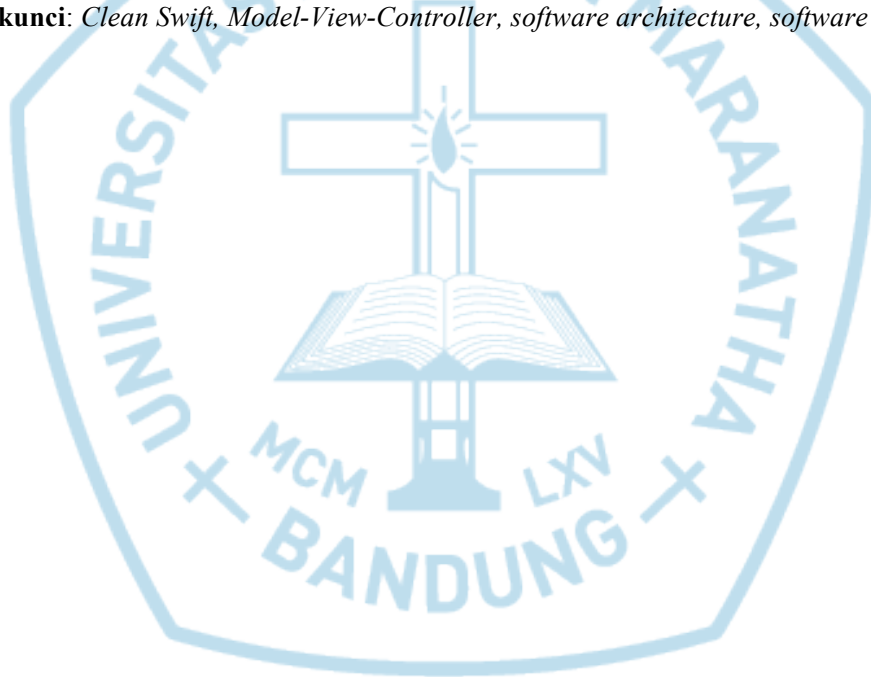


ABSTRAK

Permintaan pembuatan perangkat lunak semakin berkembang menyebabkan proses pembuatan perangkat lunak umumnya tidak selesai tepat waktu. Hal ini dikarenakan sebuah perangkat lunak memiliki nilai *software metrics* yang rendah, sehingga *programmer* mengalami kesulitan dalam mengubah kode program perangkat lunak. Penggunaan arsitektur adalah salah satu cara untuk meningkatkan nilai *software metrics* perangkat lunak. Arsitektur yang digunakan dalam pengembangan perangkat lunak akan mempengaruhi nilai *software metrics*. Penelitian ini akan menguji perbandingan nilai *software metrics* dari arsitektur Model-View-Controller dengan Clean Swift, sehingga *programmer* dapat menentukan arsitektur yang sebaiknya digunakan dalam pengembangan perangkat lunak. Pengukuran yang diukur pada penelitian ini adalah nilai *complexity* dan *Simple Component Cohesion Metrics* dengan batasan masalah kode program yang ditulis adalah kode program yang sama. Berdasarkan hasil dari penelitian didapatkan nilai rata-rata *complexity* pada arsitektur Clean Swift lebih baik dari arsitektur Model-View-Controller, sedangkan untuk nilai rata-rata *Simple Component Cohesion Metrics*, arsitektur Model-View-Controller lebih baik dari arsitektur Clean Swift.

Kata kunci: *Clean Swift, Model-View-Controller, software architecture, software metrics*



ABSTRACT

Demand for software development is increasingly. This demand causing the software development process is not completed on time. Late development process cause by software has low software metrics value. These things causing programmers have difficulty in changing and fixing code. The use of architecture is one way to increase the value of software metrics. The development of iOS software, has several architectures, such as Model-View-Controller, and Clean Swift. Each architecture affects the value of software metrics. This research will test the comparison of software metrics value from Model-View-Controller and Clean Swift, so programmer can determine architecture that need to be used in making software. Measurement values to be measure in this study are complexity and Simple Component Cohesion Metrics with limitations of code written are same. The results of this study indicate that the average value of complexity in Clean Swift is better than Model-View-Controller, whereas for Simple Component Cohesion Metrics average value, Model-View-Controller is better than Clean Swift.

Keywords: *Clean Swift, Model-View-Controller, software architecture, software metrics*



DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN ORISINALITAS LAPORAN PENELITIAN.....	ii
PERNYATAAN PUBLIKASI LAPORAN PENELITIAN	iii
PRAKATA.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiv
DAFTAR RUMUS	xv
DAFTAR NOTASI/ LAMBANG	xvi
DAFTAR SINGKATAN	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan Pembahasan	2
1.4 Ruang Lingkup.....	2
1.5 Sumber Data.....	2
1.6 Sistematika Penyajian	2
BAB 2 KAJIAN TEORI	4
2.1 Software Architecture	4
2.1.1 Model-View-Controller (MVC).....	4
2.1.2 Clean Swift.....	5
2.1.3 Perbandingan Arsitektur	6

2.1.4 SOLID Principle	8
2.1.5 Clean Architecture	9
2.2 Software Metrics.....	11
2.3 Software Architecture Metrics.....	12
2.4 RESTFul WS	12
2.5 Laravel-Passport	13
2.6 Lizard	14
2.7 XCUI Test.....	15
BAB 3 ANALISIS DAN RANCANGAN SISTEM.....	16
3.1 Profil Perusahaan	16
3.2 Pemilihan Arsitektur MVC dan Clean Swift	16
3.3 Metode Penelitian	16
3.4 <i>Use Case Diagram</i>	17
3.5 <i>Activity</i>	18
3.5.1 <i>Login</i>	18
3.5.2 <i>Logout</i>	19
3.5.3 Melihat Data Latihan	19
3.5.4 Melihat Data Pengukuran	19
3.5.5 Melihat Data Diri Pengguna	20
3.5.6 Melihat QR Code	20
3.6 Rancangan Desain Antarmuka.....	20
3.6.1 Rancangan Antarmuka Halaman Beranda	20
3.6.2 Rancangan Antarmuka Halaman <i>Login</i>	21
3.6.3 Rancangan Antarmuka Halaman Latihan	22
3.6.4 Rancangan Antarmuka Halaman Detail Latihan	22
3.6.5 Rancangan Antarmuka Halaman Pengukuran	23

3.6.6 Rancangan Antarmuka Halaman Detail Pengukuran	23
3.6.7 Rancangan Antarmuka Halaman Data Diri Pengguna.....	24
3.6.8 Rancangan Antarmuka Halaman QRCode	24
3.7 Metode Pengujian <i>Software Metrics</i>	25
3.8 Metode Pengujian <i>Architecture Metrics</i>	25
3.9 Metode Pengujian <i>Performance Test</i>	26
BAB 4 IMPLEMENTASI	27
4.1 Implementasi Model-View-Controller.....	27
4.1.1 Model	27
4.1.2 View.....	27
4.1.3 View Controller.....	27
4.2 Implementasi Clean Swift.....	28
4.2.1 Model	28
4.2.2 Interactor	29
4.2.3 Worker.....	29
4.2.4 Presenter.....	30
4.2.5 Router.....	30
4.2.6 View.....	31
4.2.7 View Controller.....	31
4.3 Implementasi <i>Class Diagram</i>	32
4.3.1 Modul <i>Login</i>	33
4.3.2 Modul Beranda.....	34
4.3.3 Modul Menampilkan Data Latihan.....	35
4.3.4 Modul Menampilkan Data Detail Latihan	36
4.3.5 Modul Menampilkan Data Pengukuran	37
4.3.6 Modul Menampilkan Data Detail Pengukuran	38

4.3.7 Modul Menampilkan <i>QRTag</i>	39
4.3.8 Modul Menampilkan Profil.....	39
4.4 Implementasi Pengujian <i>Software Metrics</i>	40
4.4.1 Pengujian Nilai <i>Complexity</i>	41
4.4.2 Perhitungan Nilai SCCM	42
4.4.3 Pengujian Nilai <i>Performance Test</i>	44
BAB 5 PENGUJIAN	46
5.1 <i>Benchmarking</i>	46
5.1.1 Modul <i>Login</i>	46
5.1.2 Modul Beranda.....	47
5.1.3 Modul Menampilkan Data Latihan.....	48
5.1.4 Modul Menampilkan Data Detail <i>Workout</i>	49
5.1.5 Modul Menampilkan Data Pengukuran.....	50
5.1.6 Modul Menampilkan Data Detail Pengukuran	51
5.1.7 Modul Menampilkan <i>QRTag</i>	52
5.1.8 Modul Menampilkan Profil.....	53
5.2 MVC dan Clean Swift.....	54
BAB 6 SIMPULAN DAN SARAN.....	57
6.1 Simpulan	57
6.2 Saran	60
DAFTAR PUSTAKA	62

DAFTAR GAMBAR

Gambar 2.1 Model-View-Controller [4].....	5
Gambar 2.2 Clean Swift <i>Pattern</i> [5].....	6
Gambar 2.3 <i>Clean Architecture</i> [8]	10
Gambar 2.4 Nilai <i>Threshold</i> SCCM ($N < 6$) [13].....	12
Gambar 2.5 Nilai <i>Threshold</i> SCCM ($N \geq 6$) [13].....	12
Gambar 2.6 <i>OAuth 2.0 Communication</i> [16].....	14
Gambar 3.1 Proses Metode Penelitian.....	16
Gambar 3.2 <i>Use Case Diagram</i> Aplikasi X.....	18
Gambar 3.3 Rancangan Antarmuka Halaman Beranda	21
Gambar 3.4 Rancangan Antarmuka Halaman <i>Login</i>	21
Gambar 3.5 Rancangan Antarmuka Halaman Latihan	22
Gambar 3.6 Rancangan Antarmuka Halaman Detail Latihan	22
Gambar 3.7 Rancangan Antarmuka Halaman Pengukuran	23
Gambar 3.8 Rancangan Antarmuka Halaman Detail Pengukuran.....	23
Gambar 3.9 Rancangan Antarmuka Halaman Data Diri Pengguna.....	24
Gambar 3.10 Rancangan Antarmuka Halaman QRCode	24
Gambar 3.11 Tampilan Pengujian Lizard [17]	25
Gambar 3.12 Contoh Modul X [13].....	26
Gambar 3.13 <i>Pseudocode</i> XCode UI Test.....	26
Gambar 4.1 Implementasi Model Arsitektur MVC.....	27
Gambar 4.2 Implementasi <i>View Controller</i> Arsitektur MVC.....	28
Gambar 4.3 Implementasi <i>Model</i> Arsitektur Clean Swift	29
Gambar 4.4 Implementasi <i>Interactor</i> Arsitektur Clean Swift	29
Gambar 4.5 Implementasi <i>Worker</i> Arsitektur Clean Swift.....	30
Gambar 4.6 Implementasi <i>Presenter</i> Arsitektur Clean Swift	30
Gambar 4.7 Implementasi <i>Router</i> Arsitektur Clean Swift.....	31
Gambar 4.8 Implementasi <i>View Controller</i> Arsitektur Clean Swift.....	32
Gambar 4.9 <i>Class Diagram</i> Modul <i>Login</i>	33
Gambar 4.10 <i>Class Diagram</i> Modul Beranda	34
Gambar 4.11 <i>Class Diagram</i> Modul Menampilkan Data Latihan.....	35

Gambar 4.12 <i>Class Diagram</i> Modul Menampilkan Data Detail Latihan.....	36
Gambar 4.13 <i>Class Diagram</i> Modul Menampilkan Data Latihan.....	37
Gambar 4.14 <i>Class Diagram</i> Modul Menampilkan Data Pengukuran.....	38
Gambar 4.15 <i>Class Diagram</i> Modul Menampilkan <i>QRTag</i>	39
Gambar 4.16 <i>Class Diagram</i> Modul Menampilkan Profil.....	40
Gambar 6.1 Diagram Perbandingan Nilai Total <i>Line of Codes</i>	57
Gambar 6.2 Diagram Perbandingan Nilai Rerata <i>Line of Codes</i>	58
Gambar 6.3 Diagram Perbandingan Nilai Rerata <i>Cyclomatic Complexity Number</i>	59
Gambar 6.4 Diagram Perbandingan <i>Performance Time</i>	60



DAFTAR TABEL

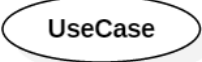
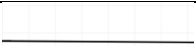
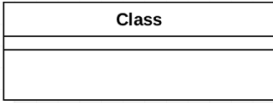
Tabel 2.1 Perbandingan Arsitektur [7].....	7
Tabel 4.1 Nilai <i>Threshold Complexity</i>	41
Tabel 4.2 Nilai <i>Complexity</i> Arsitektur MVC - 1.....	41
Tabel 4.3 Nilai <i>Complexity</i> Arsitektur MVC - 2.....	41
Tabel 4.4 Nilai <i>Complexity</i> Arsitektur Clean Swift - 1.....	42
Tabel 4.5 Nilai <i>Complexity</i> Arsitektur Clean Swift - 2.....	42
Tabel 4.6 Nilai <i>Threshold SCCM</i> Arsitektur MVC.....	43
Tabel 4.7 Nilai <i>SCCM</i> Arsitektur MVC	43
Tabel 4.8 Nilai <i>Threshold SCCM</i> Arsitektur Clean Swift.....	44
Tabel 4.9 Nilai <i>SCCM</i> Arsitektur Clean Swift	44
Tabel 4.10 Nilai Pengujian <i>Performance</i> Arsitektur MVC	45
Tabel 4.11 Nilai Pengujian <i>Performance</i> Arsitektur Clean Swift	45
Tabel 5.1 Perbandingan Nilai <i>Complexity</i> dan <i>SCCM</i> Modul <i>Login</i>	47
Tabel 5.2 Perbandingan Nilai <i>Complexity</i> dan <i>SCCM</i> Modul Beranda	48
Tabel 5.3 Perbandingan Nilai <i>Complexity</i> dan <i>SCCM</i> Modul Menampilkan Data Latihan	48
Tabel 5.4 Perbandingan Nilai <i>Complexity</i> dan <i>SCCM</i> Modul Menampilkan Data Detail Latihan.....	49
Tabel 5.5 Perbandingan Nilai <i>Complexity</i> dan <i>SCCM</i> Modul Menampilkan Data Pengukuran	50
Tabel 5.6 Perbandingan Nilai <i>Complexity</i> dan <i>SCCM</i> Modul Menampilkan Data Detail Pengukuran.....	51
Tabel 5.7 Perbandingan Nilai <i>Complexity</i> dan <i>SCCM</i> Modul Menampilkan <i>QRTag</i>	52
Tabel 5.8 Perbandingan Nilai <i>Complexity</i> dan <i>SCCM</i> Modul Menampilkan Profil	53
Tabel 5.9 Perbandingan Nilai Rerata <i>Complexity</i> dan <i>SCCM</i>	54
Tabel 6.1 Tabel Hasil Pengujian Nilai <i>SCCM</i>	59

DAFTAR RUMUS

Rumus 3.1 <i>Simple Component Cohesion Metrics</i> [13].....	25
--	----



DAFTAR NOTASI/ LAMBANG

Jenis	Notasi/ Lambang	Nama	Arti
UML (<i>Use Case</i>)		<i>Actor</i>	Pengguna sebuah sistem.
		<i>Use Case</i>	Entitas yang berhubungan dengan sistem dan pengguna.
		<i>Association</i>	Hubungan antara dua buah objek.
UML (<i>Class Diagram</i>)		<i>Class</i>	Kelas
		<i>Directed Association</i>	Hubungan satu arah antara dua buah objek.

Referensi:

Notasi/ Lambang UML dari *The Unified Modeling Language* [1]

DAFTAR SINGKATAN

MVC	Model View Controller
SCCM	Simple Component Cohesion Metrics
CCN	Cyclometric Complexity Number
UML	Unified Modelling Language
iOS	iPhone OS
RESTful WS	REpresentational State Transfer Web Service
NLOC	Number Line of Codes

