

All

Feature-Driven Development (FDD)

Dynamic Systems Development Method (DSDM)

Lean Software Development

Kanban

Scrum

Extreme Programming (XP)

Prototyping

Spiral Model

Iterative and Incremental Development

Waterfall Model

- Requirement Analysis (Analisis Kebutuhan)
 - System Design (Perancangan Sistem)
 - System Design (Perancangan Sistem)
 - Testing (Pengujian)
 - Deployment (Pengimplementasian)
 - Maintenance (Pemeliharaan)
-

Function Types - Very Important

- Storing Functions in Variables
 - Use of Functions as Parameters
 - Use of Functions as Return Types
 - Closure
 - Delegate
-

Generic Parameters and Arguments

- Generic Parameter Clause
 - Generic Where Clauses
 - Generic Argument Clause
-

○ Patterns

- Wildcard Pattern
 - Identifier Pattern
 - Value-Binding Pattern
 - Tuple Pattern
 - Enumeration Case Pattern
 - Optional Pattern
 - Type-Casting Patterns
 - Expression Pattern
-

○ Attributes

- Declaration Attributes
- attached
- available
- backDeployed
- discardableResult
- dynamicCallable
- dynamicMemberLookup
- freestanding
- frozen
- GKInspectable
- inlinable
- main
- nonobjc
- NSApplicationMain
- NSCopying
- NSManaged
- objc
- objcMembers
- propertyWrapper
- resultBuilder
- Result-Building Methods
- Result Transformations
- Custom Result-Builder Attributes
- requires_stored_property_inits
- testable
- UIApplicationMain
- unchecked
- usableFromInline
- warn_unqualified_access
- Declaration Attributes Used by Interface Builder
- Type Attributes
- autoclosure
- convention
- escaping
- Sendable
- Switch Case Attributes

unknown

Declarations

- Top-Level Code
- Code Blocks
- Import Declaration
- Constant Declaration
- Variable Declaration
- Stored Variables and Stored Variable Properties
- Computed Variables and Computed Properties
- Stored Variable Observers and Property Observers
- Type Variable Properties
- Type Alias Declaration
- Function Declaration
- Parameter Names
- Parameter Modifiers
- In-Out Parameters
- Borrowing and Consuming Parameters
- Special Kinds of Parameters
- Special Kinds of Methods
- Methods with Special Names
- Throwing Functions and Methods
- Rethrowing Functions and Methods
- Asynchronous Functions and Methods
- Functions that Never Return
- Enumeration Declaration
- Enumerations with Cases of Any Type
- Enumerations with Indirection
- Enumerations with Cases of a Raw-Value Type
- Accessing Enumeration Cases
- Structure Declaration
- Class Declaration
- Actor Declaration
- Protocol Declaration
- Protocol Property Declaration
- Protocol Method Declaration
- Protocol Initializer Declaration
- Protocol Subscript Declaration
- Protocol Associated Type Declaration
- Initializer Declaration
- Failable Initializers
- Deinitializer Declaration
- Extension Declaration
- Conditional Conformance
- Overridden Requirements Aren't Used in Some Generic Contexts
- Protocol Conformance Must Not Be Redundant
- Resolving Explicit Redundancy
- Resolving Implicit Redundancy

- Subscript Declaration
 - Type Subscript Declarations
 - Macro Declaration
 - Operator Declaration
 - Precedence Group Declaration
 - Declaration Modifiers
 - Access Control Levels
-

○ **Statements**

- Loop Statements
 - For-In Statement
 - While Statement
 - Repeat-While Statement
 - Branch Statements
 - If Statement
 - Guard Statement
 - Switch Statement
 - Switch Statements Must Be Exhaustive
 - Switching Over Future Enumeration Cases
 - Execution Does Not Fall Through Cases Implicitly
 - Labeled Statement
 - Control Transfer Statements
 - Break Statement
 - Continue Statement
 - Fallthrough Statement
 - Return Statement
 - Throw Statement
 - Defer Statement
 - Do Statement
 - Compiler Control Statements
 - Conditional Compilation Block
 - Line Control Statement
 - Compile-Time Diagnostic Statement
 - Availability Condition
-

○ **Expressions**

- Prefix Expressions
- In-Out Expression
- Try Operator
- Await Operator
- Infix Expressions
- Assignment Operator
- Ternary Conditional Operator
- Type-Casting Operators
- Primary Expressions
- Literal Expression
- Self Expression

- Superclass Expression
 - Conditional Expression
 - Closure Expression
 - Capture Lists
 - Implicit Member Expression
 - Parenthesized Expression
 - Tuple Expression
 - Wildcard Expression
 - Macro-Expansion Expression
 - Key-Path Expression
 - Selector Expression
 - Key-Path String Expression
 - Postfix Expressions
 - Function Call Expression
 - Implicit Conversion to a Pointer Type
 - Initializer Expression
 - Explicit Member Expression
 - Postfix Self Expression
 - Subscript Expression
 - Forced-Value Expression
 - Optional-Chaining Expression
-

○ **Types**

- Type Annotation
 - Type Identifier
 - Tuple Type
 - Function Type
 - Restrictions for Nonescaping Closures
 - Array Type
 - Dictionary Type
 - Optional Type
 - Implicitly Unwrapped Optional Type
 - Protocol Composition Type
 - Opaque Type
 - Boxed Protocol Type
 - Metatype Type
 - Any Type
 - Self Type
 - Type Inheritance Clause
 - Type Inference
-

○ **Lexical Structure**

- Whitespace and Comments
- Identifiers
- Keywords and Punctuation
- Literals
- Integer Literals

- Floating-Point Literals
 - String Literals
 - Regular Expression Literals
 - Operators
-

○ Advanced Operators

- Bitwise Operators
 - Bitwise NOT Operator
 - Bitwise AND Operator
 - Bitwise OR Operator
 - Bitwise XOR Operator
 - Bitwise Left and Right Shift Operators
 - Shifting Behavior for Unsigned Integers
 - Shifting Behavior for Signed Integers
 - Overflow Operators
 - Value Overflow
 - Precedence and Associativity
 - Operator Methods
 - Prefix and Postfix Operators
 - Compound Assignment Operators
 - Equivalence Operators
 - Custom Operators
 - Precedence for Custom Infix Operators
 - Result Builders
-

○ Access Control

- Modules and Source Files
- Access Levels
- Guiding Principle of Access Levels
- Default Access Levels
- Access Levels for Single-Target Apps
- Access Levels for Frameworks
- Access Levels for Unit Test Targets
- Access Control Syntax
- Custom Types
- Tuple Types
- Function Types
- Enumeration Types
- Raw Values and Associated Values
- Nested Types
- Subclassing in page link
- Constants, Variables, Properties, and Subscripts
- Getters and Setters
- Initializers
- Default Initializers
- Default Memberwise Initializers for Structure Types
- Protocols

- Protocol Inheritance
 - Protocol Conformance
 - Extensions
 - Private Members in Extensions
 - Generics
 - Type Aliases
-

○ **Memory Safety**

- Understanding Conflicting Access to Memory
 - Characteristics of Memory Access
 - Conflicting Access to In-Out Parameters
 - Conflicting Access to self in Methods
 - Conflicting Access to Properties
-

○ **Automatic Reference Counting (ARC)**

- How ARC Works
 - ARC in Action
 - Strong Reference Cycles Between Class Instances
 - Resolving Strong Reference Cycles Between Class Instances
 - Weak References
 - Unowned References
 - Unowned Optional References
 - Unowned References and Implicitly Unwrapped Optional Properties
 - Strong Reference Cycles for Closures
 - Resolving Strong Reference Cycles for Closures
 - Defining a Capture List
 - Weak and Unowned References
-

○ **Opaque and Boxed Types**

- The Problem That Opaque Types Solve
 - Returning an Opaque Type
 - Boxed Protocol Types
 - Differences Between Opaque Types and Boxed Protocol Types
-

○ **Generics**

- The Problem That Generics Solve
- Generic Functions
- Type Parameters
- Naming Type Parameters
- Generic Types
- Extending a Generic Type
- Type Constraints
- Type Constraint Syntax
- Type Constraints in Action
- Associated Types

- Associated Types in Action
 - Adding Constraints to an Associated Type
 - Using a Protocol in Its Associated Type's Constraints
 - Generic Where Clauses
 - Extensions with a Generic Where Clause
 - Contextual Where Clauses
 - Associated Types with a Generic Where Clause
 - Generic Subscripts
-

○ **Protocols**

- Protocol Syntax
 - Property Requirements
 - Method Requirements
 - Mutating Method Requirements
 - Initializer Requirements
 - Class Implementations of Protocol Initializer Requirements
 - Failable Initializer Requirements
 - Protocols as Types
 - Delegation
 - Adding Protocol Conformance with an Extension
 - Conditionally Conforming to a Protocol
 - Declaring Protocol Adoption with an Extension
 - Adopting a Protocol Using a Synthesized Implementation
 - Collections of Protocol Types
 - Protocol Inheritance
 - Class-Only Protocols
 - Protocol Composition
 - Checking for Protocol Conformance
 - Optional Protocol Requirements
 - Protocol Extensions
 - Providing Default Implementations
 - Adding Constraints to Protocol Extensions
-

○ **Extensions**

- Extension Syntax
 - Computed Properties
 - Initializers
 - Methods
 - Mutating Instance Methods
 - Subscripts
 - Nested Types
-

○ **Nested Types**

- Nested Types in Action
 - Referring to Nested Types
-

○ Type Casting

- Defining a Class Hierarchy for Type Casting
 - Checking Type
 - Downcasting
 - Type Casting for Any and AnyObject
-

○ Macros

- Freestanding Macros
 - Attached Macros
 - Macro Declarations
 - Macro Expansion
 - Implementing a Macro
 - Developing and Debugging Macros
-

○ Concurrency

- Defining and Calling Asynchronous Functions
 - Asynchronous Sequences
 - Calling Asynchronous Functions in Parallel
 - Tasks and Task Groups
 - Task Cancellation
 - Unstructured Concurrency
 - Actors
 - Sendable Types
-

○ Advanced - Modifiers

○ Shapes - Modifiers

○ Gestures - Modifiers

○ Events - Modifiers

○ Accessibility - Modifiers

○ Style - Modifiers

○ Navigation Bar - Modifiers

○ List - Modifiers

○ **Image - Modifiers**

○ **Text - Modifiers**

○ **Layout - Modifiers**

- Alignment Guide
 - Anchor Preference
 - Aspect Ratio
 - Background
 - Background Preference Value
 - Container Relative Frame
 - Content Margins
 - Coordinate Space
 - Digital Crown Accessory
 - Edges Ignoring Safe Area
 - Fixed Size
 - Frame
 - Full Screen Cover
 - Grid Cell Anchor
 - Grid Cell Columns
 - Grid Cell Unsized Axes
 - Grid Column Alignment
 - Hidden
 - Ignores Safe Area
 - Labels Hidden
 - Layout Priority
 - Menu Indicator
 - Menu Order
 - Navigation Split View Column Width
 - Overlay
 - Overlay Preference Value
 - Padding
 - Position
 - Presentation Detents
 - Safe Area Padding
 - Scaled to Fill
 - Scaled to Fit
 - Scene Padding
 - Scroll Content Background
 - Scroll Dismisses Keyboard
 - Scroll Indicators
 - Status Bar Hidden
 - Toolbar Background
 - Toolbar Role
 - Transform Anchor Preference
 - Z Index
-

○ Effects - Modifiers

- Accent Color
- Blend Mode
- Blur
- Border
- Brightness
- Button Border Shape
- Clip Shape
- Clipped
- Color Multiply
- Compositing Group
- Content Shape
- Contrast
- Corner Radius
- Drawing Group
- Foreground Color - Deprecated on iOS 17
- Foreground Style
- Grayscale
- Hover Effect
- Hue Rotation
- Invert
- Luminance to Alpha
- Mask
- Matched Geometry Effect
- Opacity
- Preferred Color Scheme
- Projection
- Rotation 3D Effect
- Rotation Effect
- Saturation
- Scale Effect
- Shadow
- Toolbar Color Scheme
- Transform
- Visual Effect

○ Controls - Modifiers

- Actions Sheet
- Alert
- Badge
- Confirmation Dialog
- Context Menu
- Control Size
- Default Focus
- Focused
- Item Provider
- Navigation Destination

- Navigation Document
 - Persistent System Overlays
 - Popover
 - Presentation Drag Indicator
 - Save Area Inset
 - Sheet
 - Submit Scope
 - Tab Item
 - Toolbar
 - Toolbar Title Menu
-

○ Auto Layout

○ Advanced - Views

- Async Image
 - Canvas
 - Capsule
 - Circle
 - Color
 - Container Relative Shape
 - Content Unavailable View
 - Divider
 - Ellipse
 - Empty View
 - Group
 - Image
 - Path
 - Rectangle
 - Rounded Rectangle
 - Timeline View
-

○ Paint - Views

- Angular Gradient
 - Elliptical Gradient
 - Linear Gradient
 - Radial Gradient
-

○ Documentation and Communication

○ Compliance

○ Privacy

○ Continuous Monitoring and Improvement

-
- Version-Specific Insights**
 -
 - Stack Traces**
 -
 - Symbolication**
 -
 - User Feedback Integration**
 -
 - Crash Investigation and Prioritization**
 -
 - Rollbar**
 -
 - Microsoft App Center**
 -
 - Raygun**
 -
 - Bugsnag**
 -
 - Sentry**
 -
 - Instabug**
 -
 - Apple Crash Reports**
 -
 - Firebase Crashlytics**
 -
 - Enjoy Your Life & Keep Learning**
 -
 - App Analytics**
 -
 - Promotion and Marketing Strategies**
 -
 - Cross-Platform Distribution**
 -
 - App Transfer and Account Access**
 -
 - App Store Connect API**

○ Customer Support and Communication

○ Version Control

○ App Archiving

○ App Store Optimization Tools

○ Legal and Compliance

○ Marketing Materials

○ App Store Badges

○ App Ratings and Reviews

○ Monetization Models

○ Pricing

○ Localization

○ App Store Optimization (ASO)

○ Geo-Location Restrictions

○ Compliance

○ App Distribution Policies

○ Security Best Practices

○ Monitoring and Analytics

○ Rollback

○ App Updates

○ App Installation Policies

○ Over-The-Air (OTA) Distribution

○ Deployment through Mobile Device Management (MDM)

○ Enterprise App Security

○ In-House App Distribution

○ Enterprise Developer Program

○ TestFlight Beta Testing

○ Subscriptions

○ In-App Purchases

○ Release Management

○ App Review Guidelines

○ App Submission Process

○ App Store Connect

○ Apple Developer Program

○ Integration with Build Phases

○ Auto-Correction

○ Git Hooks Integration

○ Pre-Commit Hooks

Continuous Improvement

Editor Integration

Visualization

Reporting

Rule Customization

Integration with CI/CD

Project Integration

Installation

Notifications

Version Bumping

Screenshots and Previews

Automated Testing

Beta Distribution

Plugin Integration

Lane Actions

Provisioning Profiles

Code Signing

Automated Builds

Fastfile Configuration

Installation and Setup

Update Management

Device Farm APIs

Cross-Browser Testing

Test Environment Simulation

Integration with Monitoring Tools

Device Farm Maintenance

Cost Optimization

Security Testing

Performance Testing

Test Result Analysis

Test Data Management

Test Framework Compatibility

Real Device Testing

Integration with CI/CD

Parallel Testing

Supported Platforms

Instrumentation Tests

Robo Tests

○ Command Line Testing

○ Test Matrix Configuration

○ Test Types

○ Integration with CI

○ Device Configuration

○ Cost Management

○ Scalability

○ Debugging

○ Logging

○ Triggering Builds

○ Sequential Execution

○ Parallel

○ Pipeline Orchestration

○ Artifact Management

○ Security Considerations

○ Integration with Other Services

○ Monitoring and Analytics

○ Distribution and Deployment

○ Test Automation

○ Build Configuration

○ App Center Account Setup

○ Status Checks

○ Notifications

○ Caching

○ Artifacts

○ Code Signing

○ Matrix Builds

○ Build and Test Jobs

○ iOS Setup

○ GitHub Actions Workflow

○ Deployment

○ Code Signing

○ Testing Integration

○ Build Script

○ Environment Setup

○ Configuration File

○ Travis CI Setup

○ Notifications and Reporting

○ Code Signing

○ Automated Testing

○ Compilation

○ Build

○ Source Code Management

○ Job Configuration

○ Jenkins Configuration

○ Jenkins Installation

○ Cultural Considerations

○ Accessibility in Localization

○ Localizing App Store Metadata

○ Dynamic Content Localization

○ Language Switching

○ Testing Localization

○ Interface Builder Localization

○ Storyboards Localization

○ Bidirectional Language Support

○ Images and Assets Localization

Currency Formatting

Number Formatting

Date and Time Localization

Pluralization Strings

Localizable Strings

Static Code Analysis

Code Signing

Security Headers

Security Training

Keep Dependencies Updated

Code Obfuscation

Handle Errors Securely

Implement Role-Based Access Control (RBAC)

Regular Code Audits

Network Security

[Back to Stage 7 - Security fo Network](#)

Data Encryption

[Back to top - Data Security Layer](#)

Use Secure Coding Practices

Avoid Hardcoding Secrets

Escape Input Data

Input Validation

Face ID

Touch ID

Security Best Practices

Secure Key Storage

Security Tokens

Federated Identity

Single Sign-On (SSO)

Token-Based Authentication

Role-Based Access Control (RBAC)

User Management

Authorization (Otorisasi)

Authentication (Otentikasi)

SSL Pinning

Exception Domains

Typosquatting

Clickjacking

- Credential Stuffing

- Smishing (SMS Phishing)

- Vishing (Voice Phishing)

- Spear Phishing

- Phishing Attacks

- Jailbreaking Exploits

- Malicious Third-Party Libraries

- Eavesdropping

- Denial-of-Service (DoS) Attacks

- Unvalidated Redirects and Forwards

- Insecure Direct Object References (IDOR)

- Broken Authentication

- Security Misconfigurations

- Session Hijacking

- Man-in-the-Middle (MitM) Attacks

- Cross-Site Request Forgery (CSRF)

- Cross-Site Scripting (XSS)

- Injection Attacks

- SQL Injection (SQLi)
 - Code Injection
-

-
- Legal Impact**
 -
 - Application Security Vulnerabilities**
 -
 - App Privacy Policy Violations**
 -
 - Source Code and Application Logs**
 -
 - Financial Information**
 -
 - Application Tokens or Access Keys**
 -
 - User Data Leakage**
 -
 - Security Monitoring**
 -
 - Cipher Suite Implementation in URLSession**
 -
 - ATS (App Transport Security)**
 -
 - HTTPS for Web Communication**
 -
 - Cipher Suite Selection**
 -
 - TLS Version**
 -
 - SSL/TLS Certificates**
 -
 - Communication Encryption**
 -
 - Remote Wipe and Data Erase**
 - Security Measures
 - Protection Against Unauthorized Access
 -
 - Developer Configuration**
 - Info.plist Settings
 - Granular Control
 -

Passcode and Biometric Authentication

- Enhanced Security
 - User Control
-

Automatic Encryption

- Transparent Encryption
 - Developer-Friendly
-

Secure Enclave Integration

- Hardware-Based Security
-

Encryption Classes

- Complete Protection
-

Integration with Keychain Services

- Keychain Access Control
-

File-Level Encryption

- Complete Protection
 - Secure Storage
-

Token-based Authentication

- OAuth Tokens
 - Secure Storage
-

Password AutoFill

- Integration with iOS Features
-

Security Framework Integration:

- APIs
 - Keychain Services API
-

Keychain Items

- Types of Items
 - Attributes
-

Access Groups

- Custom Groups

Security

Keychain Groups

- Sharing
 - Collaboration
-

Access Control

- Authentication
 - Biometric Protection
-

Secure Storage

- Encryption
 - Isolation
-

Secure Coding Practices

iOS CommonCrypto Framework

Secure Data Transmission

End-to-End Encryption

Secure Enclave

Secure Socket Layer (SSL) Deprecation

Certificate Authorities (CAs)

- DigiCert
 - Let's Encrypt
 - GlobalSign
 - Comodo (now Sectigo)
 - GoDaddy
 - Entrust
 - Symantec (now DigiCert)
 - Thawte
 - Network Solutions
 - GeoTrust
 - IdenTrust
 - Buypass
-

○ Digital Signatures

- RSA (Rivest-Shamir-Adleman)
 - DSA (Digital Signature Algorithm)
 - ECDSA (Elliptic Curve Digital Signature Algorithm)
 - EdDSA (Edwards-curve Digital Signature Algorithm)
 - RSA-PSS (RSA Probabilistic Signature Scheme)
 - HMAC with Digital Signatures
 - GOST Signature Algorithm
 - RSASSA-PKCS1-v1_5
 - DSS (Digital Signature Standard)
 - ECDSS (Elliptic Curve Digital Signature Scheme)
 - Detached and Enveloped Signatures
-

○ Hashing

- MD5 (Message Digest Algorithm 5)
 - SHA-1 (Secure Hash Algorithm 1)
 - SHA-256 (Secure Hash Algorithm 256-bit)
 - SHA-3 (Secure Hash Algorithm 3)
 - Whirlpool
 - RIPEMD-160 (RACE Integrity Primitives Evaluation Message Digest)
 - Blake2
 - SHA-256d (Double SHA-256)
 - Skein
 - Keccak
 - BLAKE3
-

○ Asymmetric Encryption

- RSA (Rivest-Shamir-Adleman)
 - DSA (Digital Signature Algorithm)
 - ECDSA (Elliptic Curve Digital Signature Algorithm)
 - Diffie-Hellman (DH)
 - ECDH (Elliptic Curve Diffie-Hellman)
 - ElGamal
 - Curve25519
 - Ed25519
 - McEliece
-

○ Symmetric Encryption

- Advanced Encryption Standard (AES)
- Data Encryption Standard (DES)
- Triple DES (3DES)
- Blowfish
- Twofish
- IDEA (International Data Encryption Algorithm)
- SEED

○ Camellia

○ Post-Release Evaluation

○ App Distribution Closure

○ Documentation and Support

○ Collaboration and Notification Settings

○ Integration with Firebase Crashlytics

○ Managing Multiple Environments

○ Branding

○ Customization

○ Permissions

○ Security

○ Monitoring Distribution Analytics

○ Integration with Continuous Integration (CI)

○ Version Tracking and History

○ Feedback and Bug Reporting

○ Over-the-Air (OTA) Updates

○ Release Notes and Distribution Groups

○ Release Notes and Distribution Groups

○ Uploading Builds to Firebase

-
- Preparing Builds for Distribution**
 - Firebase CLI for Distribution**
 - Integration with Firebase Console**
 - Introduction to Firebase App Distribution**
 - Post-Release Evaluation**
 - Preparing for App Store Submission**
 - Iteration**
 - Feedback Analysis**
 - Release Notes**
 - Documentation**
 - Communication with Testers**
 - Beta Testing Closure**
 - Data Protection**
 - Privacy**
 - Monitoring App Analytics**
 - Over-the-Air (OTA) Updates**
 - Version Control for Beta Builds**
 - Test Plans**
 - Test Cases**
-

-
- Reporting**
 -
 - Bug Tracking**
 -
 - Beta Test Feedback Collection**
 -
 - User Recruitment and Onboarding**
 -
 - App Review Guidelines**
 -
 - Internal Testing vs. External Testing**
 -
 - Build Distribution with TestFlight**
 -
 - Preparing for Beta Testing**
 -
 - Apple's TestFlight**
 -
 - Introduction to Beta Testing**
 -
 - Best Practices for Appium Testing**
 -
 - Appium and Continuous Integration**
 -
 - Appium Plugins and Extensions**
 -
 - Integration with Test Frameworks**
 -
 - Parallel Testing with Appium**
 -
 - Appium Commands and API**
 -
 - Using Appium Inspector**
 -
 - Cross-Platform Testing**
 -
 - Hybrid App Testing**

- Native App Testing

- Appium Desired Capabilities

- Appium Architecture

- Introduction to Appium

- EarlGrey Best Practices

- Parameterized Testing with EarlGrey

- Integration with Continuous Integration

- Debugging

- Error Handling

- Synchronization Strategies

- Grey Box Testing

- Matchers

- Custom Assertions

- Features of EarlGrey

- Introduction to EarlGrey

- Exploratory UI Testing

- UI Testing on Different Devices and Resolutions

- Parallel UI Testing

UI Test Code Review

Testing Accessibility

Handling Alerts and Pop-ups

Parameterized Screenshots

Continuous Integration (CI) with UI Tests

Screenshots and Recordings

Mocking

Test Data

Real Devices

Simulator

Handling Asynchronous Operations

UI Test Best Practices

Parameterized UI Tests

Assertions in UI Testing

Writing UI Test Code

Inspecting UI Elements

Playback

Recording

XCUITest Framework

○ Test Automation Patterns

○ Test Automation Patterns

○ Handling Edge Cases

○ Exploratory Testing

○ Code Review for Tests

○ Testing Dependency Injection

○ Testing in Different Environments

○ Testing Legacy Code

○ Best Practices for Unit Testing

○ UI Testing vs. Unit Testing

○ Parameterized Tests

○ Continuous Integration (CI) with Tests

○ Test-Driven Development (TDD)

○ Code Coverage

○ Stubbing

○ Mocking

○ Asynchronous Testing

○ Test Suites

○ Writing Test Methods

-
- **Test Case Structure**

 - **XCTest Framework**

 - **Introduction to Unit Testing**

 - **E-commerce Support (if applicable)**

 - **Monitoring Acquisition Channels**

 - **Page Performance Measurement (Screen Tracking)**

 - **Integration with Other Platforms**

 - **Custom Dimensions and Metrics**

 - **A/B Testing**

 - **Integration with Firebase**

 - **Retention and User Monitoring**

 - **Event Tracking**

 - **SDK Integration**

 - **Get Tracking ID**

 - **Create a Google Analytics Account**

 - **Integration with Analytics**

 - **Prioritizing Fixes**

 - **Real-time Crash Monitoring**

 - **Error Insights**

○ Crash Reporting

○ Synthetic Monitoring

○ Alerts and Notifications

○ Infrastructure Monitoring

○ Transaction Tracing

○ Error Tracking

○ Application Monitoring

○ New Relic Setup

○ Performance Monitoring

○ Other Platform

○ Conversion Tracking

○ User Engagement

○ User Properties

○ Event Tracking

○ Firebase Integration

○ Crash and Bug Analytics

○ Analytics Presentation

○ Documentation

○ Enhancements

App Updates

Data Security

Privacy

Understanding Business Metrics

A/B Testing Utilization

Measurement of Feature Success

Passive User Analytics

Active User Analytics

Application Performance Monitoring

Conversion Analysis

Funnel

User Retention and Visits

Custom Event Tracking

Installation and Usage of Analytics SDK

Analytics Platform for iOS

Cross-Functional Teams

Continuous Deployment (CD)

Continuous Integration (CI)

○ Versioning

○ Communication

○ Documentation

○ Centralized Configuration

○ Unloading

○ Dynamic Loading

○ Business Logic

○ Decoupling UI

○ Testing Each Module

○ Reusability

○ Dependency Injection

○ Clear Interfaces

○ Encapsulation

○ Abstraction

○ Independent Modules

○ Modular Code Structure

○ Maintenance

○ Updating

○ Optimizing XCFramework

○ Troubleshooting

○ Debugging

○ XCframework Documentation

○ Versioning and Compatibility

○ Integration of XCframework in iOS Project

○ XCframework Distribution

○ Creating XCframework

○ XCframework Structure

○ Purpose of Using XCframework

○ Understanding XCframework

○ User Interface Testing Framework

○ Local Notifications Framework

○ Firebase Framework

○ Networking Frameworks

○ StoreKit Framework

○ WebKit Framework

○ UIKit Dynamics Framework

○ AVFoundation Framework

○ MapKit Framework

-
- **Core Location Framework**

 - **Core Data Framework**

 - **Foundation Framework**

 - **UIKit Framework**

 - **Security Considerations for Swift iOS Apps**

 - **Scripting and Automation for Swift iOS Development**

 - **Dependency Management for Swift iOS Apps**

 - **Advanced Swift iOS Topics**

 - **Optimization for Swift iOS Apps**

 - **Debugging Swift Dynamic Libraries**

 - **Best Practices for Swift iOS Development**

 - **Testing in Swift iOS Apps**

 - **Documentation for Swift iOS Apps**

 - **Swift Versioning**

 - **Integration with iOS Applications in Swift**

 - **Distribution for iOS Apps**

 - **Building the iOS Dynamic Library (Framework) in Swift**

 - **iOS Build Process for Swift Frameworks**

 - **Swift Header Files and Module Declarations**

-
- Adding Swift Source Code

 - Setting Up an iOS Dynamic Library (Framework) Project in Swift

 - Introduction to Dynamic Libraries (Frameworks) in Swift

 - Security Considerations for iOS Apps

 - Dependency Management for iOS Apps

 - Dependency Management for iOS Apps

 - Advanced iOS Topics

 - Optimization for iOS Apps

 - Debugging iOS Libraries

 - Best Practices for iOS Development

 - Testing in iOS Apps

 - Documentation for iOS Apps

 - iOS Versioning

 - Integration with iOS Applications

 - Distribution for iOS

 - Building the iOS Static Library

 - iOS Build Process

 - iOS Header Files

 - Adding iOS-specific Source Code

- **Setting Up an iOS Static Library Project**

- **Introduction to Static Libraries in iOS**

- **Stay Updated**

- **Keep Dependencies Simple**

- **Code Reviews**

- **Continuous Integration**

- **Security Considerations**

- **Localization**

- **Internationalization**

- **Interface Builder and Storyboards**

- **Performance Optimization**

- **Dependency Management**

- **Build Configurations**

- **Documentation**

- **Testing**

- **Protocol-Oriented Programming**

- **Use Optionals Wisely**

- **Memory Management**

Error Handling

Version Control

Code Comments

Naming Conventions

Code Organization

Customization

Contributing

Versioning

Backup

Monitoring and Analytics

Caching and Offline Support

Removing Unused Code

Error Handling and Logging

Community and Support

Performance Impact

Security Considerations

Updating Libraries

Testing

Dependency Management

Integration Methods

License Compliance

Checking Compatibility

Reviewing Documentation

Choosing Third-Party Libraries

Storing Dependencies in the Repository

Documenting Dependency Versions

Managing Version Conflicts

Backing Up Dependency Configurations

Running Tests and Verifying Compatibility

Locking Versions in Production

Reading Release Notes

Using Dependabot or Automated Updates

Regularly Updating Dependencies

Using Version Ranges

Understanding Semantic Versioning (SemVer)

- Major
 - Minor
 - Patch
-

Troubleshooting SPM Issues

- **Best Practices for SPM**

- **Working with Package Collections**

- **Local Package Dependencies**

- **Cross-Platform Development**

- **Creating and Publishing Packages**

- **Continuous Integration with SPM**

- **Package Naming Conventions**

- **Package Initialization and Update**

- **Handling Binary Dependencies**

- **Customizing Build Settings**

- **Swift Package Manager Commands**

- **Dependency Resolution Strategies**

- **Handling Private Packages**

- **Package Mirroring**

- **Package Resolution**

- **Versioning with SPM**

- **Creating a Package.swift File**

- **Adding Dependencies in Xcode:**

- **Troubleshooting Carthage Issues**

○ Best Practices for Carthage

○ Handling Private Repositories

○ Carthage Configuration Files

○ Parallelizing Builds

○ Creating and Publishing Frameworks

○ Configuring Build Schemes

○ Updating Dependencies

○ GitHub Releases and Tags

○ Handling Binary Dependencies

○ Carthage Folder Structure

○ Carthage Commands

○ Versioning with Cartfile.resolved

○ Integration with Xcode

○ Building Frameworks

○ Dependency Resolution

○ Specifying Versions

○ Creating a Cartfile

○ Installation and Setup

○ Troubleshooting CocoaPods Issues

- **Best Practices for CocoaPods**

- **Using CocoaPods Plugins**

- **Handling Pod Conflicts**

- **Analyzing CocoaPods Output**

- **Understanding Podfile.lock**

- **Creating Your Own Pod**

- **Using Subspecs**

- **Podfile Hooks**

- **Handling Private Pods**

- **Podfile Configurations**

- **Integration with Xcode**

- **Updating Dependencies**

- **Installing Pods**

- **Dependency Resolution**

- **Specifying Versions**

- **Creating a Podfile**

- **Installation and Setup**

- **Documentation and Resources**

- **Continuous Improvement and Iteration**

-
- Security Considerations

 - Integration with Performance Monitoring Tools

 - Optimizing Performance with MetricKit

 - Interpreting Metric Trends

 - Error Handling and Troubleshooting

 - Handling Metric Thresholds

 - Integration with Logging Systems

 - Customizing Metric Reporting

 - Analyzing Metric Data

 - Handling Metric Updates

 - MetricKit Data Payload

 - Metric Identifiers

 - Enabling Metric Collection

 - Introduction to MetricKit

 - Simulating Network Conditions

 - Real-time Monitoring

 - Identifying Performance Bottlenecks

 - Exporting and Sharing Data

 - Network Connections Overview

○ Advanced Filtering and Sorting

○ Error Detection and Handling

○ Resource Loading Analysis

○ Bandwidth Throttling

○ SSL/TLS Handshake Analysis

○ WebSocket Inspection

○ Connection Tracing

○ Latency Analysis

○ Request and Response Inspection

○ Network Usage Instrument

○ Sampai sini broooo, lanjutkan perjuanganmu

○ Real-time Memory Monitoring and Debugging

○ Advanced Memory Analysis Techniques

○ Comparative Memory Analysis

○ Simulating Memory Scenarios

○ Memory Management Best Practices

○ Applying Smart Pointers

○ Caching Strategies

○ Optimizing Image Memory Usage

○ Heap Allocations Stack Trace

○ Automatic Reference Counting (ARC) Analysis

○ Memory Graph

○ Memory Tagging

○ Analyzing Object References

○ Understanding Retain Cycles

○ Leaks Instrument

○ Heapshot Analysis

○ Heap and Allocations Summary

○ Allocations Instrument

○ Memory Usage Instrument

○ Introduction to Memory Analysis

○ Case Studies and Practical Applications

○ Best Practices for CPU Performance

○ Real-time Monitoring and Debugging

○ Interpreting Results and Recommendations

○ Advanced Instrument Features

○ Comparative Analysis

- Simulating CPU Scenarios

- Energy Impact Analysis

- Core Animation Instrument

- Concurrency and GCD Analysis

- Thread Sanitizer

- Thread States and Analysis

- Optimizing CPU-bound Operations

- Understanding CPU Instructions

- Sampling and Instrument Configuration

- Call Tree Analysis

- Time Profiler Instrument

- CPU Usage Instrument

- Launching Instruments

- Introduction to Instruments

- Best Practices and Design Patterns

- Updates and Maintenance

- Security and Performance

- Continuous Performance Monitoring

- App Thinning

-
- Adaptive Layout and Size Classes**
 -
 - Size Optimization**
 -
 - Battery Efficiency**
 -
 - App Launch Time**
 -
 - Database Optimization**
 -
 - Responsiveness and User Experience**
 -
 - Performance Measurement Tools**
 -
 - Data Caching**
 -
 - UI Rendering Performance**
 -
 - Multithreading and Concurrency**
 -
 - Image and Asset Optimization**
 -
 - Network Performance**
 -
 - Memory Management**
 -
 - Profiling and Debugging**
 -
 - Code Optimization**
 -
 - Archiving and Distributing Targets**
 -
 - Duplicating Targets**
 -
 - Target Actions**
 -
 - App Extensions and Targets**

-
- **Target-specific Info.plist Values**

 - **Configuration Files for Targets**

 - **Target-specific Build Settings**

 - **Testing Targets**

 - **Conditional Compilation**

 - **Build Phases for Targets**

 - **Source Code and Resources for Targets**

 - **Info.plist for Targets**

 - **Dependencies Between Targets**

 - **Schemes and Targets**

 - **Target Membership**

 - **Build Configurations for Targets**

 - **Creating Targets**

 - **Default Targets**

 - **Types of Targets**

 - **Introduction to Targets**

 - **Best Practices for Schemes**

 - **Continuous Integration with Schemes**

 - **Managing Scheme Versions**

-
- **Sharing Schemes**

 - **Xcode Scheme Editor**

 - **Managing Scheme Dependencies**

 - **Distributing Apps with Schemes**

 - **Analyzing Performance with Instruments**

 - **Running and Testing Configurations**

 - **Scheme Environments**

 - **Specifying Executable**

 - **Editing Scheme Configurations**

 - **Post-actions**

 - **Pre-actions**

 - **Setting Build Targets**

 - **Configurations and Build Settings**

 - **Build Configuration in Schemes**

 - **Creating Custom Schemes**

 - **Default Schemes**

 - **Accessing Schemes in Xcode**

 - **Introduction to Schemes**

 - **Best Practices for Build Rules**

- **Dynamic Build Rules**

- **Build Rule Output Formats**

- **Build Rules and Frameworks**

- **Conditional Build Rules**

- **Integration with External Tools**

- **Importing and Exporting Build Rules**

- **Debugging Build Rule Issues**

- **Scripting in Build Rules**

- **Build Rule Variables**

- **Creating Custom Build Rules**

- **Build Rule Conditions**

- **Build Rule Precedence**

- **Output Files and Build Rules**

- **Scripted Build Rules**

- **Source Files and Build Rules**

- **File Patterns and Build Rules**

- **Build Rule Types**

- **Accessing Build Rules in Xcode**

○ Introduction to Build Rules

○ Debugging Build Phase Issues

○ Resource Processing in Build Phases

○ Build Phases

○ Static Libraries

○ Post-actions

○ Pre-actions

○ Scripting in Build Phases

○ Build Phase Output Files

○ Custom Build Phases

○ Build Phase Ordering

○ Build Phase Dependencies

○ Frameworks and Libraries in Build Phases

○ Header Files in Build Phases

○ Shell Scripting in Build Phases

○ Run Script Build Phase

○ Embed Frameworks Build Phase

○ Copy Bundle Resources Build Phase

○ Link Binary With Libraries Build Phase

○ **Compile Sources Build Phase**

○ **Build Phase Types**

○ **Accessing Build Phases in Xcode**

○ **Introduction to Build Phases**

○ **Xcode Schemes and Build Configurations**

○ **Performance Optimization Settings**

○ **Handling Build Errors**

○ **Bitcode**

○ **App Extensions and Frameworks**

○ **Resource-related Settings**

○ **Build Phase Scripts**

○ **Using xcconfig Files**

○ **Build Settings for Different Environments**

○ **Conditional Build Settings**

○ **Deployment Target**

○ **Debugging and Optimization Settings**

○ **Code Signing Settings**

○ **Linker Settings**

○ **Compiler Settings**

-
- **Default vs. Custom Build Settings**
 - **Build Configurations**
 - **Accessing Build Settings in Xcode**
 - **Project and Target Settings**
 - **Build Settings Overview**
 - **Code Signing in Different Environments**
 - **CI/CD and Code Signing**
 - **Security Best Practices**
 - **Handling Code Signing Issues**
 - **App Store Distribution**
 - **Distributing Builds for Testing**
 - **Exporting and Archiving Builds**
 - **Handling Entitlements**
 - **Wildcard App IDs**
 - **Manual Signing**
 - **Automatic Signing in Xcode**
 - **Creating Provisioning Profiles**
 - **Creating App IDs**
 - **Creating Certificates**
-

-
- **Developer Account Setup**
 - **Xcode Organizer**
 - **Provisioning Profiles**
 - **Certificates**
 - **Code Signing Identities**
 - **Introduction to Code Signing**
 - **Best Practices for Memory Management**
 - **Memory Graph Debugger in Real-world Scenarios**
 - **Testing and Simulating Memory Scenarios**
 - **Integration with Instruments**
 - **Heapshot Analysis**
 - **Optimizing Memory Usage**
 - **Memory Warnings and Auto Layout Issues**
 - **Analyzing Memory Snapshots**
 - **Heap and Allocations Summary**
 - **Filtering and Search**
 - **Navigating the Object Graph**
 - **Identifying Abandoned Memory**
 - **Viewing Object References**
-

○ Retain Cycles

○ Object Allocation Tracking

○ Using Memory Graph Debugger in Xcode

○ Introduction to Memory Graph Debugger

○ Integration with Other Debugging Tools

○ Screenshot and Export Features

○ Understanding Z-Order (Layering)

○ Memory Usage and Optimization

○ Checking Content Modes and Insets

○ View Highlighting

○ Debugging Auto Layout Constraints

○ Layer Inspection

○ Understanding Colors and Transparency

○ Identifying Overlapping and Clipping

○ Dynamic Inspection during Runtime

○ Identifying Layout Issues

○ Filtering and Search

○ Visual Inspection of UI Elements

- **Navigation in the View Hierarchy**

- **Using View Hierarchy Debugger in Xcode**

- **Introduction to View Hierarchy Debugger**

- **Breakpoint Integration with Xcode**

- **LLDB Commands Related to Breakpoints**

- **Collaborative Debugging**

- **Automated Breakpoints**

- **Breakpoint Navigator in Xcode**

- **Remove Breakpoint**

- **Enable/Disable Breakpoint**

- **Watchpoint**

- **Breakpoint Hierarchy**

- **Action Breakpoint**

- **Conditional Breakpoint**

- **Symbolic Breakpoint**

- **Exception Breakpoint**

- **Function Breakpoint**

- **Log Breakpoint**

- **Conditional Breakpoint**

Simple Breakpoint

Introduction to Breakpoint

Dependency Management Errors

Network Errors

User Interface (UI) Errors

Memory Management Errors

Logical Errors

Runtime Errors

Compile-time Errors

Performance Optimization - Advanced

Crash Reporting - Advanced

Error Handling - Advanced

Debugging - Practice

Testing - Practice

Coding Practices - Practice

Threading Errors - Basic

Memory Management Errors - Basic

Xcode Errors - Basic

Sources of Common Errors - Introduction

-
- Types of Common Errors - Introduction**

 - Common Errors - Introduction**

 - Debugging Performance Issues - Advanced**

 - Debugging Memory Leaks - Advanced**

 - Debugging Multithreading - Advanced**

 - Logging - Engineering**

 - Stack Trace - Engineering**

 - Debugger - Engineering**

 - Stepping Through Code - Basic**

 - Checking Variable Values - Basic**

 - Breakpoints - Basic**

 - Debugging Tools - Introduction**

 - Types of Debugging - Introduction**

 - Debugging - Introduction**

 - Marketing - Specific**

 - Metadata - Specific**

 - Prohibited Content - Specific**

 - In-App Purchases - Specific**

 - Performance - Principles**

-
- Security - Principles**
 -
 - Quality - Principle**
 -
 - App Review Process - Basic**
 -
 - Types of App Reviews - Basic**
 -
 - App Store Guidelines - Basic**
 -
 - Accessibility - Advanced**
 -
 - Dynamic Type in Custom Views - Advanced**
 -
 - Custom Text Styles - Advanced**
 -
 - Testing Dynamic Type - Application**
 -
 - Auto Layout - Application**
 -
 - Using Text Styles - Application**
 -
 - Adjustable Text Size - Features**
 -
 - Text Styles - Features**
 -
 - Content Size Categories - Features**
 -
 - Implementation of Dynamic Type - Basic**
 -
 - Benefits of Dynamic Type - Basic**
 -
 - Dynamic Type - Basic**
 -
 - Verbosity**
 -
 - Flicks**

-
- Rotor**
 - **Accessibility Traits - Features**
 - **Accessibility Shortcut - Features**
 - **Dictation - Features**
 - **Closed Captions & Subtitles - Features**
 - **Speak Selection - Cognitive**
 - **Reduce Motion - Cognitive**
 - **Guided Access - Cognitive**
 - **Voice Control - Mobility**
 - **Switch Control - Mobility**
 - **AssistiveTouch - Mobility**
 - **AssistiveTouch - Hearing**
 - **Live Listen - Hearing**
 - **Made for iPhone Hearing Aids - Hearing**
 - **Invert Colors - Vision**
 - **Zoom - Vision**
 - **VoiceOver - Vision**
 - **Custom Accessibility Elements - Advanced**
 - **Accessibility Testing Tools - Advanced**

○ Accessibility Inspection Scripting - Advanced

○ Creating Accessible Applications - Testing

○ Improving Accessibility - Testing

○ Verifying Accessibility - Testing

○ Accessibility Labels - Features

○ Accessibility Hints - Features

○ Accessibility Attributes - Features

○ Accessibility Elements - Features

○ View Hierarchy Structure - Basic

○ Opening and Navigating - Basic

○ Accessibility Inspector - Basic

○ Performance Optimization - Advanced

○ Debugging - Advanced

○ Testing - Advanced

○ Reactive State - Pattern

○ Immutable State - Pattern

○ Single Source of Truth - Pattern

○ Redux - Libraries

Combine - Libraries

RxSwift - Libraries

Mutable State - Basic

Immutable State - Basic

Source State - Basic

Understanding State Management - Basic

Performance Optimization - Advanced

Concurrency - Combine Advanced

Testing - Advanced

Error Handling - Networking

Parsing Responses - Networking

Make Requests - Networking

Animations - UI Development

Data Binding - UI Development

Managing Events - UI Development

Operators - Basic

- Map
 - Filter
 - Flat Map
 - Combine Latest
-

Subscribers - Basic

○ Publishers - Basic

○ Introduction to Combines - Basic

○ Performance Optimization - Advanced

○ Concurrency - ReactiveCocoa Advanced

○ Testing - Advanced

○ Error Handling - Networking

○ Parsing Responses - Networking

○ Make Requests - Networking

○ Animations - UI Development

○ Data Binding - UI Development

○ Managing Events - UI Development

○ RACSignal - Basic

○ Operators - Basic

- Map
 - Filter
 - Flatten Map
 - Combine Latest
-

○ Signals - Basic

○ Introduction to ReactiveCocoa - Basic

○ Performance Optimization - Advanced

○ Concurrency - RX Swift Advanced

Testing - Advanced

Error Handling - Networking

Parsing Responses - Networking

Make Requests - Networking

Animations - UI Development

Data Binding - UI Development

Managing Events - UI Development

Subjects - Basic

Operators - Basic

- Map
 - Filter
 - Flat Map
 - Combine Latest
-

Observables - Basic

Introduction to RxSwift - Basic

Performance Optimization - Advanced

Error Handling - Advanced

Testing - Advanced

UI Development - iOS Reactive

Networking - iOS Reactive

MVVM - iOS Reactive

○ Operators - Basic

○ Reactive Streams - Basic

○ Understanding Reactive Programming - Basic

○ Memory Leaks - Advanced

○ Livelocks - Advanced

○ Deadlocks - Advanced

○ Semaphores - Technique

○ Mutexes - Technique

○ Locks - Technique

○ Atomic Operations - Technique

○ Data Races - Basic

○ Race Condition - Basic

○ Understanding Thread Safety - Basic

○ GCD Barriers

○ GCD Queues

○ Performance Optimization - Advanced

○ Dispatch Sources - Advanced

○ Operations - Advanced

○ GCD Timers - Techniques

○ GCD Semaphores - Techniques

○ GCD Groups - Techniques

○ Serial and Parallel Execution - Techniques

○ `dispatch_barrier_async` - GCD Primitives

○ `dispatch_after` - GCD Primitives

○ `dispatch_sync` - GCD Primitives

○ `dispatch_async` - GCD Primitives

○ Threads and Queues - Basic

○ Concurrency Concept - Basic

○ Introduction to GCD - Basic

○ Code improvements

○ Debugging techniques

○ Analysis of memory leaks data - Debugging

○ Disciplined programming - Engineering

○ Application design and architecture - Engineering

○ Effective use of ARC - Technique

○ Common errors in memory management - Causes

○ Object ownership rules - Causes

- Object life cycle - Causes

- Tools to detect Memory Leaks - Basic

- Types of Memory Leaks - Basic

- Retained Cycles
 - Strong Reference Leaks
 - Unbalanced Callbacks
 - Closures Leaks
 - Blocks leaks
 - KVO leaks
 - Delegate leaks
-

- Introduction to Memory Leaks - Basic

- ARC Debugging Tools - Advanced

- Memory Management - Advanced

- Grand Central Dispatch (GCD) and ARC - Advanced

- Debugging ARC problems - Practice

- Efficient use of ARC - Practice

- Preventing memory leaks - Practice

- Unowned Reference - Rule

- Weak Reference - Rule

- Strong references - Rule

- Object life cycle - Rule

- Ownership rules - Rule

- Object Deallocation - Basic

Reference Counting - Basic

Introduction to ARC - Basic

Release Your App

Respond to Questions from The App Review Team

Submit Your App

Set Pricing and Availability

Prepare Screenshots and Video Previews

Prepare App Metadata

Upload Your App Build

Build Your App

Prepare Your App

Keychain Accessibility - Advanced

Data Synchronization - Advanced

Data Encryption - Advanced

Security.framework - Advanced

Keychain Access - Basic

Data type - Basic

Keychain Security - Basic

○ Keychain Structure - Basic

○ iCloud

○ Firebase

○ Migrations - Advanced

○ Data Synchronization - Advanced

○ Data Encryption - Advanced

○ Query Optimization - Advanced

○ Swift Data - Studio GUI

○ Swift Data - Framework

○ Data Migration - Basic

○ Data Observation - Basic

○ Data Type - Basic

○ Query Language - Basic

○ Database Structure - Basic

○ Data formats

○ Protocol

○ User Defaults - iCloud

○ Data Migration - Advanced

○ Data Encryption - Advanced

XCode User Defaults

Delete Data - Basic

Get Data - Basic

Save Data - Basic

Data Type - Basic

Data Structure - Basic

Migrations - Advanced

Data Synchronization - Advanced

Data Encryption - Advanced

Query Optimization - Advanced

Realm Studio - GUI

Realm Swift - Apple Framework

Data Synchronization - Basic

Data Observation - Basic

Data Type - Basic

Realm Query - Basic

Database Structure - Basic

Full Text Search - Advanced

Data Synchronization - Advanced

Database Encryption - Advanced

Query optimization - Advanced

SQLite Manager - GUI

FMDB - Apple Framework

Transactions - Basic

Joins - Basic

Data Type - Basic

SQL Query - Basic

- Create
- Read
- Update
- Delete

Database Structure - Basic

Keep Learning

CRUD

Data Modelling

Dependency Injection

Mocking

XCTest

App Transport Security (ATS) - App Store

Sandboxing - App Store

Static code analysis - Code

Obfuscation - Code

Penetration testing - Code

Code review - Code

Passcode - Device

Device fingerprinting - Device

Touch ID/Face ID - Device

Biometrics - Device

URL filtering - Network

VPN - Network

TLS/SSL - Network

HTTPS - Network

Access control - Data

Sandboxing - Data

Tokenization - Data

Encryption - Data

Look for real work or become a freelance 

Become a real iOS Developer

Keep Learning with a real iPhone

- Buy 1 iPhone Whatever You Want (I recommend the newest one) ★
-
- Passed 23 Stage iOS Roadmap Development ★
-
- Custom Authentication
-
- Phone Number - Firebase Authentication
-
- Provider Login - Firebase Login
-
- Email dan password - Firebase Authentication
-
- Passwordless - Apple Authentication
-
- Sign in with Apple - Apple Authentication
-
- SDWebImage
-
- YY Cache
-
- NS Cache
-
- Web Caching
-
- Disk Caching
-
- In-memory Caching
-
- Reachability
-
- NWPathMonitor
-
- GraphQL
-
- GCD by Apple
-
- AFNetworking

○ Alamofire

○ URLSession by Apple

○ TCP/IP

○ Async / Await

○ Rest API

○ JSON

○ HTTP

○ URL

○ Interface Builder Designable

○ Interface Builder Inspectable

○ Size Classes

○ Auto Layout

○ Design UI by Apple

○ Gestures

○ Navigation

○ Layout

○ Typography

○ Material Design

○ Atomic Design

○ Principle

○ Sketch

○ Figma

○ Adobe XD

○ User Experience

○ User Journey

○ Design UX

○ Layout

○ Colors

○ Tipography

○ Design UI

○ Dependency Injection (DI) - Framework

○ Typhoon - Framework

○ Swinject - Framework

○ Property Injection - Type

○ Setter Injection - Type

○ Constructor Injection - Type

○ Decoupling - Benefit

○ Testability - Benefit

○ Modularitas - Benefit

○ Injection Point - Concept

○ Injector - Concept

○ Dependency - Concept

○ Router

○ Entity

○ Presenter

○ Interactor

○ View

○ Coordinator

○ View Model

○ View

○ Model

○ View Model

○ View

○ Model

○ Presenter

○ View

-

Model

Controller

View

Model

Singleton

Prototype

Factory Method

Builder

Abstract Factory

Visitor

Template Method

Strategy

State

Singleton

Opserver

Memento

Mediator

Iterator

Interpreter

○ Command

○ Chain of Responsibility

○ Proxy

○ Flyweight

○ Facade

○ Decorator

○ Composite

○ Bridge

○ Adapter

○ Trunk Based

○ Git Flow

○ Release Train

○ Continuous Integration

○ Code Review

○ Universal Links

○ Deep Links

○ Widgets

○ App Clips

○ Home Screen Quick Actions

Push Notifications

Collaboration

Review Code

Branching

Issue

Pull Request

Fork

Repository

Platform Hosting

Git Pull

Git Push

Git Commit

Git Add

Git Clone

Git Init

Merge

Branch

Commit

Repository

○ Version Control System

○ Active

○ Background

○ Inactive

○ Not Running

○ Navigations

○ Controllers

○ Views

○ UI Kit Fundamental

○ Layout - Views

- Control Group
 - Depth Stack
 - Geometry Reader
 - Horizontal Stack
 - Lazy Horizontal Grid
 - Lazy Horizontal Stack
 - Lazy Vertical Grid
 - Lazy Vertical Stack
 - Scroll View Reader
 - Spacer
 - Vertical Stack
 - View That Fits
-

○ Controls - Views

- Button
- Color Picker
- Date Picker
- Disclosure Group
- Edit Button
- Form
- Gauge
- Group Box