The Swift Programming Language

Part 3 – Array & Tuple

iCodeWave Community

Array

What is Array?

- Arrays function to collect values and store them neatly so that they are easy to access and modify
- Array needs to be same data type
- Declaration var a = [data type]()
- Declaration with type annotation var a: [data type] = [ ] (safer)
- Array are modifiable

Array

Make repeating array

- Use Array(…) to repeat something in between
- Example:

```
var example = Array(repeating: "Hello", count: 3)
print(example)
```

```
["Hello", "Hello", "Hello"]
```

- Works perfectly on repeating LazyVGrid or LazyHGrid in SwiftUI

Array

Accessing array

- Use index to accessing element of array
- Example:

```
// Usage example:
let umur2 = 20
checkAge(umur: umur2)
print("\n\n\n\n\n\n\n\n\n")

var namaHari = ["Senin", "Selasa", "Rabu", "Kamis", "Jumat"]
let hariPertama = namaHari[0]
print(hariPertama)
```

```
Senin
```

Array

Add value to array, Part 1

- Use append to add element end of array
- Example:

```
var namaHari = ["Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu"]
namaHari.append("Minggu")
print(namaHari)
```

```
["Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu", "Minggu"]
```

Array

Add value to array, Part 2

- Use insert to add between element in array
- Example:

```
var namaHari = ["Senin", "Selasa", "Kamis", "Jumat", "Sabtu", "Minggu"]
namaHari.insert("Rabu", at: 2)
print(namaHari)
```

```
["Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu", "Minggu"]
```

Array

Count elements in array

- Use count to count elements in array
- Will returning as Integer

Array

Loop through elements in array, Part 1

- Use for loop to iterate
- Example:

```
var namaHari = ["Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu",
    "Minggu"]
for hari in namaHari {
    print("Hari", hari)
}
```

```
Hari Senin
Hari Selasa
Hari Rabu
Hari Kamis
Hari Jumat
Hari Sabtu
Hari Minggu
```

Array

Loop through elements in array, Part 2

- Use .enumerated() and tuple to loop as index
- Example:

```
var namaHari = ["Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu",
    "Minggu"]
for (indeks, hari) in namaHari.enumerated() {
    print("Hari ke-\(indeks + 1): \(hari)")
}
```

```
Hari ke-1: Senin
Hari ke-2: Selasa
Hari ke-3: Rabu
Hari ke-4: Kamis
Hari ke-5: Jumat
Hari ke-6: Sabtu
Hari ke-7: Minggu
```

Summary

By using arrays in Swift, we can easily store, access, and manipulate sets of values in a single data structure.

Tuple

What is tuple?

- Data structure used to store a set of data
- Tuple serves to group multiple values into one compound value
- Values in a tuple can consist of different data types and do not have to be the same
- Tuple are NOT modifiable

Tuple

Example of Code #1 – Simple Tuple Assignment, Part 1

```
104  let http404Error = (404, "Not Found")
105  let (statusCode, statusMessage) = http404Error
106  print("http404Error memiliki kode status \(statusCode)")
107  print("http404Error memiliki pesan error \"\(statusMessage)\"")
```

```
http404Error memiliki kode status 404
http404Error memiliki pesan error "Not Found"
```

You can use different data type to a single tuple.

Example given is Int and String in a single tuple

Tuple

Example of Code #2 – Simple Tuple Assignment, Part 2

```
103
104  let http404Error = (404, "Not Found")
105  let (justTheStatusCode, _) = http404Error
106  print("http404Error memiliki kode status \(justTheStatusCode)")
```

```
http404Error memiliki kode status 404
```

Use _ in tuple to deactive tuple variable

Tuple

Example of Code #3 – Use Indexing in Tuple

```
104  let http404Error = (404, "Not Found")
105  print("http404Error memiliki kode status \(http404Error.0)")
106  print("http404Error memiliki pesan error \"\(http404Error.1)\"")
```

```
http404Error memiliki kode status 404
http404Error memiliki pesan error "Not Found"
```

You can use by using . followed by index start from 0 (just same as array)

Tuple

Example of Code #4 – Declare Variable in Tuple

```
// Deklarasi terlebih dahulu
let http200Status = (statusCode: 200, description: "OK")

// Cetak apa yang udah dideklarasikan
print("http200Status memiliki kode status \(http200Status.statusCode)")
print("http200Status memiliki pesan error \"\(http200Status.description)\"")
```

```
http200Status memiliki kode status 200
http200Status memiliki pesan error "OK"
```

Directly declare variable inside a tuple

Tuple

Example of Code #5 – A Returning Tuple in Function

```
func hitungStatistik(_ data: [Int]) -> (rata: Double, total: Int) {
    let total = data.reduce(0, +)
    let rata = Double(total) / Double(data.count)
    return (rata, total)
}

let data = [10, 20, 30, 40, 50]
let statistik = hitungStatistik(data)
print("Rata-rata: \(statistik.rata), Total: \(statistik.total)")
```

```
Rata-rata: 30.0, Total: 150
```

Tuple vs Array

What’s the difference?

- Tuples have a fixed size and cannot be changed after being declared, while arrays have a changeable size.
- Tuples are useful for storing small amounts of related values, while arrays are suitable for storing larger collections of values.

Summary

Tuple is a powerful feature in Swift that allows you to group multiple values into one. They can be used in a variety of situations, such as returning the value of a function, temporary use of limited data, and more.

Thanks For Your Attendance Today!

iCodeWave Community