MOBILE APPLICATION 2

PERTEMUAN 2



**DESKRIPSI MATAKULIAH**

**Mata Kuliah ini mempelajari konsep bahasa pemrograman Dart, membuat aplikasi mobile dengan flutter menggunakan bahasa Dart, menerapkan widget pada flutter untuk membuat aplikasi mobile**

**CAPAIAN PEMBELAJARAN**

**Mahasiswa dapat memahami konsep dasar bahasa pemrograman Dart**

**DOSEN PENGAMPU**

**Taufikqurrohman, M.Kom**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**(STMIK) WIDYA PRATAMA**

[Pokok bahasan/bab/pertemuan]

**CAPAIAN PEMBELAJARAN**

**Mahasiswa dapat memahami konsep dasar bahasa pemrograman Dart**

**MATERI PEMBELAJARAN**

**1. Tipe data**

**2. Operator**

**MATERI**

1. **Tipe Data Pada Dart**

Salah satu karakteristik paling mendasar dari bahasa pemrograman adalah himpunan tipe data yang dimiliki oleh bahasa pemrograman. Ini adalah jenis nilai yang dapat direpresentasikan dan dimanipulasi dalam bahasa pemrograman. Dart memiliki tipe-tipe data berikut :

1. **Numbers**

Numbers(Angka) dalam Dart digunakan untuk mewakili literal numerik. Number pada Dart memiliki dua tipe :

- Integer

Adalah bilangan bulat dalam ukuran acak. Tipe data ​**int**​ digunakan untuk mewakili bilangan bulat.

- Double

Double digunakan untuk merepresentasikan angka pecahan. Sintaks untuk menyatakan nomor adalah seperti yang diberikan di bawah ini :

|  |  |
| --- | --- |
| int nama\_var;  double nama\_var; | //mendeklarasikan variable integer // mendeklarasikan a double variable |

Contoh

void main() {

// deklarasi integer

int num1 = 10;

// deklarasi double value

double num2 = 10.50;

print(num1); // print variable num1

print(num2); // print variable num2

}

Dart akan memberikan output seperti ini

10.5

|  |  |  |
| --- | --- | --- |
| **Property** | **Dekripsi** | **Contoh** |
| **hashcode** | Me-return kode hash untuk nilai numerik. | void main() { int n = 5000; print(n.hashCode); } |
| **isFinite** | ***true*** ​jika jumlahnya terbatas;***false***​, jika  tidak terbatas. | void main() { int n = 5000; print(n.isFinite); } |
| **isInfinite** | ***true*** ​jika jumlahnya adalah infinity(tidak terbatas) positif atau tidak terbatasan  negatif; ***false***​, jika tidak infinity. | void main() { int n = 5000; print(n.isInfinite); } |
| **isNegative** | ***true*** ​jika jumlahnya negatif;***false*** ​jika  sebaliknya | void main() { int posNum = 10; int negNum = -10; print(posNum.isNegative); print(negNum.isNegative); } |
| **sign** | Return minus satu, nol atau plus satu  tergantung pada tanda dan nilai numerik  dari angka tersebut. | void main() { int posNum = 10; int negNum = -12; int valZero = 0; print(posNum.sign); print(negNum.sign); print(valZero.sign); } |
| **isEven** | Me-return nilai true jika nomornya adalah  bilangan genap | void main() { int posNum = 10; print(posNum.isEven); |

NOTE:

Dart akan throw exception (memberikan error) jika nilai pecahan diletakkan pada variabel integer.

**Parsing Number**

Fungsi statis ​**parse()**​ memungkinkan penguraian string yang

berisi numerik literal menjadi angka. contoh berikut menunjukkan hal yang sama :

void main() {

print(num.parse('12'));

print(num.parse('10.91'));

}  
Code di atas akan memberikan output

12

10.91

**Property dari Number**

Tabel berikut mencantumkan properti yang dimiliki oleh Number pada Dart.

| **Property** | **Dekripsi** | **Contoh** |
| --- | --- | --- |
| **hashcode** | Me-return kode hash untuk nilai numerik. | void main() { int n = 5000; print(n.hashCode);  } |
| **isFinite** | ***true*** ​jika jumlahnya terbatas;***false***​, jika  tidak terbatas. | void main() { int n = 5000; print(n.isFinite);  } |
| **isInfinite** | ***true*** ​jika jumlahnya adalah infinity(tidak terbatas) positif atau tidak terbatasan  negatif; ***false***​, jika tidak infinity. | void main() { int n = 5000; print(n.isInfinite); } |
| **isNegative** | ***true*** ​jika jumlahnya negatif;***false*** ​jika  sebaliknya | void main() { int posNum = 10; int negNum = -10; print(posNum.isNegative); print(negNum.isNegative); } |
| **sign** | Return minus satu, nol atau plus satu  tergantung pada tanda dan nilai numerik  dari angka tersebut. | void main() { int posNum = 10; int negNum = -12; int valZero = 0; print(posNum.sign); print(negNum.sign); print(valZero.sign); } |
| **isEven** | Me-return nilai true jika nomornya adalah  bilangan genap | void main() { int posNum = 10; print(posNum.isEven);  } |
| **isOdd** | Me-return nilai true jika nomor tersebut adalah angka ganjil. | void main() { int posNum = 10; print(posNum.isOdd);  } |

​ ​​

**Strings**

Tipe data String mewakili urutan karakter. String Dart adalah urutan unit kode UTF 16 ​1​ . Nilai string di Dart dapat direpresentasikan menggunakan kutip tunggal ( ​**„** ‟​) atau ganda ( ​**“ ”**​ ) atau tiga ( ​**„ „ „** ‟ ‟ ‟​ ). String baris tunggal direpresentasikan menggunakan tanda kutip tunggal atau ganda. Kutipan rangkap(kutip tiga) digunakan untuk mewakili string multi-line.

Syntax string pada Dart //kutip satu dan dua untuk string satu line (tunggal)

String variable\_name = 'value'

String variable\_name = ''value''

//kutip rangkap (tiga) untuk string multi-line

String variable\_name = '''line1

line2'''  
String variable\_name= “””line1

Line2”””

Contoh :  
void main() {

String str1 = 'string satu baris';

String str2 = "string satu baris";

String str3 = '''multiline string''';

String str4 = """multiline string""";

print(str1);  
print(str2);  
print(str3);  
print(str4);

}

Code di atas akan memberikan ​*Output*​ seperti ini

string satu baris

string satu baris

multiline string

multiline string

1 UTF-16 (16-bit Unicode Transformation Format) adalah pengkodean karakter yang mampu mengkodekan semua 1.112.064 poin kode yang valid dari Unicode. Pengkodean panjang variabel, karena titik kode dikodekan dengan satu atau dua unit kode 16-bit (juga lihat Perbandingan pengkodean Unicode untuk perbandingan UTF-8, -16 & -32)

**String Interpolation(interpolasi)**

Proses membuat string baru dengan menambahkan nilai ke string statis disebut sebagai penggabungan atau interpolasi. Dengan kata lain, itu adalah proses menambahkan string ke string lain.

Operator plus ( ​**+**​ ) adalah mekanisme yang biasa digunakan untuk menyatukan / menyisipkan string.

Contoh 1

void main() {

String str1 = "hello";

String str2 = "world";

String tambah = str1+str2;

print("string yang digabungkan : ${tambah}");

}

Output yang akan diberikan akan seperti dibawah ini string yang digabungkan : helloworld

Contoh 2

Kita dapat menggunakan " ​**$ { }**​ ", dapat digunakan untuk menginterpolasi nilai ekspresi Dart dalam string. Contoh berikut menggambarkan hal yang sama.

Void main() {

Int n=1+1;

String str1 = "1 tambah 1 sama dengan ${n}";

print(str1);  
String str2 = "2 tambah 2 sama dengan ${2+2}";

print(str2);

}

Code di atas akan memberikan output –

1 tambah 1 sama dengan 2

2 tambah 2 sama dengan 4

**String Properties**

Properti string pada tabel berikut hanya dapat dibaca(​*read-only*​ ​*properties*​).

|  |  |
| --- | --- |
| **Properti dan Deskripsinya** | **Contoh** |
| **codeUnits** Me-return daftar unit kode UTF-16  yang tidak dapat dimodifikasi dari  string ini. | void main() { String str = "Hello"; print(str.codeUnits); } |
| **isEmpty** Return true jika string tidak memiliki  value atau *empty*​(kosong). | void main() { String str = "Hello"; print(str.isEmpty); } |
| **Length** Return panjang(jumlah karakter) pada  string termasuk spasi, tab, dan  karakter baris baru(enter). | void main() { String str = "Hello All"; print("length dari string adalah: ${str.length}"); } |

1. **Operator**

Ada 5 kelompok operator di dalam pemrograman Dart:

**Operator Aritmatika**

* Penjumlahan (+)
* Pengurangan (-)
* Pembagian (/) yang menghasilkan nilai double.
* Pembagian (~/) yang menghasilkan nilai integer.
* Perkalian (\*)
* Sisa bagi atau modulo (%)
* Increment (++)
* Derement (--)

**Operator Relasi**

* Sama dengan (==)
* Tidak samadengan (!=)
* Lebih besar (>)
* Lebih kecil (<)
* Lebih besar sama dengan (>=)
* Lebih kecil sama dengan (<=)

**Operator Logika**

* And (&&)
* Or (||)
* Not (!)

**Operator Bitwise**

* And (&)
* Or (|)
* Xor (^)
* Not (~)

**Operator Ternary (?)**

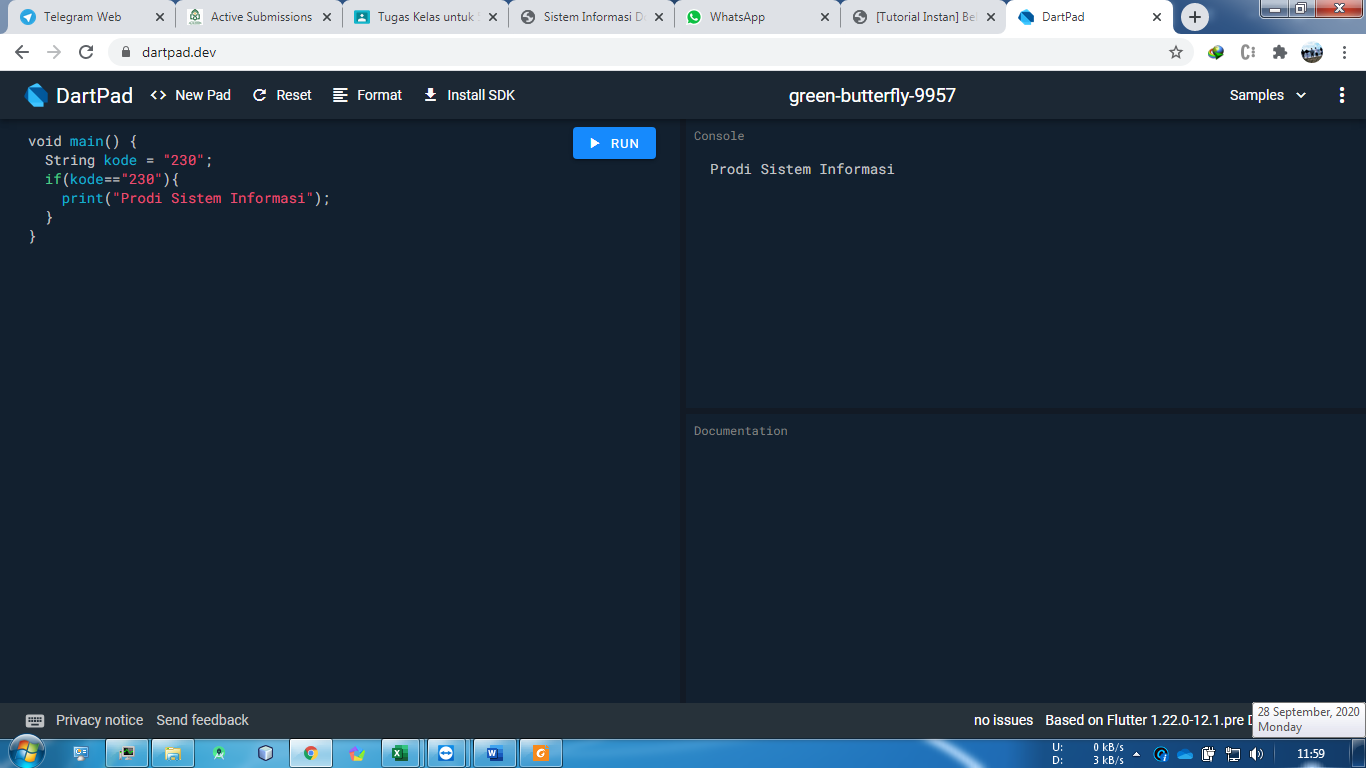
Contoh program untuk operator aritmatika:



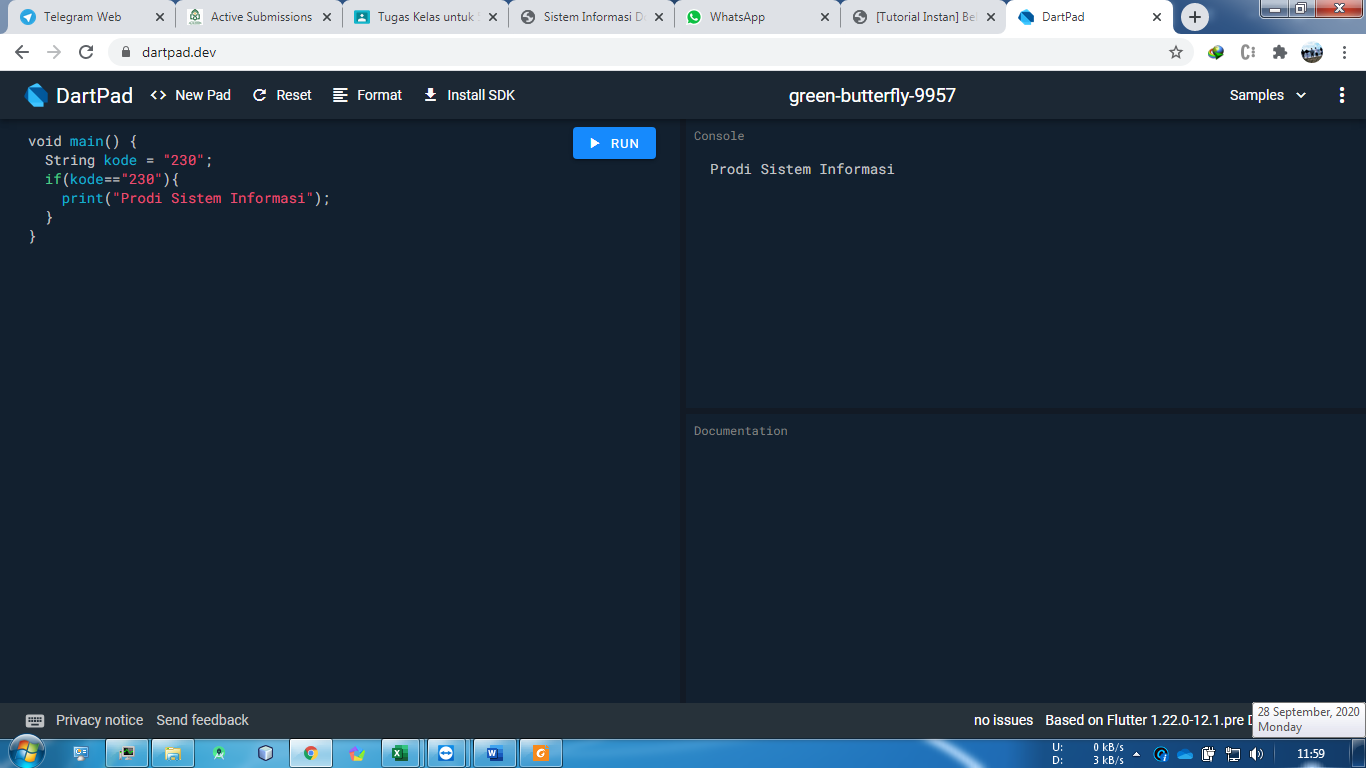
Hasil output:



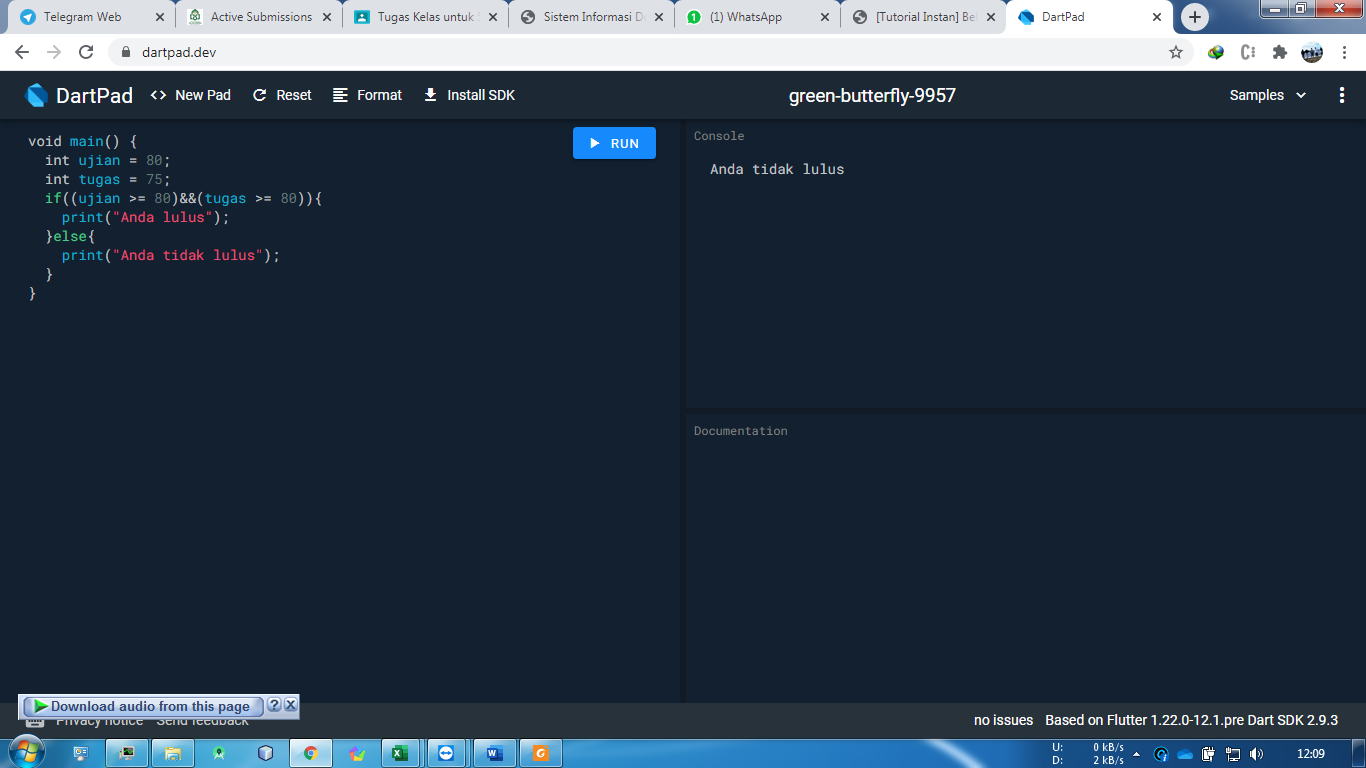
Contoh program untuk operator relasi



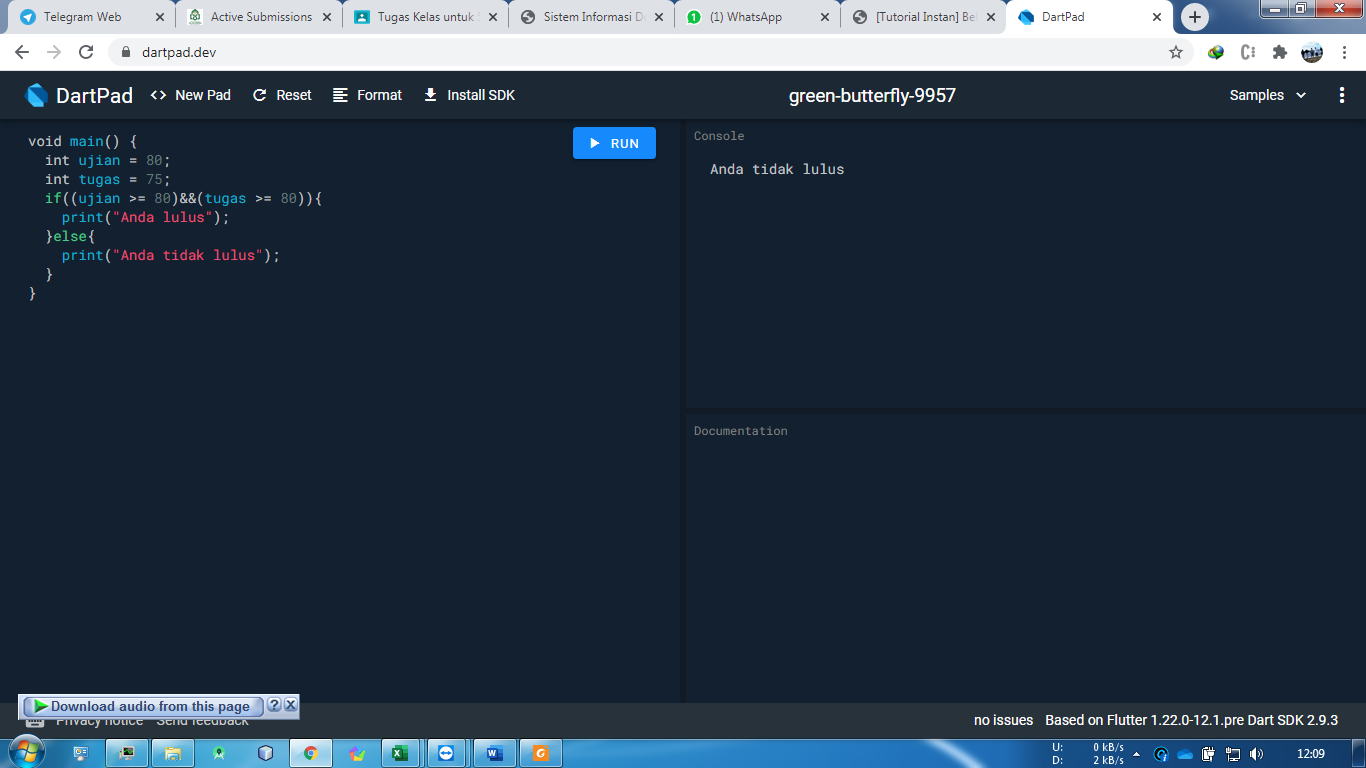
Hasil output



Contoh program untuk operator logika:

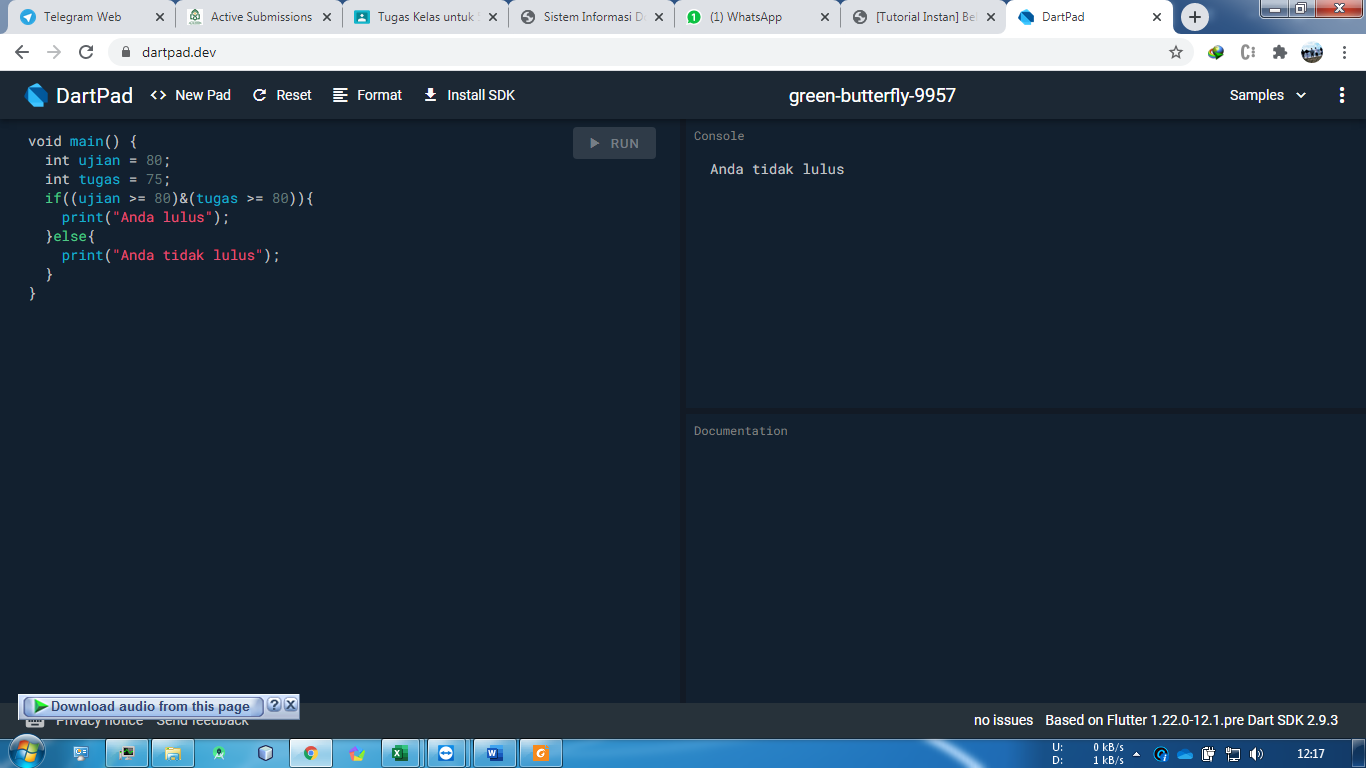


Hasil output:

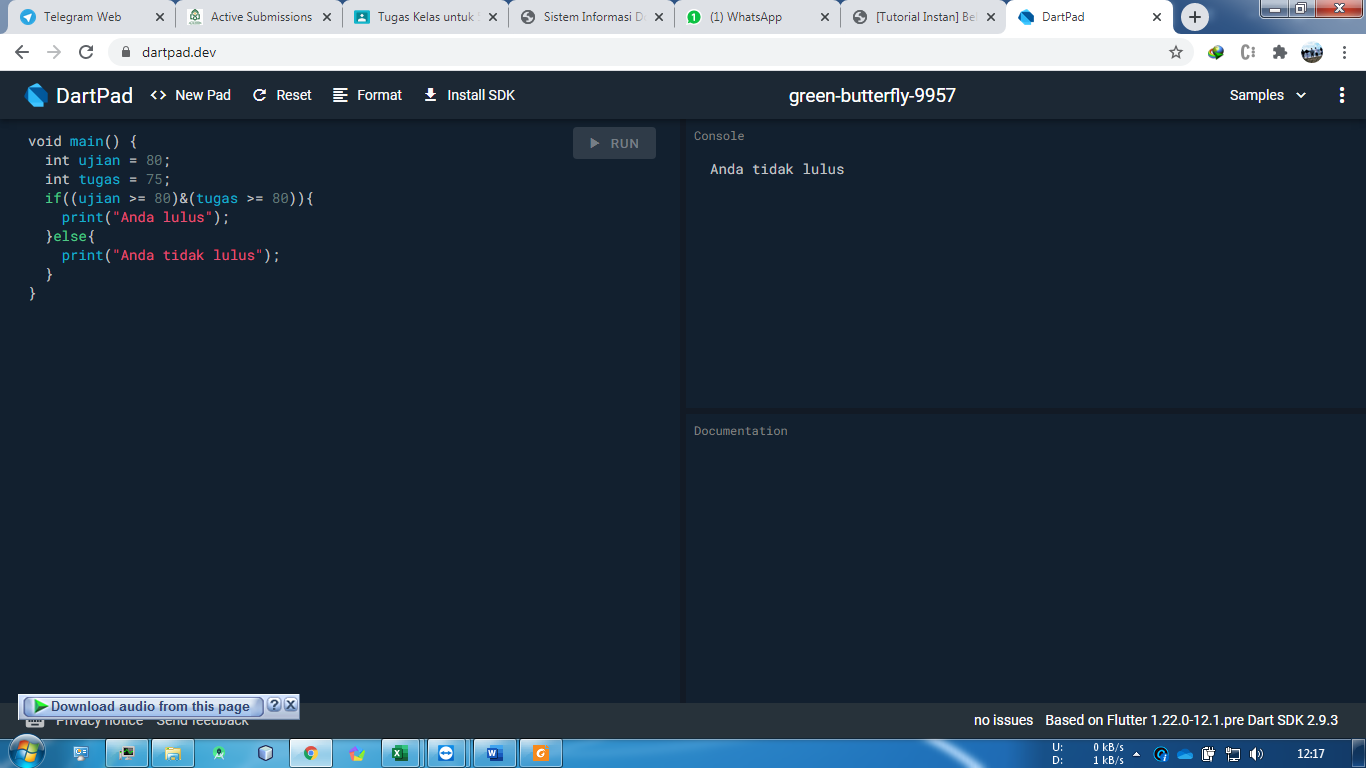


Penjelasan: operator && : bernilai true jika semua kondisi true (kondisi pertama akan dicek, jika false maka akan menghasilkan false tanpa mengecek kondisi kedua, jika true maka kondisi kedua akan dicek)

Contoh program untuk operator bitwise

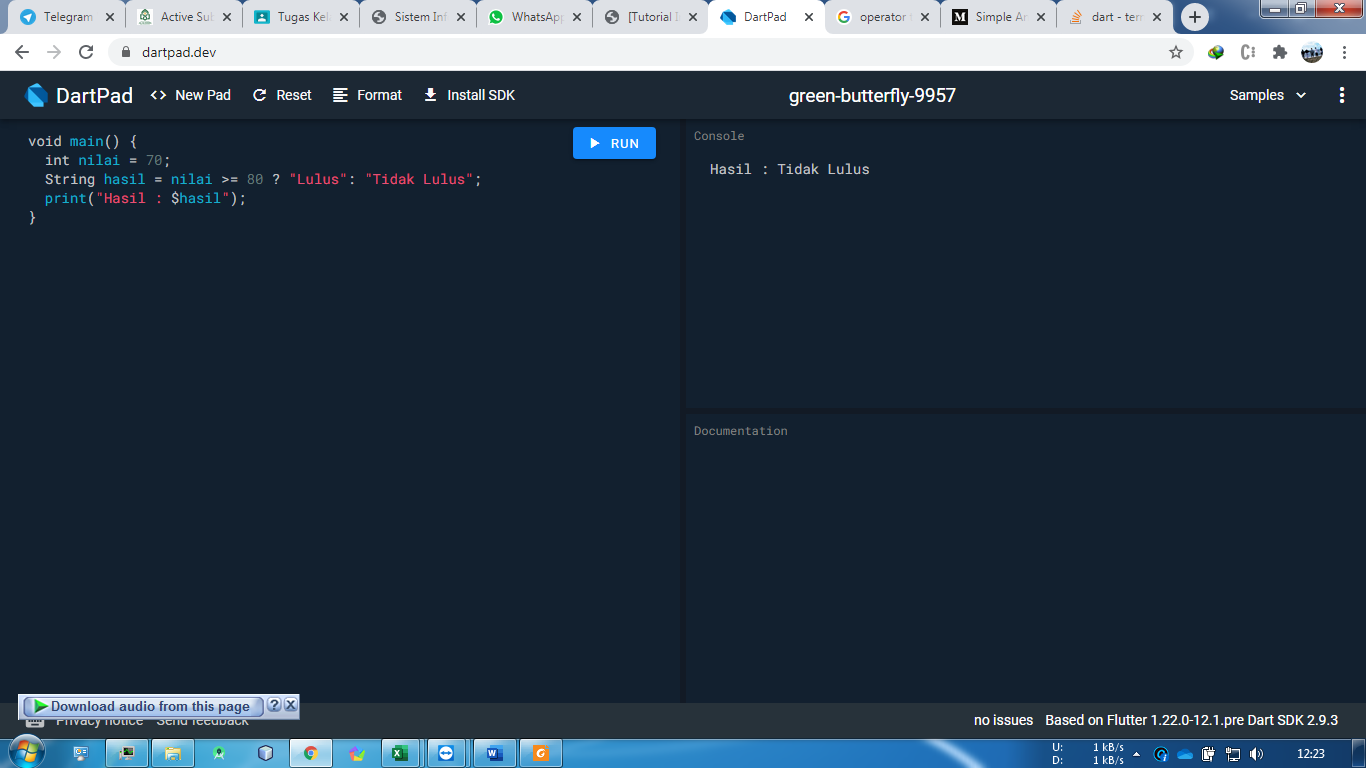


Hasil output:

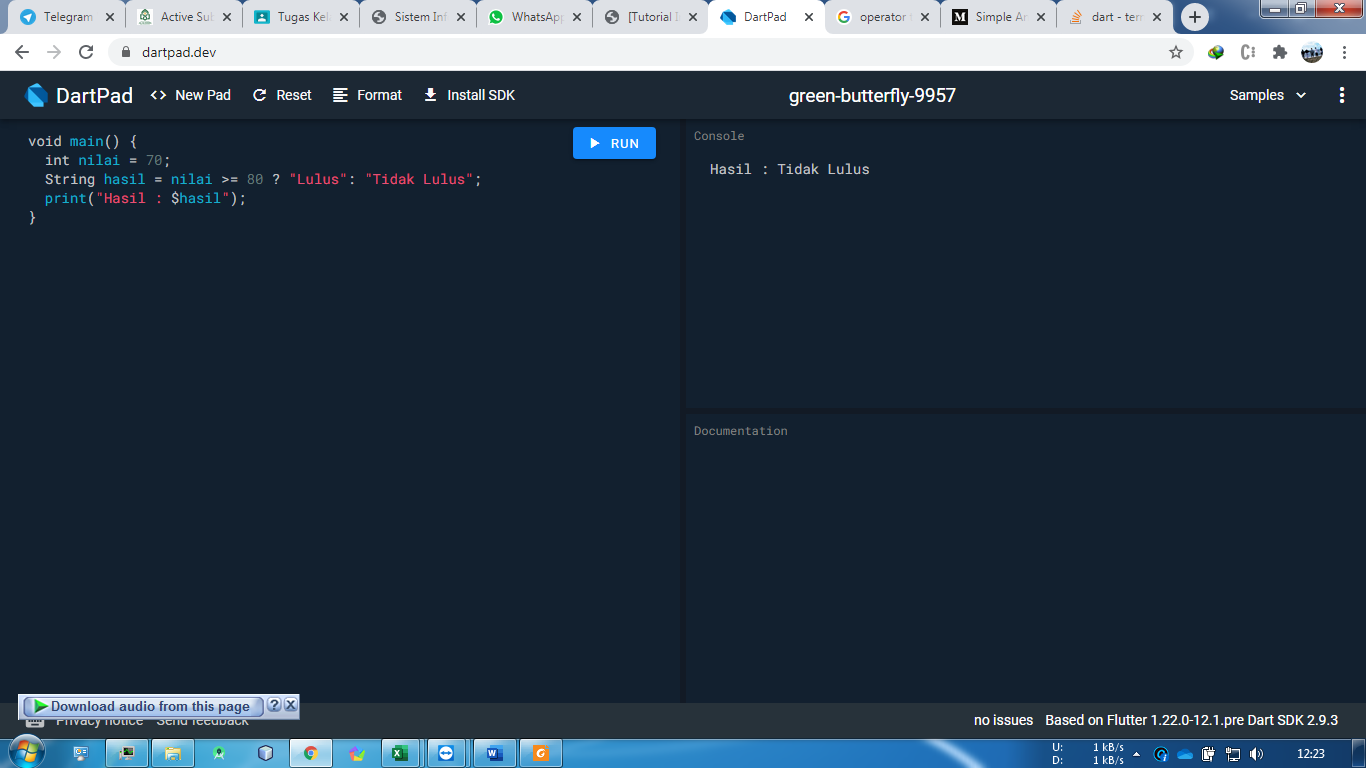


Penjelasan: operator & : bernilai true jika semua kondisi true (kondisi pertama dan kedua akan dicek semua)

Contoh program untuk operator ternary



Hasil output:



Latihan: buat/cari contoh program yang sesuai dengan materi pada modul.

1. Tipe data number (dan method)
2. Tipe data String (dan method)
3. Operator Relasi (salah satu)
4. Operator Logika (salah satu)

**RANGKUMAN**

**[Diisi dengan rangkuman materi yang telah diuraikan]**

**TES FORMATIF**

**[Diisi dengan tes formatif sesuai dengan materi yang telah diuraikan]**

**DAFTAR PUSTAKA**

**[Diisi dengan daftar pustaka yang digunakan dalam menyusun modul pembelajaran]**