

Book Series  
of  
**jago{coding}**  
Learn & Share {Your Code}

[www.jagocoding.com](http://www.jagocoding.com)

# [SWIFT] Tutorial penggunaan Realm (database management) di Ios menggunakan bahasa Swift

Oleh: Candra T. Fahmi

**Realm merupakan salah satu database NoSQL yang dapat digunakan di ios maupun android. jadi Realm dapat dijadikan alternatif untuk mengganti SQLite dan & Core data. Keuntungannya, Realm di klaim lebih mudah dalam penggunaan dan lebih cepat dalam akses dibandingkan dengan SQLite. Dan kali ini saya ingin...**

Realm merupakan salah satu database NoSQL yang dapat digunakan di ios maupun android. jadi Realm dapat dijadikan alternatif untuk mengganti SQLite dan & Core data. Keuntungannya, Realm di klaim lebih mudah dalam penggunaan dan lebih cepat dalam akses dibandingkan dengan SQLite.

Dan kali ini saya ingin berbagi tentang cara management database menggunakan Realm untuk IOS. pada tutorial ini kita akan membuat aplikasi ios sederhana yang memiliki fungsi create, read, update dan delete data pada database.

## **Tools yang dibutuhkan:**

1. xcode versi 7.x
2. Ios minimal versi 8

## **Bahasa yang digunakan:**

1. swift

## **Langkah2:**

1. menambahkan library realm menggunakan pod. Untuk menambahkan library menggunakan pod bisa dilihat disini ["Tutorial menambahkan library menggunakan cocoapod"](#). tambahkan pod 'RealmSwift' kedalam podfile

2. Buat class model

```
import Foundation
import RealmSwift

class ClassModel: Object{

    dynamic var name = ""
    dynamic var address = ""
```

```

    override static func primaryKey() -> String {
        return "name"
    }
}

```

### 3. Buat class database helper

```

import Foundation
import RealmSwift

struct DBHelper {

    static func addObject(obj: ClassModel){
        try! Realm().write(){
            try! Realm().add(obj)
        }
    }

    static func getAll() -> [ClassModel]{
        let objs: Result<ClassModel> = {
            try! Realm().objects(ClassModel)
        }()
        return Array(objs)
    }

    static func deleteByName(name: String){
        let obj = try! Realm().objects(ClassModel).filter("name == %@",
name)
        try! Realm().write(){
            try! Realm().delete(obj)
        }
    }

    static func update(obj: ClassModel){
        try! Realm().write(){
            try! Realm().add(obj, update: true)
        }
    }
}

```

### 4. Penggunaan

penggunaannya sederhana. contoh seperti ini

```
let obj = ClassModel() // membuat object dari classmodel
obj.name = "nama kamu"
obj.address = "alamat kamu"

DbHelper.update(obj) // fungsi untuk menambahkan / mengedit data
```

untuk video lengkapnya bisa dilihat di bawah ini. Terima kasih. semoga bermanfaat :)

<https://www.youtube.com/watch?v=-MKps11vMrU&index=13&list=PLVf9IL-CWZS18SL3T8tNbHdfhKbpAjKq5>

<http://catatancandra.id>

---

## Tentang Penulis



### **Candra T. Fahmi**

Sangat mencintai pemrograman. tertarik dengan hal-hal baru di bidang teknologi.