

Jobsheet 4

Pengantar Bahasa Pemrograman Dart - Bagian 3



Rendy Putra Kusuma

TI 3A / 23

POLITEKNIK NEGERI MALANG

2024

Praktikum 1

- Langkah 1

Ketik atau salin kode program berikut ke dalam fungsi `main()`.

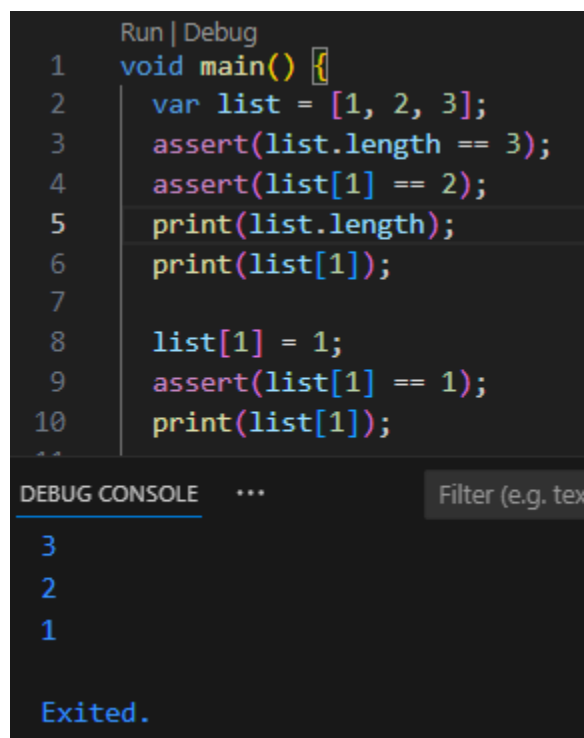
```
var list = [1, 2, 3];
assert(list.length == 3);
assert(list[1] == 2);
print(list.length);
print(list[1]);

list[1] = 1;
assert(list[1] == 1);
print(list[1]);
```

- Langkah 2

Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan!

Jawaban :



The screenshot shows a code editor with a Dart program and its debug console output. The code defines a list, asserts its length and a value, prints them, changes a value, asserts again, and prints the new value. The debug console shows the output of the print statements: 3, 2, and 1, followed by 'Exited.'.

```
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 }
```

DEBUG CONSOLE ... Filter (e.g. text)

3
2
1
Exited.

Assert pada kode tersebut berfungsi untuk mengecek, jika pengecekan tidak sesuai maka kode akan berhenti. Output 3 berasal dari panjang dari array variabel list, output 2 berasal dari index ke 1 dari variabel list yang isinya adalah angka 2, output 1 berasal dari index ke 1 dari variabel list karena index ke 1 variabel list telah diubah menjadi 1 yang sebelumnya 2.

- Langkah 3

Ubah kode pada langkah 1 menjadi variabel final yang mempunyai index = 5 dengan default value = null. Isilah nama dan NIM Anda pada elemen index ke-1 dan ke-2. Lalu print dan capture hasilnya.

Apa yang terjadi ? Jika terjadi error, silakan perbaiki.

Jawaban :

```
1 void main() {
11
12     var list = [null, 'Rendy Putra Kusuma', '2241720124', null, null];
13     assert(list.length == 5);
14     assert(list[1] == 'Rendy Putra Kusuma');
15     assert(list[2] == '2241720124');
16     print(list.length);
17     print(list[1]);
18     print(list[2]);
19 }
20
```

DEBUG CONSOLE ... Filter (e.g. text, !exclude, \escape)

5
Rendy Putra Kusuma
2241720124
Exited.

Tidak terjadi error. Null tetap dihitung sebagai isi dari array variabel list.

Praktikum 2

- Langkah 1

Ketik atau salin kode program berikut ke dalam fungsi `main()`.

```
var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine', 'astatine'};
print(halogens);
```

- Langkah 2

Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan! Lalu perbaiki jika terjadi error.

Jawaban :

```
Run | Debug
1 void main() {
2     var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine', 'astatine'};
3     print(halogens);
4 }
```

PROBLEMS OUTPUT DEBUG CONSOLE ... Filter (e.g. text, !exclude, \escape)

{fluorine, chlorine, bromine, iodine, astatine}

Exited.

Tidak terjadi error, kode tersebut melakukan print isi dari variabel halogens

- Langkah 3

Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.

```
var names1 = <String>{};
Set<String> names2 = {}; // This works, too.
var names3 = {}; // Creates a map, not a set.

print(names1);
print(names2);
print(names3);
```

Apa yang terjadi ? Jika terjadi error, silakan perbaiki namun tetap menggunakan ketiga variabel tersebut. Tambahkan elemen nama dan NIM Anda pada kedua variabel Set tersebut dengan dua fungsi berbeda yaitu .add() dan .addAll(). Untuk variabel Map dihapus, nanti kita coba di praktikum selanjutnya.

Dokumentasikan code dan hasil di console, lalu buat laporannya.

Jawaban :

```
5 var names1 = <String>{};
6 Set<String> names2 = {}; // This works, too.
7 var names3 = {}; // Creates a map, not a set.
8
9
10 print(names1);
11 print(names2);
12 print(names3);
13 }
```

PROBLEMS OUTPUT DEBUG CONSOLE ... Filter (e.g. te

3 {}

Karena variabel names1 dan names2 adalah set kosong sedangkan names3 adalah map kosong sehingga outputnya adalah {}

```
1 void main() {
5     var names1 = <String>{};
6     Set<String> names2 = {}; // This works, too.
7     var names3 = {}; // Creates a map, not a set.
8     names1.add('Rendy Putra Kusuma');
9     names1.add('2241720124');
10    names2.addAll({'Rendy Putra Kusuma', '2241720124'});
11
12    print(names1);
13    print(names2);
14    print(names3);
15 }
16
```

DEBUG CONSOLE ... Filter (e.g. text, !exclude, \escape)

2 {Rendy Putra Kusuma, 2241720124}
{}

Exited.

.add hanya bisa menambahkan 1 argumen sedangkan .addAll bisa menambahkan lebih dari 1 argumen.

Praktikum 3

- Langkah 1

Ketik atau salin kode program berikut ke dalam fungsi `main()`.

```
var gifts = {
  // Key:    Value
  'first': 'partridge',
  'second': 'turtledoves',
  'fifth': 1
};

var nobleGases = {
  2: 'helium',
  10: 'neon',
  18: 2,
};

print(gifts);
print(nobleGases);
```

- Langkah 2

Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan! Lalu perbaiki jika terjadi error.

Jawaban :

```
Run | Debug
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);

```

DEBUG CONSOLE ... Filter (e.g. text, !exclude, \escape)

```
{first: partridge, second: turtledoves, fifth: 1}
{2: helium, 10: neon, 18: 2}

Exited.
```

Gifts dan nobleGases di kode tersebut adalah Map, keduanya memiliki key dan value dan ketika dilakukan print maka akan ditampilkan key dan valuenya.

- Langkah 3

Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda..

```
var mhs1 = Map<String, String>();
gifts['first'] = 'partridge';
gifts['second'] = 'turtledoves';
gifts['fifth'] = 'golden rings';

var mhs2 = Map<int, String>();
nobleGases[2] = 'helium';
nobleGases[10] = 'neon';
```

```
nobleGases[18] = 'argon';
```

Apa yang terjadi ? Jika terjadi error, silakan perbaiki.

Tambahkan elemen nama dan NIM Anda pada tiap variabel di atas (gifts, nobleGases, mhs1, dan mhs2). Dokumentasikan hasilnya dan buat laporannya!

Jawaban :

```
1  void main() {
17
18      var mhs1 = Map<String, String>();
19      gifts['first'] = 'partridge';
20      gifts['second'] = 'turtledoves';
21      gifts['fifth'] = 'golden rings';
22      mhs1['nama'] = 'Rendy Putra Kusuma';
23      mhs1['NIM'] = '2241720124';
24
25      var mhs2 = Map<int, String>();
26      nobleGases[2] = 'helium';
27      nobleGases[10] = 'neon';
28      nobleGases[18] = 'argon';
29      mhs2[0] = 'Rendy Putra Kusuma';
30      mhs2[1] = '2241720124';
31
32      print(mhs1);
33      print(mhs2);
34  }
```

DEBUG CONSOLE ... Filter (e.g. text, !exclude, \escape)

```
{first: partridge, second: turtledoves, fifth: 1}
{2: helium, 10: neon, 18: 2}
{nama: Rendy Putra Kusuma, NIM: 2241720124}
{0: Rendy Putra Kusuma, 1: 2241720124}

Exited.
```

Praktikum 4

- Langkah 1

Ketik atau salin kode program berikut ke dalam fungsi `main()`.

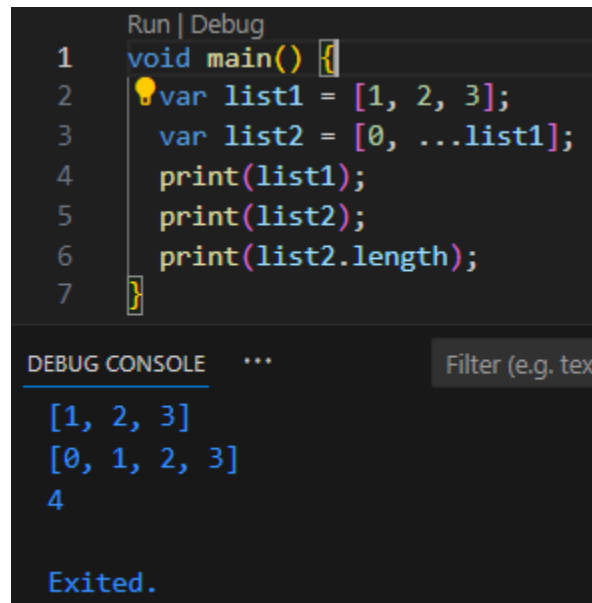
```
var list = [1, 2, 3];
var list2 = [0, ...list];
```

```
print(list1);  
print(list2);  
print(list2.length);
```

- Langkah 2

Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan! Lalu perbaiki jika terjadi error.

Jawaban :



The screenshot shows a code editor with the following JavaScript code:

```
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 }
```

Below the code editor is a 'DEBUG CONSOLE' window showing the output of the program:

```
[1, 2, 3]  
[0, 1, 2, 3]  
4  
Exited.
```

Dikarenakan spread operator (...list1) isi dari list2 menjadi 0,1,2,3

- Langkah 3

Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.

```
list1 = [1, 2, null];  
print(list1);  
var list3 = [0, ...?list1];  
print(list3.length);
```

Apa yang terjadi ? Jika terjadi error, silakan perbaiki.

Tambahkan variabel list berisi NIM Anda menggunakan Spread Operators. Dokumentasikan hasilnya dan buat laporannya!

Jawaban :


```
1 void main() {
2
3
4
5
6
7
8     var list1 = [1, 2, null, 2241720124];
9     print(list1);
10    var list3 = [0, ...?list1];
11    print(list3);
12    print(list3.length);
13 }
14
```

DEBUG CONSOLE ... Filter (e.g. text, !exclude, \esc

```
[1, 2, null, 2241720124]
[0, 1, 2, null, 2241720124]
5

Exited.
```

Kode tersebut menggunakan null-aware spread operator, yang fungsinya hampir sama dengan spread operator hanya untuk menghindari error akibat null.

- Langkah 4

Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda

```
var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];
print(nav);
```

Apa yang terjadi ? Jika terjadi error, silakan perbaiki. Tunjukkan hasilnya jika variabel promoActive ketika true dan false.

Jawaban :

```
1 void main() {
2
3
4
5
6
7
8
9
10
11
12
13
14     bool promoActive = true;
15     var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];
16     print(nav);
17 }
18
```

PROBLEMS OUTPUT DEBUG CONSOLE ... Filter (e.g. text, !exclude, \escape)

```
[Home, Furniture, Plants, Outlet]

Exited.
```

```
1 void main() {
13
14     bool promoActive = false;
15     var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];
16     print(nav);
17 }
18
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE ... Filter (e.g. text, !exclude, \escape)

[Home, Furniture, Plants]

Exited.

Apabila promoActive true maka 'Outlet' akan ditampilkan

- Langkah 5

Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.

```
var nav2 = ['Home', 'Furniture', 'Plants', if (login case 'Manager')  
'Inventory'];  
print(nav2);
```

Apa yang terjadi ? Jika terjadi error, silakan perbaiki. Tunjukkan hasilnya jika variabel login mempunyai kondisi lain.

Jawaban :

```
1 void main() {
18     var login = 'Employees';
19     var nav2 = [
20         'Home',
21         'Furniture',
22         'Plants',
23         if (login case 'Manager') 'Inventory'
24     ];
25     print(nav2);
26 }
27
```

PROBLEMS OUTPUT DEBUG CONSOLE ... Filter

[Home, Furniture, Plants]

Exited.

- Langkah 6

Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.

```
var listOfInts = [1, 2, 3];  
var listOfStrings = ['#0', for (var i in listOfInts) '#$i'];
```

```
assert(listOfStrings[1] == '#1');  
print(listOfStrings);
```

Apa yang terjadi ? Jika terjadi error, silakan perbaiki. Jelaskan manfaat Collection For dan dokumentasikan hasilnya.

Jawaban :

```
1 void main() {  
27   var listOfInts = [1, 2, 3];  
28   var listOfStrings = ['#0', for (var i in listOfInts) '#$i'];  
29   assert(listOfStrings[1] == '#1');  
30   print(listOfStrings);  
31 }  
32
```

PROBLEMS OUTPUT DEBUG CONSOLE ... Filter (e.g. text, !exclude, \escap

[#0, #1, #2, #3]

Exited.

Kode tersebut membuat sebuah list baru (listOfStrings) dengan elemen-elemen yang dihasilkan dari list lain (listOfInts). Setiap elemen dari listOfInts digunakan untuk membentuk string baru dalam format # diikuti dengan angka yang sesuai.

Praktikum 5

- Langkah 1

Ketik atau salin kode program berikut ke dalam fungsi `main()`.

```
var record = ('first', a: 2, b: true, 'last');  
print(record)
```

- Langkah 2

Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan! Lalu perbaiki jika terjadi error.

Jawaban :

```
Run | Debug  
1 void main() {  
2   var record = ('first', a: 2, b: true, 'last');  
3   print(record);  
4 }
```

PROBLEMS OUTPUT DEBUG CONSOLE ... Filter (e.g. te

(first, last, a: 2, b: true)

Exited.

Kode tersebut menerapkan fitur record yang bisa mengelompokkan beberapa nilai yang berbeda tipe ke dalam satu record tanpa perlu membuat class yang rumit.

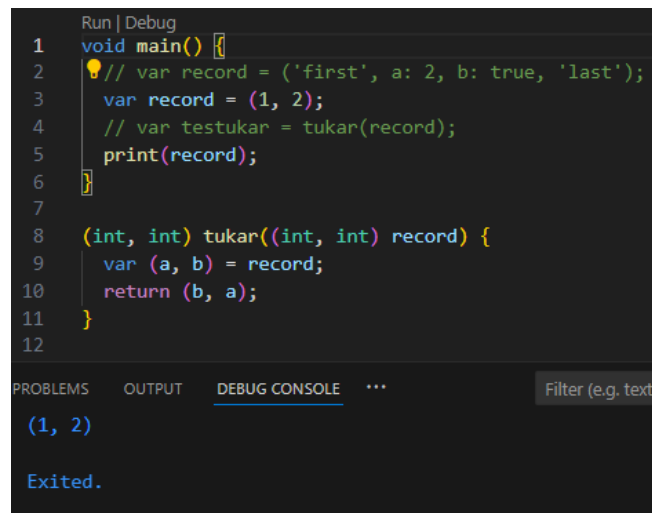
- Langkah 3

Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.

```
(int, int) tukar((int, int) record) {  
    var (a, b) = record;  
    return (b, a);  
}
```

Apa yang terjadi ? Jika terjadi error, silakan perbaiki. Gunakan fungsi tukar() di dalam main() sehingga tampak jelas proses pertukaran value field di dalam Records.

Jawaban :

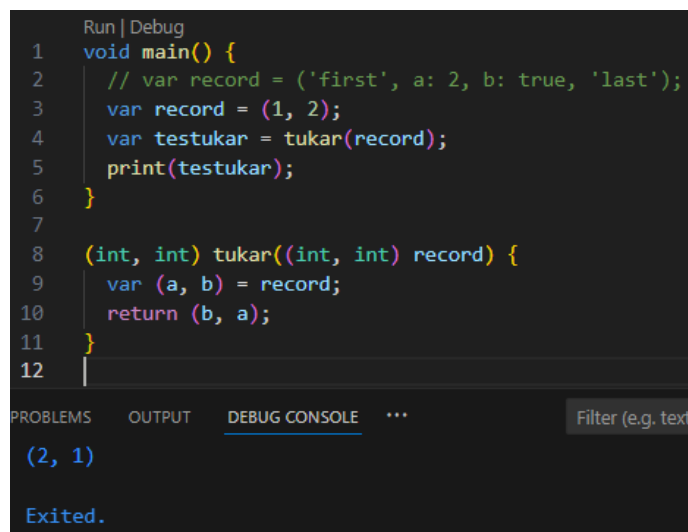


```
Run | Debug  
1 void main() {  
2     // var record = ('first', a: 2, b: true, 'last');  
3     var record = (1, 2);  
4     // var testukar = tukar(record);  
5     print(record);  
6 }  
7  
8 (int, int) tukar((int, int) record) {  
9     var (a, b) = record;  
10    return (b, a);  
11 }  
12
```

PROBLEMS OUTPUT DEBUG CONSOLE ... Filter (e.g. text, ...)

(1, 2)

Exited.



```
Run | Debug  
1 void main() {  
2     // var record = ('first', a: 2, b: true, 'last');  
3     var record = (1, 2);  
4     var testukar = tukar(record);  
5     print(testukar);  
6 }  
7  
8 (int, int) tukar((int, int) record) {  
9     var (a, b) = record;  
10    return (b, a);  
11 }  
12
```

PROBLEMS OUTPUT DEBUG CONSOLE ... Filter (e.g. text, ...)

(2, 1)

Exited.

fungsi tukar() yang menerima sebuah record dengan nilai 1,2 dan mengembalikan record yang nilainya ditukar menjadi 2,1

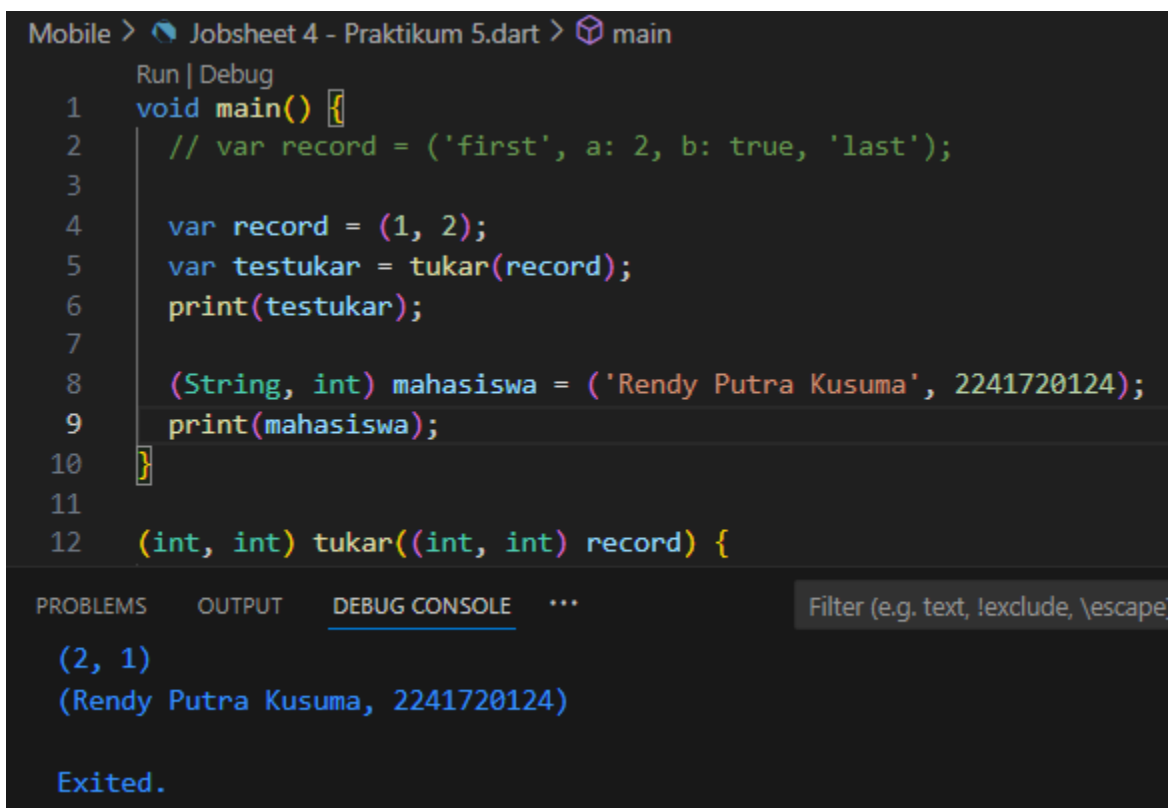
- Langkah 4

Tambahkan kode program berikut di dalam scope void main(), lalu coba eksekusi (Run) kode Anda.

```
// Record type annotation in a variable declaration:  
(String, int) mahasiswa;  
print(mahasiswa);
```

Apa yang terjadi ? Jika terjadi error, silakan perbaiki. Inisialisasi field nama dan NIM Anda pada variabel record mahasiswa di atas. Dokumentasikan hasilnya dan buat laporannya!

Jawaban :



The screenshot shows an IDE window titled 'Jobsheet 4 - Praktikum 5.dart' with a 'main' function. The code defines a record type `(String, int)` for `mahasiswa`, creates a `record` variable with values `(1, 2)`, uses a `tukar` function to swap them, and prints the result. The `tukar` function is defined below. The output console shows the swapped values `(2, 1)` and the student information `(Rendy Putra Kusuma, 2241720124)`, followed by `Exited.`

```
Mobile > Jobsheet 4 - Praktikum 5.dart > main  
Run | Debug  
1 void main() {  
2   // var record = ('first', a: 2, b: true, 'last');  
3  
4   var record = (1, 2);  
5   var testukar = tukar(record);  
6   print(testukar);  
7  
8   (String, int) mahasiswa = ('Rendy Putra Kusuma', 2241720124);  
9   print(mahasiswa);  
10 }  
11  
12 (int, int) tukar((int, int) record) {  
13  
14 }  
15 }  
16 }  
17 }  
18 }  
19 }  
20 }  
21 }  
22 }  
23 }  
24 }  
25 }  
26 }  
27 }  
28 }  
29 }  
30 }  
31 }  
32 }  
33 }  
34 }  
35 }  
36 }  
37 }  
38 }  
39 }  
40 }  
41 }  
42 }  
43 }  
44 }  
45 }  
46 }  
47 }  
48 }  
49 }  
50 }  
51 }  
52 }  
53 }  
54 }  
55 }  
56 }  
57 }  
58 }  
59 }  
60 }  
61 }  
62 }  
63 }  
64 }  
65 }  
66 }  
67 }  
68 }  
69 }  
70 }  
71 }  
72 }  
73 }  
74 }  
75 }  
76 }  
77 }  
78 }  
79 }  
80 }  
81 }  
82 }  
83 }  
84 }  
85 }  
86 }  
87 }  
88 }  
89 }  
90 }  
91 }  
92 }  
93 }  
94 }  
95 }  
96 }  
97 }  
98 }  
99 }  
100 }  
101 }  
102 }  
103 }  
104 }  
105 }  
106 }  
107 }  
108 }  
109 }  
110 }  
111 }  
112 }  
113 }  
114 }  
115 }  
116 }  
117 }  
118 }  
119 }  
120 }  
121 }  
122 }  
123 }  
124 }  
125 }  
126 }  
127 }  
128 }  
129 }  
130 }  
131 }  
132 }  
133 }  
134 }  
135 }  
136 }  
137 }  
138 }  
139 }  
140 }  
141 }  
142 }  
143 }  
144 }  
145 }  
146 }  
147 }  
148 }  
149 }  
150 }  
151 }  
152 }  
153 }  
154 }  
155 }  
156 }  
157 }  
158 }  
159 }  
160 }  
161 }  
162 }  
163 }  
164 }  
165 }  
166 }  
167 }  
168 }  
169 }  
170 }  
171 }  
172 }  
173 }  
174 }  
175 }  
176 }  
177 }  
178 }  
179 }  
180 }  
181 }  
182 }  
183 }  
184 }  
185 }  
186 }  
187 }  
188 }  
189 }  
190 }  
191 }  
192 }  
193 }  
194 }  
195 }  
196 }  
197 }  
198 }  
199 }  
200 }  
201 }  
202 }  
203 }  
204 }  
205 }  
206 }  
207 }  
208 }  
209 }  
210 }  
211 }  
212 }  
213 }  
214 }  
215 }  
216 }  
217 }  
218 }  
219 }  
220 }  
221 }  
222 }  
223 }  
224 }  
225 }  
226 }  
227 }  
228 }  
229 }  
230 }  
231 }  
232 }  
233 }  
234 }  
235 }  
236 }  
237 }  
238 }  
239 }  
240 }  
241 }  
242 }  
243 }  
244 }  
245 }  
246 }  
247 }  
248 }  
249 }  
250 }  
251 }  
252 }  
253 }  
254 }  
255 }  
256 }  
257 }  
258 }  
259 }  
260 }  
261 }  
262 }  
263 }  
264 }  
265 }  
266 }  
267 }  
268 }  
269 }  
270 }  
271 }  
272 }  
273 }  
274 }  
275 }  
276 }  
277 }  
278 }  
279 }  
280 }  
281 }  
282 }  
283 }  
284 }  
285 }  
286 }  
287 }  
288 }  
289 }  
290 }  
291 }  
292 }  
293 }  
294 }  
295 }  
296 }  
297 }  
298 }  
299 }  
300 }  
301 }  
302 }  
303 }  
304 }  
305 }  
306 }  
307 }  
308 }  
309 }  
310 }  
311 }  
312 }  
313 }  
314 }  
315 }  
316 }  
317 }  
318 }  
319 }  
320 }  
321 }  
322 }  
323 }  
324 }  
325 }  
326 }  
327 }  
328 }  
329 }  
330 }  
331 }  
332 }  
333 }  
334 }  
335 }  
336 }  
337 }  
338 }  
339 }  
340 }  
341 }  
342 }  
343 }  
344 }  
345 }  
346 }  
347 }  
348 }  
349 }  
350 }  
351 }  
352 }  
353 }  
354 }  
355 }  
356 }  
357 }  
358 }  
359 }  
360 }  
361 }  
362 }  
363 }  
364 }  
365 }  
366 }  
367 }  
368 }  
369 }  
370 }  
371 }  
372 }  
373 }  
374 }  
375 }  
376 }  
377 }  
378 }  
379 }  
380 }  
381 }  
382 }  
383 }  
384 }  
385 }  
386 }  
387 }  
388 }  
389 }  
390 }  
391 }  
392 }  
393 }  
394 }  
395 }  
396 }  
397 }  
398 }  
399 }  
400 }  
401 }  
402 }  
403 }  
404 }  
405 }  
406 }  
407 }  
408 }  
409 }  
410 }  
411 }  
412 }  
413 }  
414 }  
415 }  
416 }  
417 }  
418 }  
419 }  
420 }  
421 }  
422 }  
423 }  
424 }  
425 }  
426 }  
427 }  
428 }  
429 }  
430 }  
431 }  
432 }  
433 }  
434 }  
435 }  
436 }  
437 }  
438 }  
439 }  
440 }  
441 }  
442 }  
443 }  
444 }  
445 }  
446 }  
447 }  
448 }  
449 }  
450 }  
451 }  
452 }  
453 }  
454 }  
455 }  
456 }  
457 }  
458 }  
459 }  
460 }  
461 }  
462 }  
463 }  
464 }  
465 }  
466 }  
467 }  
468 }  
469 }  
470 }  
471 }  
472 }  
473 }  
474 }  
475 }  
476 }  
477 }  
478 }  
479 }  
480 }  
481 }  
482 }  
483 }  
484 }  
485 }  
486 }  
487 }  
488 }  
489 }  
490 }  
491 }  
492 }  
493 }  
494 }  
495 }  
496 }  
497 }  
498 }  
499 }  
500 }  
501 }  
502 }  
503 }  
504 }  
505 }  
506 }  
507 }  
508 }  
509 }  
510 }  
511 }  
512 }  
513 }  
514 }  
515 }  
516 }  
517 }  
518 }  
519 }  
520 }  
521 }  
522 }  
523 }  
524 }  
525 }  
526 }  
527 }  
528 }  
529 }  
530 }  
531 }  
532 }  
533 }  
534 }  
535 }  
536 }  
537 }  
538 }  
539 }  
540 }  
541 }  
542 }  
543 }  
544 }  
545 }  
546 }  
547 }  
548 }  
549 }  
550 }  
551 }  
552 }  
553 }  
554 }  
555 }  
556 }  
557 }  
558 }  
559 }  
560 }  
561 }  
562 }  
563 }  
564 }  
565 }  
566 }  
567 }  
568 }  
569 }  
570 }  
571 }  
572 }  
573 }  
574 }  
575 }  
576 }  
577 }  
578 }  
579 }  
580 }  
581 }  
582 }  
583 }  
584 }  
585 }  
586 }  
587 }  
588 }  
589 }  
590 }  
591 }  
592 }  
593 }  
594 }  
595 }  
596 }  
597 }  
598 }  
599 }  
600 }  
601 }  
602 }  
603 }  
604 }  
605 }  
606 }  
607 }  
608 }  
609 }  
610 }  
611 }  
612 }  
613 }  
614 }  
615 }  
616 }  
617 }  
618 }  
619 }  
620 }  
621 }  
622 }  
623 }  
624 }  
625 }  
626 }  
627 }  
628 }  
629 }  
630 }  
631 }  
632 }  
633 }  
634 }  
635 }  
636 }  
637 }  
638 }  
639 }  
640 }  
641 }  
642 }  
643 }  
644 }  
645 }  
646 }  
647 }  
648 }  
649 }  
650 }  
651 }  
652 }  
653 }  
654 }  
655 }  
656 }  
657 }  
658 }  
659 }  
660 }  
661 }  
662 }  
663 }  
664 }  
665 }  
666 }  
667 }  
668 }  
669 }  
670 }  
671 }  
672 }  
673 }  
674 }  
675 }  
676 }  
677 }  
678 }  
679 }  
680 }  
681 }  
682 }  
683 }  
684 }  
685 }  
686 }  
687 }  
688 }  
689 }  
690 }  
691 }  
692 }  
693 }  
694 }  
695 }  
696 }  
697 }  
698 }  
699 }  
700 }  
701 }  
702 }  
703 }  
704 }  
705 }  
706 }  
707 }  
708 }  
709 }  
710 }  
711 }  
712 }  
713 }  
714 }  
715 }  
716 }  
717 }  
718 }  
719 }  
720 }  
721 }  
722 }  
723 }  
724 }  
725 }  
726 }  
727 }  
728 }  
729 }  
730 }  
731 }  
732 }  
733 }  
734 }  
735 }  
736 }  
737 }  
738 }  
739 }  
740 }  
741 }  
742 }  
743 }  
744 }  
745 }  
746 }  
747 }  
748 }  
749 }  
750 }  
751 }  
752 }  
753 }  
754 }  
755 }  
756 }  
757 }  
758 }  
759 }  
760 }  
761 }  
762 }  
763 }  
764 }  
765 }  
766 }  
767 }  
768 }  
769 }  
770 }  
771 }  
772 }  
773 }  
774 }  
775 }  
776 }  
777 }  
778 }  
779 }  
780 }  
781 }  
782 }  
783 }  
784 }  
785 }  
786 }  
787 }  
788 }  
789 }  
790 }  
791 }  
792 }  
793 }  
794 }  
795 }  
796 }  
797 }  
798 }  
799 }  
800 }  
801 }  
802 }  
803 }  
804 }  
805 }  
806 }  
807 }  
808 }  
809 }  
810 }  
811 }  
812 }  
813 }  
814 }  
815 }  
816 }  
817 }  
818 }  
819 }  
820 }  
821 }  
822 }  
823 }  
824 }  
825 }  
826 }  
827 }  
828 }  
829 }  
830 }  
831 }  
832 }  
833 }  
834 }  
835 }  
836 }  
837 }  
838 }  
839 }  
840 }  
841 }  
842 }  
843 }  
844 }  
845 }  
846 }  
847 }  
848 }  
849 }  
850 }  
851 }  
852 }  
853 }  
854 }  
855 }  
856 }  
857 }  
858 }  
859 }  
860 }  
861 }  
862 }  
863 }  
864 }  
865 }  
866 }  
867 }  
868 }  
869 }  
870 }  
871 }  
872 }  
873 }  
874 }  
875 }  
876 }  
877 }  
878 }  
879 }  
880 }  
881 }  
882 }  
883 }  
884 }  
885 }  
886 }  
887 }  
888 }  
889 }  
890 }  
891 }  
892 }  
893 }  
894 }  
895 }  
896 }  
897 }  
898 }  
899 }  
900 }  
901 }  
902 }  
903 }  
904 }  
905 }  
906 }  
907 }  
908 }  
909 }  
910 }  
911 }  
912 }  
913 }  
914 }  
915 }  
916 }  
917 }  
918 }  
919 }  
920 }  
921 }  
922 }  
923 }  
924 }  
925 }  
926 }  
927 }  
928 }  
929 }  
930 }  
931 }  
932 }  
933 }  
934 }  
935 }  
936 }  
937 }  
938 }  
939 }  
940 }  
941 }  
942 }  
943 }  
944 }  
945 }  
946 }  
947 }  
948 }  
949 }  
950 }  
951 }  
952 }  
953 }  
954 }  
955 }  
956 }  
957 }  
958 }  
959 }  
960 }  
961 }  
962 }  
963 }  
964 }  
965 }  
966 }  
967 }  
968 }  
969 }  
970 }  
971 }  
972 }  
973 }  
974 }  
975 }  
976 }  
977 }  
978 }  
979 }  
980 }  
981 }  
982 }  
983 }  
984 }  
985 }  
986 }  
987 }  
988 }  
989 }  
990 }  
991 }  
992 }  
993 }  
994 }  
995 }  
996 }  
997 }  
998 }  
999 }  
1000 }
```

Variabel record akan error apabila hanya dideklarasikan saja namun tidak diinisialisasi.

- Langkah 5

Tambahkan kode program berikut di dalam scope void main(), lalu coba eksekusi (Run) kode Anda.

```
var mahasiswa2 = ('first', a: 2, b: true, 'last');  
  
print(mahasiswa2.$1); // Prints 'first'  
print(mahasiswa2.a); // Prints 2  
print(mahasiswa2.b); // Prints true
```

```
print(mahasiswa2.$2); // Prints 'last'
```

Apa yang terjadi ? Jika terjadi error, silakan perbaiki. Gantilah salah satu isi record dengan nama dan NIM Anda, lalu dokumentasikan hasilnya dan buat laporannya!

Jawaban :

```
1  void main() {
11     var mahasiswa2 = ('Rendy Putra Kusuma', a: 2241720124, b: true, 'last');
12
13     print(mahasiswa2.$1);
14     print(mahasiswa2.a);
15     print(mahasiswa2.b);
16     print(mahasiswa2.$2);
17 }
18
```

PROBLEMS OUTPUT DEBUG CONSOLE ... Filter (e.g. text, !exclude, \escape)

Rendy Putra Kusuma
2241720124
true
last

Exited.

Kode tersebut menggunakan positional fields dan named fields dari mahasiswa2. \$1 dan \$2 adalah positional fields untuk elemen pertama dan terakhir, sedangkan a dan b adalah named fields.

Tugas

1. Jelaskan yang dimaksud Functions dalam bahasa Dart!

Functions adalah kode yang dapat dipanggil untuk menjalankan satu atau lebih instruksi. Functions digunakan untuk mengelompokkan logika program dan membuat kode lebih mudah dipakai kembali. Functions bisa menerima input (parameter) dan mengembalikan output (nilai return).

2. Jelaskan jenis-jenis parameter di Functions beserta contoh sintaksnya!

Positional Parameter adalah parameter yang harus diberikan secara berurutan saat fungsi dipanggil. Parameter ini wajib diisi jika tidak ada nilai default.

```
void cetakPesan(String pesan, int jumlah) {  
  for (var i = 0; i < jumlah; i++) {  
    print(pesan);  
  }  
}
```

Optional Positional Parameter dideklarasikan dengan tanda kurung siku [] di sekitar parameter. Parameter ini tidak wajib diisi saat fungsi dipanggil, tetapi dapat diberikan nilai default.

```
void cetakPesan(String pesan, [int jumlah = 1]) {  
  for (var i = 0; i < jumlah; i++) {  
    print(pesan);  
  }  
}
```

Named Parameter dapat diberikan tanpa memperhatikan urutan saat pemanggilan fungsi. Parameter ini dideklarasikan dengan tanda kurung kurawa {}

```
void kirimPesan({String pesan = 'Pesan Kosong', int jumlah = 1}) {  
  for (var i = 0; i < jumlah; i++) {  
    print(pesan);  
  }  
}
```

Required Named Parameter hampir sama yang membedakan ditambahkan kata kunci required sehingga parameter harus diisi

```
void kirimPesan({required String pesan, int jumlah = 1}) {  
  for (var i = 0; i < jumlah; i++) {  
    print(pesan);  
  }  
}
```

3. Jelaskan maksud Functions sebagai first-class objects beserta contoh sintaknya!

Functions sebagai first class objects artinya functions dapat disimpan dalam variabel, dilewatkan sebagai parameter, atau dikembalikan dari function lain.

```
void halo(String nama) {  
    print('Halo, $nama!');  
}  
  
void main() {  
    var salam = halo;  
  
    salam('Rendy');  
}
```

```
void olah(int data, Function fungsi) {  
    fungsi(data);  
}  
  
void cetak(int angka) {  
    print('cetak dari $angka adalah ${angka * 2}');  
}  
  
void main() {  
    olah(5, cetak);  
}
```

4. Apa itu Anonymous Functions? Jelaskan dan berikan contohnya!

Anonymous Functions adalah function tanpa nama yang dapat digunakan secara langsung, biasanya ketika hanya dibutuhkan sekali dalam konteks tertentu

```
var list = [1, 2, 3];  
list.forEach((item) {  
    print(item);  
});
```

5. Jelaskan perbedaan Lexical scope dan Lexical closures! Berikan contohnya!

Lexical Scope artinya setiap variabel yang dideklarasikan dalam suatu scope hanya bisa diakses dalam scope tempat variabel itu dideklarasikan. Sedangkan **Lexical Closure** adalah function yang dapat mengakses variabel di luar scope-nya

```
void main() {  
    var name = 'Rendy';  
  
    print(name);  
}
```

Lexical Scope


```
Function buatCounter() {
  int hitungan = 0;
  return () {
    hitungan++;
    return hitungan;
  };
}
```

Run | Debug

```
void main() {
  var counter = buatCounter();
  print(counter());
}
```

Lexical Closure

6. Jelaskan dengan contoh cara membuat return multiple value di Functions!

Return multiple value bisa menggunakan record, list dan map

```
(String, int) getNamaDanUmur() {
  return ('Rendy', 20);
}
```

Run | Debug

```
void main() {
  var (nama, umur) = getNamaDanUmur();
  print('Nama: $nama, Umur: $umur');
}
```

Record

```
List getNamaDanUmur() {
  return ['Rendy', 20];
}
```

Run | Debug

```
void main() {
  var result = getNamaDanUmur();
  print('Nama: ${result[0]}, Umur: ${result[1]}');
}
```

List

```
Map<String, dynamic> getNamaDanUmur() {
  return {'nama': 'Rendy', 'umur': 20};
}
```

Run | Debug

```
void main() {
  var result = getNamaDanUmur();
  print('Nama: ${result['nama']}, Umur: ${result['umur']}');
}
```

Map