**Laporan Workshop Pemrograman Perangkat Bergerak Pertemuan 5 : Function, List, dan Map**

Sebuah gambar berisi Grafis, kartun, teks, papan klip

Deskripsi dibuat secara otomatis

Oleh:

Lutfi Zadeh Filoshof

3123521012

Dosen Pengampu:

Mohammad Robihul Mufid S.ST., M.Tr.Kom.

Teknik Informatika PSDKU Lamongan

Politeknik Elektronika Negeri Surabaya

2025

Tugas Workshop Pemrograman Perangkat Bergerak Pertemuan 5

**Link repository Github:** [**https://github.com/Lutfizadeh/belajar-widget-dan-button/**](https://github.com/Lutfizadeh/belajar-widget-dan-button/)

**Function**

**Arrow Syntax**

Membuat arrow function

void main(List<String> arguments) {

  int number1 = 10;

  int number2 = 20;

  var result = add(number1, number2);

  printNumber(result);

}

int add(int number1, int number2) => number1 + number2;

void printNumber(int number) => print('The number is $number');



**First-Class Object**

Tipe data function yang mengarah ke function lain

void main(List<String> arguments) {

  int number1 = 10;

  int number2 = 20;

  void Function(int) myFunc = printNumber;

  var result = add(number1, number2);

  myFunc(result);

}

int add(int number1, int number2) => number1 + number2;

void printNumber(int number) => print('The number is $number');



Function sebagai nilai kembalian

void main(List<String> arguments) {

  int number1 = 10;

  int number2 = 20;

  int Function(int, int) myFunc = getMathFunc();

  var result = doMathOperator(number1, number2, myFunc);

  printNumber(result);

}

int add(int number1, int number2) => number1 + number2;

void printNumber(int number) => print('The number is $number');

int doMathOperator(int number1, int number2, int Function(int, int) operator) {

  return operator(number1, number2);

}

int Function(int, int) getMathFunc() {

  return add;

}



**Anonymous Function**

Anonymous function sebagai parameter

void main(List<String> arguments) {

  int number1 = 10;

  int number2 = 20;

  int Function(int, int) myFunc = getMathFunc();

  var result = doMathOperator(number1, number2, (int number1, int number2) => number1 + number2);

  printNumber(result);

}

int add(int number1, int number2) => number1 + number2;

void printNumber(int number) => print('The number is $number');

int doMathOperator(int number1, int number2, int Function(int n1, int n2) operator) {

  return operator(number1, number2);

}

int Function(int, int) getMathFunc() {

  return add;

}



Menetapkan nama parameter default

void main(List<String> arguments) {

  int number1 = 10;

  int number2 = 20;

  int Function(int, int) myFunc = getMathFunc();

  var result = doMathOperator(number1, number2, (num1, num2) => num1 + num2);

  printNumber(result);

}

int add(int number1, int number2) => number1 + number2;

void printNumber(int number) => print('The number is $number');

int doMathOperator(int number1, int number2, int Function(int n1, int n2) operator) {

  return operator(number1, number2);

}

int Function(int, int) getMathFunc() {

  return add;

}



**List**

**List of Integer**

Mengakses list dengan index

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5];

  print(numbers[0]);

}



Print panjang list

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5];

  print(numbers.length);

}



Iterasi list dengan bebas

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5];

  for(int i=0; i<numbers.length; i++) {

    print(numbers[i]);

  }

}

A screen shot of a computer

AI-generated content may be incorrect.

Iterasi berurutan

void main(List<String> arguments) {

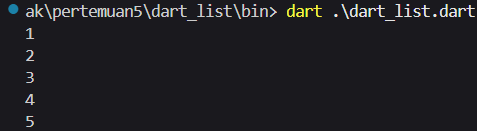
  List<int> numbers = [1, 2, 3, 4, 5];

  for(int element in numbers) {

    print(element);

  }

}



Iterasi dengan forEach

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5];

  numbers.forEach((number) => print(number));

}

A screen shot of a computer

AI-generated content may be incorrect.

Tambah elemen list

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5];

  numbers.add(6);

  numbers.forEach((number) => print(number));

}

A screen shot of a computer

AI-generated content may be incorrect.

Tambah list dengan list lain

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5];

  List<int> numbers2 = [6, 7, 8];

  numbers.addAll(numbers2);

  numbers.forEach((number) => print(number));

}

A computer screen with white text

AI-generated content may be incorrect.

Menyisipkan elemen di list

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5];

  List<int> numbers2 = [6, 7, 8];

  numbers.insert(2, 6);

  numbers.forEach((number) => print(number));

}

A screen shot of a computer

AI-generated content may be incorrect.

Menyisipkan list di list lain

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5];

  List<int> numbers2 = [6, 7, 8];

  numbers.insertAll(2, numbers2);

  numbers.forEach((number) => print(number));

}

A computer screen with white text

AI-generated content may be incorrect.

Menghapus elemen yang ditemukan pertama di list

void main(List<String> arguments) {

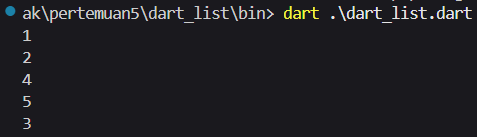
  List<int> numbers = [1, 2, 3, 4, 5, 3];

  List<int> numbers2 = [6, 7, 8];

  numbers.remove(3);

  numbers.forEach((number) => print(number));

}



Menghapus elemen terakhir di list

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5];

  List<int> numbers2 = [6, 7, 8];

  numbers.removeLast();

  numbers.forEach((number) => print(number));

}

A computer screen with white text

AI-generated content may be incorrect.

Menghapus elemen dengan indeks

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5];

  List<int> numbers2 = [6, 7, 8];

  numbers.removeAt(2);

  numbers.forEach((number) => print(number));

}

A screen shot of a computer

AI-generated content may be incorrect.

Menghapus elemen dengan range

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5];

  List<int> numbers2 = [6, 7, 8];

  numbers.removeRange(2, 4);

  numbers.forEach((number) => print(number));

}

A computer screen shot of a black background

AI-generated content may be incorrect.

Menghapus elemen dengan kondisi

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5];

  List<int> numbers2 = [6, 7, 8];

  numbers.removeWhere((number) => number % 2 == 0);

  numbers.forEach((number) => print(number));

}

A screen shot of a computer

AI-generated content may be incorrect.

Melihat isi list dengan contains

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5];

  List<int> numbers2 = [6, 7, 8];

  if(numbers.contains(4)) print("found");

  numbers.forEach((number) => print(number));

}

A screen shot of a computer

AI-generated content may be incorrect.

Mengambil bagian list dengan sublist

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5];

  List<int> numbers2 = numbers.sublist(2);

  numbers2.forEach((number) => print(number));

}

A black screen with white text

AI-generated content may be incorrect.

Mengambil bagian list dengan sublist menggunakan batas

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5];

  List<int> numbers2 = numbers.sublist(2, 4);

  numbers2.forEach((number) => print(number));

}

A black screen with white text

AI-generated content may be incorrect.

Menghapus seluruh elemen list

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5];

*// List<int> numbers2 = numbers.sublist(2, 4);*

  numbers.clear();

  numbers.forEach((number) => print(number));

}



Mengurutkan elemen di list dari yang terkecil

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5, 3];

*// List<int> numbers2 = numbers.sublist(2, 4);*

  numbers.sort();

  numbers.forEach((number) => print(number));

}

A screen shot of a computer

AI-generated content may be incorrect.

Menguji seluruh elemen di list dengan suatu kondisi

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5, 3];

  if(numbers.every((element) => element % 2 == 0)) {

    print("Semua genap");

  } else {

    print("Tidak semua genap");

  }

*// numbers.forEach((number) => print(number));*

}



Menguji list apakah kosong

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5, 3];

  if(numbers.isEmpty) {

    print("Kosong");

  } else {

    print("Tidak kosong");

  }

*// numbers.forEach((number) => print(number));*

}



Menguji list apakah tidak kosong

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5, 3];

  numbers.clear();

  if(numbers.isNotEmpty) {

    print("Tidak Kosong");

  } else {

    print("Kosong");

  }

*// numbers.forEach((number) => print(number));*

}



Mengubah list menjadi set

void main(List<String> arguments) {

  List<int> numbers = [1, 2, 3, 4, 5, 3, 1];

  Set<int> uniqueNumbers = numbers.toSet();

  uniqueNumbers.forEach((number) => print(number));

}

A screen shot of a computer

AI-generated content may be incorrect.

**Map**

Membuat map

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

    'onion': 'bawang bombay',

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  print(dictionary['garlic']);

}



Menampilkan keys dari map

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

    'onion': 'bawang bombay',

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

*// print(dictionary['garlic']);*

  print(dictionary.keys);

}



Menampilkan value dari map

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

    'onion': 'bawang bombay',

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

*// print(dictionary['garlic']);*

  print(dictionary.values);

}



Iterasi keys dari map

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

    'onion': 'bawang bombay',

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  for(int i=0; i<dictionary.length; i++) {

    print(dictionary.keys.elementAt(i));

  }

}

A screen shot of a computer

AI-generated content may be incorrect.

Iterasi keys yang diubah menjadi list

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

    'onion': 'bawang bombay',

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  for(int i=0; i<dictionary.length; i++) {

    print(dictionary.keys.toList()[i]);

  }

}

A screen shot of a computer

AI-generated content may be incorrect.

Iterasi values yang diubah menjadi list

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

    'onion': 'bawang bombay',

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  for(int i=0; i<dictionary.length; i++) {

    print(dictionary.values.toList()[i]);

  }

}

A screen shot of a computer

AI-generated content may be incorrect.

Iterasi keys dan values menggunakan entries

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

    'onion': 'bawang bombay',

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  for(int i=0; i<dictionary.length; i++) {

    print(dictionary.entries.elementAt(i));

  }

}

A screen shot of a computer code

AI-generated content may be incorrect.

Iterasi map menggunakan forEach

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

    'onion': 'bawang bombay',

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  dictionary.forEach((key,value) {

    print('$key: $value');

  });

}

A screen shot of a computer code

AI-generated content may be incorrect.

Menambah element di map

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

    'onion': 'bawang bombay',

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  dictionary['potato'] = 'kentang';

  dictionary.forEach((key,value) {

    print('$key: $value');

  });

}

A screen shot of a computer

AI-generated content may be incorrect.

Mengubah value dari map

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

    'onion': 'bawang bombay',

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  dictionary['onion'] = 'bombay';

  dictionary.forEach((key,value) {

    print('$key: $value');

  });

}

A screen shot of a computer code

AI-generated content may be incorrect.

Menambah element di map dengan element dari map yang lain

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

    'onion': 'bawang bombay',

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  Map<String, String> dictionary2 = {

    'cabbage': 'kol',

    'potato': 'kentang',

    'cucumber': 'mentimun',

    'onion': 'bombay',

  };

  dictionary.addAll(dictionary2);

  dictionary.forEach((key,value) {

    print('$key: $value');

  });

}

A screenshot of a computer program

AI-generated content may be incorrect.

Menambah element di map dengan MapEntry

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

    'onion': 'bawang bombay',

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  Map<String, String> dictionary2 = {

    'cabbage': 'kol',

    'potato': 'kentang',

    'cucumber': 'mentimun',

    'onion': 'bombay',

  };

  List<MapEntry<String, String>> entries = [

    MapEntry('cabbage', 'kol'),

    MapEntry('potato', 'kentang'),

  ];

  dictionary.addEntries(entries);

  dictionary.forEach((key,value) {

    print('$key: $value');

  });

}

A screen shot of a computer

AI-generated content may be incorrect.

Menambah element di map dengan map yang diubah menjadi entry

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

    'onion': 'bawang bombay',

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  Map<String, String> dictionary2 = {

    'cabbage': 'kol',

    'potato': 'kentang',

    'cucumber': 'mentimun',

    'onion': 'bombay',

  };

  List<MapEntry<String, String>> entries = [

    MapEntry('cabbage', 'kol'),

    MapEntry('potato', 'kentang'),

  ];

  dictionary.addEntries(dictionary2.entries);

  dictionary.forEach((key,value) {

    print('$key: $value');

  });

}

A screen shot of a computer

AI-generated content may be incorrect.

Menggunakan method where di entry

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

    'onion': 'bawang bombay',

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  Map<String, String> dictionary2 = {

    'cabbage': 'kol',

    'potato': 'kentang',

    'cucumber': 'mentimun',

    'onion': 'bombay',

  };

  List<MapEntry<String, String>> entries = [

    MapEntry('cabbage', 'kol'),

    MapEntry('potato', 'kentang'),

  ];

  dictionary.addEntries(dictionary2.entries.where((element) => element.key.startsWith('c')));

  dictionary.forEach((key,value) {

    print('$key: $value');

  });

}

A screen shot of a computer code

AI-generated content may be incorrect.

Menampilkan element jika sudah ada di map

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

    'onion': 'bawang bombay',

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  Map<String, String> dictionary2 = {

    'cabbage': 'kol',

    'potato': 'kentang',

    'cucumber': 'mentimun',

    'onion': 'bombay',

  };

  List<MapEntry<String, String>> entries = [

    MapEntry('cabbage', 'kol'),

    MapEntry('potato', 'kentang'),

  ];

  dictionary.addEntries(dictionary2.entries.where((element) => element.key.startsWith('c')));

  String value = dictionary.putIfAbsent('onion', () => 'bombay');

  print(value);

*// dictionary.forEach((key,value) {*

*//   print('$key: $value');*

*// });*

}



Menambahkan element jika belum ada di map

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

    'onion': 'bawang bombay',

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  Map<String, String> dictionary2 = {

    'cabbage': 'kol',

    'potato': 'kentang',

    'cucumber': 'mentimun',

    'onion': 'bombay',

  };

  List<MapEntry<String, String>> entries = [

    MapEntry('cabbage', 'kol'),

    MapEntry('potato', 'kentang'),

  ];

  dictionary.addEntries(dictionary2.entries.where((element) => element.key.startsWith('c')));

  String value = dictionary.putIfAbsent('chili', () => 'cabai');

  print(value);

  dictionary.forEach((key,value) {

    print('$key: $value');

  });

}

A screen shot of a computer

AI-generated content may be incorrect.

Update element yang sudah ada di map

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

    'onion': 'bawang bombay',

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  Map<String, String> dictionary2 = {

    'cabbage': 'kol',

    'potato': 'kentang',

    'cucumber': 'mentimun',

    'onion': 'bombay',

  };

  List<MapEntry<String, String>> entries = [

    MapEntry('cabbage', 'kol'),

    MapEntry('potato', 'kentang'),

  ];

  dictionary.addEntries(dictionary2.entries.where((element) => element.key.startsWith('c')));

  dictionary.update('onion', (value) => 'new $value');

  dictionary.forEach((key,value) {

    print('$key: $value');

  });

}

A screen shot of a computer code

AI-generated content may be incorrect.

Update element yang belum ada di map

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

    'onion': 'bawang bombay',

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  Map<String, String> dictionary2 = {

    'cabbage': 'kol',

    'potato': 'kentang',

    'cucumber': 'mentimun',

    'onion': 'bombay',

  };

  List<MapEntry<String, String>> entries = [

    MapEntry('cabbage', 'kol'),

    MapEntry('potato', 'kentang'),

  ];

  dictionary.addEntries(dictionary2.entries.where((element) => element.key.startsWith('c')));

  dictionary.update('chili', (value) => 'new $value', ifAbsent: () => 'cabai');

  dictionary.forEach((key,value) {

    print('$key: $value');

  });

}

A screen shot of a computer code

AI-generated content may be incorrect.

Mengubah value dengan suatu kondisi (key mengandung huruf ‘c’)

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

*// 'onion': 'bawang bombay',*

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  Map<String, String> dictionary2 = {

    'cabbage': 'kol',

    'potato': 'kentang',

    'cucumber': 'mentimun',

    'onion': 'bombay',

  };

  List<MapEntry<String, String>> entries = [

    MapEntry('cabbage', 'kol'),

    MapEntry('potato', 'kentang'),

  ];

  dictionary.addEntries(dictionary2.entries.where((element) => element.key.startsWith('c')));

  dictionary.updateAll((key, value) => key.contains('c') ? 'Contains c' : value);

  dictionary.forEach((key,value) {

    print('$key: $value');

  });

}

A screen shot of a computer

AI-generated content may be incorrect.

Menghapus element dari suatu map

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

*// 'onion': 'bawang bombay',*

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  Map<String, String> dictionary2 = {

    'cabbage': 'kol',

    'potato': 'kentang',

    'cucumber': 'mentimun',

    'onion': 'bombay',

  };

  List<MapEntry<String, String>> entries = [

    MapEntry('cabbage', 'kol'),

    MapEntry('potato', 'kentang'),

  ];

  dictionary.addEntries(dictionary2.entries.where((element) => element.key.startsWith('c')));

  dictionary.updateAll((key, value) => key.contains('c') ? 'Contains c' : value);

  dictionary.remove('cabbage');

  dictionary.forEach((key,value) {

    print('$key: $value');

  });

}

A screen shot of a computer

AI-generated content may be incorrect.

Menghapus element dari suaru map dengan kondisi tertentu

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

*// 'onion': 'bawang bombay',*

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  Map<String, String> dictionary2 = {

    'cabbage': 'kol',

    'potato': 'kentang',

    'cucumber': 'mentimun',

    'onion': 'bombay',

  };

  List<MapEntry<String, String>> entries = [

    MapEntry('cabbage', 'kol'),

    MapEntry('potato', 'kentang'),

  ];

  dictionary.addEntries(dictionary2.entries.where((element) => element.key.startsWith('c')));

  dictionary.updateAll((key, value) => key.contains('c') ? 'Contains c' : value);

  dictionary.removeWhere((key, value) => key.startsWith('c'));

  dictionary.forEach((key,value) {

    print('$key: $value');

  });

}

A black screen with white text

AI-generated content may be incorrect.

Mengecek element dengan key atau value tertentu

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

*// 'onion': 'bawang bombay',*

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  Map<String, String> dictionary2 = {

    'cabbage': 'kol',

    'potato': 'kentang',

    'cucumber': 'mentimun',

    'onion': 'bombay',

  };

  List<MapEntry<String, String>> entries = [

    MapEntry('cabbage', 'kol'),

    MapEntry('potato', 'kentang'),

  ];

  dictionary.addEntries(dictionary2.entries.where((element) => element.key.startsWith('c')));

  dictionary.updateAll((key, value) => key.contains('c') ? 'Contains c' : value);

  dictionary.removeWhere((key, value) => key.contains('c'));

  dictionary.forEach((key,value) {

    print('$key: $value');

  });

  print(dictionary.containsKey('garlic'));

  print(dictionary.containsValue('tomat'));

}

A black screen with white text

AI-generated content may be incorrect.

Menghapus seluruh element di map

void main(List<String> arguments) {

  Map<String, String> dictionary;

  dictionary = {

*// 'onion': 'bawang bombay',*

    'garlic': 'bawang putih',

    'tomato': 'tomat',

    'carrot': 'wortel',

  };

  Map<String, String> dictionary2 = {

    'cabbage': 'kol',

    'potato': 'kentang',

    'cucumber': 'mentimun',

    'onion': 'bombay',

  };

  List<MapEntry<String, String>> entries = [

    MapEntry('cabbage', 'kol'),

    MapEntry('potato', 'kentang'),

  ];

  dictionary.addEntries(dictionary2.entries.where((element) => element.key.startsWith('c')));

  dictionary.updateAll((key, value) => key.contains('c') ? 'Contains c' : value);

*// dictionary.removeWhere((key, value) => key.contains('c'));*

  dictionary.clear();

  dictionary.forEach((key,value) {

    print('$key: $value');

  });

  print(dictionary.containsKey('garlic'));

  print(dictionary.containsValue('tomat'));

}

A black screen with white text

AI-generated content may be incorrect.

**Mapping**

Menampilkan list hasil mapping

void main(List<String> arguments) {

  List<String> names = [

    'Lutfi',

    'Zadeh',

    'Filoshof',

    'Ade',

  ];

  List<String> initials = names.map((name) => name.substring(0, 2).toUpperCase()).toList();

  for(String initial in initials) {

    print(initial);

  }

}

A screen shot of a computer

AI-generated content may be incorrect.

Menampilkan list hasil mapping yang diubah tipe datanya (String ke int)

void main(List<String> arguments) {

  List<String> names = [

    'Lutfi',

    'Zadeh',

    'Filoshof',

    'Ade',

  ];

  List<String> initials = names.map((name) => name.substring(0, 2).toUpperCase()).toList();

  List<int> nameLengths = names.map((name) => name.length).toList();

  for(int nameLength in nameLengths) {

    print(nameLength);

  }

}

A screen shot of a computer

AI-generated content may be incorrect.

Mapping dengan key dan value yang berbeda tipe data

void main(List<String> arguments) {

*// List<String> names = [*

*//   'Lutfi',*

*//   'Zadeh',*

*//   'Filoshof',*

*//   'Ade',*

*// ];*

*// List<String> initials = names.map((name) => name.substring(0, 2).toUpperCase()).toList();*

*// List<int> nameLengths = names.map((name) => name.length).toList();*

*// for(int nameLength in nameLengths) {*

*//   print(nameLength);*

*// }*

  Map<int, String> days = {

    1: 'Monday',

    2: 'Tuesday',

    3: 'Wednesday',

    4: 'Thursday',

    5: 'Friday',

    6: 'Saturday',

    7: 'Sunday',

  };

  Map<int, String> shortenedDays = days.map((key, value) => MapEntry(key, value.substring(0, 2)));

  print(shortenedDays);

}

A black background with white text

AI-generated content may be incorrect.