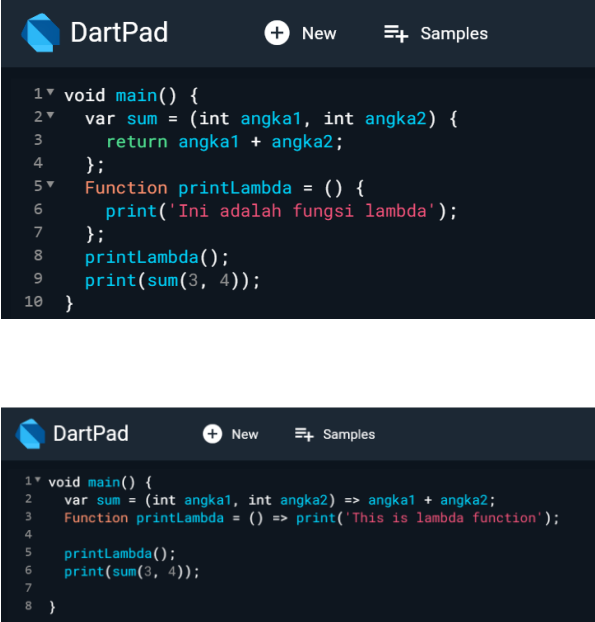
```

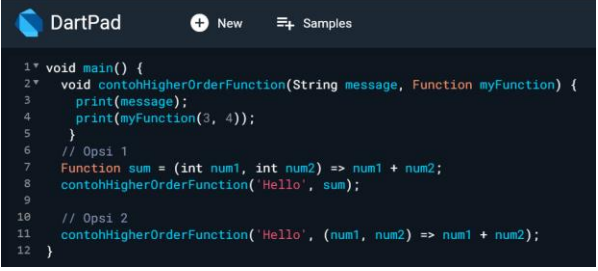

Pembahasan dari Gambar Langkah 18 yaitu program ini menggunakan kelas abstract Hewan yang memiliki tiga property nama, umur, dan berat untuk mendeklarasikan pesan yang akan di sebutkan oleh program tersebut. Dan di sini juga menggunakan kelas turunan Kucing dari kelas Hewan. Sedangkan fungsi main ini objek dari semua kelas yang akan di panggil oleh program.

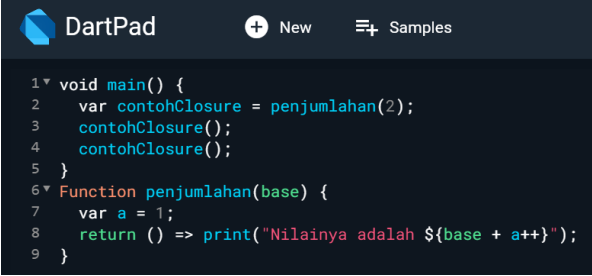
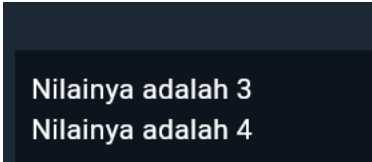
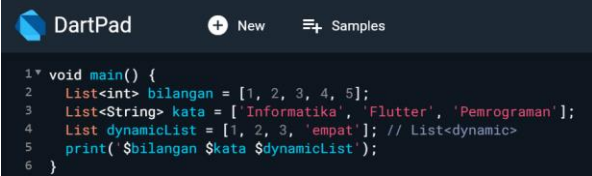
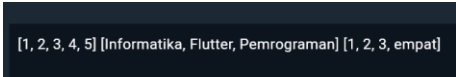
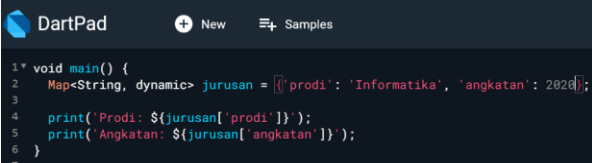
	<pre> Melati makan. Melati sedang tidur Nama: Melati Umur: 2 tahun Berat: 2.2 kg </pre>
<p>Gambar Langkah 19</p>  <pre> bin > dart burung.dart > ... 1 //[[1]].Implicit Interface 2 import 'hewan.dart'; 3 4 class Flyable { 5 void fly() {} 6 } 7 8 class Burung extends Hewan implements Flyable { 9 String warna; 10 11 Burung(String nama, int umur, double berat, this.warna) 12 : super(nama, umur, berat); 13 14 @override 15 void fly() { 16 print('\$nama is flying'); 17 } 18 } 19 20 Run Debug 21 void main() { 22 Burung elang = Burung('Elang', 5, 2.5, 'brown'); 23 elang.fly(); // Output: Elang is flying 24 } </pre>	<p>Pembahasan dari Gambar Langkah 19 yaitu program ini mendemostrasikan konsep pewarisan dan implementasi interface. Dalam import kelas Hewan untuk implementasi dari kelas Hewan tidak diberikan kode yang di berika. Kemudian kelas Burung ini merupakan turunan dari kelas Hewan dan mengimplementasikan kelas Flyable. Sedangkan fungsi main untuk mengimplementasikan objek yang mau di panggil di dalam program tersebut.</p> <pre> Elang is flying </pre>
<p>Gambar Langkah 20</p>  <pre> DartPad + New ≡ Samples 1 enum Pelangi { 2 merah, jingga, kuning, hijau, biru, nila, ungu 3 } 4 enum Status { 5 Todo, In_Progress, In_Review, Done 6 } 7 8 void main() { 9 print(Pelangi.values); 10 print(Pelangi.kuning); 11 print(Pelangi.biru.index); 12 } </pre>	<p>Pembahasan dari Gambar Langkah 20 yaitu program ini mendemostrasikan pengguna enumerasi di sini ada dua jenis yaitu Pelangi dan Status yang mendefinisikan warna – warna dalam Pelangi dan juga mendefinisikan status dari suatu tugas atau pekerjaan Todo,</p>

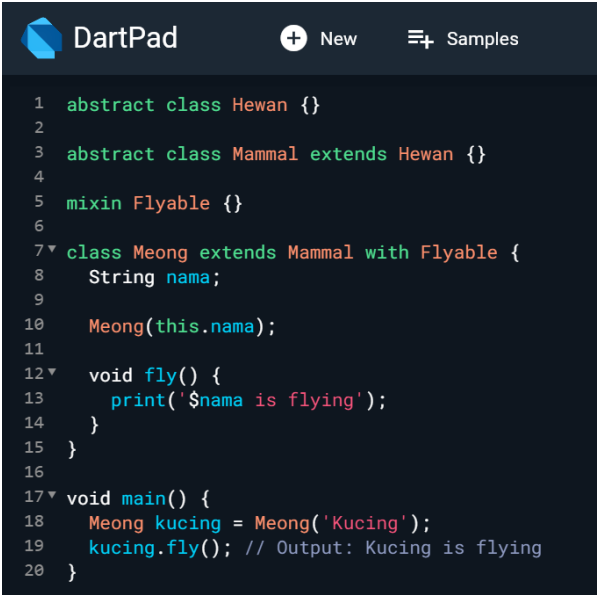
	<p>In_Progress, In_Review, dan Done. Sedangkan fungsi main ini memanggil program yang akan di panggil.</p> <pre>[Pelangi.merah, Pelangi.jingga, Pelangi.kuning, Pelangi.hijau, Pelangi.biru, Pelangi.nila, Pelangi.ungu] Pelangi.kuning 4</pre>
<p>Gambar Langkah 21</p>  <pre>1 ▾ int sum(int angka1, int angka2) { 2 return angka1 + angka2; 3 } 4 5 ▾ void main() { 6 int hasil = sum(10, 20); 7 print('Hasil penjumlahan: \$hasil'); 8 }</pre>	<p>Pembahasan dari Gambar Langkah 21 yaitu program ini mendefinisikan fungsi sum yang menerima dua parameter bertipe integer (angka 1 dan angka 2) dan mengembalikan hasil penjumlahan keduanya. Kemudian fungsi main menggunakan fungsi sum() tersebut untuk menjumlahkan dua angka (10 dan 20) dan menyimpan hasilnya dalam variabel hasil yang akan mencetak hasil penjumlahan tersebut menggunakan string interpolation.</p>  <pre>Hasil penjumlahan: 30</pre>
<p>Gambar Langkah 22</p>	<p>Pembahasan dari Gambar Langkah 22 yaitu program ini menggunakan fungsi Fibonacci yang menerima satu parameter n merupakan bilangan bulat</p>

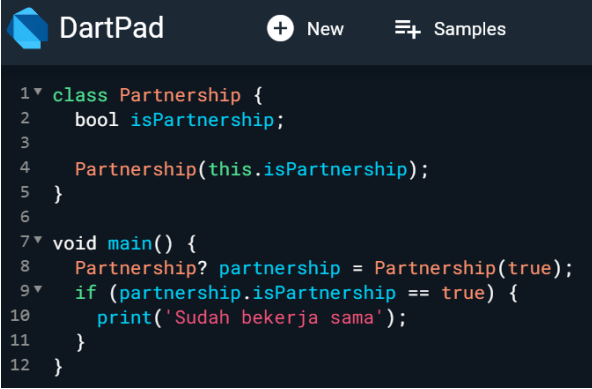

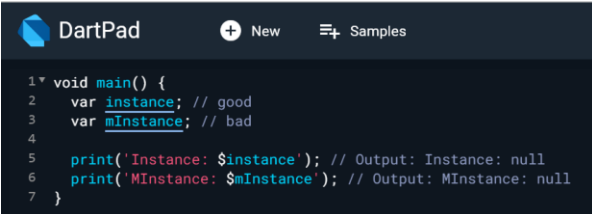
 <pre> 1* int fibonacci(int n) { 2* if (n <= 0) { 3* return 0; 4* } else if (n == 1) { 5* return 1; 6* } else { 7* return fibonacci(n - 1) + fibonacci(n - 2); 8* } 9* } 10 11* void main() { 12* int n = 10; 13* int result = fibonacci(n); // Perbaikan penambahan variabel 14* print('Nilai Fibonacci untuk n = \$n adalah \$result'); 15* } </pre>	<p>yang akan dicari nilai fibonaccinya. Fungsi main nilai Fibonacci untuk n dihitung menggunakan fungsi Fibonacci dan disimpan dalam variabel result yang akan dicetak dengan menggunakan string interpolation.</p> <div data-bbox="880 705 1339 801" style="background-color: #2e3436; color: white; padding: 5px; text-align: center;"> Nilai Fibonacci untuk n = 10 adalah 55 </div>
--	---

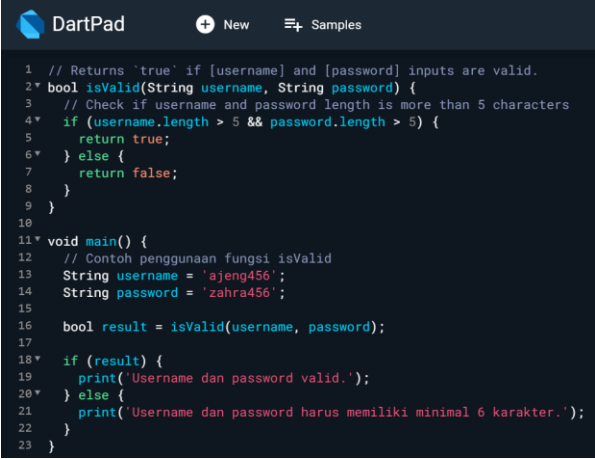
<p>Gambar Langkah 23</p>  <pre> 1* void main() { 2* var sum = (int angka1, int angka2) { 3* return angka1 + angka2; 4* }; 5* Function printLambda = () { 6* print('Ini adalah fungsi lambda'); 7* }; 8* printLambda(); 9* print(sum(3, 4)); 10* } </pre> <pre> 1* void main() { 2* var sum = (int angka1, int angka2) => angka1 + angka2; 3* Function printLambda = () => print('This is lambda function'); 4* 5* printLambda(); 6* print(sum(3, 4)); 7* 8* } </pre>	<p>Pembahasan dari Gambar Langkah 23 yaitu program ini menggunakan fungsi lambda, pada variabel sum ini mendeklarasikan dan diinisialisasi dengan fungsi lambda. Sedangkan variabel printLambda dideklarasikan dan diinisialisasi dengan fungsi lambda tersebut tidak menerima parameter apa pun dan hanya mencetak sebuah pesan.</p> <div data-bbox="880 1579 1339 1753" style="background-color: #2e3436; color: white; padding: 5px; text-align: center;"> Ini adalah fungsi lambda 7 </div> <p>Program selanjutnya yaitu penggunaan fungsi lambda (arrow function), variabel sum dideklarasikan dan diinisialisasi</p>
--	--

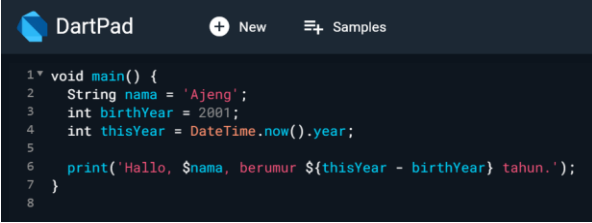
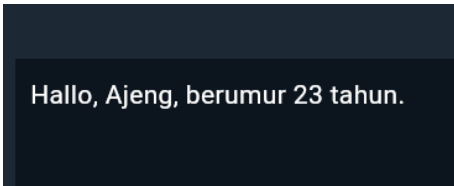
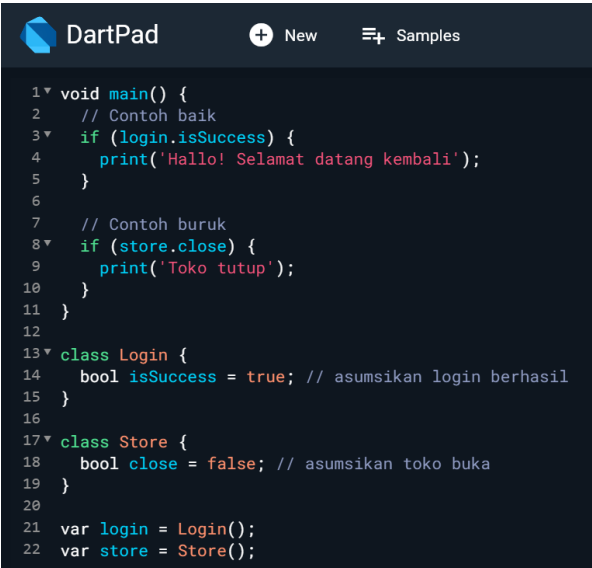
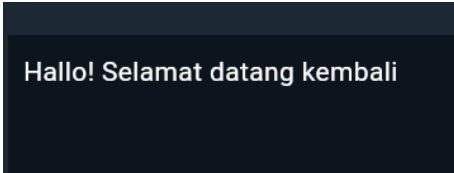
	<p>fungsi lambda yang menerima dua parameter bertipe integer (angka 1 dan angka 2) dan mengembalikan hasil penjumlahan keduanya. Variabel printLambda mendeklarasikan dan diinisialisasi dengan fungsi lambda tidak menerima parameter apapun dan mencetak sebuah pesan.</p> <div data-bbox="879 887 1262 1032"> <pre>This is lambda function 7</pre> </div>
<p>Gambar Langkah 24</p> <div data-bbox="256 1122 855 1391">  <pre> 1 void main() { 2 void contohHigherOrderFunction(String message, Function myFunction) { 3 print(message); 4 print(myFunction(3, 4)); 5 } 6 // Opsi 1 7 Function sum = (int num1, int num2) => num1 + num2; 8 contohHigherOrderFunction('Hello', sum); 9 10 // Opsi 2 11 contohHigherOrderFunction('Hello', (num1, num2) => num1 + num2); 12 } </pre> </div>	<p>Pembahasan dari Gambar Langkah 24 yaitu program ini menggunakan fungsi higher-order yang menerima dua parameter yaitu message yang bertipe string dan myFunction. Sedangkan fungsi opsi 1 dan opsi 2 ini mendeklarasikan fungsi HigherOrderFunction untuk melakukan penjumlahan dua angka tanpa nama.</p> <div data-bbox="879 1733 1302 1895">  <pre> Hello 7 Hello 7 </pre> </div>
<p>Gambar Langkah 25</p>	<p>Pembahasan dari Gambar</p>

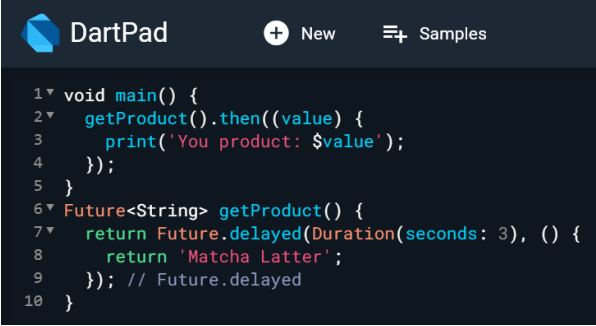

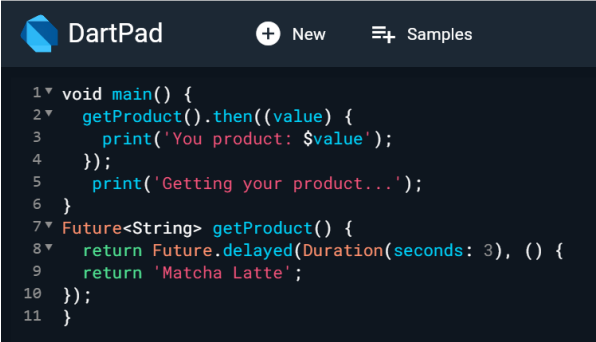
 <pre> 1* void main() { 2 var contohClosure = penjumlahan(2); 3 contohClosure(); 4 contohClosure(); 5 } 6* Function penjumlahan(base) { 7 var a = 1; 8 return () => print("Nilainya adalah \${base + a++}"); 9 } </pre>	<p>Langkah 25 yaitu program ini penggunaan closure, pada fungsi closure ini memiliki parameter yang mencetak nilai dari penjumlahan antara base dan a kemudian menambahkan nilai a setiap kali fungsi yang di panggil.</p> 
<p>Gambar Langkah 26</p>  <pre> 1* void main() { 2 List<int> bilangan = [1, 2, 3, 4, 5]; 3 List<String> kata = ['Informatika', 'Flutter', 'Pemrograman']; 4 List<dynamic> dynamicList = [1, 2, 3, 'empat']; // List<dynamic> 5 print('\$bilangan \$kata \$dynamicList'); 6 } </pre>	<p>Pembahasan dari Gambar Langkah 26 yaitu program ini menggunakan tiga tipe data yang berbeda adalah List<int>, List<String>, List<dynamic>. Pada print('\$bilangan \$kata \$dynamicList') ini mencetak isi dari ketiga list tersebut ke dalam satu baris dengan menggunakan string interpolation.</p> 
<p>Gambar Langkah 27</p>  <pre> 1* void main() { 2 Map<String, dynamic> jurusan = {'prodi': 'Informatika', 'angkatan': 2020}; 3 4 print('Prodi: \${jurusan['prodi']}'); 5 print('Angkatan: \${jurusan['angkatan']}'); 6 } </pre>	<p>Pembahasan dari Gambar Langkah 27 yaitu pemrograman ini membuat sebuah map yang disebut jurusan yang memiliki dua pasangan kunci nilai prodi dan angkatan. Kunci prodi</p>


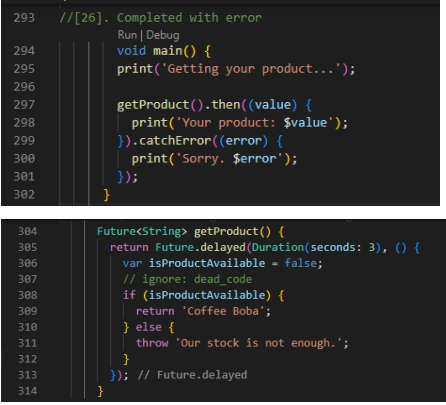
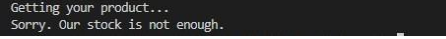
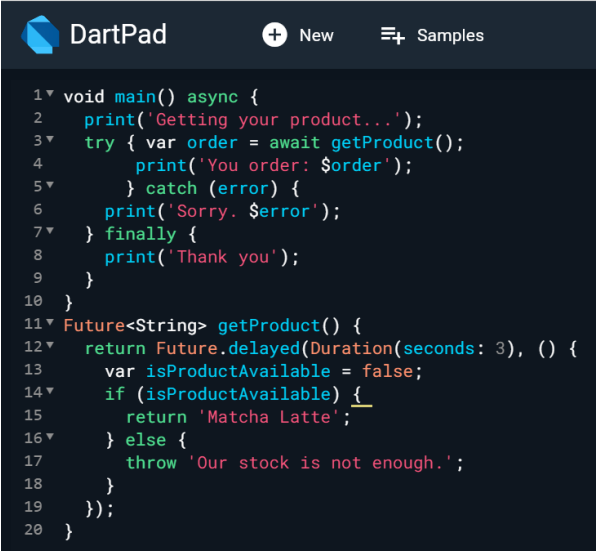
	<p>memiliki nilai Informatika yang merupakan sebuah string, sedangkan kunci Angkatan memiliki nilai 2020 yang merupakan sebuah integer. Kemudian program mencetak nilai dari kedua pasangan kunci nilai tersebut.</p> <div data-bbox="879 696 1244 853"> Prodi: Informatika Angkatan: 2020 </div>
<p>Gambar Langkah 28</p> <div data-bbox="255 934 853 1527">  <pre> 1 abstract class Hewan {} 2 3 abstract class Mammal extends Hewan {} 4 5 mixin Flyable {} 6 7 class Meong extends Mammal with Flyable { 8 String nama; 9 10 Meong(this.nama); 11 12 void fly() { 13 print('\$nama is flying'); 14 } 15 } 16 17 void main() { 18 Meong kucing = Meong('Kucing'); 19 kucing.fly(); // Output: Kucing is flying 20 } </pre> </div>	<p>Pembahasan dari Gambar Langkah 28 yaitu program ini mendemonstrasikan konsep inheritance dan mixim. Dalam fungsi main ini untuk memanggil sebuah objek Meong dengan nama Kucing dibuat dan kemudian metode fly() di panggil pada objek tersebut. Meskipun kucing tidak bisa terbang dalam kehidupan nyata namun di sini karena menggunakan mixin Flyable maka hasil cetakan adalah Kucing is flying.</p> <div data-bbox="879 1733 1197 1890"> Kucing is flying </div>
<p>Gambar Langkah 29</p>	<p>Pembahasan dari Gambar Langkah 29 yaitu pemrogram ini</p>

 <pre> 1 class Partnership { 2 bool isPartnership; 3 4 Partnership(this.isPartnership); 5 } 6 7 void main() { 8 Partnership? partnership = Partnership(true); 9 if (partnership.isPartnership == true) { 10 print('Sudah bekerja sama'); 11 } 12 } </pre>	<p>menggunakan objek nullable. Terdapat tanda tanya ? di program setelah tipe data Partnership pada deklarasi objek mengindikasikan bahwa objek tersebut bisa bernilai null. Dengan demikian kita dapat mengakses property isPartnership dengan menggunakan operator akses atribut. Jika objek tersebut adalah null maka akan dianggap sebagai null safe sehingga tidak terjadi error saat mengakses propertinya.</p> 
<p>Gambar Langkah 30</p>  <pre> 1 void main() { 2 var instance; // good 3 var mInstance; // bad 4 5 print('Instance: \$instance'); // Output: Instance: null 6 print('MInstance: \$mInstance'); // Output: MInstance: null 7 } </pre>	<p>Pembahasan dari Gambar Langkah 30 yaitu pemrograman ini menggunakan dua variabel yang pertama variabel instance yang mendeklarasikan dengan baik dan tidak melanggar konversi penamaan yang umum. Sedangkan variabel mInstance ini mendeklarasikan variabel yang tampaknya tidak memiliki makna yang jelas dan</p>

	<p>dapat membingungkan pembaca kode.</p> <pre>Instance: null MInstance: null</pre>
<p>Gambar Langkah 31</p>  <pre> 1 // Returns 'true' if [username] and [password] inputs are valid. 2* bool isValid(String username, String password) { 3 // Check if username and password length is more than 5 characters 4* if (username.length > 5 && password.length > 5) { 5 return true; 6* } else { 7 return false; 8* } 9 } 10 11* void main() { 12 // Contoh penggunaan fungsi isValid 13 String username = 'ajeng456'; 14 String password = 'zahra456'; 15 16 bool result = isValid(username, password); 17 18* if (result) { 19 print('Username dan password valid.');</pre>	<p>Pembahasan dari Gambar Langkah 31 yaitu program ini contoh sederhana dari sebuah fungsi isValid yang bertujuan untuk memvalidasi username dan password. Fungsi ini mengembalikan nilai true jika kedua input memiliki Panjang lebih dari 5 karakter dan false jika tidak. Dalam fungsi main isValid dilakukan dengan mengirimkan dua string yaitu username dan password sebagai argument. Hasil dari pemanggilan fungsi ini disimpan dalam variabel result yang kemudian digunakan untuk menentukan pesan yang akan dicetak berdasarkan apakah validasi berhasil atau tidak.</p> <pre>Username dan password valid.</pre>
<p>Gambar Langkah 32</p>	<p>Pembahasan dari Gambar Langkah 32 yaitu program ini</p>

 <pre> 1 void main() { 2 String nama = 'Ajeng'; 3 int birthYear = 2001; 4 int thisYear = DateTime.now().year; 5 6 print('Hallo, \$nama, berumur \${thisYear - birthYear} tahun.');</pre>	<p>berisikan variabel nama untuk menyimpan nama seseorang, variabel birthYear ini untuk menyimpan tahun kelahiran seseorang. Sedangkan fungsi print dilakukan perhitungan umur dengan mengurangi tahun saat ini.</p> 
<p>Gambar Langkah 33</p>  <pre> 1 void main() { 2 // Contoh baik 3 if (login.isSuccess) { 4 print('Hallo! Selamat datang kembali');</pre> <pre> 5 } 6 7 // Contoh buruk 8 if (store.close) { 9 print('Toko tutup');</pre>	<p>Pembahasan dari Gambar Langkah 33 yaitu program ini untuk mengatasi contoh buruk untuk memperbaiki nama variabel close menjadi isOpen yang secara langsung menyatakan apakah tokoh buka atau tidak.</p> 
<p>Gambar Langkah 34</p>	<p>Pembahasan dari Gambar Langkah 34 yaitu program ini menggunakan Future dapat melanjutkan eksekusi lainnya selama menunggu hasil dari operasi yang asinkron seperti</p>

 <pre> 1 void main() { 2 getProduct().then((value) { 3 print('You product: \$value'); 4 }); 5 } 6 Future<String> getProduct() { 7 return Future.delayed(Duration(seconds: 3), () { 8 return 'Matcha Latter'; 9 }); // Future.delayed 10 } </pre>	<p>pembacaan dari jaringan atau penyimpanan local. Ini memungkinkan aplikasi untuk tetap responsive dan tidak terblokir saat menunggu hasil operasi yang memakan waktu.</p> 
<p>Gambar Langkah 35</p>  <pre> 1 void main() { 2 getProduct().then((value) { 3 print('You product: \$value'); 4 }); 5 print('Getting your product...'); 6 } 7 Future<String> getProduct() { 8 return Future.delayed(Duration(seconds: 3), () { 9 return 'Matcha Latte'; 10 }); 11 } </pre>	<p>Pembahasan dari Gambar Langkah 35 yaitu Di dalam fungsi main(), terdapat pemanggilan fungsi getProduct() yang mengembalikan sebuah future. Ketika future tersebut selesai dieksekusi, fungsi then() akan dijalankan untuk mencetak produk yang diperoleh. Pada bagian akhir fungsi main(), terdapat pesan 'Getting your product...' yang akan dicetak sebelum produk didapatkan. Fungsi getProduct() mengembalikan sebuah future yang selesai dalam waktu 3 detik. Hasil future tersebut adalah 'Matcha Latte'.</p>

	
<p>Gambar Langkah 36</p> 	<p>Pembahasan dari Gambar Langkah 36 yaitu pada pemrograman fungsi main akan di cetak terlebih dahulu untuk memberi tahu pengguna bahwa proses pengambilan produk sedang berlangsung.</p> 
<p>Gambar Langkah 37</p> 	<p>Pembahasan dari Gambar Langkah 37 yaitu pada fungsi main(), kata kunci async digunakan untuk menandai bahwa fungsi tersebut adalah asynchronous. Ini memungkinkan penggunaan await di dalamnya untuk menunggu hasil dari operasi asynchronous. Fungsi getProduct() mengembalikan sebuah future yang selesai dalam waktu 3 detik. Di dalamnya, terdapat variabel isProductAvailable yang menandakan apakah produk tersedia atau tidak. Jika produk</p>

	<p>tidak tersedia, maka fungsi akan melemparkan error dengan pesan 'Our stock is not enough.'</p> <div><p>Getting your product... Sorry. Our stock is not enough. Thank you</p></div>
--	---