

The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer lists files under 'BELAJAR PROGRAMMING ALTERRA', including 'belajar_dart_dasar/basic_dart_1/bin/segitiga_bintang.dart'. The code editor displays the following Dart code:

```
19
20 import 'dart:io';
21
22 void main(List<String> args) {
23   int i, j, k = 0, n=8;
24
25   for (i = 1; i <= n; i++, k = 0) {
26     for (j = 1; j <= n - i; j++) {
27       | stdout.write(" ");
28     }
29     while (k != 2 * i - 1) {
30       | stdout.write("* ");
31     }
32     k++;
33   }
34   stdout.writeln("");
35 }
```

The output of the program is displayed in the 'DEBUG CONSOLE' tab, showing a triangle pattern of asterisks:

```

      *
     * * *
    * * * * *
   * * * * * * *
  * * * * * * * *
 * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
```

The output ends with 'Exited'.

The image shows a screenshot of an IDE interface. On the left is the Explorer panel with a file tree. The main editor displays a Dart file named `hourglass.dart` with the following code:

```
1 import 'dart:io';
2 void main(List<String> args) {
3     int n=6, i, j, z, k = 0;
4
5     for (i = n; i >= 1; --i) {
6         for (z = 0; z < n - i; z++){
7             stdout.write(" ");
8
9
10
11             for (j = i; j <= 2 * i - 1; j++){
12                 stdout.write("* ");
13             }
14
15             for (j = 0; j < i - 1; j++){
16                 stdout.write("* ");
17             }
18
19             stdout.writeln("");
20         }
21
22         for (i = 2; i <= n; ++i, k = 0) {
23             for (z = 1; z <= n - i; z++) {
24                 stdout.write(" ");
```

Below the code editor is a panel with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The DEBUG CONSOLE tab is active, showing the output of the program:

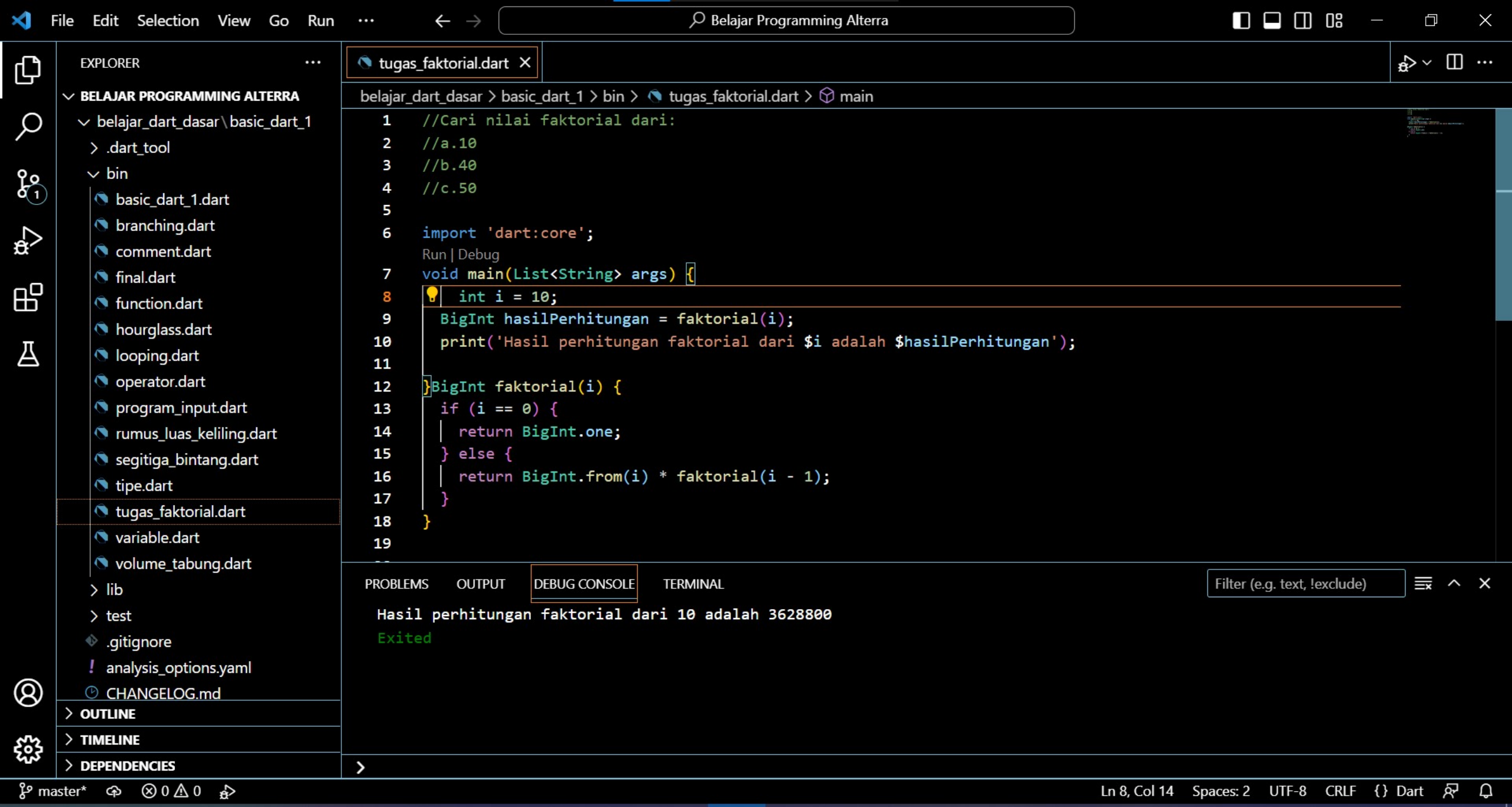
```
* * * * *
Exited
```

The image shows a screenshot of an IDE interface. On the left is a sidebar with icons for Explorer, Search, Source Control, Run and Debug, Extensions, Testing, and Account. The Explorer panel shows a project structure under 'BELAJAR PROGRAMMING ...' with a folder 'belajar_dart_dasar' containing a 'bin' subfolder. The 'bin' folder is expanded, showing several Dart files, with 'hourglass.dart' selected. The main editor area displays the code for 'hourglass.dart' at the path 'belajar_dart_dasar > basic_dart_1 > bin > hourglass.dart > main'. The code is a Dart program that prints an hourglass pattern using nested loops. The pattern consists of 11 lines of asterisks. The code includes comments in Indonesian. The bottom of the IDE has a panel with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The DEBUG CONSOLE tab is active, showing the output '*****' and 'Exited'. The status bar at the bottom indicates 'Ln 9, Col 12', 'Spaces: 2', 'UTF-8', 'CRLF', and the language is 'Dart'.

The screenshot shows an IDE with a file explorer on the left, a code editor in the center, and a debug console at the bottom. The file explorer shows a project named 'BELAJAR PROGRAMMING ...' with a subdirectory 'belajar_dart_dasar' containing a 'bin' folder. The 'bin' folder contains several Dart files, including 'hourglass.dart' which is currently open in the editor. The code in 'hourglass.dart' is a Dart program that prints a star pattern. The debug console at the bottom shows the output of the program, which is a star pattern. The pattern consists of 11 lines of stars, with the number of stars per line decreasing from 11 to 1 in the first half and then increasing back to 11 in the second half. The output is as follows:

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

The debug console also shows the text 'Exited' at the end of the output. The IDE interface includes a sidebar with icons for Explorer, Search, Run and Debug, Source Control, Extensions, Testing, and Account. The top bar shows the current file 'hourglass.dart' and the current directory 'belajar_dart_dasar > basic_dart_1 > bin > hourglass.dart > main'.



EXPLORER

BELAJAR PROGRAMMING ALTERRA

belajar_dart_dasar\basic_dart_1

> .dart_tool

bin

basic_dart_1.dart

branching.dart

comment.dart

final.dart

function.dart

hourglass.dart

looping.dart

operator.dart

program_input.dart

rumus_luas_keliling.dart

segitiga_bintang.dart

tipe.dart

tugas_faktorial.dart

variable.dart

volume_tabung.dart

> lib

> test

.gitignore

! analysis_options.yaml

CHANGELOG.md

> OUTLINE

> TIMELINE

> DEPENDENCIES

tugas_faktorial.dart X

belajar_dart_dasar > basic_dart_1 > bin > tugas_faktorial.dart > main

```
1 //Cari nilai faktorial dari:
2 //a.10
3 //b.40
4 //c.50
5
6 import 'dart:core';
7 void main(List<String> args) {
8   int i = 10;
9   BigInt hasilPerhitungan = faktorial(i);
10  print('Hasil perhitungan faktorial dari $i adalah $hasilPerhitungan');
11
12  BigInt faktorial(i) {
13    if (i == 0) {
14      return BigInt.one;
15    } else {
16      return BigInt.from(i) * faktorial(i - 1);
17    }
18  }
19 }
```

PROBLEMS

OUTPUT

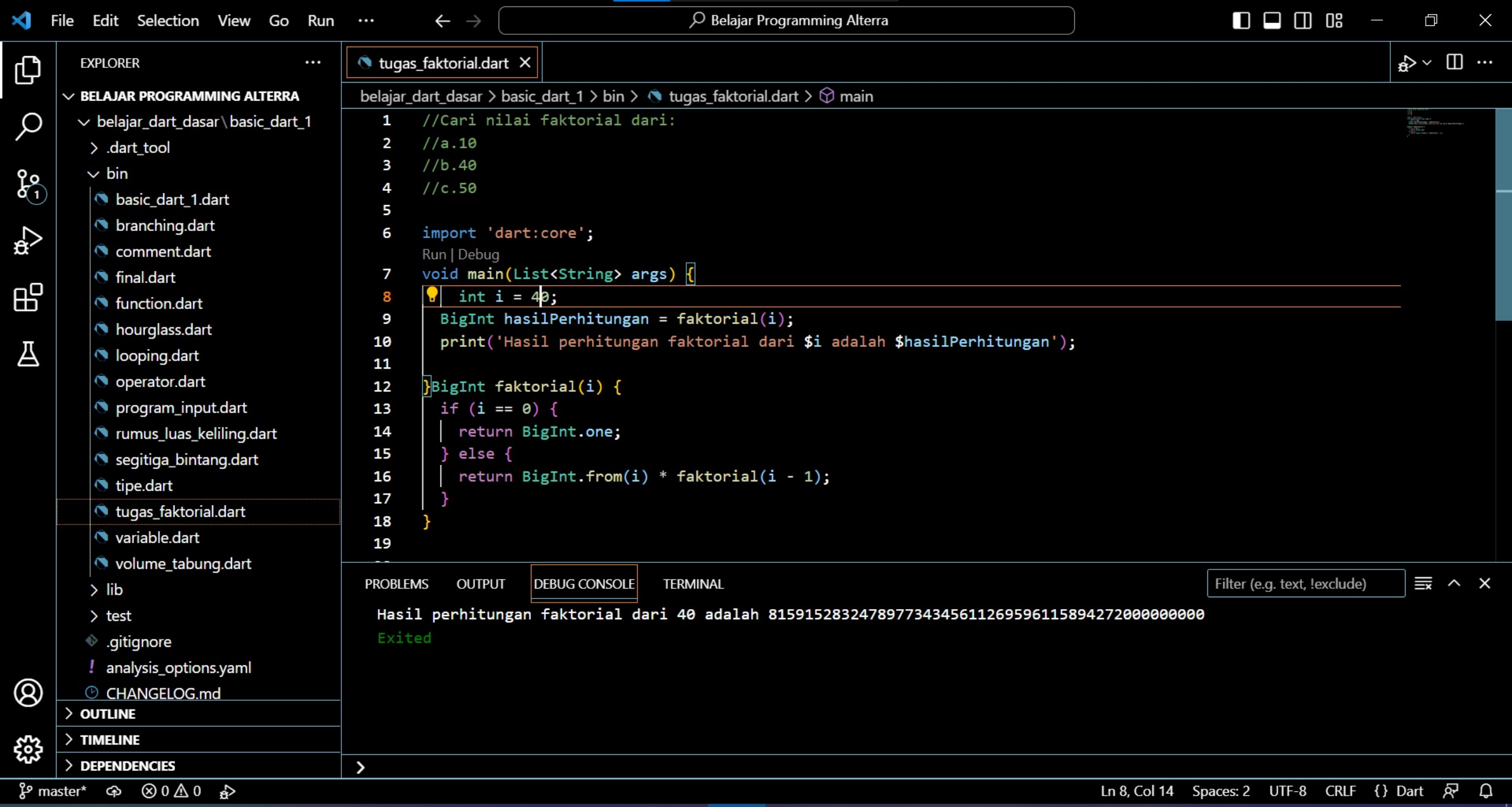
DEBUG CONSOLE

TERMINAL

Filter (e.g. text, !exclude)

```
Hasil perhitungan faktorial dari 10 adalah 3628800
Exited
```

>



EXPLORER

BELAJAR PROGRAMMING ALTERRA

belajar_dart_dasar\basic_dart_1

> .dart_tool

bin

basic_dart_1.dart

branching.dart

comment.dart

final.dart

function.dart

hourglass.dart

looping.dart

operator.dart

program_input.dart

rumus_luas_keliling.dart

segitiga_bintang.dart

tipe.dart

tugas_faktorial.dart

variable.dart

volume_tabung.dart

> lib

> test

.gitignore

! analysis_options.yaml

CHANGELOG.md

> OUTLINE

> TIMELINE

> DEPENDENCIES

tugas_faktorial.dart X

belajar_dart_dasar > basic_dart_1 > bin > tugas_faktorial.dart > main

```
1 //Cari nilai faktorial dari:
2 //a.10
3 //b.40
4 //c.50
5
6 import 'dart:core';
7 void main(List<String> args) {
8   int i = 40;
9   BigInt hasilPerhitungan = faktorial(i);
10  print('Hasil perhitungan faktorial dari $i adalah $hasilPerhitungan');
11
12  BigInt faktorial(i) {
13    if (i == 0) {
14      return BigInt.one;
15    } else {
16      return BigInt.from(i) * faktorial(i - 1);
17    }
18  }
19 }
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

Filter (e.g. text, !exclude)

```
Hasil perhitungan faktorial dari 40 adalah 815915283247897734345611269596115894272000000000
Exited
```

>

The image shows a screenshot of an IDE interface. On the left is the 'EXPLORER' sidebar with a file tree. The tree is expanded to show the file 'tugas_faktorial.dart' under the path 'belajar_dart_dasar > basic_dart_1 > bin'. The main editor area displays the code for 'tugas_faktorial.dart'. The code is in Dart and calculates the factorial of 50. It includes comments in Indonesian, imports 'dart:core', and defines a 'main' function and a recursive 'faktorial' function. The 'DEBUG CONSOLE' tab at the bottom shows the output of the program: 'Hasil perhitungan faktorial dari 50 adalah 30414093201713378043612608166064768844377641568960512000000000000' followed by 'Exited'.

EXPLORER

- ▼ **BELAJAR PROGRAMMING ALTERRA**
 - ▼ belajar_dart_dasar\basic_dart_1
 - > .dart_tool
 - ▼ bin
 - basic_dart_1.dart
 - branching.dart
 - comment.dart
 - final.dart
 - function.dart
 - hourglass.dart
 - looping.dart
 - operator.dart
 - program_input.dart
 - rumus_luas_keliling.dart
 - segitiga_bintang.dart
 - tipe.dart
 - tugas_faktorial.dart**
 - variable.dart
 - volume_tabung.dart
 - > lib
 - > test
 - ! .gitignore
 - ! analysis_options.yaml
 - 🕒 CHANGELOG.md
 - > **OUTLINE**
 - > **TIMELINE**
 - > **DEPENDENCIES**

tugas_faktorial.dart

```
belajar_dart_dasar > basic_dart_1 > bin > tugas_faktorial.dart > main

1 //Cari nilai faktorial dari:
2 //a.10
3 //b.40
4 //c.50
5
6 import 'dart:core';
7 void main(List<String> args) {
8   int i = 50;
9   BigInt hasilPerhitungan = faktorial(i);
10  print('Hasil perhitungan faktorial dari $i adalah $hasilPerhitungan');
11
12  BigInt faktorial(i) {
13    if (i == 0) {
14      return BigInt.one;
15    } else {
16      return BigInt.from(i) * faktorial(i - 1);
17    }
18  }
19 }
```

DEBUG CONSOLE

Hasil perhitungan faktorial dari 50 adalah 30414093201713378043612608166064768844377641568960512000000000000
Exited

