

**APLIKASI MANAJEMEN KERENTANAN BERDASARKAN HASIL UJI  
KEAMANAN BERBASIS WEBSITE**

**SKRIPSI**

**Oleh**

**Stanley Halim                    2301859146**

**Alfredo Thomas Nofanski  2301924536**

**Ryo Chris Samuel                2301931503**



**Cyber Security Program  
Computer Science Study Program  
School of Computer Science  
Universitas Bina Nusantara  
Jakarta  
2023**

**APLIKASI MANAJEMEN KERENTANAN BERDASARKAN HASIL UJI  
KEAMANAN BERBASIS WEBSITE**

**SKRIPSI**

**diajukan sebagai salah satu syarat  
untuk gelar kesarjanaan pada  
Program Studi Cyber Security  
Jenjang Pendidikan Strata-1**

**Oleh**

**Stanley Halim                    2301859146  
Alfredo Thomas Nofanski  2301924536  
Ryo Chris Samuel                2301931503**



**Cyber Security Program  
Computer Science Study Program  
School of Computer Science  
Universitas Bina Nusantara  
Jakarta  
2023**

**Pernyataan Kesiapan Skripsi untuk Sidang Skripsi**

**Pernyataan Penyusunan Skripsi**

**Kami, Stanley Halim**

**Alfredo Thomas Nofanski,**

**Ryo Chris Samuel,**

dengan ini menyatakan bahwa Skripsi yang berjudul:

**APLIKASI MANAJEMEN KERENTANAN BERDASARKAN HASIL UJI  
KEAMANAN BERBASIS WEBSITE  
VULNERABILITY MANAGEMENT WEB APPLICATION BASED ON  
PENETRATION TESTING RESULT**

adalah benar hasil karya kami dan belum pernah diajukan sebagai karya ilmiah, sebagian atau seluruhnya, atas nama kami atau pihak lain



**Stanley Halim**  
**2301859146**



**Alfredo Thomas Nofanski**  
**2301924536**



**Ryo Chris Samuel**  
**2301931503**

**Disetujui oleh Pembimbing**

**Kami setuju Skripsi tersebut layak diajukan untuk Sidang Skripsi**

**Yohan Muliono, S.Kom., M.TI**

**D5543**

**6 Februari 2021**

---

## UNIVERSITAS BINA NUSANTARA

---

School of Computer Science  
Cyber Security Program  
Skripsi Sarjana Teknik Informatika  
Semester Genap Tahun 2022/2023

### AUTOMASI TERHADAP ANALISIS DINAMIS PADA UJI KEAMANAN APLIKASI SELULER BERBASIS ANDROID

Stanley Halim                    2301859146  
Alfredo Thomas Nofanski      2301924536  
Ryo Chris Samuel                2301931503

#### *ABSTRACT*

*As the number of mobile or web applications developed increases, the threat of attacks on applications also becomes greater. One effort is to conduct security testing of applications by penetration testers. However, the large number of tests requires security testing to be completed faster and still maintain quality, as well as the technical capability gap of penetration testers and collaboration with clients, making it difficult to focus on testing. This research focuses on the development of a vulnerability management application to assist penetration testers in testing faster and more efficiently with ease of collaboration, automation of security test report generation, and knowledge sharing among penetration testers. The result of the application development is a web-based application that helps both penetration testers and clients.*

**Keywords:** Vulnerability Management, Penetration Testing, Collaboration

#### **ABSTRAK**

Dengan naiknya jumlah aplikasi *mobile* atau *web* yang dikembangkan, maka ancaman serangan terhadap aplikasi juga semakin besar. Salah satu upaya adalah melakukan uji keamanan aplikasi oleh *penetration tester*. Namun, banyaknya jumlah pengujian menuntut uji keamanan diselesaikan lebih cepat dan tetap menjaga kualitas, serta permasalahan gap kemampuan teknis *penetration tester* dan kolaborasi dengan klien sehingga membuat sulit fokus dalam melakukan pengujian. Penelitian ini berfokus pada pengembangan aplikasi manajemen *vulnerability* untuk membantu *penetration tester* dalam melakukan pengujian secara lebih cepat dan efisien dengan kemudahan kolaborasi, automasi pembuatan laporan hasil uji keamanan, dan *knowledge sharing* antar *penetration tester*. Hasil dari pengembangan aplikasi berupa aplikasi berbasis *web* yang membantu *penetration tester* maupun klien.

**Kata kunci:** Manajemen Kerentanan, Pengujian Kebobolan, Kolaborasi

## KATA PENGANTAR

Puji dan syukur kepada Tuhan Yang Maha Esa karena Rahmat-nya yang berlimpah sehingga penulis dapat menyelesaikan skripsi yang berjudul “**APLIKASI MANAJEMEN KERENTANAN BERDASARKAN HASIL UJI KEAMANAN BERBASIS WEBSITE**” dengan tepat waktu. Skripsi ini dibuat untuk memenuhi persyaratan kelulusan serta mendapatkan Gelar Sarjana Komputer di jurusan Cyber Security, Universitas Bina Nusantara jenjang studi strata-1.

Penulisan skripsi ini dapat selesai dengan baik karena tidak lepas dari bantuan berbagai pihak. Untuk itu penulis mengucapkan Terima kasih sebesar-besarnya kepada pihak-pihak yang sudah memberi dukungan dan bantuan selama penyusunan skripsi ini khususnya:

1. Bapak Prof. Dr. Ir. Harjanto Prabowo, MM., selaku Rektor Universitas Bina Nusantara.
2. Bapak Fredy Purnomo, S.Kom., M.CompSc., selaku *Dean of School of Computer Science* Universitas Bina Nusantara.
3. Bapak Yohan Muliono, S.Kom., M.TI., selaku *Head of Program Cyber Security* serta sebagai dosen pembimbing yang telah memberikan dukungan, bimbingan dan pengarahan untuk menyelesaikan skripsi ini.
4. Seluruh dosen Universitas Bina Nusantara yang telah memberikan ilmu yang bermanfaat dalam penulisan skripsi ini.
5. Seluruh orang tua dan keluarga dari para penulis yang selalu memberikan dukungan baik moral maupun materiil serta selalu mendoakan dan memberi semangat bagi penulis selama penulisan skripsi ini berlangsung.
6. Teman-teman dan sahabat yang selalu mendukung, menyemangati dan mendoakan penulis.
7. Semua pihak yang telah membantu dan memberi dukungan kepada penulis dalam menyelesaikan skripsi ini.

Akhir kata, semoga skripsi ini dapat memberikan manfaat bagi para pembaca. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna karena keterbatasan pengalaman dan pengetahuan yang dimiliki oleh penulis. Oleh karena itu penulis memohon maaf jika terdapat kesalahan dan kekurangan pada skripsi ini. Penulis

mengharapkan kritik dan saran untuk memperoleh hasil yang lebih baik lagi di kemudian hari.

Jakarta, 11 Januari 2023

Penulis

## DAFTAR ISI

<b>HALAMAN SAMPUL .....</b>	i
<b>HALAMAN JUDUL .....</b>	ii
<b>HALAMAN PERNYATAAN ORISINALITAS .....</b>	iii
<b>ABSTRAK .....</b>	iv
<b>KATA PENGANTAR.....</b>	v
<b>DAFTAR ISI.....</b>	vii
<b>DAFTAR TABEL .....</b>	xi
<b>DAFTAR GAMBAR.....</b>	xii
<b>DAFTAR LAMPIRAN .....</b>	xvii
<b>BAB 1 PENDAHULUAN .....</b>	1
1.1    Latar Belakang .....	1
1.2    Rumusan Masalah .....	3
1.3    Ruang Lingkup .....	3
1.4    Tujuan dan Manfaat.....	4
1.4.1    Tujuan .....	4
1.4.2    Manfaat .....	4
1.5    Metodologi .....	4
1.5.1    Metode Penelitian.....	4
1.5.2    Metode Perancangan .....	4
1.5.3    Metode Pengembangan .....	5
1.6    Sistematika Penulisan.....	5
1.6.1    BAB 1 PENDAHULUAN .....	5
1.6.2    BAB 2 TINJAUAN REFERENSI .....	6
1.6.3    BAB 3 METODE PENELITIAN.....	6
1.6.4    BAB 4 HASIL DAN PEMBAHASAN.....	6
1.6.5    BAB 5 SIMPULAN DAN SARAN.....	6
<b>BAB 2 TINJAUAN REFERENSI .....</b>	7
2.1 <i>Penetration Testing .....</i>	7
2.1.1    Web Penetration Testing .....	7
2.1.2    Mobile Penetration Testing .....	8
2.2 <i>Penetration Tester .....</i>	8

2.3	<i>Vulnerability</i> .....	8
2.4	<i>Vulnerability Management</i> .....	9
2.5	<i>Applikasi</i> .....	10
2.6	<i>Website</i> .....	10
2.7	<i>Docker</i> .....	10
2.7.1	<i>Docker Compose</i> .....	10
2.8	<i>Github</i> .....	11
2.9	<i>Common Vulnerability Scoring System (CVSS)</i> .....	11
2.10	<i>Common Weakness Enumeration (CWE)</i> .....	11
2.11	<i>Create, Read, Update, Delete (CRUD)</i> .....	12
2.12	<i>Prototyping</i> .....	12
2.13	<i>SDLC Waterfall</i> .....	12
2.14	<i>Unified Modeling Language (UML)</i> .....	13
2.14.1	<i>Use Case Diagram</i> .....	13
2.14.2	<i>Activity Diagram</i> .....	15
2.14.3	<i>Use Case Description</i> .....	17
2.14.4	<i>Class Diagram</i> .....	18
2.14.5	<i>Sequence Diagram</i> .....	19
2.15	<i>Entity Relationship Diagram (ERD)</i> .....	21
2.16	<i>Application Development Tools</i> .....	21
2.16.1	<i>Python</i> .....	21
2.16.2	<i>HyperText Markup Language (HTML)</i> .....	22
2.16.3	<i>Cascading Style Sheets (CSS)</i> .....	22
2.16.4	<i>Javascript</i> .....	22
2.17	<i>Framework</i> .....	22
2.17.1	<i>Django</i> .....	22
2.17.2	<i>Bootstrap</i> .....	22
2.17.3	<i>SB Admin 2</i> .....	22
2.18	<i>Model View Controller (MVC)</i> .....	23
2.19	<i>Blackbox Testing</i> .....	23
2.20	<i>8 Golden Rules of Interface Design</i> .....	23
2.21	<i>Gantt Chart</i> .....	24
	<b>BAB 3 METODE PENELITIAN</b> .....	25

3.1	Kerangka Berpikir .....	25
3.2	Analisis Masalah .....	26
3.2.1	Permasalahan dan Kebutuhan .....	26
3.2.2	Analisis Aplikasi Sejenis .....	27
3.2.3	Analisis Berdasarkan Wawancara.....	28
3.2.4	Usulan Pemecahan Masalah.....	34
3.2.5	Testing Methodology .....	38
3.3	Perancangan Sistem.....	38
3.3.1	Use Case Diagram.....	39
3.3.2	Use Case Description .....	42
3.3.3	Activity Diagram.....	61
3.3.4	Class Diagram .....	82
3.3.5	Sequence Diagram .....	83
3.3.6	Entity Relationship Diagram.....	99
3.3.7	User Interface .....	100
3.4	Gantt Chart .....	135
3.5	Teknologi yang digunakan .....	136
<b>BAB 4</b>	<b>HASIL DAN PEMBAHASAN .....</b>	139
4.1	Spesifikasi Sistem.....	139
4.2	Hasil Perancangan .....	140
4.2.1	Login .....	140
4.2.2	Home Page .....	141
4.2.3	Project .....	144
4.2.4	Resources .....	167
4.2.5	Admin Panel.....	177
4.3	<i>Black Box Testing</i> .....	183
4.4	Evaluasi Aplikasi berdasarkan <i>user feedback</i> .....	190
4.5	Evaluasi Aplikasi berdasarkan <i>Eight Golden Rules</i> .....	196
4.5.1	Strive for Consistency .....	196
4.5.2	Cater to Universal Usability.....	198
4.5.3	Offer Informative Feedback.....	199
4.5.4	Design Dialogs to Yield Closure.....	201
4.5.5	Prevent Errors.....	202

4.5.6	Permit Easy Reversal of Actions.....	202
4.5.7	Support Internal Locus of Control .....	204
4.5.8	Reduce Short-Term Memory Load .....	204
4.6	Perbandingan Aplikasi Sejenis .....	205
<b>BAB 5</b>	<b>SIMPULAN DAN SARAN .....</b>	<b>207</b>
5.1	Simpulan.....	207
5.2	Saran .....	207
<b>REFERENSI.....</b>		<b>209</b>
<b>LAMPIRAN.....</b>		<b>213</b>
LAMPIRAN 1 - Transkrip Wawancara mengenai <i>User Requirement</i> .....		213
Transkrip Wawancara dengan Henky Tornado .....		213
Transkrip Wawancara dengan Rafi Ramzy .....		217
Transkrip Wawancara dengan Christoval Leaved .....		220
Transkrip Wawancara dengan Ming Hong .....		222
Transkrip Wawancara dengan Ferry Alifani .....		224
LAMPIRAN 2 - Transkrip Wawancara mengenai <i>User Feedback</i> .....		226
Transkrip Wawancara dengan Henky Tornado .....		226
Transkrip Wawancara dengan Rafi Ramzy .....		228
Transkrip Wawancara dengan Christoval Leaved .....		229
Transkrip Wawancara dengan Ming Hong .....		230
Transkrip Wawancara dengan Ferry Alifani .....		231
LAMPIRAN 3 - Persetujuan Transkrip Wawancara dengan Narasumber.....		232
Persetujuan Transkrip Wawancara dengan Henky Tornado.....		232
Persetujuan Transkrip Wawancara dengan Rafi Ramzy.....		233
Persetujuan Transkrip Wawancara dengan Christoval Leaved .....		234
Persetujuan Transkrip Wawancara dengan Ming Hong .....		235
Persetujuan Transkrip Wawancara dengan Ferry Alifani.....		236
LAMPIRAN 4 - Surat Non Survei .....		237
<b>RIWAYAT HIDUP .....</b>		<b>238</b>

## DAFTAR TABEL

Tabel 2.1 Penjelasan Simbol <i>Use Case Diagram</i> .....	14
Tabel 2.2 Penjelasan Simbol <i>Activity Diagram</i> .....	16
Tabel 2.3 Penjelasan Simbol <i>Class Diagram</i> .....	18
Tabel 2.4 Penjelasan simbol <i>Sequence Diagram</i> .....	20
Tabel 3.1 Penjabaran Aktor pada <i>Use Case Diagram</i> .....	39
Tabel 3.2 Penjabaran <i>Use Case</i> pada <i>Use Case Diagram</i> .....	40
Tabel 3.3 <i>Use Case Description: Login</i> .....	42
Tabel 3.4 <i>Use Case Description: Manage User</i> .....	43
Tabel 3.5 <i>Use Case Description: Add User</i> .....	43
Tabel 3.6 <i>Use Case Description: Edit User</i> .....	44
Tabel 3.7 <i>Use Case Description: Delete User</i> .....	45
Tabel 3.8 <i>Use Case Description: View User</i> .....	46
Tabel 3.9 <i>Use Case Description: Manage Project</i> .....	46
Tabel 3.10 <i>Use Case Description: Search Project</i> .....	47
Tabel 3.11 <i>Use Case Description: Add Project</i> .....	47
Tabel 3.12 <i>Use Case Description: Edit Project</i> .....	48
Tabel 3.13 <i>Use Case Description: Delete Project</i> .....	49
Tabel 3.14 <i>Use Case Description: View Project</i> .....	50
Tabel 3.15 <i>Use Case Description: Manage Finding</i> .....	50
Tabel 3.16 <i>Use Case Description: Add Finding</i> .....	51
Tabel 3.17 <i>Use Case Description: Edit Finding</i> .....	52
Tabel 3.18 <i>Use Case Description: Delete Finding</i> .....	53
Tabel 3.19 <i>Use Case Description: Change Finding Status</i> .....	54
Tabel 3.20 <i>Use Case Diagram: View Finding</i> .....	55
Tabel 3.21 <i>Use Case Description: Generate Report</i> .....	56
Tabel 3.22 <i>Use Case Description: Manage Testing Guide</i> .....	57
Tabel 3.23 <i>Use Case Description: Add Testing Guide</i> .....	57
Tabel 3.24 <i>Use Case Description: Edit Testing Guide</i> .....	58
Tabel 3.25 <i>Use Case Description: Delete Testing Guide</i> .....	59
Tabel 3.26 <i>Use Case Description: View Testing Guide</i> .....	60
Tabel 4.1 Spesifikasi Aplikasi <i>vulnerability management</i> .....	139
Tabel 4.2 Spesifikasi server Aplikasi <i>vulnerability management</i> .....	139
Tabel 4.3 Spesifikasi <i>user</i> Aplikasi <i>vulnerability management</i> .....	140
Tabel 4.4 Evaluasi <i>Black Box Testing</i> .....	183
Tabel 4.5 Spesifikasi aplikasi <i>vulnerability management</i> .....	205

## DAFTAR GAMBAR

Gambar 1.1 Metode <i>SDLC Waterfall</i> .....	5
Gambar 2.1 Contoh <i>Use Case Diagram</i> .....	14
Gambar 2.2 Contoh <i>Activity Diagram</i> .....	15
Gambar 2.3 Contoh <i>Use Case Description</i> .....	17
Gambar 2.4 Contoh <i>Class Diagram</i> .....	18
Gambar 3.1 Metode <i>SDLC Waterfall</i> .....	25
Gambar 3.2 Perbandingan alur <i>penetration testing</i> tanpa dan dengan aplikasi <i>Vulnerability Management</i> .....	35
Gambar 3.3 Perbandingan kolaborasi tanpa dan dengan aplikasi <i>Vulnerability Management</i> .....	36
Gambar 3.4 Perbandingan <i>transfer knowledge</i> tanpa dan dengan aplikasi <i>Vulnerability Management</i> .....	37
Gambar 3.5 <i>Use Case Diagram</i> .....	39
Gambar 3.6 <i>Activity Diagram Login</i> .....	61
Gambar 3.7 <i>Activity Diagram Manage User</i> .....	62
Gambar 3.8 <i>Activity Diagram Add User</i> .....	63
Gambar 3.9 <i>Activity Diagram Edit User</i> .....	64
Gambar 3.10 <i>Activity Diagram Delete User</i> .....	65
Gambar 3.11 <i>Activity Diagram View User</i> .....	65
Gambar 3.12 <i>Activity Diagram Manage Project</i> .....	66
Gambar 3.13 <i>Activity Diagram Search Project</i> .....	67
Gambar 3.14 <i>Activity Diagram Add Project</i> .....	68
Gambar 3.15 <i>Activity Diagram Edit Project</i> .....	69
Gambar 3.16 <i>Activity Diagram Delete Project</i> .....	70
Gambar 3.17 <i>Activity Diagram View Project</i> .....	70
Gambar 3.18 <i>Activity Diagram Manage Finding</i> .....	71
Gambar 3.19 <i>Activity Diagram Add Finding</i> .....	72
Gambar 3.20 <i>Activity Diagram Edit Finding</i> .....	73
Gambar 3.21 <i>Activity Diagram Delete Finding</i> .....	74
Gambar 3.22 <i>Activity Diagram Change Finding Status</i> .....	75
Gambar 3.23 <i>Activity Diagram View Finding</i> .....	76
Gambar 3.24 <i>Activity Diagram Generate Report</i> .....	76
Gambar 3.25 <i>Activity Diagram Manage Testing Guide</i> .....	77
Gambar 3.26 <i>Activity Diagram Add Testing Guide</i> .....	78
Gambar 3.27 <i>Activity Diagram Edit Testing Guide</i> .....	79
Gambar 3.28 <i>Activity Diagram Delete Testing Guide</i> .....	80
Gambar 3.29 <i>Activity Diagram View Testing Guide</i> .....	81
Gambar 3.30 <i>Class Diagram</i> .....	82
Gambar 3.31 <i>Sequence Diagram Login</i> .....	83
Gambar 3.32 <i>Sequence Diagram Manage User</i> .....	84
Gambar 3.33 <i>Sequence Diagram Add User</i> .....	84
Gambar 3.34 <i>Sequence Diagram Edit User</i> .....	85
Gambar 3.35 <i>Sequence Diagram Delete User</i> .....	85
Gambar 3.36 <i>Sequence Diagram View User</i> .....	86
Gambar 3.37 <i>Sequence Diagram Manage Project</i> .....	86
Gambar 3.38 <i>Sequence Diagram Search Project</i> .....	87
Gambar 3.39 <i>Sequence Diagram Add Project</i> .....	88

Gambar 3.40 Sequence Diagram Edit Project .....	88
Gambar 3.41 Sequence Diagram Delete Project .....	89
Gambar 3.42 Sequence Diagram View Project.....	89
Gambar 3.43 Sequence Diagram Manage Finding.....	90
Gambar 3.44 Sequence Diagram Add Finding .....	90
Gambar 3.45 Sequence Diagram Edit Finding .....	91
Gambar 3.46 Sequence Diagram Delete Finding .....	92
Gambar 3.47 Sequence Diagram View Finding.....	92
Gambar 3.48 Sequence Diagram Change Finding Status .....	93
Gambar 3.49 Sequence Diagram Generate Report.....	93
Gambar 3.50 Sequence Diagram Manage Testing Guide.....	94
Gambar 3.51 Sequence Diagram Add Testing Guide .....	95
Gambar 3.52 Sequence Diagram Edit Testing Guide .....	96
Gambar 3.53 Sequence Diagram Delete Testing Guide .....	97
Gambar 3.54 Sequence Diagram View Testing Guide.....	98
Gambar 3.55 Entity Relationship Diagram.....	99
Gambar 3.56 Login Page .....	100
Gambar 3.57 Home Page ( senior penetration tester ) .....	101
Gambar 3.58 Home Page ( junior penetration tester ) .....	102
Gambar 3.59 Home Page ( client ).....	102
Gambar 3.60 Project Lists To Do ( senior penetration tester ) .....	103
Gambar 3.61 Project Lists To Do ( junior penetration tester ).....	104
Gambar 3.62 Project Lists To Do ( client ) .....	104
Gambar 3.63 Project Lists On Going ( senior penetration tester ).....	105
Gambar 3.64 Project Lists On Going ( junior penetration tester ).....	106
Gambar 3.65 Project Lists On Going ( client ) .....	106
Gambar 3.66 Project Lists Done ( junior penetration tester ) .....	107
Gambar 3.67 Project Lists Done .....	108
Gambar 3.68 Project Lists Done .....	108
Gambar 3.69 Project setelah klik icon 3 titik ( senior penetration tester ) .....	109
Gambar 3.70 Project setelah klik icon 3 titik ( junior penetration tester ).....	109
Gambar 3.71 Project setelah klik icon 3 titik ( client ) .....	110
Gambar 3.72 Form Create New Project ( senior penetration tester ) .....	111
Gambar 3.73 Modal Add Testing Object di Form New Create Project dan Form Edit Project .....	112
Gambar 3.74 Modal Edit Testing Object di Form New Create Project dan Form Edit Project .....	112
Gambar 3.75 Modal Select Team Member di Form New Create Project dan Form Edit Project .....	113
Gambar 3.76 Modal Select Client di Form Create Project dan Form Edit Project .....	113
Gambar 3.77 Form Edit Project.....	114
Gambar 3.78 Modal Delete Project .....	115
Gambar 3.79 View Detail Project with Not Fixed Finding Lists ( senior penetration tester dan junior penetration tester ).....	116
Gambar 3.80 View Detail Project with Not Fixed Finding Lists ( client ) .....	117
Gambar 3.81 View Detail Project with Not Fixed Finding Lists setelah klik button action ( senior penetration tester ).....	118

Gambar 3.82 View Detail Project with Not Fixed Finding Lists setelah klik button action ( junior penetration tester ) .....	118
Gambar 3.83 View Detail Project with Fixed Finding Lists ( senior penetration tester dan junior penetration tester ).....	119
Gambar 3.84 View Detail Project with Fixed Finding Lists ( Client ) .....	119
Gambar 3.85 View Detail Project with Not Fixed Finding Lists setelah klik button action ( senior penetration tester ).....	119
Gambar 3.86 View Detail Project with Not Fixed Finding Lists setelah klik button action ( junior penetration tester ) .....	120
Gambar 3.87 Form Add New Finding ( senior penetration tester dan junior penetration tester ) .....	121
Gambar 3.88 Modal Add Evidence ( senior penetration tester dan junior penetration tester ) .....	122
Gambar 3.89 Modal Edit Evidence ( senior penetration tester dan junior penetration tester ) .....	123
Gambar 3.90 Form Edit Finding ( senior penetration tester dan junior penetration tester ).....	124
Gambar 3.91 View Detail Finding .....	125
Gambar 3.92 Modal Delete Finding .....	126
Gambar 3.93 Testing Guide Lists.....	127
Gambar 3.94 Detail Testing Guide .....	128
Gambar 3.95 Detail Testing .....	128
Gambar 3.96 Detail Testing setelah klik icon 3 titik .....	129
Gambar 3.97 Form Add Testing Guide .....	130
Gambar 3.98 Form Edit Testing Guide .....	131
Gambar 3.99 Modal Delete Testing Guide .....	132
Gambar 3.100 Users Management.....	132
Gambar 3.101 User Management setelah klik button Action .....	133
Gambar 3.102 Form Add New User.....	133
Gambar 3.103 Form Edit User.....	134
Gambar 3.104 Modal Change Password User .....	135
Gambar 3.105 Modal Delete User .....	135
Gambar 4.1 Tampilan halaman Login.....	141
Gambar 4.2 Tampilan Home Page.....	141
Gambar 4.3 Tampilan Menu Logout .....	142
Gambar 4.4 Tampilan Side Navigation Bar Principal Penetration Tester.....	143
Gambar 4.5 Tampilan Side Navigation Bar Penetration Tester.....	143
Gambar 4.6 Tampilan Side Navigation Bar Client .....	143
Gambar 4.7 Tampilan Search Project.....	144
Gambar 4.8 Tampilan Hasil Pencarian dari Search Project .....	144
Gambar 4.9 Tampilan Project List dengan status To Do .....	145
Gambar 4.10 Tampilan Project List dengan status On Going .....	145
Gambar 4.11 Tampilan Project List dengan status Done .....	146
Gambar 4.12 Tampilan Icon Titik Tiga Project di role Principal Penetration Tester .....	147
Gambar 4.13 Tampilan Icon Titik Tiga Project di role Penetration Tester .....	147
Gambar 4.14 Tampilan Icon Titik Tiga Project di role Client .....	147
Gambar 4.15 Tampilan Create New Project.....	148
Gambar 4.16 Tampilan Alert Sukses Create New Project.....	149

Gambar 4.17 Tampilan Alert Gagal Create New Project .....	149
Gambar 4.18 Tampilan Add Testing Object pada Create New Project dan Edit Project .....	149
Gambar 4.19 Tampilan Edit Testing Object pada Create New Project dan Edit Project .....	150
Gambar 4.20 Tampilan Add Team Member pada Create New Project dan Edit Project .....	150
Gambar 4.21 Tampilan Add Client pada Create New Project dan Edit Project ...	151
Gambar 4.22 Tampilan Edit Project .....	152
Gambar 4.23 Tampilan Alert Sukses Edit Project .....	153
Gambar 4.24 Tampilan Alert Gagal Edit Project.....	153
Gambar 4.25 Tampilan Konfirmasi Delete Project .....	153
Gambar 4.26 Tampilan Alert Sukses Delete Project .....	154
Gambar 4.27 Tampilan Alert Gagal Delete Project.....	154
Gambar 4.28 Tampilan View Detail Project.....	155
Gambar 4.29 Tampilan Not Fixed Finding Lists .....	156
Gambar 4.30 Tampilan Fixed Finding Lists .....	156
Gambar 4.31 Tampilan Action Finding dengan status Not Fixed di role Principal Penetration Tester.....	157
Gambar 4.32 Tampilan Action Finding dengan status Fixed di role Principal Penetration Tester.....	157
Gambar 4.33 Tampilan Action Finding pada semua status di role Principal Penetration Tester.....	157
Gambar 4.34 Tampilan Add New Finding .....	159
Gambar 4.35 Tampilan Add New Finding lanjutan .....	160
Gambar 4.36 Tampilan Alert Sukses Add New Finding .....	160
Gambar 4.37 Tampilan Alert Gagal Add New Finding .....	160
Gambar 4.38 Tampilan Add Evidence .....	161
Gambar 4.39 Tampilan Edit Evidence .....	161
Gambar 4.40 Tampilan Edit Finding .....	162
Gambar 4.41 Tampilan Edit Finding lanjutan .....	163
Gambar 4.42 Tampilan Alert Sukses Edit Finding .....	164
Gambar 4.43 Tampilan Alert Gagal Edit Finding.....	164
Gambar 4.44 Tampilan Konfirmasi Delete Finding .....	164
Gambar 4.45 Tampilan Alert Sukses Delete Finding .....	165
Gambar 4.46 Tampilan Alert Gagal Delete Finding .....	165
Gambar 4.47 Tampilan View Detail Finding .....	166
Gambar 4.48 Tampilan Alert Sukses Change Finding Status .....	167
Gambar 4.49 Tampilan Alert Gagal Change Finding Status .....	167
Gambar 4.50 Tampilan Generate Report.....	167
Gambar 4.51 Tampilan Testing Guide List .....	168
Gambar 4.52 Tampilan View Detail Testing Guide .....	169
Gambar 4.53 Tampilan Salah Satu Testing Guide .....	169
Gambar 4.54 Tampilan Icon Titik Tiga Testing Guide.....	170
Gambar 4.55 Tampilan Add Testing Guide.....	171
Gambar 4.56 Tampilan Add Testing Guide lanjutan.....	172
Gambar 4.57 Tampilan Alert Sukses Add Testing Guide .....	173
Gambar 4.58 Tampilan Alert Gagal Add Testing Guide .....	173
Gambar 4.59 Tampilan Edit Testing Guide .....	174

Gambar 4.60 Tampilan <i>Edit Testing Guide</i> lanjutan .....	175
Gambar 4.61 Tampilan <i>Alert Sukses Edit Testing Guide</i> .....	176
Gambar 4.62 Tampilan <i>Alert Gagal Edit Testing Guide</i> .....	176
Gambar 4.63 Tampilan <i>Delete Testing Guide</i> .....	176
Gambar 4.64 Tampilan <i>Alert Sukses Delete Testing Guide</i> .....	177
Gambar 4.65 Tampilan <i>Alert Gagal Delete Testing Guide</i> .....	177
Gambar 4.66 Tampilan <i>User Lists</i> .....	178
Gambar 4.67 Tampilan <i>Add New User</i> .....	179
Gambar 4.68 Tampilan <i>Alert Sukses Add User</i> .....	179
Gambar 4.69 Tampilan <i>Alert Gagal Add User</i> .....	179
Gambar 4.70 Tampilan <i>Action User</i> .....	180
Gambar 4.71 Tampilan <i>Edit User</i> .....	180
Gambar 4.72 Tampilan <i>Alert Sukses Edit User</i> .....	181
Gambar 4.73 Tampilan <i>Alert Gagal Edit User</i> .....	181
Gambar 4.74 Tampilan <i>Change Password</i> .....	181
Gambar 4.75 Tampilan <i>Alert Sukses Change Password</i> .....	181
Gambar 4.76 Tampilan <i>Alert Gagal Edit Change Password</i> .....	182
Gambar 4.77 Tampilan Konfirmasi <i>Delete User</i> .....	182
Gambar 4.78 Tampilan <i>Alert Sukses Delete User</i> .....	182
Gambar 4.79 Tampilan <i>Alert Gagal Delete User</i> .....	182
Gambar 4.80 Tampilan Konsistensi Letak Menu, Judul, dan Tombol di <i>Project Lists</i> .....	197
Gambar 4.81 Tampilan Konsistensi Letak Menu, Judul, dan Tombol di <i>Resources</i> .....	197
Gambar 4.82 Tampilan Konsistensi Letak Menu, Judul, dan Tombol di <i>Admin Panel</i> .....	198
Gambar 4.83 Tampilan Letak Menu Navigasi, tombol <i>Profile</i> dan <i>Search box</i> ... 199	199
Gambar 4.84 Tampilan Konfirmasi <i>Logout</i> .....	199
Gambar 4.85 Tampilan Konfirmasi <i>Delete Project</i> .....	200
Gambar 4.86 Tampilan Konfirmasi <i>Delete Finding</i> .....	200
Gambar 4.87 Tampilan Konfirmasi <i>Delete Resources</i> .....	200
Gambar 4.88 Tampilan Konfirmasi <i>Delete User</i> .....	201
Gambar 4.89 Tampilan Sukses <i>Add Project</i> .....	201
Gambar 4.90 Tampilan Sukses <i>Edit Project</i> .....	201
Gambar 4.91 Tampilan Sukses <i>Delete Project</i> .....	202
Gambar 4.92 Tampilan <i>User</i> Salah Memasukkan <i>Username</i> dan <i>Password</i> .....	202
Gambar 4.93 Tampilan <i>user</i> setelah klik tombol <i>Add Team Member</i> .....	203
Gambar 4.94 Tampilan <i>user</i> setelah klik tombol <i>Delete Project</i> di salah satu <i>action project</i> .....	203
Gambar 4.95 Tampilan Pilihan Status pada <i>Project Lists</i> .....	204
Gambar 4.96 Tampilan Pilihan Status pada <i>Finding Lists</i> .....	204
Gambar 4.97 Tampilan Daftar Fitur pada Menu Navigasi .....	205
Gambar 4.98 Tampilan <i>Icon</i> Titik Tiga pada <i>Project</i> .....	205

## **DAFTAR LAMPIRAN**

<b>LAMPIRAN.....</b>	213
LAMPIRAN 1 - Transkrip Wawancara mengenai <i>User Requirement</i> .....	213
LAMPIRAN 2 - Transkrip Wawancara mengenai <i>User Feedback</i> .....	226
LAMPIRAN 3 - Persetujuan Transkrip Wawancara dengan Narasumber.....	232
LAMPIRAN 4 - Surat Non Survei .....	237



# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Di era modern sekarang ini, semua negara di seluruh dunia ikut berkembang dengan adanya teknologi dan inovasi baru yang hadir di tengah kita. Termasuk dengan digitalisasi kegiatan manusia yang dapat dilakukan secara daring, mulai dari sekolah dan kuliah *online*, bekerja *online*, berbelanja *online* dan masih banyak kegiatan lainnya yang dapat diselesaikan hanya dengan menggunakan perangkat elektronik saja. Di Indonesia sendiri digitalisasi ini bisa dikatakan bisa dirasakan oleh seluruh penduduk Indonesia, dilansir dari data yang dimiliki oleh DataReportal menunjukkan bahwa di Indonesia terdapat sekitar 204.7 juta pengguna internet per Januari 2022 kemarin, dibandingkan tahun 2021 pengguna internet di Indonesia bertambah sekitar 2.1 juta pengguna (Data Reportal, 2022) yang artinya perkembangan pesat dari digitalisasi yang mengharuskan kita melakukan kegiatan secara daring, ini didorong karena adanya pandemi COVID-19 yang melanda seluruh dunia termasuk Indonesia sehingga memberlakukan beberapa peraturan baru seperti PPKM, kegiatan daring, dan lainnya.

Karena kondisi tersebut, mendorong pengembang aplikasi untuk mengembangkan aplikasi baru baik *mobile* maupun *website* yang dapat membantu masyarakat dalam melakukan kegiatan sehari-hari, ini berakibat permintaan aplikasi rilis dalam kurun waktu yang singkat sehingga seringkali sisi keamanan dari aplikasi sering terabaikan. Menurut data yang dihimpun Badan Siber Sandi Negara (BSSN) mencatat lebih dari 1.6 miliar trafik anomali terhadap Indonesia dalam kurun waktu 1 Januari 2021 sampai 31 Desember 2021 (BSSN, 2021). Kondisi tersebut mendorong penegembang harus lebih *aware* terhadap cyber security untuk menjaga data-data pengguna dan juga data perusahaan agar tetap aman. Menurut survei yang dilakukan oleh Palo Alto terhadap 100 perusahaan yang ada di Indonesia, sebanyak 84 persen dari perusahaan tersebut meningkatkan anggaran untuk *cyber security* di tahun 2020 (Palo Alto, 2020). Oleh karena itu, seiring dengan meningkatnya anggaran perusahaan untuk *cyber security*, tuntutan pengujian aplikasi oleh vendor juga semakin tinggi.

Sebagai *penetration tester* yang bekerja pada vendor tentunya permintaan pengujian aplikasi akan meningkat sehingga *penetration tester* juga perlu beradaptasi dengan jumlah pengujian keamanan yang banyak dengan waktu yang relatif cepat dan efisien. Tentu saja ini akan menjadi tantangan dan kesulitan untuk para *penetration tester* mengingat ada beberapa tahapan dalam pengujian keamanan yang harus dilakukan.

Dari masalah tersebut ada beberapa opsi yang bisa diambil, salah satunya adalah memfokuskan *penetration tester* dalam melakukan pengujian sehingga hasil pengujian dapat memenuhi standar, tetapi jika *penetration tester* tersebut menguji banyak aplikasi secara bersamaan maka akan menimbulkan masalah baru seperti report tidak memenuhi standar atau ada bagian pengujian yang terlewat. Maka dari itu dalam melakukan pengujian perlu adanya kolaborasi bersama penguji lain dan memudahkan kolaborasi dengan klien terkait, sehingga dalam proses *penetration testing* diharapkan akan mempercepat waktu dan juga meningkatkan kualitas pengujian serta memudahkan klien dalam mengetahui perkembangan hasil pengujian.

Para *penetration tester* juga mempunyai kemampuan yang berbeda secara teknis dalam melakukan pengujian, asesmen, analisis, dan eksploitasi suatu kerentanan. Sering terjadinya gap kemampuan teknis antara *penetration tester* yang membuat *penetration tester* tidak bisa melakukan pengujian secara menyeluruh dan mendalam, hal ini tentu saja berpengaruh kepada hasil *penetration testing* yang dilakukan. Maka dari itu diperlukannya sebuah acuan atau petunjuk dalam melakukan *penetration testing*.

Dari penjabaran yang sudah dijelaskan, aplikasi *mobile* dan *website* sangat berperan penting dalam kegiatan setiap orang secara daring. Dari sisi keamanan mendorong *penetration tester* melakukan pengujian keamanan dengan cepat dan efisien. Tetapi ada beberapa masalah yang dihadapi untuk tetap menjaga kualitas pengujian dengan banyaknya aplikasi yang harus diuji. Dengan memfokuskan *penetration tester* pada saat pengujian dan juga melakukan kolaborasi bersama penguji lain dalam melakukan penetration test ini diharapkan tetap menjaga kualitas hasil pengujian. Selain itu, dengan mempermudah kolaborasi antara penetration tester dengan klien dalam memberikan dan mendapatkan perkembangan hasil pengujian.

Untuk menengahi permasalahan tersebut, penulis membuat solusi berupa sebuah aplikasi manajemen *vulnerability* yang memudahkan para *penetration tester*

melakukan kolaborasi baik dengan *penetration tester* lain ataupun dengan klien dan juga memudahkan dalam melakukan pembuatan laporan secara otomatis serta mempermudah *transfer knowledge* antar *penetration tester*. Ini diharapkan dapat membantu *penetration tester* agar tetap fokus melakukan pengujian sehingga menghasilkan kualitas pengujian yang memenuhi standar.

## 1.2 Rumusan Masalah

Dari penjelasan latar belakang yang sudah dijelaskan maka berikut poin-poin permasalahan yang akan sebagai berikut:

1. Bagaimana merancang dan mengimplementasikan sebuah aplikasi yang dapat memudahkan kerja sama antar sesama *penetration tester* dan antar *penetration tester* dengan *client*?
2. Bagaimana merancang dan mengimplementasikan proses automasi dalam pembuatan *report penetration testing*?
3. Bagaimana merancang dan mengimplementasikan sebuah aplikasi yang dapat membantu *transfer knowledge* sesama *penetration tester*?

## 1.3 Ruang Lingkup

1. Aplikasi *vulnerability management* berbasis *website*.
2. *Back end* aplikasi dibangun menggunakan bahasa pemrograman *python* dengan bantuan *framework django*.
3. *Front end* aplikasi dibangun menggunakan HTML, CSS dan *javascript* menggunakan bantuan *framework CSS Bootstrap* dengan tema *SB Admin 2* dan *library javascript jQuery*.
4. *Database Management System (DBMS)* yang digunakan adalah *MySQL database*.
5. *Deployment tools* yang digunakan untuk menjalankan aplikasi *vulnerability management* adalah Docker.
6. Aplikasi *vulnerability management* difokuskan sebagai platform kolaborasi antar *penetration tester* dengan *penetration tester* lain dan antar *penetration tester* dengan *client*. Kolaborasi tersebut meliputi pengelolaan *project penetration testing*, dan pengelolaan *finding* setiap *project penetration testing*.

7. Aplikasi *vulnerability management* difokuskan juga sebagai tempat menampung *knowledge penetration testing* yang terdiri dari pengetahuan yang dapat membantu dalam pengujian keamanan aplikasi dan penulisan *finding*.
8. Jenis aplikasi yang digunakan sebagai contoh untuk aplikasi *vulnerability management* adalah aplikasi berbasis web dan aplikasi berbasis *mobile*.

## 1.4 Tujuan dan Manfaat

### 1.4.1 Tujuan

1. Membuat platform yang dapat menyimpan *knowledge Penetration Testing*.
2. Membuat platform yang dapat menghasilkan *report Penetration Testing* secara automasi.
3. Membuat platform yang dapat digunakan untuk kolaborasi antar *Penetration Tester* dan klien.

### 1.4.2 Manfaat

1. Membantu *Penetration Tester* lebih mudah dalam mendapatkan *knowledge Penetration Testing*.
2. Membantu *Penetration Tester* agar lebih efisien dalam pembuatan *report Penetration Testing*.
3. Membantu *Penetration Tester* dalam melakukan kolaborasi dengan *Penetration Tester* lain atau klien.

## 1.5 Metodologi

### 1.5.1 Metode Penelitian

1. Melakukan pengamatan berupa perbandingan antara aplikasi *vulnerability management* yang dikembangkan penulis dengan tools *vulnerability management* lain.
2. Melakukan pengamatan yang berkaitan dengan ketertarikan *user* yaitu *penetration tester* dan klien tentang aplikasi *vulnerability management* berbasis *website*.

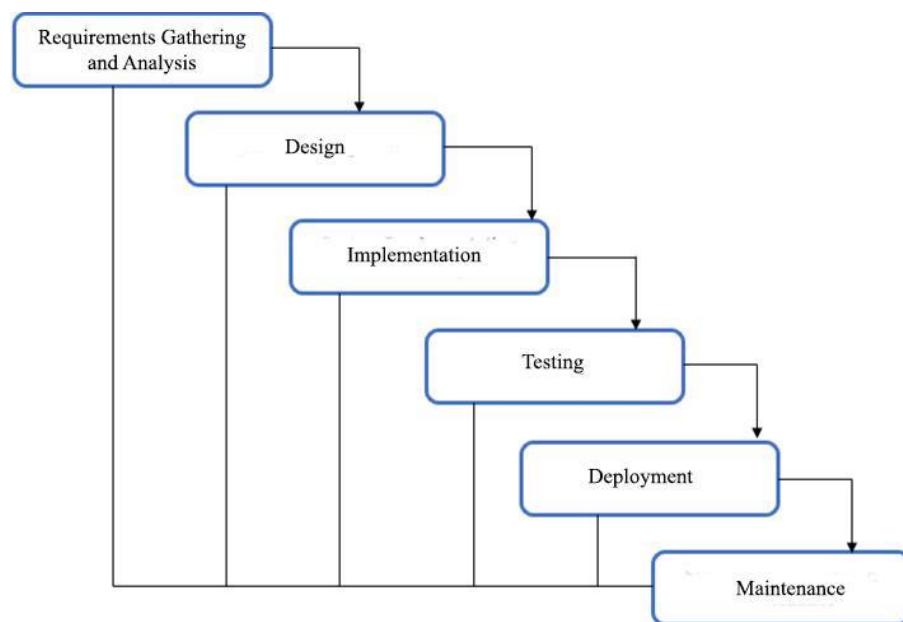
### 1.5.2 Metode Perancangan

1. Membuat desain *user interface* untuk tools *vulnerability management* ini agar mudah untuk digunakan.

2. Membuat desain sistem yang sesuai dengan *user requirement* agar tools ini dapat bekerja secara efisien menggunakan *Use Case Diagram*, *Use Case Description*, *Activity Diagram*, *Class Diagram*, dan *Entity Relationship Diagram*.

### 1.5.3 Metode Pengembangan

*Software Development Life Cycle* (SDLC) yang akan digunakan adalah *Waterfall Model*. Dalam *Waterfall Model* terbagi menjadi 6 tahapan dalam mengembangkan aplikasi, yaitu *Requirement Gathering and Analysis*, *Design*, *Implementation*, *Testing*, *Deployment*, dan *Maintenance*.



**Gambar 1.1 Metode SDLC Waterfall**

## 1.6 Sistematika Penulisan

Penulisan skripsi ini terbagi menjadi beberapa bab, yaitu:

### 1.6.1 BAB 1 PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang, rumusan masalah, ruang lingkup, tujuan dan manfaat, metode penelitian yang digunakan, dan juga sistematika penulisan skripsi dalam pembuatan aplikasi *tools vulnerability management*.

### **1.6.2 BAB 2 TINJAUAN REFERENSI**

Bab ini menjelaskan mengenai landasan teori yang digunakan sebagai panduan dalam pembuatan aplikasi ini serta metode yang digunakan dalam penelitian ini.

### **1.6.3 BAB 3 METODE PENELITIAN**

Bab ini menjelaskan lebih dalam mengenai metode yang digunakan dalam pembuatan aplikasi *tools vulnerability management* ini beserta dengan penjabaran kerangka berpikir, analisis masalah, dan juga rancangan-rancangan aplikasi yang dibuat.

### **1.6.4 BAB 4 HASIL DAN PEMBAHASAN**

Bab ini menjelaskan mengenai spesifikasi yang digunakan dalam pembuatan aplikasi yang dibuat, tata cara penggunaan aplikasi *tools vulnerability management*, dan hasil evaluasi aplikasi yang dibuat berdasarkan hasil interview.

### **1.6.5 BAB 5 SIMPULAN DAN SARAN**

Bab ini berisi kesimpulan yang didapatkan dari perancangan dan pembuatan aplikasi yang telah dilakukan, serta saran dan masukan yang dapat digunakan untuk pengembangan aplikasi lebih lanjut di masa mendatang.

## **BAB 2**

### **TINJAUAN REFERENSI**

#### **2.1 *Penetration Testing***

*Penetration Testing* merupakan tipe dari *security assessment* yang menggunakan simulasi *real-world attack* pada sistem komputer, jaringan, atau aplikasi untuk mengidentifikasi kerentanan dan potensi eksploitasi (Nagpure & Kurkure, 2017).. *Penetration Testing* dapat dilakukan pada berbagai macam lingkungan atau platform seperti *mobile*, website, infrastruktur, dan masih banyak lagi. Karena *Penetration Testing* ini bisa dilakukan di bermacam lingkungan atau platform, sebuah yayasan bernama *The Open Web Application Security Project* atau disingkat OWASP membuat sebuah *framework* yang bisa digunakan oleh penguji sebagai acuan dalam melakukan *Penetration Testing*. *The OWASP Testing Project* telah dikembangkan selama bertahun-tahun, proyek ini bertujuan membantu penguji memahami apa, mengapa, kapan, di mana, dan bagaimana menguji sebuah aplikasi. *The OWASP Testing Project* ini telah memberikan cara pengujian yang lengkap serta panduan pengujian yang menjelaskan secara rinci baik kerangka pengujian umum dan teknik yang diperlukan untuk mengimplementasikan panduan ini dalam praktik. (Saad & Mitchell, 2020).

##### **2.1.1 *Web Penetration Testing***

*Web Penetration Testing* adalah kegiatan pengujian keamanan dengan cakupan pengujian yaitu website. Berdasarkan klasifikasinya, *Web Penetration Testing* dibagi menjadi dua yaitu *Client Side Attack* atau serangan yang dilakukan penyerang terhadap klien situs website tertentu dan *Server Side Attack* atau serangan terhadap *server* website, sedangkan untuk cara pengujian dibagi menjadi dua yaitu *Manual Testing* dan *Automated Testing*. (Nagendr, Adithyan, Chethana, Camillus, & Bala, 2019). *Web Security Testing Guide* (WSTG) adalah sebuah panduan untuk melakukan pengujian terhadap web aplikasi atau web *service*, yang dibuat dari berbagai kolaborasi antar *cybersecurity profesional* dan *volunteer* yang berdedikasi.

### 2.1.2 *Mobile Penetration Testing*

*Mobile Penetration Testing* adalah kegiatan pengujian keamanan dengan cakupan pengujian yaitu aplikasi *android*. Diambil dari banyaknya pasar *smartphone* tahun 2015, *Android* dan *IOS* adalah yang paling terlaris, juga aplikasi *mobile* dibagi menjadi tiga kategori yaitu *Native*, *Mobile web*, dan *Hybrid* (Velu, 2016). *Mobile Security Testing Guide* (MSTG) adalah manual komprehensif untuk menguji keamanan dari aplikasi *mobile*. Pada MSTG menjelaskan proses dan teknik untuk memverifikasi persyaratan yang tercantum dalam *Mobile Application Security Verification Standard* (MASVS), dan memberikan garis dasar untuk pengujian keamanan yang lengkap dan konsisten.

## 2.2 *Penetration Tester*

*Penetration Tester* merupakan keamanan profesional yang melakukan vulnerability assessments dan penetration testing untuk mengidentifikasi kerentanan dan potensi eksloitasi. Tujuan dari evaluasi keamanan tersebut adalah sebagai peningkatan keamanan dan menjaga dari ancaman keamanan (Nagpure & Kurkure, 2017).

## 2.3 *Vulnerability*

*Vulnerability* merupakan sebuah kelemahan atau kerentanan pada keamanan yang bisa dieksloitasi untuk mendapatkan akses tidak sah ke sebuah sistem atau jaringan (McKinnel, Dargahi, Dehghantanha, Choo, 2019). Dengan banyaknya *open source code* yang tersedia untuk dianalisis, ada peluang untuk mempelajari tentang pola *bug* yang dapat menyebabkan kerentanan keamanan untuk membantu dalam penemuan kerentanan. Kemajuan terbaru dalam *deep learning*, *natural language processing*, *speech recognition*, dan *image processing* telah menunjukkan potensi besar model saraf untuk memahami bahasa alami. Hal ini telah mendorong para peneliti di sektor keamanan siber dan rekayasa perangkat lunak untuk memanfaatkan *deep learning* ini untuk mempelajari dan memahami pola kode yang rentan dan yang terindikasi memiliki kode yang rentan. (Bahaa, Kamal, Ghoneim, 2022).

## 2.4 Vulnerability Management

*Vulnerability Management* adalah proses dari identifikasi, menentukan, dan mitigasi kerentanan pada sistem komputer, jaringan, atau aplikasi. Tujuannya adalah untuk mengurangi resiko dari eksploitasi pada kerentanan tersebut. *Program vulnerability management* secara keseluruhan melibatkan *scanning*, identifikasi kerentanan baru, memprioritaskan status *critical-low*, dan implementasi remediasi bagaimana menyelesaikan kerentanan tersebut (Nanda & Ghugar, 2017).

Proses *Vulnerability Management* sendiri terbagi menjadi empat langkah berikut. Pertama, *Identifying Vulnerabilities* atau mengidentifikasi kerentanan. Mengidentifikasi kerentanan pada sistem dapat dilakukan dengan cara *scanning*, *scanning* ini dapat berupa *scanning* jaringan dengan mengirimkan paket TCP atau UDP, menemukan *port* yang terbuka dan layanan yang ada pada sistem, masuk ke dalam sistem untuk mengumpulkan informasi, dan menghubungkan informasi yang ditemukan dengan kerentanan yang diketahui. Kedua adalah *Evaluating Vulnerabilities*, setelah kerentanan diidentifikasi, kerentanan tersebut perlu dievaluasi sehingga risiko yang ditimbulkannya ditangani dengan tepat dan sesuai dengan strategi manajemen risiko. Solusi manajemen kerentanan akan memberikan peringkat risiko dan skor yang berbeda untuk kerentanan, skor ini membantu dalam memberi tahu organisasi kerentanan mana yang harus mereka fokuskan terlebih dahulu. Ketiga adalah *Treating Vulnerabilities*, ketika kerentanan sudah divalidasi dan dianggap sebagai risiko, langkah selanjutnya adalah bagaimana memperlakukan kerentanan tersebut, terdapat 3 cara dalam melakukannya, yaitu *Remediation* atau melakukan patching sehingga kerentanan tidak dapat dieksploitasi lagi, *Mitigation* atau mengurangi kemungkinan atau dampak yang bisa ditimbulkan dari eksploitasi, hal ini dilakukan ketika perbaikan atau patch belum tersedia, kemudian *Acceptance* atau menerima kerentanan tanpa melakukan perbaikan, hal ini dilakukan jika kerentanan dianggap memiliki risiko rendah atau biaya perbaikannya lebih mahal dibanding dampak yang ada. Keempat adalah *Reporting Vulnerabilities*, dengan melakukan *vulnerability assessments* secara teratur dapat membuat perusahaan atau organisasi mengerti efisiensi dari program *vulnerability management* mereka. Dengan adanya visualisasi laporan data kerentanan membuat bagian IT mudah mengerti bagaimana cara menangani kerentanan tersebut.

## 2.5 Aplikasi

Aplikasi adalah suatu perangkat lunak yang dalam proses pengembangannya menggabungkan beberapa fitur sehingga dalam setiap fitur dapat diakses dan digunakan secara bersamaan dalam sebuah bentuk bernama aplikasi. (Tolle, Pinandito, Kharisma, & Dewi, 2017). Aplikasi merupakan sebuah bentuk alat yang dapat membantu manusia dalam kegiatan sehari-hari sehingga banyak orang mengembangkan aplikasi untuk menyelesaikan masalah yang timbul dalam masyarakat. Sebuah aplikasi dapat diintegrasikan dengan berbagai modul dan *interface* sehingga mudah dalam penggunaannya.

## 2.6 Website

*Website* adalah sejumlah halaman *web* yang memiliki topik saling terkait, terkadang disertai pula dengan berkas-berkas gambar, video, atau berkas lainnya. (Ardhana, 2012). *Website* dapat diakses secara *online* yang berisikan konten tertentu dan dijadikan kesatuan informasi.

## 2.7 Docker

*Docker* merupakan sebuah platform *open source* yang menjalankan aplikasi dan membuat prosesnya lebih mudah untuk dikembangkan dan didistribusikan. Aplikasi-aplikasi tersebut dibangun dan disatukan di *docker* dalam bentuk *container*. *Container* tersebut berjalan terisolasi dari sistem operasi *kernel* sehingga jika terjadi sesuatu pada *container*, tidak berefek pada sistem operasi (Rad, Bhatti, Ahmadi, 2017).

### 2.7.1 Docker Compose

Menurut *Docker* (2022), *Compose* merupakan sebuah alat untuk mendefinisikan dan menjalankan beberapa *container* pada aplikasi *docker*. Dengan *compose*, konfigurasi atau *command* dari beberapa *container* dapat disatukan menjadi sebuah *file*. Penggunaan *docker compose* memberikan kemudahan pada *developer* untuk mengelola sebuah aplikasi yang memiliki beberapa *container* sehingga memudahkan dalam *build*, *test*, dan *deploy* aplikasi yang menggunakan *container*.

## 2.8 Github

*Github* adalah layanan *open source* untuk proyek pengembangan perangkat lunak yang menggunakan sistem kendali versi *Git* dan layanan *hosting* internet. *Github* banyak digunakan untuk bahasa pemrograman karena memberikan kontrol akses dan beberapa fitur kolaborasi seperti pelacakan *bug*, permintaan fitur, manajemen tugas, dan wiki untuk setiap proyek. (Irfan, 2020).

## 2.9 Common Vulnerability Scoring System (CVSS)

*Common Vulnerability Scoring System* atau CVSS adalah sebuah *framework* terbuka untuk menggambarkan karakteristik dan tingkat keparahan dari kerentanan perangkat lunak. CVSS terdiri dari tiga kelompok metrik: *Base*, *Temporal*, dan *Environmental*. *Base* mewakili kualitas yang membangun kerentanan dan konstan sepanjang waktu juga di seluruh lingkungan pengguna, *Temporal* mencerminkan karakteristik kerentanan yang berubah seiring waktu, dan *Environmental* mewakili karakteristik kerentanan yang unik bagi pengguna lingkungan. Metrik *Base* menghasilkan skor mulai dari 0 hingga 10, yang kemudian dapat dimodifikasi dengan menilai metrik *Temporal* dan *Environmental*. Skor CVSS juga direpresentasikan sebagai *string* vektor, representasi tekstual dari nilai yang digunakan untuk mendapatkan skor (Mell & Gueye, 2020).

## 2.10 Common Weakness Enumeration (CWE)

*Common Weakness Enumeration* atau CWE adalah daftar penting dari jenis kelemahan perangkat lunak. Daftar ini digunakan sebagai *vulnerability database* untuk menggambarkan kelemahan keamanan yang mendasari dalam kerentanan yang dianalisis. Dengan daftar ini, membuka kemungkinan menggunakan *vulnerability analysis* untuk mengidentifikasi kelemahan paling signifikan pada sebuah kerentanan. (Mell & Gueye, 2020). CWE ini merupakan inisiatif strategis Jaminan Perangkat Lunak yang disponsori oleh *National Cyber Security Division of the U.S. Department of Homeland Security*. (MITRE, 2022).

## 2.11 *Create, Read, Update, Delete (CRUD)*

*Create, Read, Update, Delete* atau *CRUD* adalah akronim yang digunakan untuk mendefinisikan operasi dasar yang bisa dilakukan pada data dalam sebuah *database* atau sistem penyimpanan lain dan digunakan secara luas pada industri pengembangan *web* (Satzinger J, Jackson R, & Burd S, 2015).

## 2.12 *Prototyping*

*Prototyping* adalah proses dari membuat model awal dari sebuah produk atau sistem dengan tujuan untuk mencari desain dan *testing* (Greenberg, 2018). *High-fidelity prototype* sangat detail, fungsional, dan sering digunakan untuk simulasi produk akhir dengan sangat mirip (Carroll, 2016). Sementara *low-fidelity prototype* kurang detail atau simpel dan lebih digunakan untuk tujuan eksplorasi seperti mengetes dasar interaksi atau mengumpulkan *feedback* dari pengguna (Nielsen, Molich, 2014).

## 2.13 *SDLC Waterfall*

Penggunaan model *waterfall* dalam pengembangan aplikasi bertujuan agar hasil akhir dari aplikasi lebih terstruktur sehingga dalam pengembangannya lebih mudah dalam menentukan kebutuhan yang diperlukan aplikasi tersebut. Penggunaan model *waterfall* juga cocok untuk *project* yang memiliki *requirement* dan penjelasan yang jelas untuk setiap fase pengembangan dan gambaran *project* secara keseluruhan sehingga mudah dalam melacak *progress* dan mengurangi risiko kegagalan *project* (Kristanto, Andrayana, Benramhman, 2020).. Menurut Sommerville (2019), terdapat 6 tahapan dari model *waterfall*:

1. *Requirement Gathering and Analysis*, merupakan tahapan pertama dalam pengembangan aplikasi yaitu dengan mengumpulkan dan mengidentifikasi kebutuhan dari aplikasi ini. Serta mengumpulkan informasi yang berkaitan dengan kebutuhan *user* sehingga dalam pengembangan dapat dengan tepat memenuhi apa yang diinginkan oleh *user*.
2. *Design*, merupakan tahapan selanjutnya setelah mengumpulkan informasi yang dapat digunakan dalam mengembangkan rancangan dari aplikasi. Pada

tahap ini dengan melakukan perancangan desain dari aplikasi termasuk arsitektur, *user interface*, dan desain *database*.

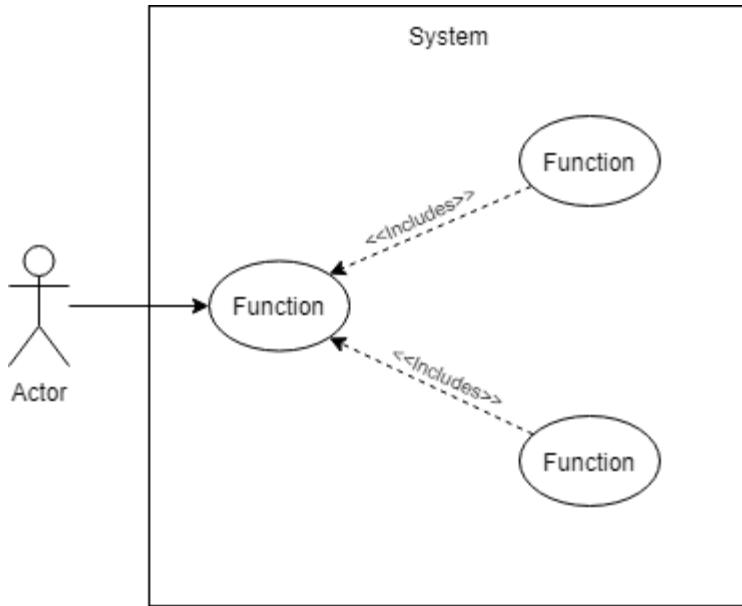
3. *Implementation*, merupakan tahapan dimana mengacu pada pengimplementasian dari rancangan aplikasi yang sudah dibuat pada tahap sebelumnya. Pada tahap ini *developer* akan mengembangkan aplikasi dari sisi *coding*.
4. *Testing*, pada tahap ini akan ada pengujian hasil pengembangan aplikasi yang telah dikembangkan pada tahap sebelumnya. Pengujian disini mencakup apakah ada kendala atau *error* saat menjalankan aplikasi yang telah dikembangkan dan juga apakah fungsionalitas dari aplikasi ini sudah tercapai.
5. *Deployment*, pada tahapan ini setelah melakukan pengujian terhadap aplikasi yang telah dikembangkan dan juga memastikan aplikasi sudah berjalan dengan baik, maka langkah selanjutnya adalah mendeploy aplikasi ke bagian *production* sehingga bisa digunakan oleh *user*.
6. *Maintenance*, merupakan tahapan dimana adanya proses pengoperasian aplikasi oleh *user*. Aplikasi yang sudah dikembangkan dilakukan pemeliharaan mulai dari menambahkan fitur, melakukan pembaharuan modul, serta memperbaiki aplikasi saat terdapat *error* atau malfungsi saat aplikasi dijalankan.

## **2.14 Unified Modeling Language (UML)**

Dalam pembuatan aplikasi, terdapat rancangan atau model yang telah dibuat dan dijadikan panduan dalam pengembangan aplikasi. Menurut Larman (2018), *Unified Modeling Language* atau *UML* adalah suatu bentuk pemodelan standar yang merepresentasikan sebuah sistem yang dibuat dalam berbagai jenis diagram . Diagram-diagram tersebut diantaranya adalah:

### **2.14.1 Use Case Diagram**

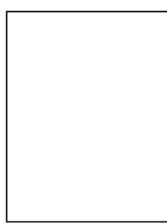
*Use Case Diagram* digunakan untuk merepresentasikan interaksi antara pengguna dan sistem digambarkan sebagai *actor* dan *use case* (Larman, 2018). Contoh gambaran dari *Use Case Diagram* sebagai berikut.

**Gambar 2.1 Contoh Use Case Diagram**

Simbol-simbol yang digunakan pada *use case diagram* diatas dapat dijelaskan sebagai berikut:

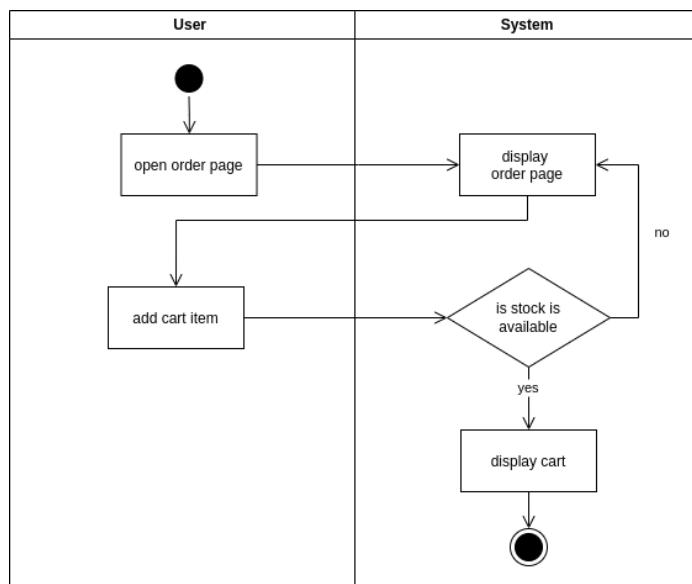
**Tabel 2.1 Penjelasan Simbol Use Case Diagram**

No	Gambar	Nama	Keterangan
1		Actor	Actor merepresentasikan pengguna yang digambarkan dalam bentuk subjek yang berinteraksi dengan sistem dimana terdapat fungsi-fungsi yang direpresentasikan sebagai use case.
2		Use Case	Use Case merupakan simbol yang merepresentasikan suatu fungsi dari sistem.

3	_____	<i>Association</i>	Simbol <i>Association</i> merepresentasikan interaksi yang menghubungkan antara <i>actor</i> dan <i>use case</i> .
4	<<Includes>> -----	<i>Includes</i>	Simbol <i>Includes</i> merepresentasikan hubungan fungsionalitas antara <i>use case</i> utama dengan <i>use case</i> yang menjadi bagian <i>use case</i> utama.
5		<i>System Boundary</i>	Simbol <i>System Boundary</i> merupakan pemisah antara sistem internal dan <i>actor</i> yang berada di luar sistem.

#### 2.14.2 Activity Diagram

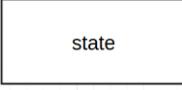
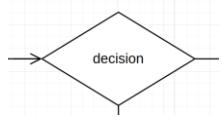
*Activity Diagram* digunakan untuk merepresentasikan urutan aktivitas yang dilakukan dari sebuah sistem. (Larman, 2018). Contoh gambaran *Activity Diagram* sebagai berikut.

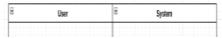


Gambar 2.2 Contoh *Activity Diagram*

Simbol yang digunakan pada *Activity Diagram* diatas dijelaskan sebagai berikut.

**Tabel 2.2 Penjelasan Simbol *Activity Diagram***

No	Gambar	Nama	Keterangan
1		<i>Start Point</i>	<i>Start Point</i> menggambarkan <i>state</i> awal sebuah aktivitas.
2		<i>End Point</i>	<i>End Point</i> menggambarkan <i>state</i> akhir dari aktivitas yang telah selesai.
3		<i>Activity</i>	<i>Activity</i> menggambarkan aktivitas yang dilakukan oleh <i>user</i> atau sistem.
4		<i>Decision</i>	<i>Decision</i> menggambarkan dari sebuah aktivitas ke aktivitas lain.

5		<i>Swimlane</i>	<i>Swimlane</i> menggambarkan <i>actor</i> yang melakukan aktivitas pada <i>activity diagram</i> .
---	---	-----------------	--

### 2.14.3 Use Case Description

*Use Case Description* adalah bagian spesifik dari *use case* dimana aktivitas dijelaskan dengan detail (Larman, 2018). Contoh gambaran *Use Case Description* sebagai berikut.

Use case name:	Add Item	
Actor:	User	
Trigger:	Actor wants to add an item to the cart.	
Flow of activities:	Actor:	System:
	1. Actor open the order page 2. Actor click add item in one of the item	1.1 System display order page 2.1 System check the item stock is still available or not 2.2 If the stock is not available redirected to order page, if available display cart

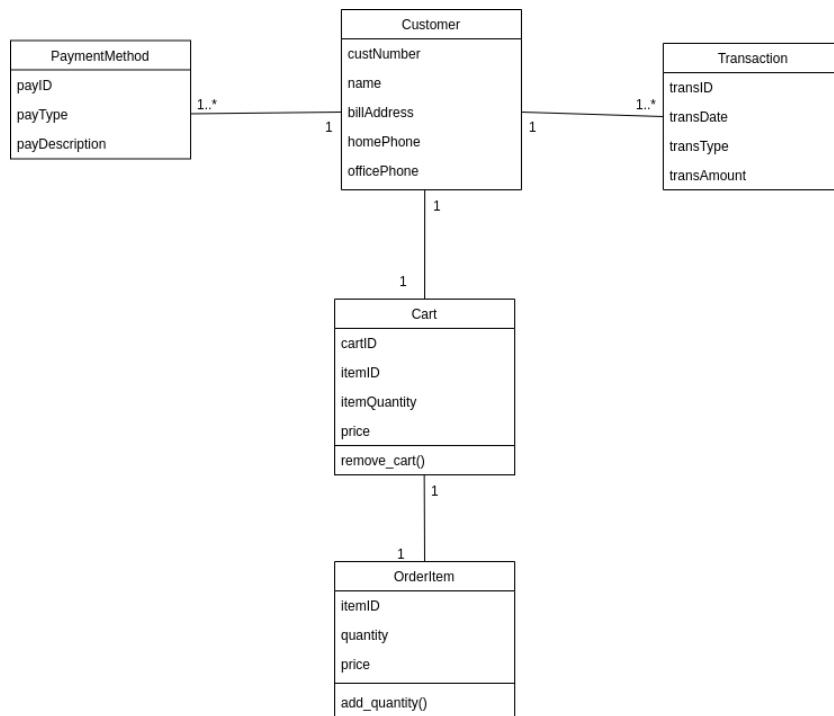
**Gambar 2.3 Contoh Use Case Description**

Berikut adalah penjelasan dari setiap entitas pada *Use Case Description* di atas sebagai berikut:

1. *Use Case Name*, merupakan nama dari proses *use case* yang akan dilakukan dan mewakilkan sebuah skenario yang akan terjadi.
2. *Actor*, merupakan individu yang berperan dalam dalam interaksi.
3. *Trigger*, merupakan suatu hal yang dapat memicu *use case* tertentu berjalan.
4. *Flow of activities*, merupakan penjabaran hal yang dilakukan oleh *Actor*.

#### 2.14.4 Class Diagram

*Class Diagram* digunakan untuk merepresentasikan struktur dari sebuah sistem dimana hal tersebut adalah struktur dari *class*-nya dan relasi dengan *class* lain (Larman, 2018). Contoh gambaran *Activity Diagram* sebagai berikut.

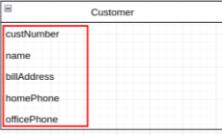
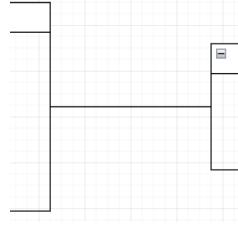
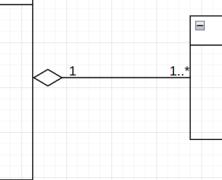
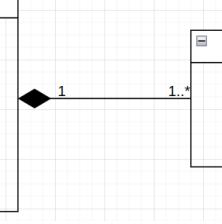


**Gambar 2.4 Contoh Class Diagram**

Simbol yang digunakan pada *Class Diagram* diatas dijelaskan sebagai berikut:

**Tabel 2.3 Penjelasan Simbol Class Diagram**

No	Gambar	Nama	Keterangan
1		<i>Class Name</i>	<i>Class Name</i> merupakan nama pada sebuah <i>class</i> yang bertujuan untuk membedakan dengan <i>class</i> lain dan menentukan fungsi dari komponen pada <i>class</i> tersebut.

2		Attributes	<i>Attributes</i> merupakan komponen yang terdapat pada sebuah <i>class</i> sehingga dapat dijabarkan dengan jelas.
3		Operation	<i>Operation</i> merupakan fungsi yang bekerja pada sebuah <i>class</i> .
4		Association	<i>Association</i> merupakan simbol yang merepresentasikan hubungan antar <i>class</i> sehingga menandakan adanya relasi dan fungsionalitas.
5		Aggregation	<i>Aggregation</i> merupakan simbol yang merepresentasikan hubungan yang ada atau terdiri dari aksi tertentu.
6		Composition	<i>Composition</i> merupakan simbol yang merepresentasikan hubungan bagian dari aksi atau <i>class</i> tertentu.

#### 2.14.5 Sequence Diagram

*Sequence Diagram* digunakan untuk memodelkan interaksi antar objek dan juga digunakan untuk mengidentifikasi potensi masalah dan ketidakefiesienan

dari desain dengan memperlihatkan bagaimana 1 atau lebih objek berinteraksi dalam sebuah kolaborasi secara urut dari inisiasi sampai respon (Larman, 2018).

Simbol yang digunakan pada *Class Diagram* diatas dijelaskan sebagai berikut:

**Tabel 2.4 Penjelasan simbol *Sequence Diagram***

No	Gambar	Nama	Keterangan
1		Actor	Orang atau sistem eksternal yang menggunakan sistem.
2		Object	Objek yang terlibat pada sistem.
3		Lifeline	Menggambarkan masa hidup dari sebuah objek.
4		Activation	Menggambarkan proses durasi aktivasi sebuah operasi.
5		Recursive	Pesan untuk dirinya.
6		Message	Menggambarkan informasi tentang aktivitas yang terjadi antar objek.

7	 $\leftarrow$ 1:1	Return Message	Menggambarkan infomasi yang dikirim kembali ke objek sebelumnya.
---	----------------------	----------------	--

## 2.15 Entity Relationship Diagram (ERD)

*Entity Relationship (ER) Diagram* adalah sebuah representasi grafis dari entitas dan hubungannya satu sama lain. *Entity* merupakan objek atau konsep dunia nyata yang dapat diidentifikasi, *attributes* adalah karakteristik dari sebuah entitas yang dapat digunakan untuk mendeskripsikan atau mengidentifikasinya, sementara *relationship* adalah koneksi atau asosiasi dari dua atau lebih entitas. *Relationship* memiliki 3 tipe, yaitu: *one-to-one* (1:1), relasi ini menunjukkan kemunculan satu entitas dikaitkan dengan kemunculan satu entitas di sisi lain relasi, dan sebaliknya, *one-to-many* (1:N), menunjukkan kemunculan entitas pada sisi *one* diasosiasikan dengan satu atau lebih entitas di sisi relasi *many* namun kemunculan entitas pada sisi *many* diasosiasikan hanya dengan satu kemunculan entitas disisi *one*, *many-to-many* (M:N) menunjukkan setiap kemunculan entitas di satu sisi relasi diasosiasikan dengan satu atau lebih kemunculan entitas di sisi lain. Selain itu terdapat *identifier* berupa *primary key* yang digunakan untuk menunjukkan batasan integritas dari entitas yang menyatakan bahwa tidak ada kemunculan entitas yang memiliki *primary key* yang sama dan *foreign key* merupakan satu atau lebih atribut dari entitas yang digunakan untuk membuat koneksi antara dua entitas (Elmasri & Navathe, 2019, p.4).

## 2.16 Application Development Tools

### 2.16.1 Python

*Python* merupakan salah satu bahasa pemrograman dengan perkembangan pesat dikarenakan mudah digunakan dan serbaguna, hal tersebut membuat pengguna mudah dalam implementasi dan penggunaan di berbagai macam bidang seperti *scientific computing*, analisis data, *artificial intelligence*, dan lainnya. Kebutuhan akan *developer python* juga meningkat, sehingga membuatnya terus berkembang (Srinath, 2017).

### **2.16.2 *HyperText Markup Language (HTML)***

*HyperText Markup Language* atau HTML merupakan bahasa yang digunakan untuk mendesain dan merancang halaman *web*. (Connoly & Begg, 2015). HTML mudah dalam digunakan untuk melakukan kontrol terhadap tampilan *web* berupa teks, gambar, suara, ataupun video dengan menggunakan *tag* penanda pada sebuah dokumen.

### **2.16.3 *Cascading Style Sheets (CSS)***

*Cascading Style Sheets* atau CSS merupakan aturan dalam mengendalikan elemen-elemen dalam sebuah halaman *web* agar lebih terstruktur (Sianipar, 2015). CSS digunakan untuk mengatur tampilan *web* sesuai dengan kebutuhan penggunanya.

### **2.16.4 *Javascript***

Menurut Hejlsberg (2016), *Javascript* merupakan bahasa pemrograman tingkat tinggi, dinamis, dan diinterpretasikan. Bahasa pemrograman ini cocok digunakan dalam pengembangan *website* dan menjadi salah satu teknologi inti dari *World Wide Web*.

## **2.17 *Framework***

### **2.17.1 *Django***

*Django* merupakan web framework pada bahasa pemrograman python dan digunakan untuk mengembangkan aplikasi web dan menyediakan fitur seperti URL routing, database models, dan template rendering (Saputra, 2018).

### **2.17.2 *Bootstrap***

*Bootstrap* merupakan sebuah *front-end framework* yang membantu proses pengembangan aplikasi *web* sehingga menjadi lebih responsif (Rahman, 2014). Dengan penggunaan *bootstrap*, *developer* bisa berfokus pada penulisan *HTML* ataupun *back-end* tanpa memikirkan *CSS* yang rumit.

### **2.17.3 *SB Admin 2***

Menurut website [startbootstrap.com](http://startbootstrap.com), *SB Admin 2* adalah sebuah tema berbasis *bootstrap 4* dan bersifat *open-source* untuk membantu membuat halaman dasbor admin pada sebuah aplikasi berbasis *web*.

## 2.18 Model View Controller (MVC)

*Model View Controller (MVC)* adalah arsitektur perangkat lunak yang memisahkan representasi informasi dari interaksi pengguna dengannya (Fowler, 2015). MVC terbagi menjadi 3 komponen: *model*, yang merepresentasikan data dan *business logic* dari aplikasi; *view*, yang merepresentasikan tampilan data ke pengguna, dan *controller*, yang menjadi penengah dengan mengubah *input* menjadi perintah untuk *model* atau *view*.

## 2.19 Blackbox Testing

*Blackbox testing* merupakan teknik dalam *software testing* dimana bagian struktur internal dan implementasi dari sistem yang diuji tidak diketahui oleh penguji (Beller, 2021). Penguji berinteraksi dengan sistem melalui *user interface* dan mengevaluasi hasil dari sistem berdasarkan hasil yang diharapkan, tanpa mempertimbangkan proses internal atau logika yang mengeluarkan hasil (Jorgensen, 2019). *Blackbox testing* sering digunakan untuk memvalidasi fungsionalitas dan kegunaan dari sistem, dan bisa dilakukan pada beberapa level *testing*, termasuk *unit*, *integration*, dan *acceptance testing*.

## 2.20 8 Golden Rules of Interface Design

Menurut Shneiderman (2020), terdapat *8 Golden Rules of Interface Design* sehingga bisa menghasilkan *user interface* yang efektif harus:

1. *Strive for consistency*: memastikan konsistensi pada desain dan *layout* dari *interface* untuk performa dan kepuasan pengguna.
2. *Cater to Universal Usability*: mengarahkan untuk mengidentifikasi kebutuhan beragam setiap pengguna agar dapat digunakan secara universal.
3. *Offer informative feedback*: memberikan *feedback* yang singkat dan padat mengenai apa yang mereka lakukan.
4. *Design dialogue to yield closure*: memberikan *interface* yang fleksibel dan efisien yang memungkinkan pengguna mencapai tujuan mereka dalam waktu sesingkat mungkin, dengan sedikit usaha.

5. *Offer simple error handling:* merancang *error* untuk mengurangi jumlah kesalahan yang dibuat oleh pengguna dan menyediakan opsi pemulihan yang jelas.
6. *Permit easy reversal of actions:* izinkan pengguna membatalkan atau mengulangi tindakan untuk memberikan rasa kontrol.
7. *Support internal locus of control:* memberikan pengguna kemungkinan untuk mereka menyesuaikan dan mempersonalisasi *interface* sesuai kebutuhan dan preferensi mereka.
8. *Reduce short-term memory load:* meminimalkan jumlah informasi yang perlu diingat pengguna dengan memberikan isyarat yang terlihat, instruksi yang jelas, dan informasi yang relevan.

Prinsip-prinsip tersebut, pertama kali dikenalkan oleh Ben Shneiderman pada tahun 1980-an, dan telah digunakan secara luas sebagai panduan dalam desain *user interface* yang *user-friendly* dan efektif (Kolko, 2018).

## 2.21 Gantt Chart

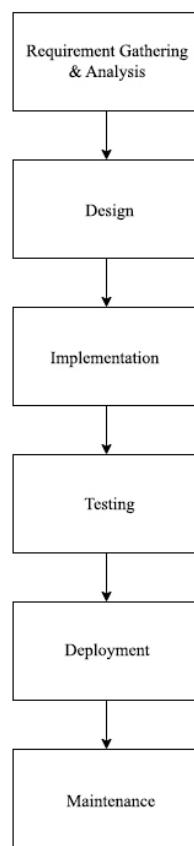
*Gantt Chart* digunakan pada projek manajemen untuk mengvisualisasikan urutan jadwal atau *timeline* dari projek dan melacak progress secara berkala. Selain itu, *Gantt Chart* juga memberi gambaran yang jelas dari tugas dari projek dan *deadline* sehingga bisa membuat *project manager* untuk mengidentifikasi potensi *delay*. Namun, *Gantt Chart* bisa menjadi kompleks dan sulit untuk dijaga ketika digunakan untuk projek besar ataupun beberapa *stakeholder* (Mamun, Rahman, & Islam, 2021).

## **BAB 3**

### **METODE PENELITIAN**

#### **3.1 Kerangka Berpikir**

Dalam proses pengembangan aplikasi yang akan dibuat oleh penulis untuk skripsi ini, penulis menggunakan metode SDLC *Waterfall* sehingga dapat mengatur alur pengembangan secara terstruktur. Dengan menggunakan SDLC *Waterfall*, ruang lingkup dari aplikasi yang akan dicakup tidak terlalu besar, sehingga dari kebutuhan aplikasi tidak akan banyak berubah saat memasuki tahap pengembangan.



**Gambar 3.1 Metode SDLC Waterfall**

Pada metode SDLC *Waterfall* terdapat 6 tahapan yang digunakan penulis sebagai panduan dalam mengembangkan aplikasi ini:

1. *Requirement Gathering and Analysis*, merupakan tahapan pertama dalam pengembangan aplikasi yaitu dengan mengumpulkan dan mengidentifikasi kebutuhan dari aplikasi ini. Serta mengumpulkan informasi yang berkaitan dengan kebutuhan *user* sehingga dalam pengembangan dapat dengan tepat memenuhi apa yang diinginkan oleh *user*.

2. *Design*, merupakan tahapan selanjutnya setelah mengumpulkan informasi yang dapat digunakan dalam mengembangkan rancangan dari aplikasi. Pada tahap ini dengan melakukan perancangan desain dari aplikasi termasuk arsitektur, *user interface*, dan desain *database*.
3. *Implementation*, merupakan tahapan dimana mengacu pada pengimplementasian dari rancangan aplikasi yang sudah dibuat pada tahap sebelumnya. Pada tahap ini *developer* akan mengembangkan aplikasi dari sisi penulisan kode.
4. *Testing*, pada tahap ini akan ada pengujian hasil pengembangan aplikasi yang telah dikembangkan pada tahap sebelumnya. Pengujian disini mencakup apakah ada kendala atau *error* saat menjalankan aplikasi yang telah dikembangkan dan juga apakah fungsionalitas dari aplikasi ini sudah tercapai.
5. *Deployment*, pada tahapan ini setelah melakukan pengujian terhadap aplikasi yang telah dikembangkan dan juga memastikan aplikasi sudah berjalan dengan baik, maka langkah selanjutnya adalah mendeploy aplikasi ke bagian *production* sehingga bisa digunakan oleh *user*.
6. *Maintenance*, merupakan tahapan dimana adanya proses pengoperasian aplikasi oleh *user*. Aplikasi yang sudah dikembangkan dilakukan pemeliharaan mulai dari menambahkan fitur, melakukan pembaharuan modul, serta memperbaiki aplikasi saat terdapat *error* atau malfungsi saat aplikasi dijalankan.

### 3.2 Analisis Masalah

Dalam proses analisis masalah saat pengembangan aplikasi ini, penulis menggunakan aplikasi sejenis sebagai pembanding dari aplikasi yang penulis kembangkan. Penulis juga melakukan wawancara kepada Senior *Penetration Tester*, *Junior Penetration Tester*, dan *Client* dimana hasil wawancara digunakan sebagai pengumpulan data terkait ketertarikan *user* untuk menggunakan aplikasi ini sebagai *vulnerability management*.

#### 3.2.1 Permasalahan dan Kebutuhan

Pada aplikasi *vulnerability management* harus memiliki dasar pengujian yang mengacu kepada hasil *penetration testing* yang dilakukan oleh *penetration*

*tester* sehingga saat aplikasi digunakan dapat mencapai *goal* utama yaitu sebagai automasi *report penetration testing*, membantu kolaborasi yang dilakukan antar *penetration tester* serta klien, dan membantu proses *transfer knowledge* antar *penetration tester*. Dengan adanya aplikasi ini, *penetration tester* dapat lebih berfokus pada aspek teknikal proses *penetration testing*. Cakupan dari fungsi-fungsi yang akan dibuat pada aplikasi rancangan harus mencangkup klasifikasi *project*, *generate report* berdasarkan hasil *penetration testing*, dan *list testing guide*. Fitur tersebut menjadi prioritas bagi penulis, karena selain dari rancangan awal yang sudah dibuat, beberapa fitur juga ditambahkan berdasarkan hasil wawancara yang dilakukan oleh penulis dalam mengumpulkan data. Cakupan fungsi yang dipilih juga dipersempit dengan tujuan untuk membangun aplikasi yang mudah dipahami dan mudah digunakan.

### 3.2.2 Analisis Aplikasi Sejenis

Dalam membuat *report penetration testing*, perusahaan biasanya memakai *Google Docs* sebagai platform yang bisa digunakan namun penggunaannya kurang efektif karena *file* akan tersimpan pada akun pribadi, selain itu terdapat platform lain yang digunakan bernama *Confluence*, yang merupakan sebuah platform yang digunakan dalam kolaborasi dan *transfer knowledge* dalam sebuah tim secara umum. (*Confluence Atlassian*).

Meskipun demikian, aplikasi pembanding bagi aplikasi *vulnerability management* yang tim penulis kembangkan adalah *Confluence*. Penulis merasa bahwa *Confluence* memiliki fitur yang sesuai dengan penulis rencanakan untuk dibuat pada aplikasi *vulnerability management* sebagai alat automasi *report*, *vulnerability management*, dan *transfer knowledge* bagi *penetration tester*.

*Confluence* merupakan aplikasi kolaborasi antar individu dalam satu perusahaan, konteks dari kolaborasi ini adalah secara umum sehingga tidak berfokus pada *vulnerability management*. Sesuai dengan demo dan dokumentasi dari aplikasi *Confluence*, penulis menemukan beberapa fitur yang menjadi fokus penulis dalam membuat aplikasi *vulnerability management* ini yaitu kolaborasi antar individu, pembuatan dokumentasi secara terstruktur, automasi pembuatan suatu *report* dari satu *pages* ke format PDF. Diluar itu masih banyak fitur yang dimiliki oleh aplikasi *Confluence* yang tidak menjadi fokus penulis, seperti

integrasi ke *tools* lainnya baik *slack*, *jira*, *figma* dan lain sebagainya. Dengan segala kelebihan yang dimiliki oleh aplikasi *Confluence*, aplikasi *Confluence* tidak berfokus pada *vulnerability management* sehingga menyulitkan *user* ketika ingin menjadikan aplikasi *Confluence* sebagai *vulnerability management*, terlebih lagi dalam menghitung *severity* dan menentukan *weakness* pada suatu *vulnerability*. Aplikasi *Confluence* juga sangat kompleks sehingga menyulitkan untuk pengguna baru memahami fitur-fitur yang tersedia. yang jarang menggunakan aplikasi kolaborasi

### **3.2.3 Analisis Berdasarkan Wawancara**

Penulis menggunakan wawancara dalam mengetahui *user requirements* dari narasumber sehingga dapat menjadikan informasi hasil wawancara dalam mengembangkan aplikasi yang dibuat oleh penulis. Wawancara dilakukan kepada 3 profesi yang terkait dengan aplikasi penulis, yaitu senior *penetration tester* sebanyak 2 orang, junior *penetration tester* sebanyak 3 orang, dan klien sebanyak 1 orang sehingga segmentasi *user* jelas karena aplikasi automasi ini dikhusruskan pada kegiatan *penetration testing*.

#### **3.2.3.1 Junior Penetration Tester**

Pertanyaan pertama yang diajukan penulis adalah bagaimana tahapan saat mengerjakan suatu *penetration testing* untuk mengetahui proses yang dilakukan oleh narasumber dan mengetahui apakah pengembangan aplikasi automasi ini diperlukan dalam mempercepat proses *penetration testing*. Narasumber menjawab jika proses dimulai dari RFP (*Request For Proposal*), kemudian menentukan *scoping* fitur yang akan dites dan *consultant* akan mulai identifikasi, eksloitasi, membuat *report*, dan *regression testing*.

Pertanyaan kedua yang diajukan penulis untuk mengetahui pendapat narasumber adalah apakah ada pekerjaan yang berulang dalam *penetration testing* dan merasa perlu ada *tools* sehingga nantinya aplikasi yang dikembangkan bisa menjadi solusi dari hal tersebut. Dua narasumber menjawab dengan adanya pekerjaan berulang pada sisi teknikal atau *static analysis* dan pada saat *reporting*.

Pertanyaan ketiga yang diajukan penulis untuk mengetahui apa kendala yang dihadapi dalam melakukan *penetration testing* sebagai *penetration tester* junior agar penulis dapat melakukan *research* fitur apa yang bisa ditambahkan dalam pengembangan aplikasi. Narasumber pertama menjawab masalah saat mempelajari *guide* yang ada dan saat penulisan *report*, sementara narasumber kedua menjawab sering lupa *checklist* apa yang akan dites dan penulisan *report* yang masih manual.

Pertanyaan keempat yang diajukan penulis untuk mengetahui bagaimana cara yang dilakukan dalam kolaborasi antar *penetration tester* dan pertanyaan ini untuk mengetahui platform yang digunakan oleh narasumber dalam kolaborasi suatu *project*. Kedua narasumber menjawab jika platform yang mereka gunakan adalah *Microsoft Teams*, satu narasumber menjawab biasanya kolaborasi yang dilakukan adalah pembagian tugas fitur mana yang akan dilakukan *penetration testing*, narasumber lain menjawab jika menemukan *vulnerability SQLI* pada fitur A, *payload* yang digunakan akan dibagi ke *penetration tester* lain untuk dicoba pada fitur lain.

Pertanyaan kelima yang diajukan penulis untuk mengetahui kendala dalam melakukan kolaborasi antar *penetration tester* sehingga saat pengembangan aplikasi ini bisa menjadi solusi yang dapat digunakan sebagai jembatan antar *penetration tester*. Jawaban dari kedua narasumber hampir sama yaitu tidak mengetahui apa yang sedang dilakukan oleh *penetration tester* lain, apakah sedang tidak di depan komputer atau mengerjakan bagian yang sama sehingga waktu yang ada menjadi sia-sia.

Pertanyaan keenam yang diajukan oleh penulis adalah mengetahui apa kegiatan yang dilakukan saat berhubungan dengan klien sehingga pengembangan aplikasi ini bisa membantu menjembatani antara *penetration tester* dengan klien. Kedua narasumber menjawab pada saat setelah selesai *reporting*, biasanya ada tahap *retest* yang langsung komunikasi ke klien misal kalau ini masih *vuln* atau cara *fix*-nya sampai status *vulnerability*nya *close*.

Pertanyaan ketujuh yang diajukan oleh penulis adalah apakah kendala yang dihadapi saat berhubungan dengan klien yang diketahui. Kedua narasumber menjawab jika kendala yang dihadapi seringkali pada bagian

*reporting* yang dirasa kurang, pemakaian kata salah, bahasa yang digunakan terlalu teknikal, serta seringkali klien menolak *finding* atau dianggap tidak *vuln*.

### 3.2.3.2 Senior *Penetration Tester*

Pertanyaan pertama yang diajukan penulis adalah bagaimana tahapan saat mengerjakan suatu *penetration testing* untuk mengetahui proses yang dilakukan oleh narasumber dan mengetahui apakah pengembangan aplikasi automasi ini diperlukan dalam mempercepat proses *penetration testing*. Kedua narasumber menjawab dengan proses dimulai dengan *prerequisite meeting* dengan klien untuk membahas *service* yang vendor tawarkan dan ekspektasi dari klien. Beberapa perusahaan vendor kemudian dipilih melalui proses *tender*, jika perusahaan menang *tender* kemudian masuk ke fase *scoping* untuk menjelaskan lebih detail mengenai *service* yang akan dilakukan *penetration testing*, setelah itu akan diadakan *kick-off meeting* dengan *developer* untuk membahas detail *penetration testing*. Setelah itu baru dimulai proses *penetration testing* sampai selesai dan *meeting* lagi untuk presentasi *finding* dan sebagai *consultant* menjelaskan *best practice* yang bisa dilakukan dilanjutkan dengan proses *retest* sampai semua status *finding* menjadi *close*. Setelah semua proses selesai, nanti dibuat *final report*.

Pertanyaan kedua yang diajukan penulis untuk mengetahui pendapat narasumber adalah apakah ada pekerjaan yang berulang dalam *penetration testing* dan merasa perlu ada *tools* sehingga nantinya aplikasi yang dikembangkan bisa menjadi solusi dari hal tersebut. Narasumber pertama menjawab dari sisi teknikal dengan memberi contoh tidak mungkin melakukan pengecekan *injection* dengan *payload* di 100 *parameter* secara manual dan dari sisi manajemen mengenai *follow-up project* yang sering terjadi *miskom* karena dari sisi klien berkomunikasi melalui *project manager* mereka dan dari sisi vendor juga melalui *project manager* vendor. Sementara narasumber kedua menjawab pada saat proses *reporting*, seringkali melakukan penulisan berulang mengenai rekomendasi, *script* yang digunakan, dan risk yang ada.

Pertanyaan ketiga yang diajukan penulis untuk mengetahui apa kendala yang dihadapi dalam melakukan *penetration testing* sebagai *penetration tester* senior agar penulis dapat melakukan *research* fitur apa yang bisa ditambahkan dalam pengembangan aplikasi. Narasumber pertama menjawab mengenai masalah pribadi yaitu *time management* sebagai *team leader* dan senior *penetration tester consultant* yang mempunyai tanggung jawab menyelesaikan *project* yang ada namun disisi lain harus memimpin tim. Sementara dari sisi teknis, banyak perusahaan *cyber security* menjunjung tinggi dan mendukung *cyber security* untuk dikembangkan namun kebanyakan memilih untuk *quick fix* daripada mengatasi masalah dasar keamanannya serta kurangnya *cyber security awareness* dan *responsibility* dari sisi manajemen dan *developer* yang berakibat kurangnya kesadaran akan resiko dari aplikasi yang mereka buat. Narasumber kedua menjawab pada saat penentuan *risk* dari sisi bisnis dan teknikal serta dari sisi fundamental *penetration tester*, sementara dari sisi manajemen tidak ada karena perusahaan narasumber telah menggunakan *technical writer* untuk penulisan *report*.

Pertanyaan keempat yang diajukan penulis untuk mengetahui pendapat narasumber mengenai kendala yang dihadapi *penetration tester* junior dari sisi *penetration tester* senior. Narasumber pertama menjawab pada saat transisi dari mahasiswa menjadi pekerja dimana harus membuktikan diri sendiri dan kurangnya rasa ingin untuk mengamankan aplikasi klien. Selain itu juga susahnya menjelaskan *finding* yang tidak ada di OWASP, manajemen waktu pada saat kuliah, organisasi, dan bekerja, serta lebih mudah menyerah dan gengsi untuk bertanya. Sementara narasumber kedua menjawab mengenai masalah teknikal membaca *logic business*, memberi rekomendasi, dan adu argumen dengan klien.

Pertanyaan kelima yang diajukan penulis untuk mengetahui bagaimana cara kolaborasi yang dilakukan dengan *penetration tester* lain di sebuah *project*. Narasumber pertama menjawab jika dia yang mengecek mengenai *finding* aneh dari sisi *business logic*, sementara junior *penetration tester* diberi tugas untuk mengecek *checklist OWASP WSTG* atau *MSTG* dan membicarakan apa yang ditemukan dan masalah yang ada, platform yang

digunakan adalah *Confluence*. Narasumber kedua memberi jawaban yang hampir sama, jika konsultan dengan konsultan akan mengecek aplikasi yang akan dilakukan *penetration testing* secara keseluruhan, sementara junior diberi tugas tertentu. Untuk platform tidak tahu apa boleh disebutkan, tapi fungsi platform ini untuk *tracking* dan *sharing* informasi mengenai *project penetration testing* yang berisi *proof of concept*, deskripsi, *threat risk*, rekomendasi dan list *retest* yang bisa dilihat *penetration tester* lain.

Pertanyaan keenam yang diajukan penulis mengenai kendala yang dihadapi saat melakukan kolaborasi dengan *penetration tester* lain. Kedua narasumber menjawab masalah mereka ada di bagian komunikasi. Narasumber pertama menjelaskan masalah komunikasi sering terjadi karena beberapa sebab dan bisa diselesaikan dengan platform seperti *excel* yang berisi data *penetration testing* agar tidak bentrok. Sementara narasumber kedua menjelaskan masalah komunikasi dimana platform yang sekarang digunakan tidak mempunyai fitur yang menampilkan bagian mana saja yang telah di cek dan sering ada yang terlewat saat proses *penetration testing*.

Pertanyaan ketujuh yang diajukan oleh penulis adalah kegiatan yang dilakukan saat berhubungan dengan klien. Kedua narasumber memberi jawaban yang sama yaitu pada saat selesai proses *penetration testing*, akan diadakan *meeting* untuk presentasi atau pemaparan mengenai *vulnerability* yang ditemukan, *risk* sejauh apa, rekomendasinya bagaimana. Narasumber kedua menambahkan jika pada saat proses *penetration testing* terjadi *error*, maka klien akan dihubungi untuk melakukan perbaikan.

Pertanyaan kedelapan yang diajukan oleh penulis adalah kendala saat berhubungan dengan klien yang diketahui. Narasumber pertama menjawab jika klien bebal, kurang *awareness*, dan ada klien yang tidak suka diberi rekomendasi konsep dan lebih meminta langsung *coding*-nya. Sementara narasumber kedua hampir sama yaitu klien yang bebal dan menolak *finding* sehingga harus dijelaskan *risk* dan *impact*-nya pada skenario asli sehingga butuh pengalaman.

### 3.2.3.3 Client

Pertanyaan pertama yang diajukan oleh penulis adalah bagaimana tahapan saat menggunakan jasa *penetration testing* pada produk. Narasumber menjawab proses dimulai dengan *open tender*, beberapa *vendor* diundang untuk melakukan *scoping* dari *project* dan *vendor* memberikan harga dari jasa mereka. Kami dari klien kemudian memilih *vendor* yang sesuai, setelah itu akan dilakukan diskusi lebih lanjut mengenai *scope*, waktu, siapa yang terlibat, sampai jadi *reportnya*. Kemudian menentukan metode *penetration testing* yang akan dilakukan, jika *white-box* maka klien akan memberi *resource* seperti VPN atau *credentials*. Untuk melihat *progress*, klien diberi akses *spreadsheet* yang memperlihatkan *finding* termasuk nomor, nama *finding*, dan *severitynya*. Setelah proses selesai akan dikirimkan *report*-nya.

Pertanyaan kedua yang diajukan oleh penulis adalah apakah ada kendala dari sisi klien ke *penetration tester* pada saat proses *penetration testing*. Narasumber menjawab masalah mengenai *VPS* atau pertanyaan mengenai fitur, kalau dari sisi *finding*, terkadang informasi yang diberikan kurang karena hanya nomor, nama *finding*, dan *severity* karena dari sisi klien tidak paham *impact* atau rekomendasinya.

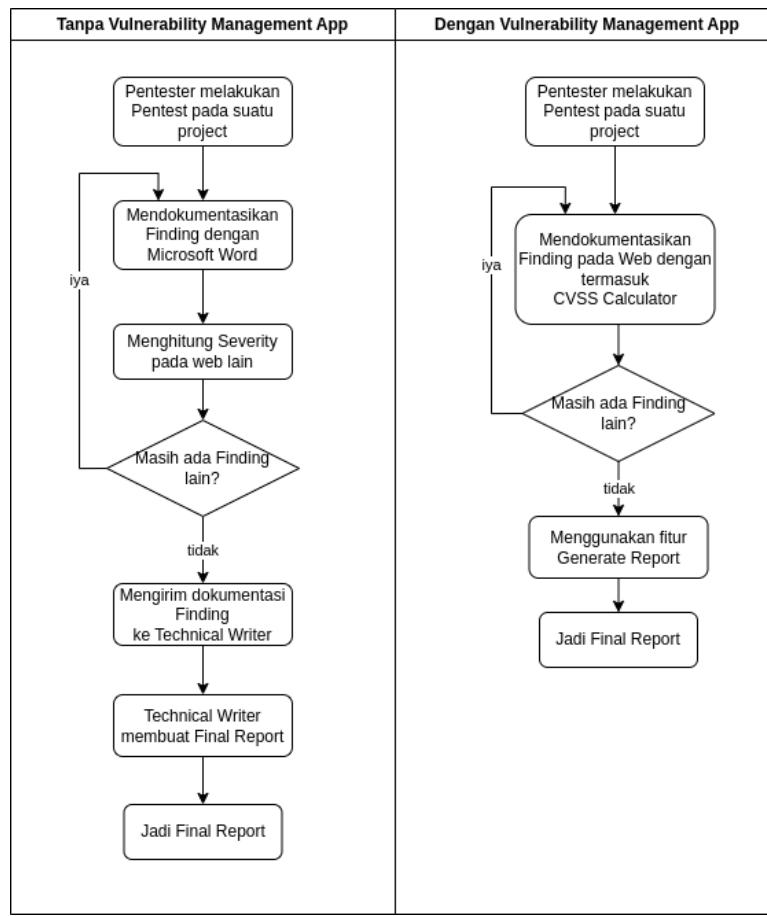
Pertanyaan ketiga yang diajukan oleh penulis adalah bagaimana cara melakukan diskusi dengan *penetration tester* selama proses *penetration testing*. Narasumber menjawab dengan 3 platform yaitu *whatsapp*, *email*, dan *telegram*.

Pertanyaan keempat yang diajukan oleh penulis adalah apakah ada hal yang ingin dipermudah dalam kolaborasi dengan *penetration tester*. Narasumber menjawab mengenai masalah *finding* yang kurang jelas.

Pertanyaan kelima yang diajukan oleh penulis adalah apakah ada informasi yang dirasa kurang dari *penetration tester*. Narasumber menjawab pada saat *final report*, terkadang masih ada *finding* yang kurang jelas seperti bagaimana cara mendapatkannya atau mengeksplorasinya.

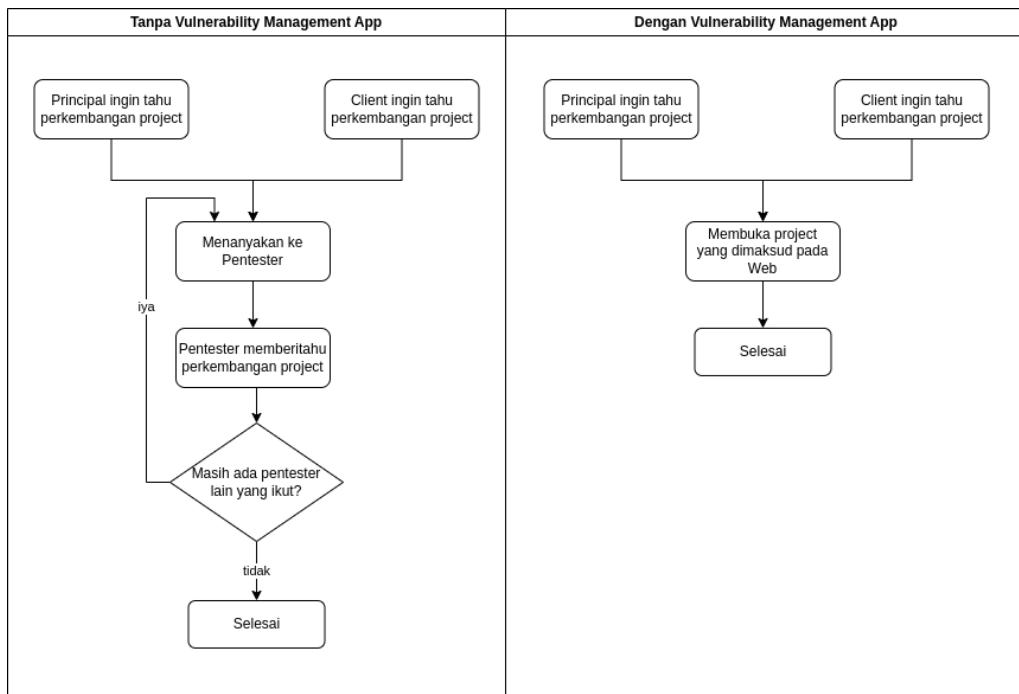
### 3.2.4 Usulan Pemecahan Masalah

Dari permasalahan dan kebutuhan yang telah terkumpul, tim penulis merancang solusi dari permasalahan yang ada. Untuk permasalahan mengenai kolaborasi dalam sebuah *project*, penulis mengusulkan untuk membuat sebuah aplikasi yang menjadi penghubung antara *penetration tester* yang terlibat dalam sebuah *project* dengan klien yang aplikasinya ingin dilakukan *penetration testing*. Untuk masalah automasi *report*, penulis membuat masing-masing *project* yang ada berisi hasil *finding* berdasarkan *penetration test*, *penetration tester* dan klien yang terlibat, objektif, dan WSTG atau MSTG *checklist* sehingga klien bisa melihat *progress* dari *penetration test* yang dilakukan. Untuk permasalahan *transfer knowledge* antar *penetration tester*, penulis membuat *resources* yang berisi *testing guide* berdasarkan WSTG maupun MSTG ataupun *testing guide* lain yang ingin ditambahkan oleh *penetration tester* sehingga *penetration tester* lain bisa melihat dan membaca *guide* yang bisa membantu dalam melakukan *penetration test*. Berikut perbandingan solusi dengan tidak menggunakan aplikasi dan dengan menggunakan aplikasi yang dibuat.



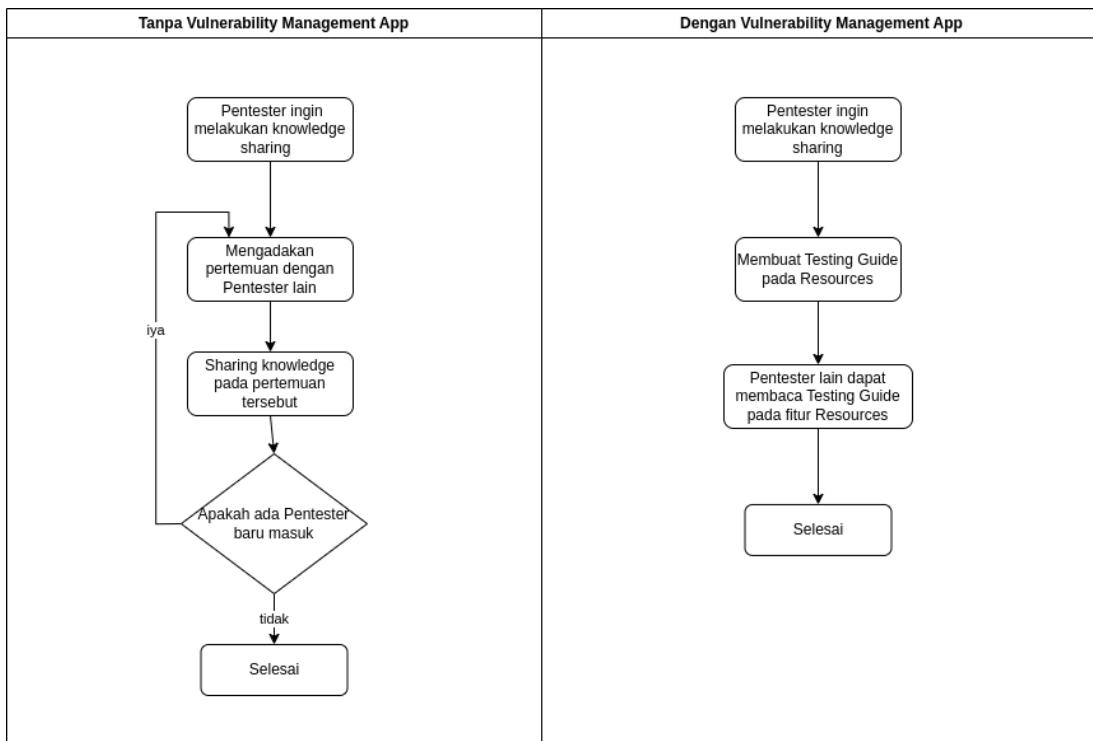
**Gambar 3.2 Perbandingan alur *penetration testing* tanpa dan dengan aplikasi *Vulnerability Management***

Gambar diatas menunjukkan aplikasi kami yang melakukan efisiensi pada proses *penetration testing* dari mulai pengujian sampai tahap akhir yaitu *final report*. Jika tanpa aplikasi *vulnerability management* terdapat sedikitnya tujuh tahap untuk jadi *final report*, sedangkan menggunakan aplikasi *vulnerability management* kami melakukan efisiensi pada tahap menghitung *severity* pada web lain dan mengganti pembuatan *final report* menjadi automasi menggunakan fitur *generate report*. Sehingga aplikasi *vulnerability management* kami bisa melakukannya dengan sedikitnya lima tahap saja.



**Gambar 3.3 Perbandingan kolaborasi tanpa dan dengan aplikasi *Vulnerability Management***

Gambar diatas menunjukkan aplikasi kami yang melakukan efisiensi pada kolaborasi antar *principal*, *penetration tester* dan *client*. Jika tanpa aplikasi *vulnerability management* terdapat sedikitnya lima tahap hingga selesai, sedangkan menggunakan aplikasi *vulnerability management* kami melakukan efisiensi dengan hanya tiga tahap saja. *Principal*, *penetration tester* dan *client* hanya perlu membuka *project* yang dimaksud pada web untuk melihat perkembangan dari *project*.



**Gambar 3.4 Perbandingan *transfer knowledge* tanpa dan dengan aplikasi *Vulnerability Management***

Gambar diatas menunjukkan aplikasi kami yang melakukan efisiensi pada *transfer knowledge* antar *penetration tester*. Jika tanpa aplikasi *vulnerability management* terdapat sedikitnya lima tahap hingga selesai, sedangkan menggunakan aplikasi *vulnerability management* kami melakukan efisiensi dengan hanya empat tahap saja. Para *penetration tester* hanya perlu membuat dan membuka *testing guide* pada fitur *resources* untuk melakukan *knowledge sharing*.

Fungsi-fungsi utama pada aplikasi dapat dijelaskan sebagai berikut:

#### **3.2.4.1 Project**

*Project* adalah satu lingkup kerja dalam *project penetration testing*. Para *penetration tester* menguji keamanan dari aplikasi yang diberikan oleh klien. Pada *project*, *penetration tester* dapat berkolaborasi untuk mengerjakan sebuah *project* dan klien dapat melihat perkembangan yang terjadi dari *project* tersebut. *Project* memiliki 3 status yaitu *To Do*, *On Going*, dan *Done*. Fungsi untuk membuat, mengedit, dan menghapus *project* hanya bisa dilakukan oleh principal untuk menjaga agar tetap terstruktur, sementara untuk melihat dan mencari *project* bisa dilakukan semua *user*. Di dalam setiap

*project* berisi detail mengenai *project* tersebut, seperti: tipe *pentest*, *deadline*, objek *pentest*, *penetration tester* dan *client* yang terlibat, WSTG atau MSTG *checklist*, dan daftar *finding* dari *project* tersebut.

#### 3.2.4.2 *Finding*

*Finding* merupakan kerentanan yang ditemukan pada suatu *object pentest* dalam satu *project*. *Finding* dibuat sebagai pembuktian dengan tujuan mempermudah klien dalam mereproduksi ulang *finding* yang ditemukan dan bagaimana cara remediasinya. Didalam setiap *finding* berisi detail mengenai *finding* tersebut, seperti: *finder*, di objek mana *finding* ditemukan, nama *finding*, CWE, severity CVSS, description, impact, evidence, remediation, dan references. *Finding* dibagi menjadi 2 status yaitu *Fixed* dan *Not Fixed*.

#### 3.2.4.3 *Resources*

Fungsi *Resources* merupakan fungsi yang ditujukan untuk permasalahan *transfer knowledge* dan hanya bisa diakses oleh *penetration tester*. Fungsi ini berisi tutorial atau cara dalam melakukan *penetration test* yang dapat membantu bagi junior *penetration tester*. *Guide* yang ada pada fungsi ini diambil dari OWASP WSTG dan OWASP MSTG sebagai sumber utama sehingga terjamin kualitasnya.

#### 3.2.5 *Testing Methodology*

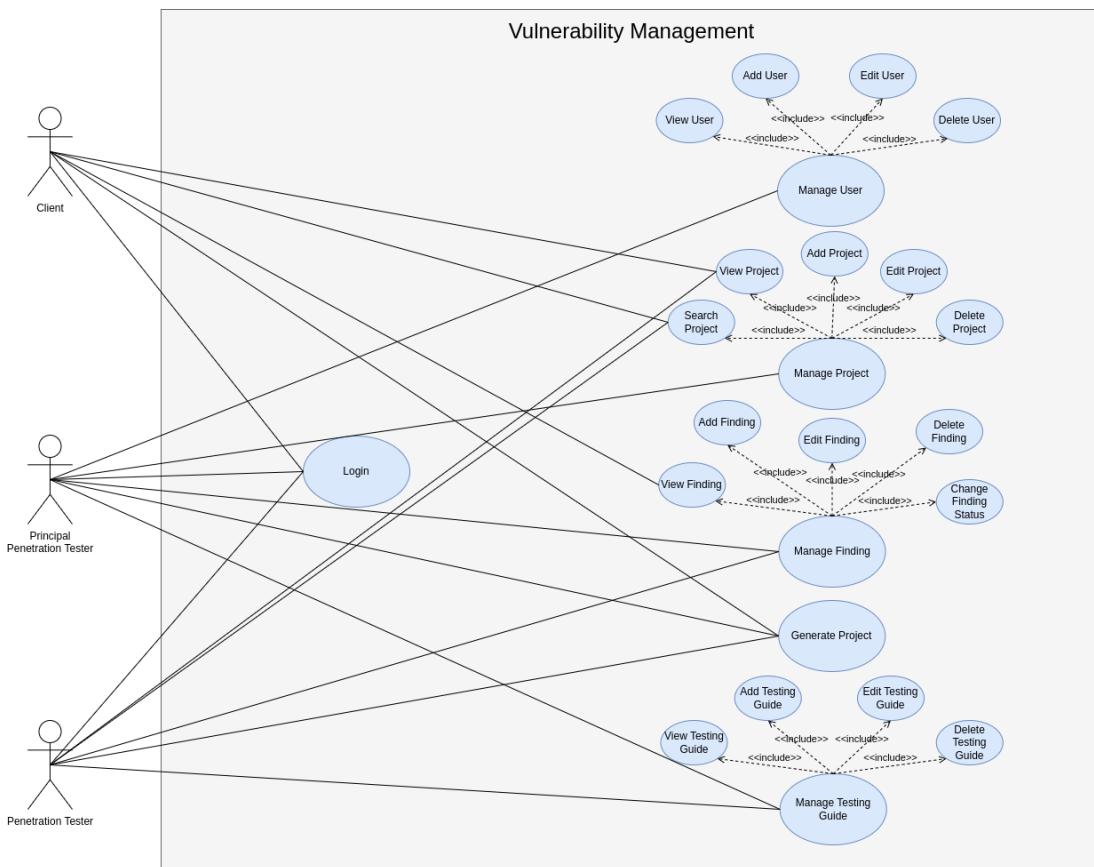
Penulis akan melakukan pengujian terhadap solusi yang dikembangkan yaitu aplikasi *vulnerability management* dengan pengujian keseluruhan fungsionalitas secara *Black Box Testing* dan evaluasi berdasarkan 8 *Golden Rules of Interface Design*. Dengan pengujian ini, penulis dapat memastikan bahwa aplikasi berjalan dengan baik dan menjamin kualitas aplikasi sebelum memasuki tahapan *deployment*.

### 3.3 Perancangan Sistem

Aplikasi ini adalah sebuah aplikasi *vulnerability management* yang bertujuan untuk mempermudah kolaborasi antar *penetration tester* dan juga berfungsi untuk automasi report hasil *penetration testing* dan *transfer knowledge* antar *penetration*

*tester*. Pada aplikasi ini terdapat fitur-fitur inti yang akan dikembangkan dan dijabarkan pada sub bab selanjutnya.

### 3.3.1 Use Case Diagram



**Gambar 3.5 Use Case Diagram**

Berikut adalah penjelasan dari *use case diagram* diatas yang telah dibuat oleh penulis:

**Tabel 3.1 Penjabaran Aktor pada Use Case Diagram**

No.	Aktor	Keterangan
1.	<i>Client</i>	<i>Client</i> merupakan individu dari perusahaan yang produknya dilakukan <i>penetration testing</i> sebagai aktor yang menggunakan fungsi-fungsi dari aplikasi seperti <i>View Project</i> , <i>View Finding</i> , dan <i>Generate Report</i> .
2.	Principal atau Penetration Tester	Principal merupakan seorang senior

	Senior	<i>penetration tester</i> pada perusahaan vendor sebagai aktor yang menggunakan fungsi-fungsi dari aplikasi seperti <i>Manage User</i> , <i>Manage Project</i> , <i>Manage Finding</i> , <i>Generate Report</i> , dan <i>Manage Resources</i> .
3.	Pentester atau Junior	<i>Pentester</i> merupakan seorang junior <i>penetration tester</i> pada perusahaan vendor sebagai aktor yang menggunakan fungsi-fungsi dari aplikasi seperti <i>View Project</i> , <i>Manage Finding</i> , <i>Generate Report</i> , dan <i>Manage Resources</i> .

**Tabel 3.2 Penjabaran *Use Case* pada *Use Case Diagram***

No.	<i>Use Case</i>	Keterangan
1.	<i>Login</i>	Menampilkan form <i>username</i> dan <i>password</i> sebagai autentikasi masuk ke aplikasi.
2.	<i>Manage User</i>	Mengelola <i>user</i> yang terdaftar pada aplikasi dengan role masing-masing.
3.	<i>Add User</i>	Menambahkan <i>user</i> baru pada aplikasi.
4.	<i>Edit User</i>	Mengganti informasi mengenai <i>user</i> tertentu pada aplikasi.
5.	<i>Delete User</i>	Menghapus data <i>user</i> dari aplikasi.
6.	<i>View User</i>	Menampilkan daftar <i>user</i> yang terdaftar pada aplikasi.
7.	<i>Manage Project</i>	Mengelola <i>project</i> yang akan dimulai, sedang dikerjakan, dan sudah selesai.

8.	<i>Search Project</i>	Mencari dan menampilkan <i>project</i> menggunakan kata kunci tertentu.
9.	<i>Add Project</i>	Menambahkan <i>project</i> baru pada aplikasi.
10.	<i>Edit Project</i>	Mengganti data <i>project</i> tertentu pada aplikasi.
11.	<i>Delete Project</i>	Menghapus data <i>project</i> dari aplikasi.
12.	<i>View Project</i>	Menampilkan daftar <i>project</i> yang akan dimulai, sedang dikerjakan, dan sudah selesai untuk <i>user</i> tersebut.
13.	<i>Manage Finding</i>	Mengelola <i>finding</i> yang ditemukan pada saat proses <i>penetration test</i> .
14.	<i>Add Finding</i>	Menambahkan <i>finding</i> baru yang ditemukan pada proses <i>penetration test</i> .
15.	<i>Edit Finding</i>	Mengganti detail <i>finding</i> pada sebuah <i>project</i> .
16.	<i>Delete Finding</i>	Menghapus <i>finding</i> dari sebuah <i>project</i> .
17.	<i>View Finding</i>	Menampilkan daftar <i>finding</i> yang ditemukan pada saat proses <i>penetration test</i> sebuah <i>project</i> .
18.	<i>Change Finding Status</i>	Mengganti <i>status finding</i> menjadi <i>Fixed</i> atau <i>Not Fixed</i> .
19.	<i>Generate Report</i>	Mengekspor atau menghasilkan file berekstensi pdf berdasarkan sebuah <i>project</i> .
20.	<i>Manage Testing Guide</i>	Menampilkan <i>guide</i> untuk <i>penetration testing</i> berdasarkan OWASP MSTG dan WSTG serta daftar <i>report</i> dari <i>project</i> yang

		sudah dikumpulkan.
21.	<i>Add Testing Guide</i>	Menambahkan <i>testing guide</i> baru pada <i>resources</i> .
22.	<i>Edit Testing Guide</i>	Mengganti data isi <i>testing guide</i> pada <i>resources</i> .
23.	<i>Delete Testing Guide</i>	Menghapus data <i>testing guide</i> pada <i>resources</i> .
24.	<i>View Testing Guide</i>	Menampilkan daftar dan isi dari <i>testing guide</i> pada <i>resources</i> .

### 3.3.2 Use Case Description

Dari *use case diagram* yang telah dibuat oleh penulis, penulis juga membuat *use case description* untuk menjelaskan lebih rinci terkait proses apa saja yang terjadi yaitu:

**Tabel 3.3 Use Case Description: Login**

Use case name:	<i>Login</i>	
Actor:	<i>Principal, Pentester, Client</i>	
Trigger:	<i>Actor</i> ingin masuk ke aplikasi melalui autentikasi dengan <i>username</i> dan <i>password</i> .	
Flow of activities:	Actor:	System:
	<ol style="list-style-type: none"> <li>1. <i>Actor</i> masuk ke halaman <i>login</i>.</li> <li>2. <i>Actor</i> memasukkan <i>username</i> dan <i>password</i>.</li> <li>3. <i>Actor</i> menekan tombol <i>login</i>.</li> </ol>	<ol style="list-style-type: none"> <li>1.1 Sistem menampilkan halaman <i>login</i>.</li> <li>2.1 Sistem menerima <i>input</i> <i>username</i> dan <i>password</i>.</li> <li>3.1 Sistem mengecek dan meng-autentikasi <i>input user</i>.</li> </ol>

		4.1 Jika ter-autentikasi, sistem akan melakukan <i>redirect</i> ke dalam aplikasi, jika tidak, sistem memunculkan <i>alert</i> <i>username</i> atau <i>password</i> salah.
--	--	--

**Tabel 3.4 Use Case Description: Manage User**

Use case name:	<i>Manage User</i>	
Actor:	Principal	
Trigger:	<i>Actor</i> ingin mengelola <i>user</i> yang terdaftar pada aplikasi.	
Flow of activities:	Actor:	System:
	1. <i>Actor</i> masuk ke menu <i>Admin Panel</i> dan memilih opsi <i>Users Management</i> . 2. <i>Actor</i> memilih aksi yang ingin dilakukan pada halaman <i>Users Management</i> ,	1.1 Sistem menampilkan halaman <i>Users Management</i> berisi daftar <i>user</i> yang terdaftar, 2.1 Sistem memproses aksi yang dipilih.

**Tabel 3.5 Use Case Description: Add User**

Use case name:	<i>Add User</i>	
Actor:	Principal	
Trigger:	<i>Actor</i> ingin menambahkan <i>user</i> baru pada aplikasi.	

Flow of activities:	Actor:	System:
	<p>1. <i>Actor</i> masuk ke menu <i>Admin Panel</i> dan memilih opsi <i>Users Management.</i></p> <p>2. <i>Actor</i> menekan tombol <i>Add New User</i>.</p> <p>3. <i>Actor</i> mengisi data mengenai <i>user</i> pada form dan menekan tombol <i>Add User</i> jika telah selesai.</p>	<p>1.1 Sistem menampilkan halaman <i>Users Management.</i></p> <p>2.1 Sistem menampilkan halaman <i>Add New User</i> berisi <i>form.</i></p> <p>3.1 Data yang terisi akan disimpan ke dalam <i>database.</i></p>

**Tabel 3.6 Use Case Description: Edit User**

Use case name:	<i>Edit User</i>	
Actor:	Principal	
Trigger:	<i>Actor</i> ingin mengubah data <i>user</i> yang telah terdaftar.	
Flow of activities:	Actor:	System:
	<p>1. <i>Actor</i> masuk ke menu <i>Admin Panel</i> dan memilih opsi <i>Users Management.</i></p>	<p>1.1 Sistem menampilkan halaman <i>Users Management.</i></p> <p>2.1 Sistem menampilkan halaman <i>Edit User</i> berisi data <i>user</i> yang dipilih.</p>

	<p>2. <i>Actor</i> menekan tombol <i>Action</i> pada salah satu <i>user</i> dan memilih opsi <i>Edit</i>.</p> <p>3. <i>Actor</i> mengubah data yang ada dan menekan tombol <i>Save</i> jika telah selesai.</p>	<p>3.1 Data yang diubah akan tersimpan ke dalam <i>database</i>.</p>
--	--	--

**Tabel 3.7 Use Case Description: Delete User**

Use case name:	<i>Delete User</i>	
Actor:	Principal	
Trigger:	<i>Actor</i> ingin menghapus data <i>user</i> tertentu dari aplikasi.	
Flow of activities:	Actor:	System:
	<p>1. <i>Actor</i> masuk ke menu <i>Admin Panel</i> dan memilih opsi <i>Users Management</i>.</p> <p>2. <i>Actor</i> menekan tombol <i>Action</i> pada salah satu <i>user</i> dan memilih opsi <i>Delete User</i>.</p> <p>3. <i>Actor</i> menekan tombol <i>Delete</i></p>	<p>1.1 Sistem menampilkan halaman <i>Users Management</i>.</p> <p>2.1 Sistem memunculkan <i>pop-up</i> sebagai konfirmasi penghapusan data <i>user</i>.</p> <p>3.1 Sistem menghapus data <i>user</i> dari <i>database</i>.</p>

	pada <i>pop-up</i> konfirmasi.	
--	-----------------------------------	--

**Tabel 3.8 Use Case Description: View User**

Use case name:	<i>View User</i>	
Actor:	Principal	
Trigger:	<i>Actor</i> ingin melihat daftar <i>user</i> yang terdaftar pada aplikasi.	
Flow of activities:	Actor:	System:
	1. <i>Actor</i> masuk ke menu <i>Admin Panel</i> dan memilih opsi <i>Users Management</i>	1,1 Sistem menampilkan halaman <i>Users Management</i> berisi daftar <i>user</i> yang terdaftar,

**Tabel 3.9 Use Case Description: Manage Project**

Use case name:	<i>Manage Project</i>	
Actor:	Principal	
Trigger:	<i>Actor</i> ingin mengelola projek yang ada pada aplikasi.	
Flow of activities:	Actor:	System:
	1. <i>Actor</i> masuk ke menu <i>Project</i> dan memilih opsi <i>Project List</i> .	1.1 Sistem menampilkan halaman <i>Project</i> 2.1 Sistem memproses aksi yang dipilih.

	<p>2. <i>Actor</i> memilih aksi yang ingin dilakukan pada halaman <i>Project List</i>.</p>	
--	--	--

**Tabel 3.10 Use Case Description: Search Project**

Use case name:	<i>Search Project</i>	
Actor:	Principal, Pentester, Client	
Trigger:	<i>Actor</i> ingin mencari nama <i>project</i> dengan menggunakan kata kunci.	
Flow of activities:	<p>Actor:</p> <ol style="list-style-type: none"> <li>1. <i>Actor</i> masuk ke menu <i>Project</i> dan memilih opsi <i>Project List</i>.</li> <li>2. <i>Actor</i> mengisi kata kunci yang ingin dicari pada <i>search bar</i> dan menekan <i>icon</i> kaca pembesar atau menekan tombol <i>Enter</i>.</li> </ol>	<p>System:</p> <ol style="list-style-type: none"> <li>1.1 Sistem menampilkan halaman <i>Project Lists</i> berisi daftar <i>project</i>.</li> <li>2.1 Sistem menampilkan daftar <i>project</i> yang terdapat kata kunci yang dicari.</li> </ol>

**Tabel 3.11 Use Case Description: Add Project**

Use case name:	<i>Add Project</i>
Actor:	Principal

Trigger:	<i>Actor</i> ingin menambah <i>project</i> baru ke dalam aplikasi.	
Flow of activities:	Actor:	System:
	<ol style="list-style-type: none"> <li>1. <i>Actor</i> masuk ke menu <i>Project</i> dan memilih opsi <i>Project List</i>.</li> <li>2. <i>Actor</i> menekan tombol <i>Create New Project</i>.</li> <li>3. <i>Actor</i> mengisi detail data <i>project</i> dan <i>Object Pentest</i> serta memilih <i>user pentester</i> dan <i>client</i> yang terlibat, kemudian menekan tombol <i>Save New Project</i> jika telah selesai.</li> </ol>	<p>1.1 Sistem menampilkan halaman <i>Project Lists</i> berisi daftar <i>project</i>.</p> <p>2.1 Sistem menampilkan halaman <i>Create New Project</i> berisi <i>form</i> dan opsi.</p> <p>3.1 Data yang telah diubah akan disimpan ke dalam <i>database</i>.</p>

**Tabel 3.12 Use Case Description: Edit Project**

Use case name:	<i>Edit Project</i>	
Actor:	Principal	
Trigger:	<i>Actor</i> ingin mengubah data sebuah <i>project</i> .	
Flow of activities:	Actor:	System:
	<ol style="list-style-type: none"> <li>1. <i>Actor</i> masuk ke menu <i>Project</i> dan memilih opsi</li> </ol>	<p>1.1 Sistem menampilkan halaman <i>Project Lists</i> berisi daftar <i>project</i>.</p>

	<p><i>Project List.</i></p> <p>2. <i>Actor</i> menekan tombol <i>icon titik tiga</i> pada salah satu <i>project</i> dan memilih opsi <i>Edit Project</i>.</p> <p>3. <i>Actor</i> mengubah data dari <i>project</i> tersebut dan menekan tombol <i>Edit Project</i> jika telah selesai.</p>	<p>2.1 Sistem menampilkan halaman <i>Edit Project</i> berisi data <i>project</i> yang dipilih.</p> <p>3.1 Perubahan data akan disimpan ke dalam <i>database</i>.</p>
--	--	--

**Tabel 3.13 Use Case Description: Delete Project**

Use case name:	<i>Delete Project</i>	
Actor:	Principal	
Trigger:	<i>Actor</i> ingin menghapus <i>project</i> dari aplikasi	
Flow of activities:	Actor:	System:
	<p>1. <i>Actor</i> masuk ke menu <i>Project</i> dan memilih opsi <i>Project List</i>.</p> <p>2. <i>Actor</i> menekan tombol <i>icon titik tiga</i> pada salah satu <i>project</i> dan memilih opsi <i>Delete Project</i>.</p>	<p>1.1 Sistem menampilkan halaman <i>Project Lists</i> berisi daftar <i>project</i>.</p> <p>2.1 Sistem memunculkan <i>pop-up</i> sebagai konfirmasi penghapusan data <i>project</i>.</p> <p>3.1 Data <i>project</i> dihapus dari <i>database</i>.</p>

	3. <i>Actor</i> menekan tombol <i>Delete</i> pada <i>pop-up</i> konfirmasi.	
--	---	--

**Tabel 3.14 Use Case Description: View Project**

Use case name:	<i>View Project</i>	
Actor:	Principal, Pentester, Client	
Trigger:	<i>Actor</i> ingin melihat detail atau <i>progress</i> projek yang akan dimulai, sedang dikerjakan, atau sudah selesai.	
Flow of activities:	Actor:	System:
	1. <i>Actor</i> ke menu <i>Project</i> dan memilih opsi <i>Project List</i> .	1.1 Sistem menampilkan halaman <i>Project Lists</i> berisi daftar <i>project</i> yang ada,

**Tabel 3.15 Use Case Description: Manage Finding**

Use case name:	<i>Manage Finding</i>	
Actor:	Principal, Pentester	
Trigger:	<i>Actor</i> ingin mengelola <i>finding</i> hasil dari <i>penetration test</i> .	
Flow of activities:	Actor:	System:
	1. <i>Actor</i> masuk ke menu <i>Project</i> dan memilih opsi <i>Project List</i> .	1.1 Sistem menampilkan halaman <i>Project Lists</i> berisi daftar <i>project</i> .
	2. <i>Actor</i> memilih dan	2.1 Sistem menampilkan

	<p>menekan salah satu <i>project</i>.</p> <p>3. <i>Actor</i> memilih aksi yang ingin dilakukan pada bagian <i>Findings Lists</i>.</p>	<p>data detail mengenai <i>project</i> yang dipilih.</p> <p>3.1 Sistem memproses aksi yang dipilih.</p>
--	---	---

**Tabel 3.16 Use Case Description: Add Finding**

Use case name:	<i>Add Finding</i>	
Actor:	Principal, Pentester	
Trigger:	<i>Actor</i> ingin menambah <i>finding</i> baru dari hasil <i>penetration test</i> .	
Flow of activities:	<p>Actor:</p> <ol style="list-style-type: none"> <li><i>Actor</i> masuk ke menu <i>Project</i> dan memilih opsi <i>Project List</i>.</li> <li>a.1. <i>Actor</i> memilih dan menekan salah satu <i>project</i>.</li> <li>a.2. <i>Actor</i> menekan tombol <i>Add New Finding</i>.</li> <li>a.3. <i>Actor</i> mengisi data detail mengenai <i>finding</i> dan menekan tombol <i>Add Finding</i> jika telah</li> </ol>	<p>System:</p> <ol style="list-style-type: none"> <li>1.1 Sistem menampilkan halaman <i>Project Lists</i> berisi daftar <i>project</i>.</li> <li>a.1.1 Sistem menampilkan data detail mengenai <i>project</i> yang dipilih.</li> <li>a.2.1. Sistem menampilkan halaman <i>Add New Finding</i> berisi form dan opsi.</li> <li>a.3.1. Data mengenai <i>finding</i> disimpan ke dalam <i>database</i>.</li> <li>b.1.1. Sistem menampilkan halaman</li> </ol>

	<p>selesai.</p> <p>2.b.1. <i>Actor</i> menekan <i>icon</i> titik tiga pada sebuah <i>project</i> dan memilih opsi <i>Add Finding</i>.</p> <p>2,b,2, <i>Actor</i> mengisi data detail mengenai <i>finding</i> dan menekan tombol <i>Add Finding</i> jika telah selesai.</p>	<p><i>Add New Finding</i> berisi form dan opsi.</p> <p>2.b.2.1. Data mengenai <i>finding</i> disimpan ke dalam <i>database</i>.</p>
--	--	---

**Tabel 3.17 Use Case Description: Edit Finding**

Use case name:	<i>Edit Finding</i>	
Actor:	Principal, Pentester	
Trigger:	<i>Actor</i> ingin mengubah data mengenai sebuah <i>finding</i> .	
Flow of activities:	Actor:	System:
	<ol style="list-style-type: none"> <li>1. <i>Actor</i> masuk ke menu <i>Project</i> dan memilih opsi <i>Project List</i>.</li> <li>2. <i>Actor</i> memilih dan menekan salah satu <i>project</i>.</li> <li>3. <i>Actor</i> menekan tombol <i>Action</i></li> </ol>	<ol style="list-style-type: none"> <li>1.1. Sistem menampilkan halaman <i>Project Lists</i> berisi daftar <i>project</i>.</li> <li>2.1. Sistem menampilkan data detail mengenai <i>project</i> yang dipilih.</li> <li>3.1. Sistem menampilkan data <i>finding</i> yang dipilih.</li> <li>4.1. Data yang telah</li> </ol>

	<p>pada <i>finding</i> yang dipilih dan memilih opsi <i>Edit Finding</i>.</p> <p>4. <i>Actor</i> mengubah data <i>finding</i> dan menekan tombol <i>Save Finding</i> jika telah selesai.</p>	diubah akan disimpan ke dalam <i>database</i> .
--	--	---

**Tabel 3.18 Use Case Description: Delete Finding**

Use case name:	<i>Delete Finding</i>	
Actor:	Principal, Pentester	
Trigger:	<i>Actor</i> ingin menghapus <i>finding</i> dari sebuah <i>project</i> .	
Flow of activities:	<p>Actor:</p> <ol style="list-style-type: none"> <li><i>Actor</i> masuk ke menu <i>Project</i> dan memilih opsi <i>Project List</i>.</li> <li><i>Actor</i> memilih dan menekan salah satu <i>project</i>.</li> <li><i>Actor</i> menekan tombol <i>Action</i> pada <i>finding</i> yang ingin dihapus dan memilih opsi <i>Delete Finding</i>.</li> <li><i>Actor</i> menekan</li> </ol>	<p>System:</p> <ol style="list-style-type: none"> <li>1. Sistem menampilkan halaman <i>Project Lists</i> berisi daftar <i>project</i>.</li> <li>2. Sistem menampilkan data detail mengenai <i>project</i> yang dipilih.</li> <li>3. Sistem memunculkan <i>pop-up</i> konfirmasi penghapusan.</li> <li>4. Data <i>finding</i> akan dihapus dari database.</li> </ol>

	tombol <i>Delete</i> pada <i>pop-up</i> konfirmasi.	
--	---	--

**Tabel 3.19 Use Case Description: Change Finding Status**

Use case name:	<i>Resolve Finding</i>	
Actor:	Principal	
Trigger:	<i>Actor</i> ingin mengubah status dari sebuah <i>finding</i> .	
Flow of activities:	Actor:	System:
	1. <i>Actor</i> masuk ke menu <i>Project</i> dan memilih opsi <i>Project List</i> . 2. <i>Actor</i> memilih dan menekan salah satu <i>project</i> . 3. <i>Actor</i> memilih status <i>finding</i> mana yang ingin diubah. 3.a.1. <i>Actor</i> memilih untuk mengubah status <i>finding</i> menjadi <i>Fixed</i> dengan menekan bar status <i>finding</i> <i>Not Fixed</i> .	1.1. Sistem menampilkan halaman <i>Project Lists</i> berisi daftar <i>project</i> . 2.1. Sistem menampilkan data detail mengenai <i>project</i> yang dipilih. 3.1. Sistem menampilkan daftar status <i>finding</i> yang dipilih. 3.a.1.1. Sistem menampilkan daftar <i>finding</i> dengan status <i>Not Fixed</i> . 3.a.2.1. Sistem memindahkan <i>finding</i> ke status <i>Fixed</i> . 3.b.1.1. Sistem menampilkan daftar <i>finding</i> dengan status <i>Fixed</i> .

	<p>3.a.2. <i>Actor</i> menekan tombol <i>Action</i> pada <i>finding</i> yang ingin diubah dan memilih opsi <i>Resolve to Fixed.</i></p> <p>3.b.1. <i>Actor</i> memilih untuk mengubah status <i>finding</i> menjadi <i>Not Fixed</i> dengan menekan bar status <i>finding Fixed.</i></p> <p>3.b.2. <i>Actor</i> menekan tombol <i>Action</i> pada <i>finding</i> yang ingin diubah dan memilih opsi <i>Re-open to Not Fixed.</i></p>	<p>3.b.2.1. Sistem memindahkan <i>finding</i> ke status <i>Not Fixed.</i></p>
--	--	---

**Tabel 3.20 Use Case Diagram: View Finding**

Use case name:	<i>View Finding</i>
Actor:	Principal, Pentester, Client
Trigger:	Actor ingin melihat keseluruhan atau <i>progress finding</i>

	dari hasil <i>penetration test</i> .	
Flow of activities:	Actor:	System:
	<ol style="list-style-type: none"> <li>1. <i>Actor</i> masuk ke menu <i>Project</i> dan memilih opsi <i>Project List</i>.</li> <li>2. <i>Actor</i> memilih dan menekan salah satu <i>project</i>.</li> <li>3. <i>Actor</i> mengecek bagian <i>Findings Lists</i>.</li> </ol>	<ol style="list-style-type: none"> <li>1.1. Sistem menampilkan halaman <i>Project Lists</i> berisi daftar <i>project</i>.</li> <li>2.1. Sistem menampilkan data detail mengenai <i>project</i> yang dipilih.</li> <li>3.1. Sistem menampilkan daftar <i>finding</i> dari hasil <i>penetration test</i>.</li> </ol>

**Tabel 3.21 Use Case Description: Generate Report**

Use case name:	<i>Generate Report</i>	
Actor:	Principal, Pentester, Client	
Trigger:	<i>Actor</i> ingin mengekspor hasil <i>penetration test</i> dalam bentuk file pdf.	
Flow of activities:	Actor:	System:
	<ol style="list-style-type: none"> <li>1. <i>Actor</i> masuk ke menu <i>Project</i> dan memilih opsi <i>Project List</i>.</li> <li>2. <i>Actor</i> menekan icon titik tiga sebuah <i>project</i> dan memilih opsi <i>Generate Report</i>.</li> </ol>	<ol style="list-style-type: none"> <li>1.1. Sistem menampilkan halaman <i>Project Lists</i> berisi daftar <i>project</i>.</li> <li>2.1. Sistem akan membuat file pdf berisi data <i>project</i> yang dipilih dan mengirimkan file.</li> </ol>

**Tabel 3.22 Use Case Description: Manage Testing Guide**

Use case name:	<i>Manage Testing Guide</i>	
Actor:	Principal, Pentester	
Trigger:	<i>Actor</i> ingin mengelola <i>resource</i> pada <i>testing guide</i> .	
Flow of activities:	Actor:	System:
	<ol style="list-style-type: none"> <li>1. <i>Actor</i> masuk ke menu <i>Resources</i> dan memilih opsi <i>List Testing Guide</i>.</li> <li>2. <i>Actor</i> memilih <i>List Testing Guide</i>.</li> <li>3. <i>Actor</i> memilih aksi yang ingin dilakukan dengan <i>Testing Guide</i>.</li> </ol>	<ol style="list-style-type: none"> <li>1.1. Sistem menampilkan halaman <i>Testing Guide Lists</i>.</li> <li>2.1. Sistem menampilkan halaman <i>Testing Guide</i> yang dipilih.</li> <li>3.1. Sistem memproses aksi yang dipilih.</li> </ol>

**Tabel 3.23 Use Case Description: Add Testing Guide**

Use case name:	<i>Add Testing Guide</i>	
Actor:	Principal, Pentester	
Trigger:	<i>Actor</i> ingin menambahkan <i>guide</i> baru ke <i>testing guide</i> .	
Flow of activities:	Actor:	System:
	<ol style="list-style-type: none"> <li>1. <i>Actor</i> masuk ke menu <i>Resources</i></li> </ol>	<ol style="list-style-type: none"> <li>1.1. Sistem menampilkan halaman <i>Testing Guide</i></li> </ol>

	<p>dan memilih opsi <i>List Testing Guide</i>.</p> <p>2. <i>Actor</i> memilih opsi <i>Mobile Testing Guide</i> atau <i>Web Testing Guide</i></p> <p>3. <i>Actor</i> menekan tombol <i>Add Resource</i>.</p> <p>4. <i>Actor</i> mengisi data terkait <i>guide</i> yang ingin ditambahkan dan menekan tombol <i>Add Testing Guide</i> jika telah selesai.</p>	<p><i>Lists.</i></p> <p>2.1. Sistem menampilkan opsi yang dipilih.</p> <p>3.1. Sistem menampilkan <i>form</i> terkait isi <i>testing guide</i>.</p> <p>4.1. Data yang telah diisi disimpan ke dalam <i>database</i>.</p>
--	---	--

**Tabel 3.24 Use Case Description: Edit Testing Guide**

Use case name:	<i>Edit Testing Guide</i>	
Actor:	Principal, Pentester	
Trigger:	<i>Actor</i> ingin mengubah data sebuah <i>testing guide</i> .	
Flow of activities:	Actor:	System:
	<p>1. <i>Actor</i> masuk ke menu <i>Resources</i> dan memilih opsi <i>List Testing Guide</i>.</p>	<p>1.1. Sistem menampilkan halaman <i>Testing Guide Lists</i>.</p> <p>2.1. Sistem menampilkan opsi yang dipilih.</p>

	<p>2. <i>Actor</i> memilih opsi <i>Mobile Testing Guide</i> atau <i>Web Testing Guide</i>.</p> <p>3. <i>Actor</i> memilih dan menekan salah satu <i>testing guide</i>.</p> <p>4. <i>Actor</i> menekan icon titik tiga dan memilih opsi <i>Edit Resource</i> dan menekan tombol <i>Edit Testing Guide</i> jika telah selesai.</p>	<p>3.1. Sistem menampilkan data isi <i>testing guide</i>.</p> <p>4.1. Sistem menyimpan perubahan data pada ke <i>database</i>.</p>
--	--	--

**Tabel 3.25 Use Case Description: Delete Testing Guide**

Use case name:	<i>Delete Testing Guide</i>	
Actor:	Principal, Pentester	
Trigger:	<i>Actor</i> ingin menghapus <i>guide</i> dari <i>testing guide</i> .	
Flow of activities:	Actor:	System:
	<p>1. <i>Actor</i> masuk ke menu <i>Resources</i> dan memilih opsi <i>List Testing Guide</i>.</p> <p>2. <i>Actor</i> memilih opsi <i>Mobile Testing Guide</i> atau</p>	<p>1.1. Sistem menampilkan halaman <i>Testing Guide Lists</i>.</p> <p>2.1. Sistem menampilkan opsi yang dipilih.</p> <p>3.1. Sistem menampilkan data isi <i>testing guide</i>.</p> <p>4.1. Sistem memunculkan</p>

	<p><i>Web Testing Guide.</i></p> <p>3. Principal, Pentester memilih dan menekan salah satu <i>testing guide</i>.</p> <p>4. Principal, Pentester menekan <i>icon</i> titik tiga dan memilih opsi <i>Delete Resource</i>.</p> <p>5. Principal, Pentester menekan tombol <i>Delete</i> pada <i>pop-up</i> konfirmasi.</p>	<p><i>pop-up</i> konfirmasi penghapusan.</p> <p>5.1. Sistem menghapus data <i>guide</i> dari <i>database</i>.</p>
--	--	---

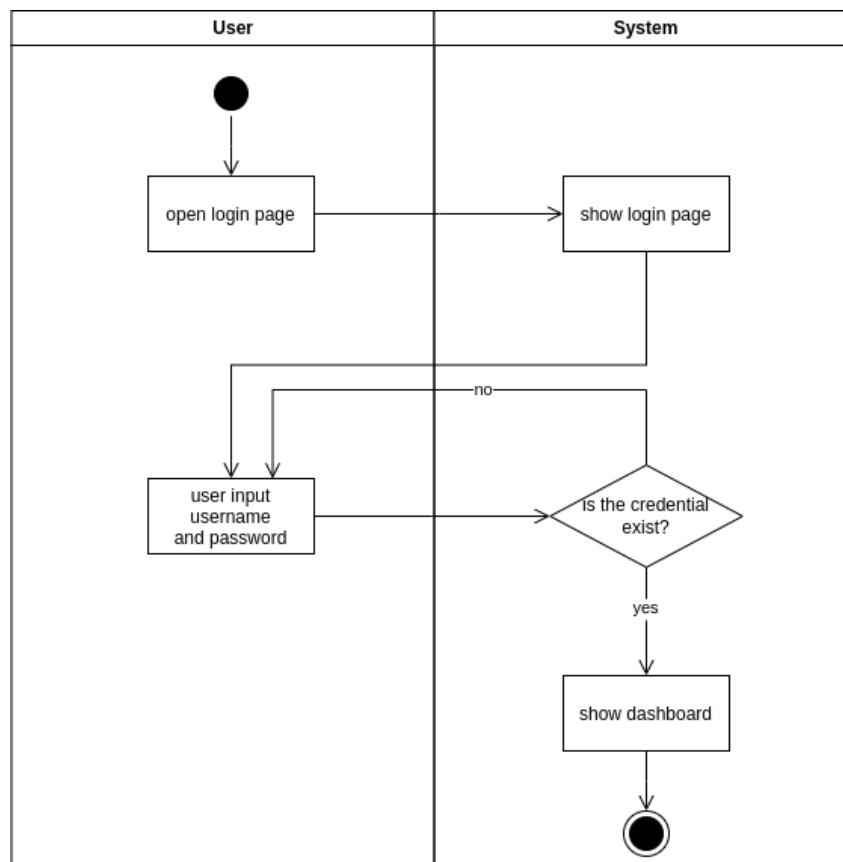
**Tabel 3.26 Use Case Description: View Testing Guide**

Use case name:	<i>View Testing Guide</i>	
Actor:	Principal, Pentester	
Trigger:	<i>Actor</i> ingin melihat <i>testing guide</i> .	
Flow of activities:	Actor:	System:
	<p>1. <i>Actor</i> masuk ke menu <i>Resources</i> dan memilih opsi <i>List Testing Guide</i>.</p> <p>2. <i>Actor</i> memilih</p>	<p>1.1. Sistem menampilkan halaman <i>Testing Guide Lists</i>.</p> <p>2.1. Sistem menampilkan opsi yang dipilih.</p> <p>3.1. Sistem menampilkan</p>

	<p>opsi <i>Mobile Testing Guide</i> atau <i>Web Testing Guide</i>.</p> <p>3. <i>Actor</i> memilih dan menekan salah satu <i>testing guide</i>.</p>	
--	--	--

### 3.3.3 *Activity Diagram*

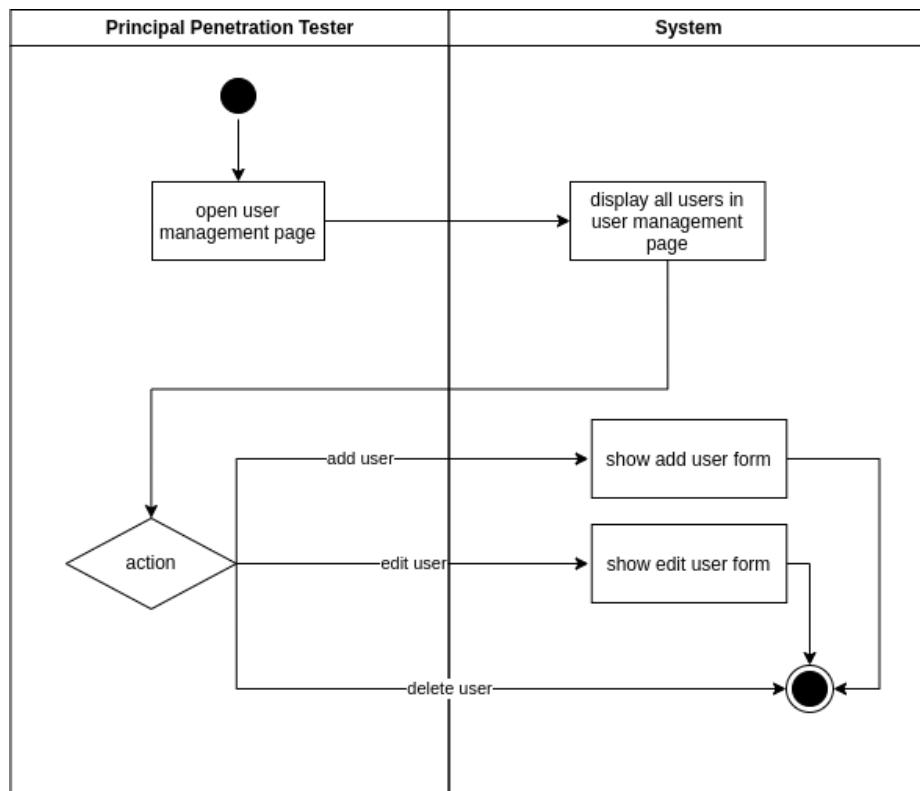
Penulis juga membuat *activity diagram* agar dapat menunjukkan aktivitas apa saja yang terjadi dari interaksi *user* terhadap sistem yang dikembangkan:



Gambar 3.6 *Activity Diagram Login*

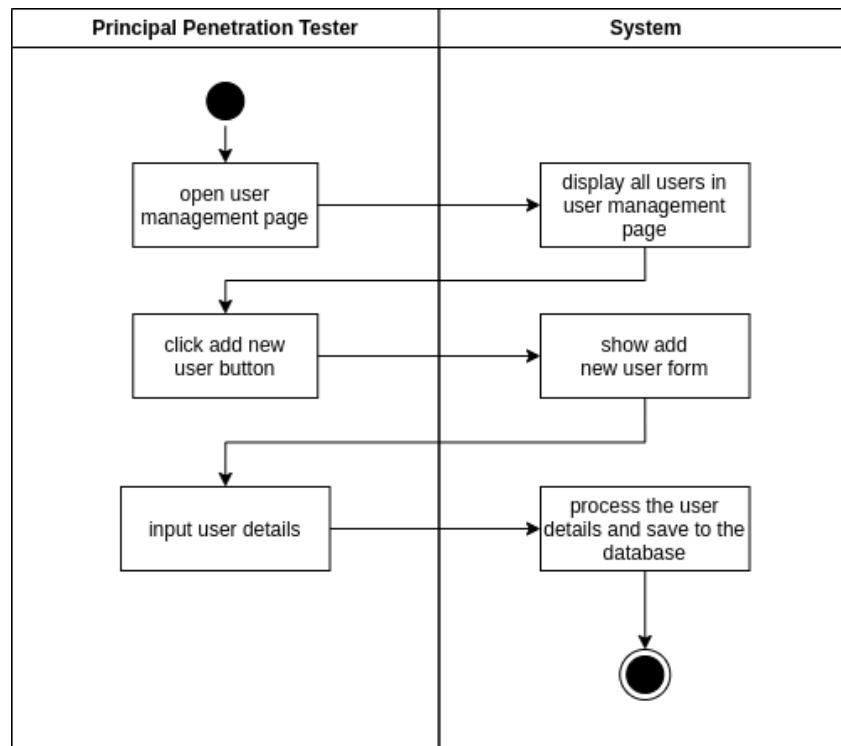
Untuk dapat menggunakan aplikasi, *User* wajib *login* terlebih dahulu ke dalam aplikasi. Pada fitur *Login* ini, *actor* diminta untuk membuka halaman *login* terlebih dahulu yang nantinya sistem akan menampilkan halaman *login* pada

*actor*, setelah itu *actor* akan memasukkan *username* dan *password*, kemudian sistem akan memproses *username* dan *password* dengan mengecek apakah kredensial tersebut atau tidak. Jika tidak, *actor* akan dikembalikan untuk memasukkan *username* dan *password* kembali dan jika benar sistem akan menampilkan *dashboard* dari aplikasi.



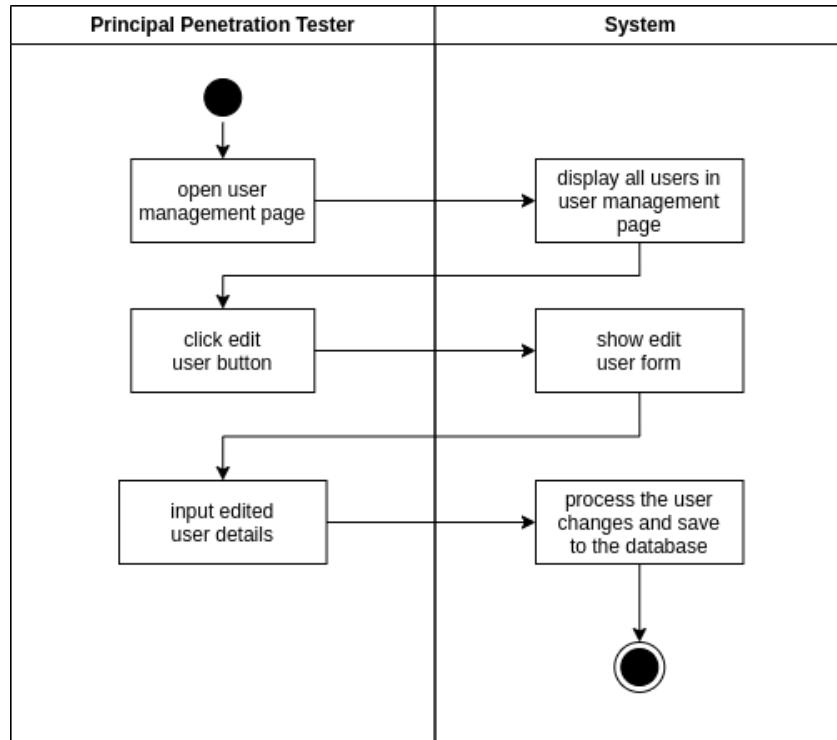
**Gambar 3.7 Activity Diagram Manage User**

Fitur *Manage User* ini berguna bagi *Principal Penetration Tester* untuk mengelola *user* yang terdaftar pada aplikasi. Pada fitur *Manage User* ini, *actor* membuka halaman *user management* yang nantinya sistem akan menampilkan semua *user* yang terdaftar pada halaman *user management*. Kemudian *actor* dapat memilih beberapa aksi, yang pertama adalah *add user* yang akan menampilkan *add user form* untuk menambahkan *user*, kedua *edit user* yang akan menampilkan *edit user form* untuk mengubah *user* dan ketiga adalah *delete user* untuk menghapus *user*.



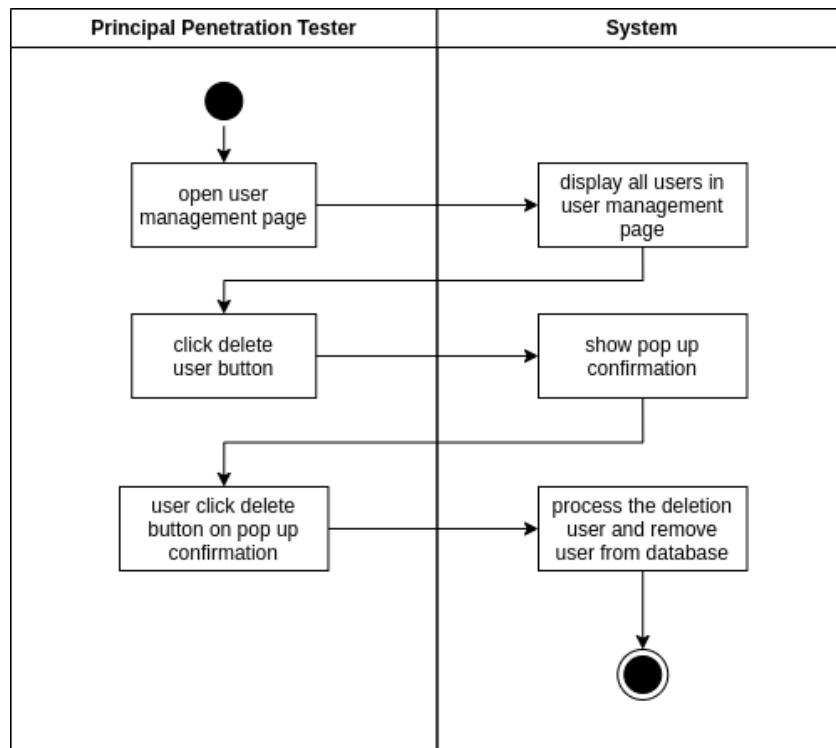
**Gambar 3.8 Activity Diagram Add User**

Fitur *Add User* ini berguna bagi *Principal Penetration Tester* untuk menambahkan *user*. Pada fitur *Add User* ini, *actor* membuka halaman *user management* yang nantinya sistem akan menampilkan semua *user* yang terdaftar pada halaman *user management*. Kemudian *actor* dapat menekan tombol *add new user* yang nantinya sistem akan menampilkan *add new user form*. *Actor* menginputkan data *user* yang ingin ditambahkan, lalu sistem akan memproses dan menyimpan data *user* tersebut ke *database*.



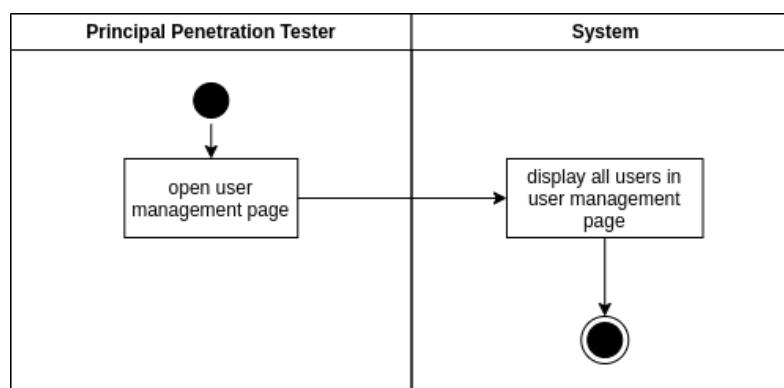
**Gambar 3.9 Activity Diagram Edit User**

Fitur *Edit User* ini berguna bagi *Principal Penetration Tester* untuk mengubah data *user*. Pada fitur *Edit User* ini, *actor* membuka halaman *user management* yang nantinya sistem akan menampilkan semua *user* yang terdaftar pada halaman *user management*. Kemudian *actor* dapat menekan tombol *edit user* yang nantinya sistem akan menampilkan *edit user form*. *Actor* menginputkan data *user* yang sudah diubah, lalu sistem akan memproses dan menyimpan perubahan data *user* tersebut ke *database*.



**Gambar 3.10 Activity Diagram Delete User**

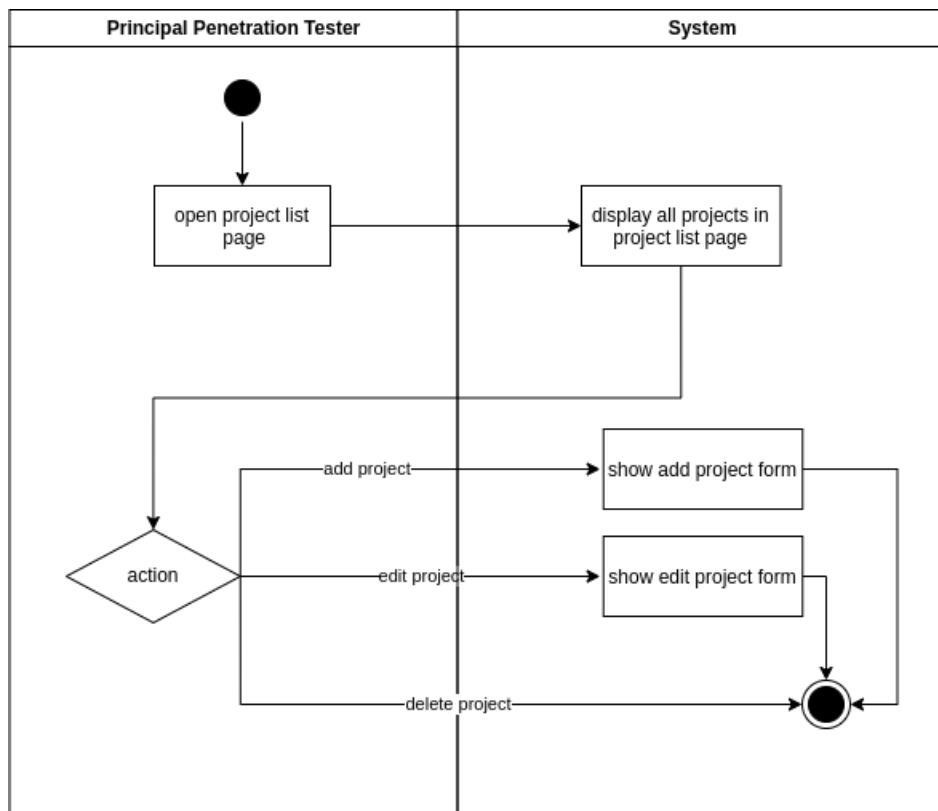
Fitur *Delete User* ini berguna bagi *Principal Penetration Tester* untuk menghapus *user*. Pada fitur *Delete User* ini, *actor* membuka halaman *user management* yang nantinya sistem akan menampilkan semua *user* yang terdaftar pada halaman *user management*. Kemudian *actor* dapat menekan tombol *delete user* yang nantinya sistem akan menampilkan *pop-up* konfirmasi penghapusan. Setelah itu *actor* menekan tombol *delete*, lalu sistem akan memproses dan menghapus *user* tersebut dari *database*.



**Gambar 3.11 Activity Diagram View User**

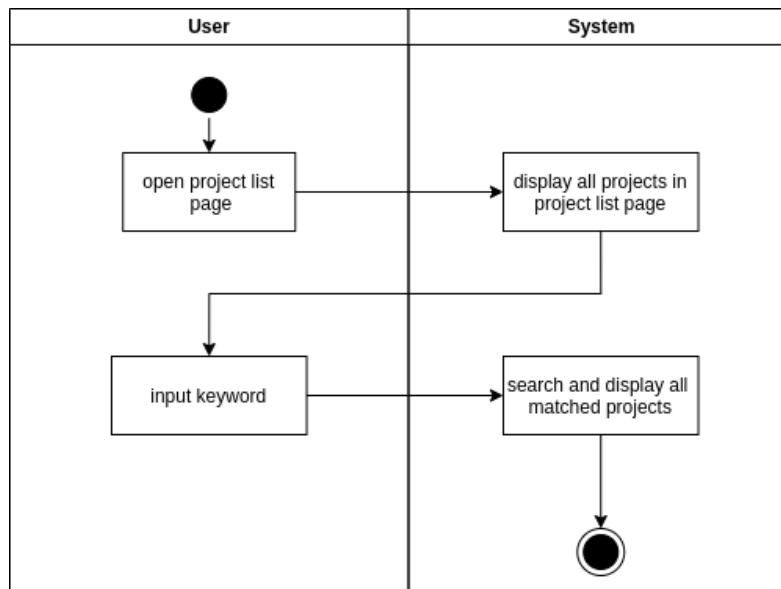
Fitur *View User* ini berguna bagi *Principal Penetration Tester* untuk melihat *user*. Pada fitur *View User* ini, *actor* membuka halaman *user management*

yang nantinya sistem akan menampilkan semua *user* yang terdaftar pada halaman *user management*.



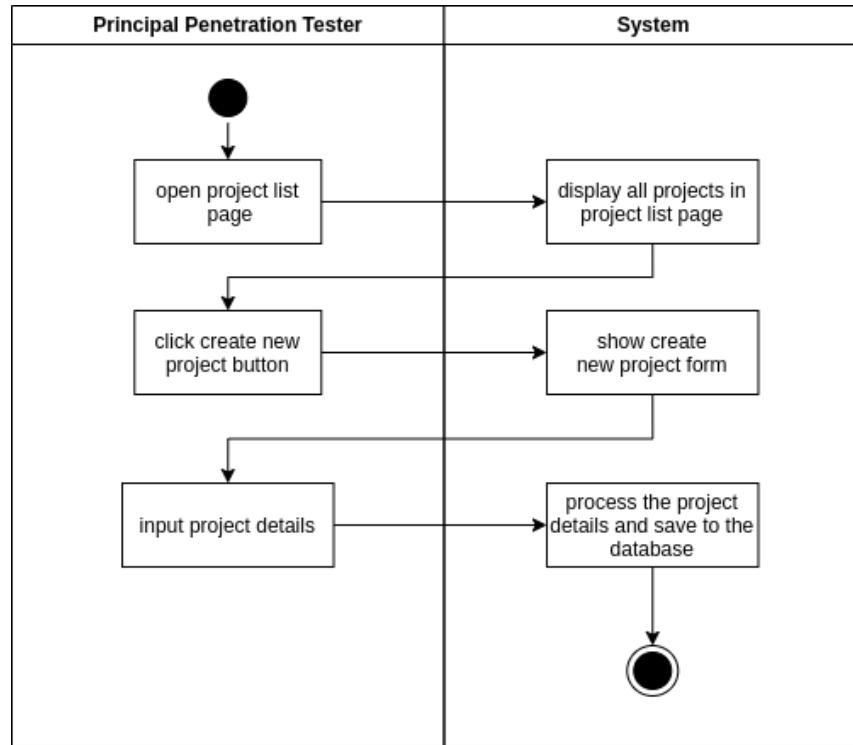
**Gambar 3.12 Activity Diagram Manage Project**

Fitur *Manage Project* ini berguna bagi *Principal Penetration Tester* untuk mengelola *project* yang ada pada aplikasi. Pada fitur *Manage Project* ini, *actor* membuka halaman *project list* yang nantinya sistem akan menampilkan semua *project* yang ada pada halaman *project list*. Kemudian *actor* dapat memilih beberapa aksi, yang pertama adalah *add project* yang akan menampilkan *add project form* untuk menambahkan *project*, kedua *edit project* yang akan menampilkan *edit project form* untuk mengubah *project* dan ketiga adalah *delete project* untuk menghapus *project*.



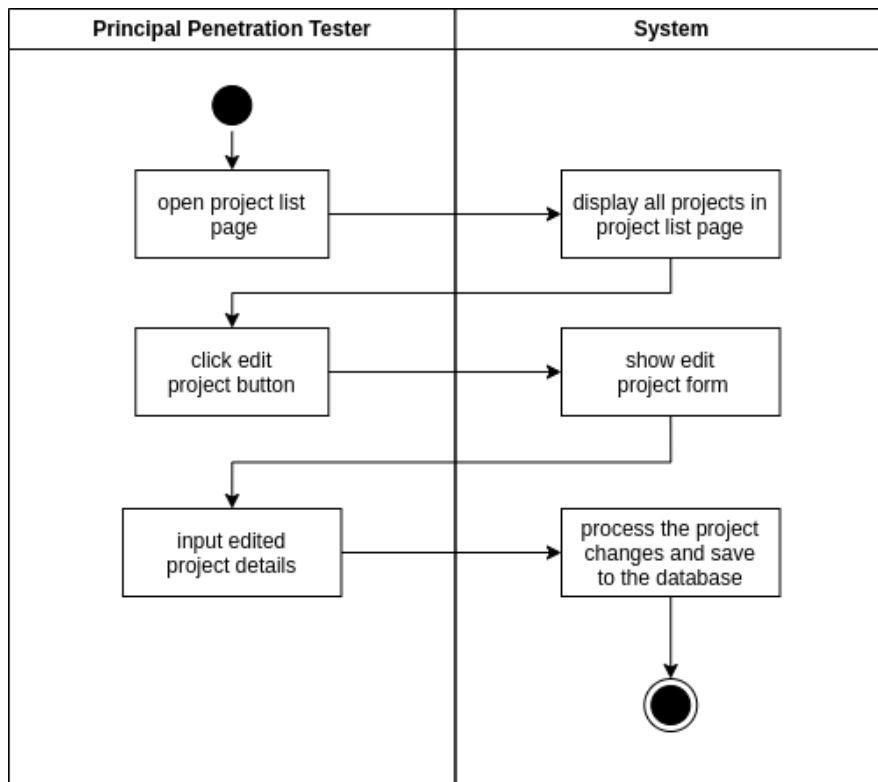
**Gambar 3.13 Activity Diagram Search Project**

Fitur *Search Project* ini berguna bagi *User* untuk mencari *project*. Pada fitur *Search Project* ini, *actor* membuka halaman *project list* yang nantinya sistem akan menampilkan semua *project* yang ada pada halaman *project list*. Kemudian *actor* dapat menginputkan kata kunci pada form *Search Project* dan menekan tombol kaca pembesar yang nantinya sistem akan mencari dan menampilkan *project* yang sesuai dengan kata kunci yang dicari.



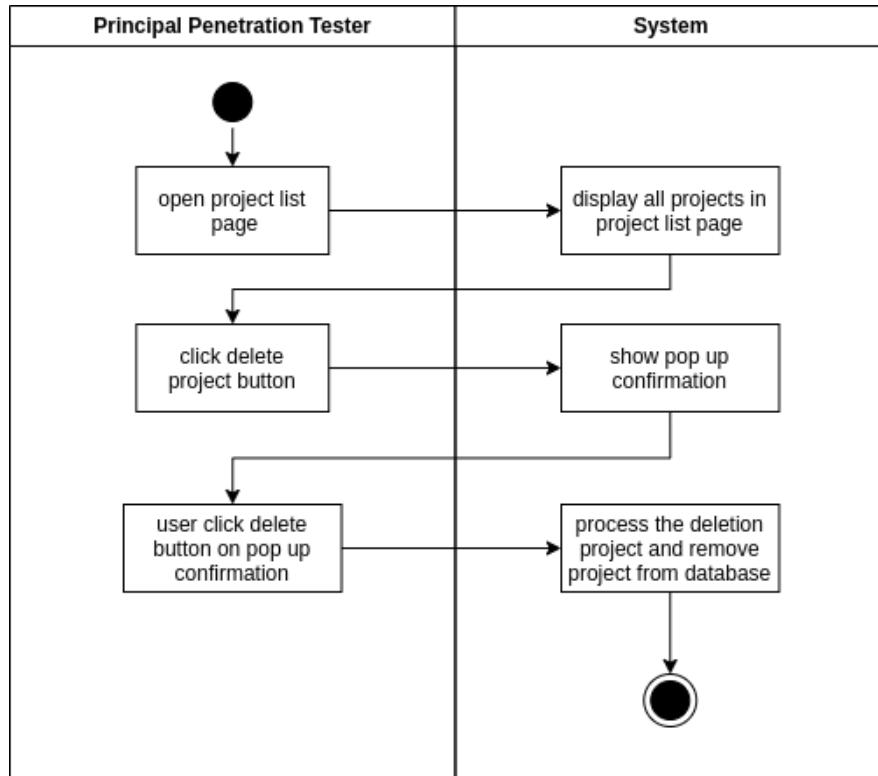
**Gambar 3.14 Activity Diagram Add Project**

Fitur *Add Project* ini berguna bagi *Principal Penetration Tester* untuk menambahkan *project*. Pada fitur *Add Project* ini, *actor* membuka halaman *project list* yang nantinya sistem akan menampilkan semua *project* yang ada pada halaman *project list*. Kemudian *actor* dapat menekan tombol *create new project* yang nantinya sistem akan menampilkan *create new project form*. *Actor* menginputkan data *project* yang ingin ditambahkan, lalu sistem akan memproses dan menyimpan data *project* tersebut ke *database*.



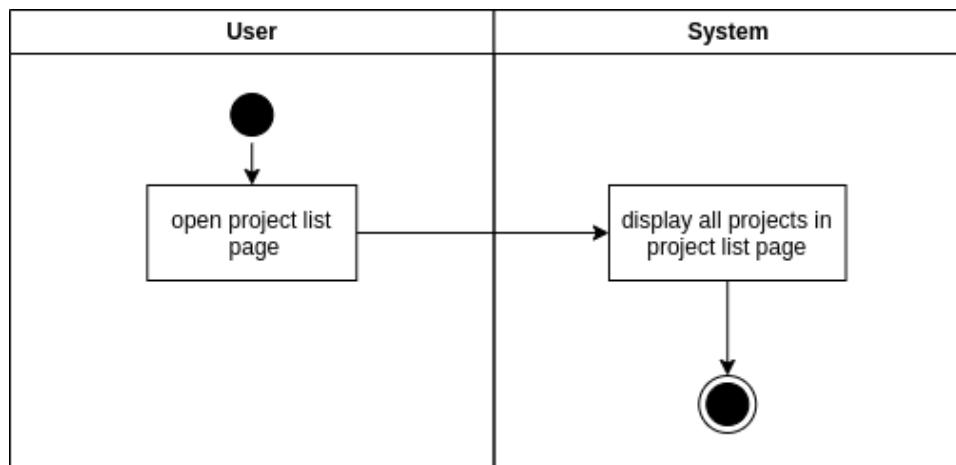
**Gambar 3.15 Activity Diagram Edit Project**

Fitur *Edit Project* ini berguna bagi *Principal Penetration Tester* untuk mengubah data *project*. Pada fitur *Edit Project* ini, *actor* membuka halaman *project list* yang nantinya sistem akan menampilkan semua *project* yang ada pada halaman *project list*. Kemudian *actor* dapat menekan tombol *edit project* yang nantinya sistem akan menampilkan *edit project form*. *Actor* menginputkan data *project* yang sudah diubah, lalu sistem akan memproses dan menyimpan perubahan data *project* tersebut ke *database*.



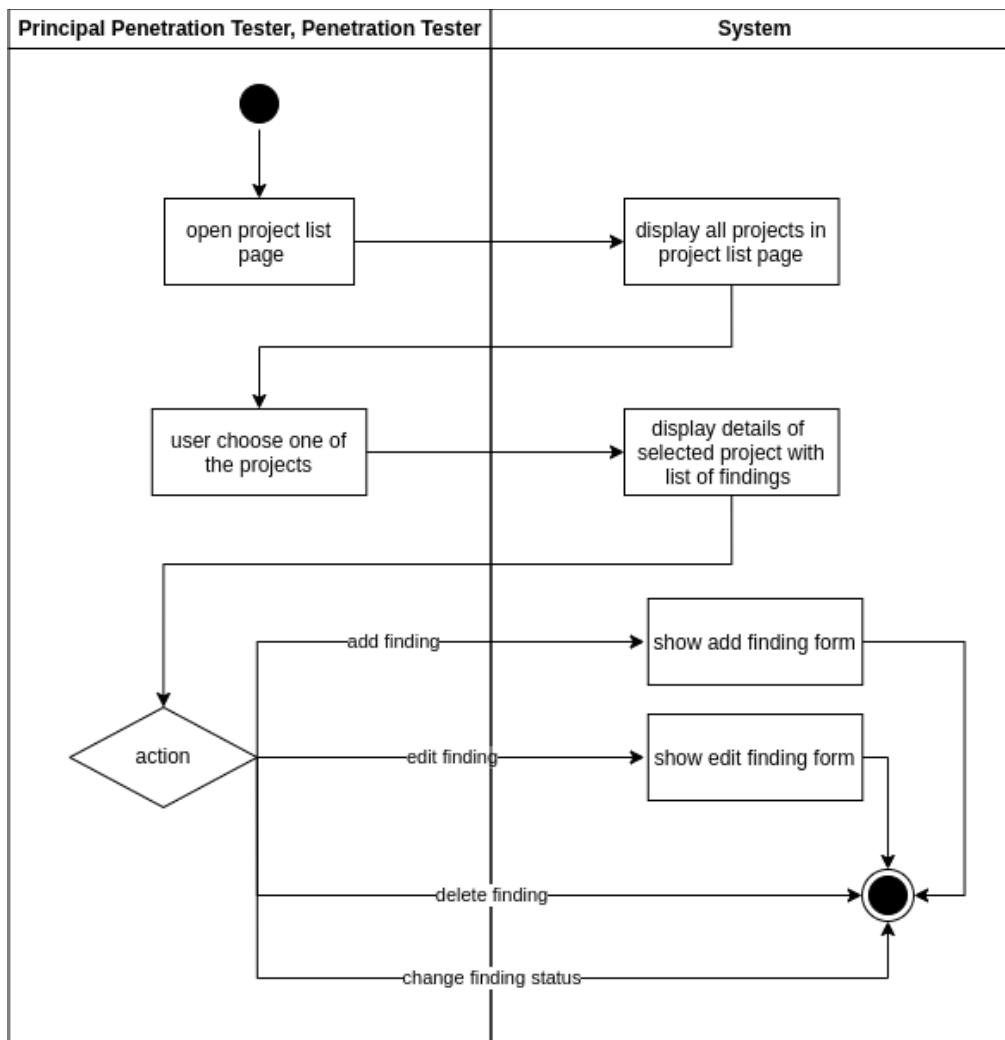
**Gambar 3.16 Activity Diagram Delete Project**

Fitur *Delete Project* ini berguna bagi *Principal Penetration Tester* untuk menghapus *project*. Pada fitur *Delete Project* ini, *actor* membuka halaman *project list* yang nantinya sistem akan menampilkan semua *project* yang ada pada halaman *project list*. Kemudian *actor* dapat menekan tombol *delete project* yang nantinya sistem akan menampilkan *pop-up* konfirmasi penghapusan. Setelah itu *actor* menekan tombol *delete*, lalu sistem akan memproses dan menghapus *project* tersebut dari *database*.



**Gambar 3.17 Activity Diagram View Project**

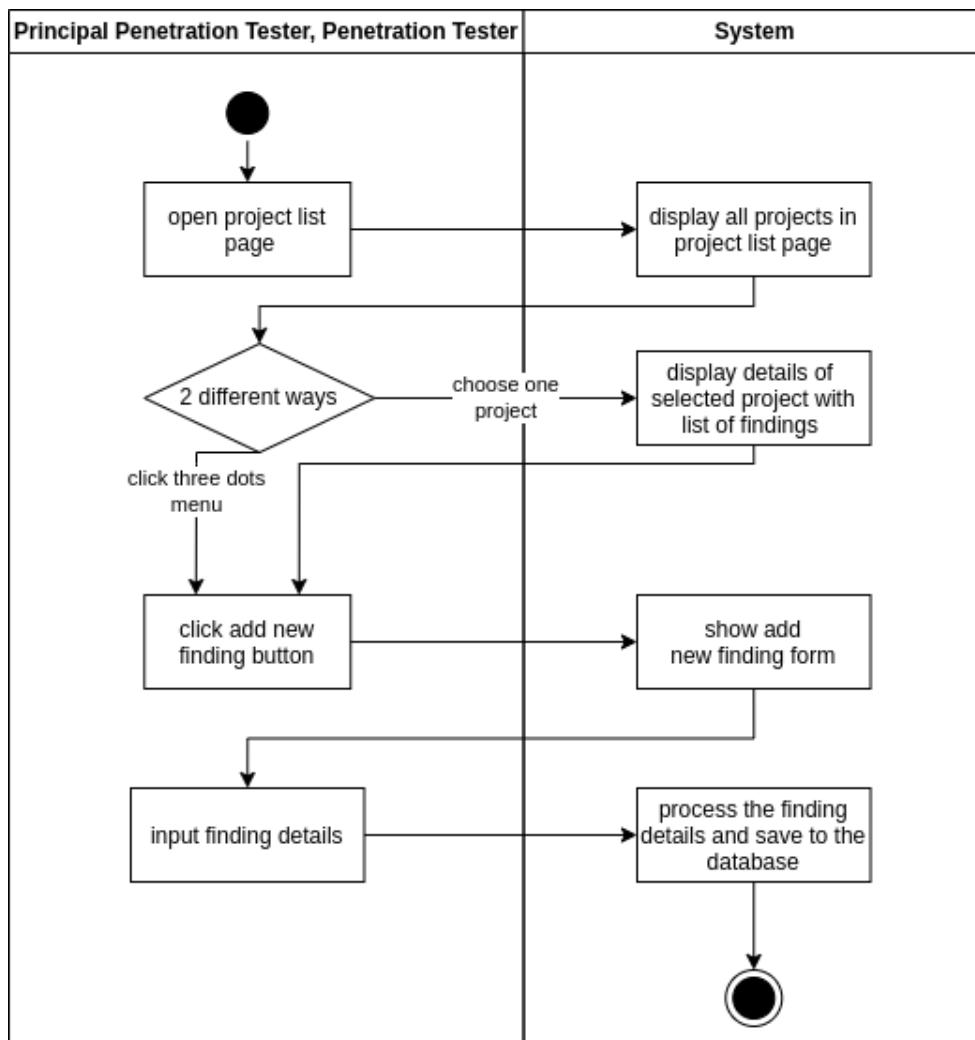
Fitur *View Project* ini berguna bagi *User* untuk melihat *project*. Pada fitur *View Project* ini, *actor* membuka halaman *project list* yang nantinya sistem akan menampilkan semua *project* yang ada pada halaman *project list*.



**Gambar 3.18 Activity Diagram Manage Finding**

Fitur *Manage Finding* ini berguna bagi *Principal Penetration Tester* dan *Penetration Tester* untuk mengelola *finding* yang ada pada suatu *project*. Pada fitur *Manage Finding* ini, *actor* membuka halaman *project list* yang nantinya sistem akan menampilkan semua *project* yang ada pada halaman *project list*. Lalu *actor* akan memilih salah satu *project* yang ada dan sistem akan menampilkan detail dari *project* yang dipilih beserta dengan daftar *finding* yang ditemukan. Kemudian *actor* dapat memilih beberapa aksi, yang pertama adalah *add finding* yang akan menampilkan *add finding form* untuk menambahkan *finding*, kedua *edit finding* yang akan menampilkan *edit finding form* untuk mengubah *finding*,

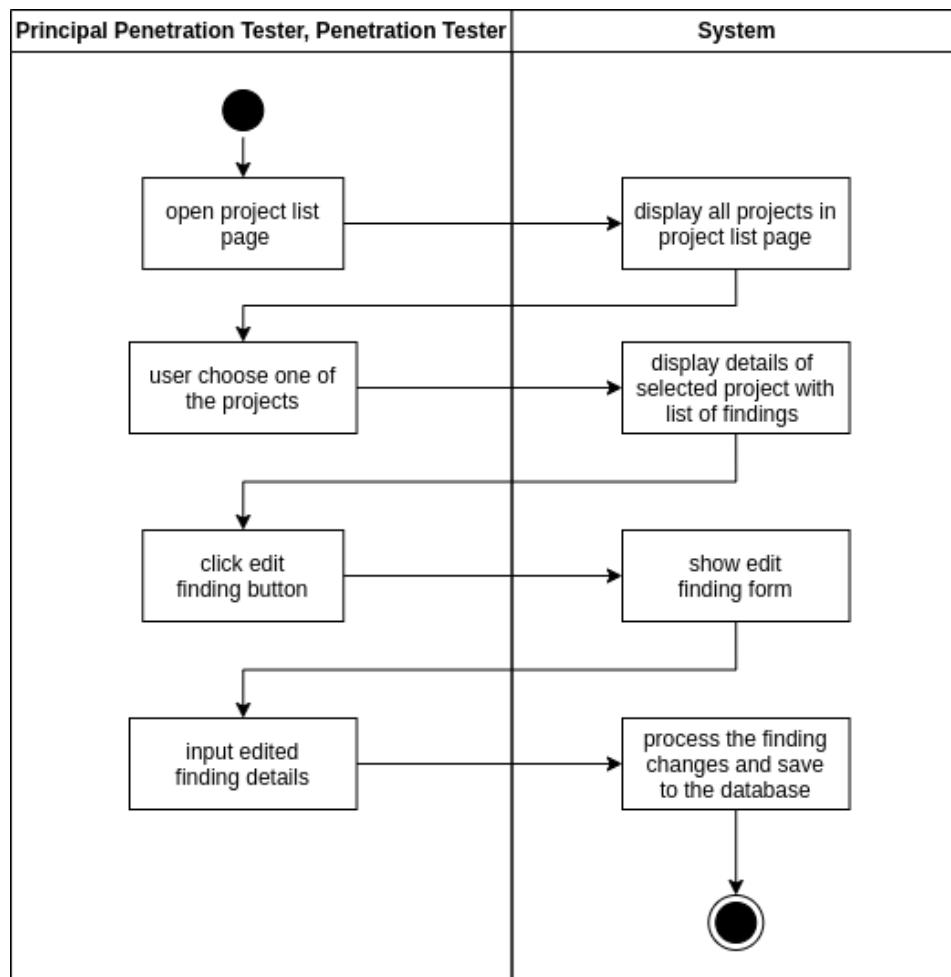
ketiga adalah *delete finding* untuk menghapus *finding* dan yang keempat adalah *change finding status* untuk mengubah status *finding*.



**Gambar 3.19 Activity Diagram Add Finding**

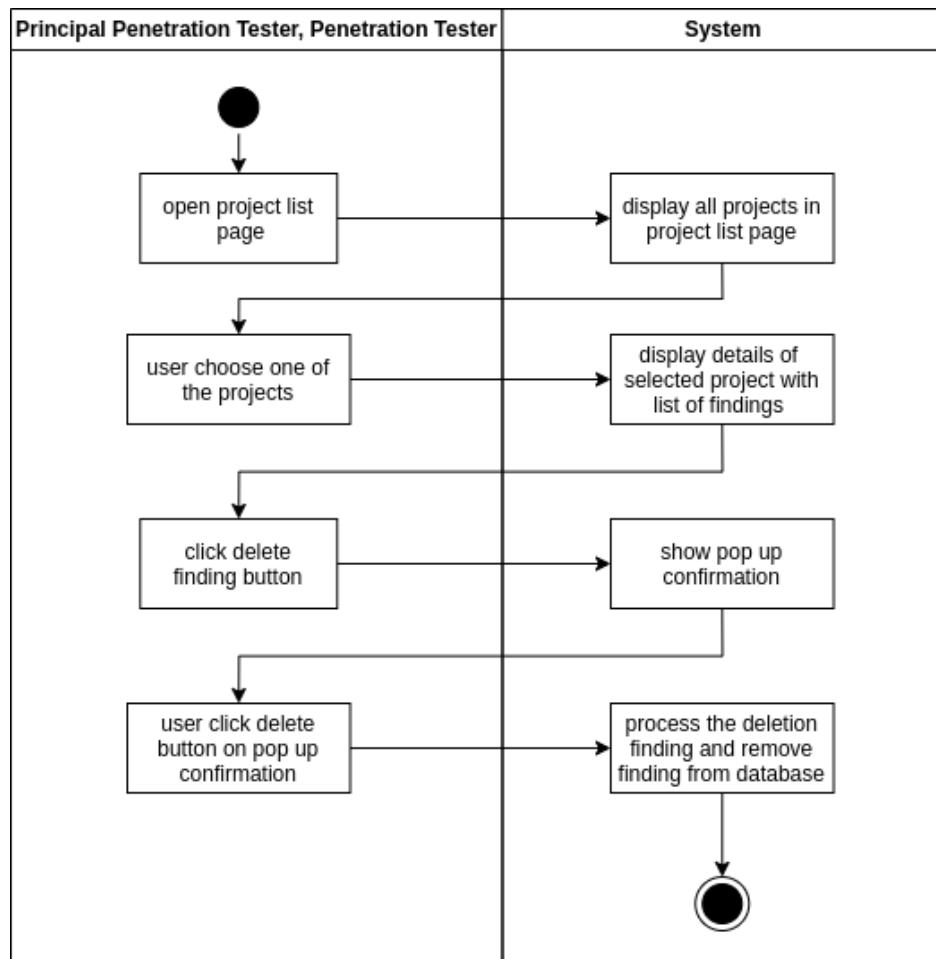
Fitur *Add Finding* ini berguna bagi *Principal Penetration Tester* dan *Penetration Tester* untuk menambahkan *finding* yang ada pada suatu *project*. Pada fitur *Add Finding* ini, *actor* membuka halaman *project list* yang nantinya sistem akan menampilkan semua *project* yang ada pada halaman *project list*. Ada dua cara untuk menambahkan *finding*, cara pertama yaitu melalui tiga titik menu pada *project* atau *actor* memilih salah satu *project* yang ada dan sistem akan menampilkan detail dari *project* yang dipilih beserta dengan daftar *finding* yang ditemukan. Kemudian *actor* dapat menekan tombol *add new finding* yang nantinya sistem akan menampilkan *add new finding form*. *Actor* menginputkan

data *finding* yang ingin ditambahkan, lalu sistem akan memproses dan menyimpan data *finding* tersebut ke *database*.



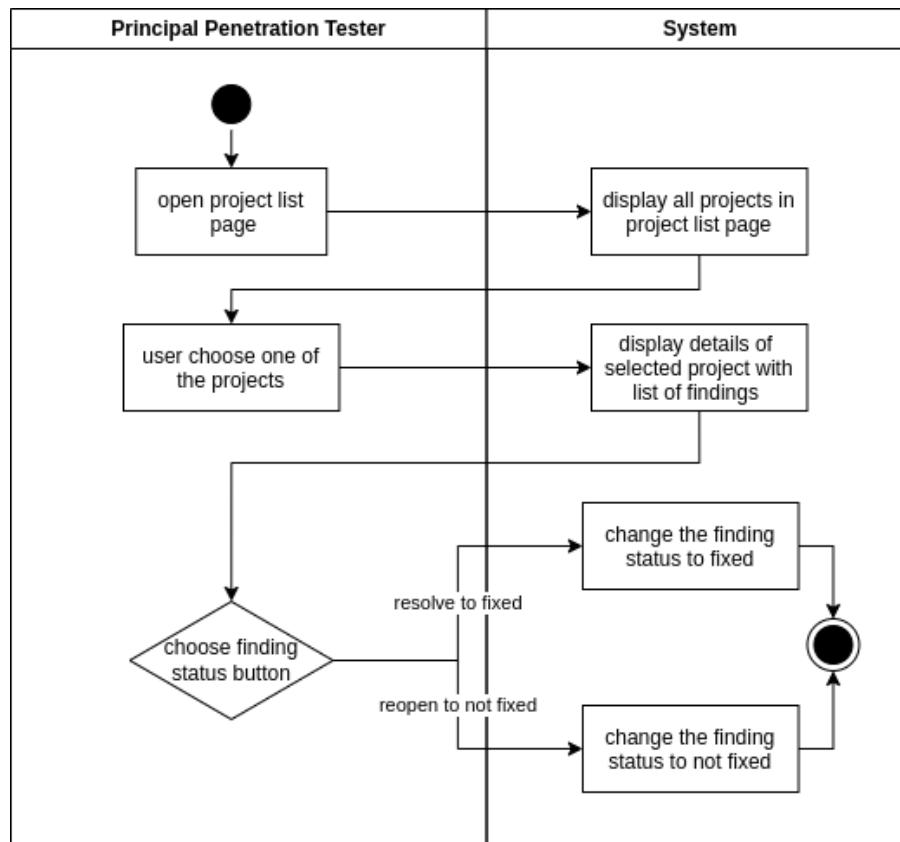
**Gambar 3.20 Activity Diagram Edit Finding**

Fitur *Edit Finding* ini berguna bagi *Principal Penetration Tester* dan *Penetration Tester* untuk mengubah data *finding* yang ada pada suatu *project*. Pada fitur *Edit Finding* ini, *actor* membuka halaman *project list* yang nantinya sistem akan menampilkan semua *project* yang ada pada halaman *project list*. Lalu *actor* memilih salah satu *project* yang ada dan sistem akan menampilkan detail dari *project* yang dipilih beserta dengan daftar *finding* yang ditemukan. Kemudian *actor* dapat menekan tombol *edit finding* yang nantinya sistem akan menampilkan *edit finding form*. *Actor* menginputkan data *finding* yang telah diubah, lalu sistem akan memproses dan menyimpan perubahan data *finding* tersebut ke *database*.



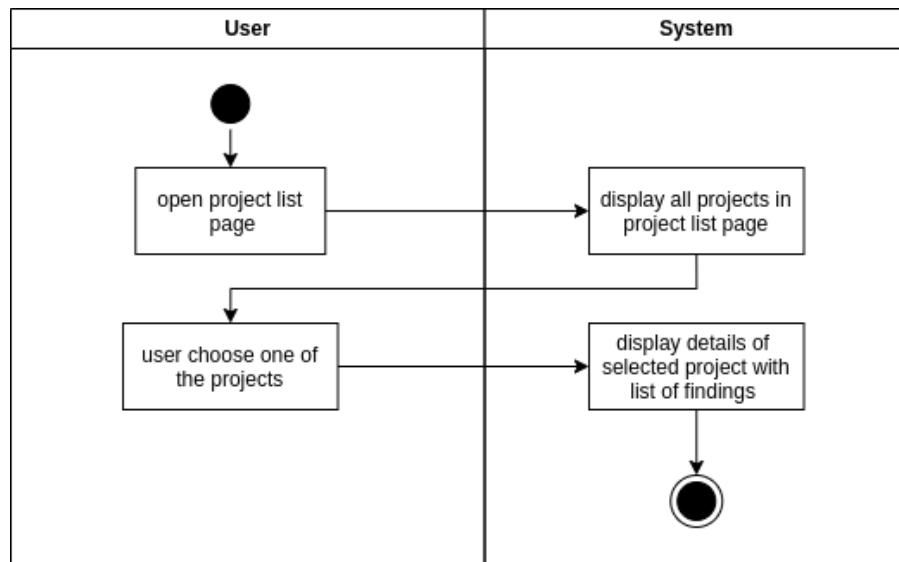
**Gambar 3.21 Activity Diagram Delete Finding**

Fitur *Delete Finding* ini berguna bagi *Principal Penetration Tester* dan *Penetration Tester* untuk menghapus *finding* yang ada pada suatu *project*. Pada fitur *Delete Finding* ini, *actor* membuka halaman *project list* yang nantinya sistem akan menampilkan semua *project* yang ada pada halaman *project list*. Lalu *actor* memilih salah satu *project* yang ada dan sistem akan menampilkan detail dari *project* yang dipilih beserta dengan daftar *finding* yang ditemukan. Kemudian *actor* dapat menekan tombol *delete finding* yang nantinya sistem akan menampilkan *pop-up* konfirmasi penghapusan. Setelah itu *actor* menekan tombol *delete*, lalu sistem akan memproses dan menghapus *finding* tersebut dari *database*.



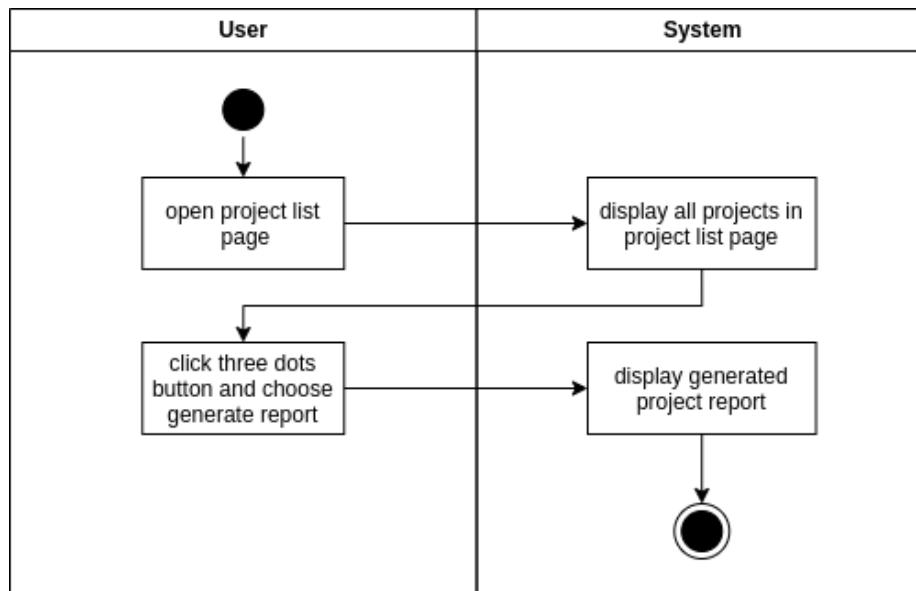
**Gambar 3.22 Activity Diagram Change Finding Status**

Fitur *Change Finding Status* ini berguna bagi *Principal Penetration Tester* untuk mengganti status *finding*. Pada fitur *Change Finding Status* ini, *actor* membuka halaman *project list* yang nantinya sistem akan menampilkan semua *project* yang ada pada halaman *project list*. Lalu *actor* memilih salah satu *project* yang ada dan sistem akan menampilkan detail dari *project* yang dipilih beserta dengan daftar *finding* yang ditemukan. Kemudian *actor* dapat memilih tombol antara *resolve to fixed* atau *reopen to not fixed*. Setelah *actor* menekan tombol, sistem akan memprosesnya dengan mengganti status *finding* ke *fixed* atau *not fixed*.



**Gambar 3.23 Activity Diagram View Finding**

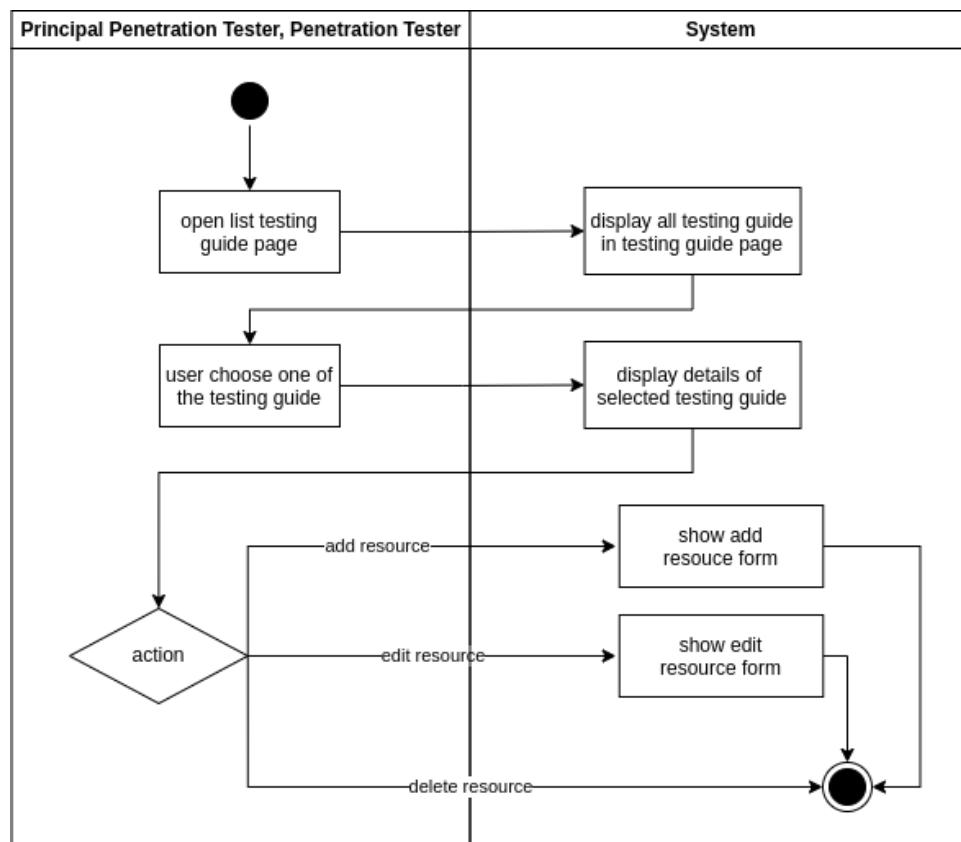
Fitur *View Finding* ini berguna bagi *User* untuk melihat *finding* yang ada pada suatu *project*. Pada fitur *View Finding* ini, *actor* membuka halaman *project list* yang nantinya sistem akan menampilkan semua *project* yang ada pada halaman *project list*. Lalu *actor* memilih salah satu *project* yang ada dan sistem akan menampilkan detail dari *project* yang dipilih beserta dengan daftar *finding* yang ditemukan.



**Gambar 3.24 Activity Diagram Generate Report**

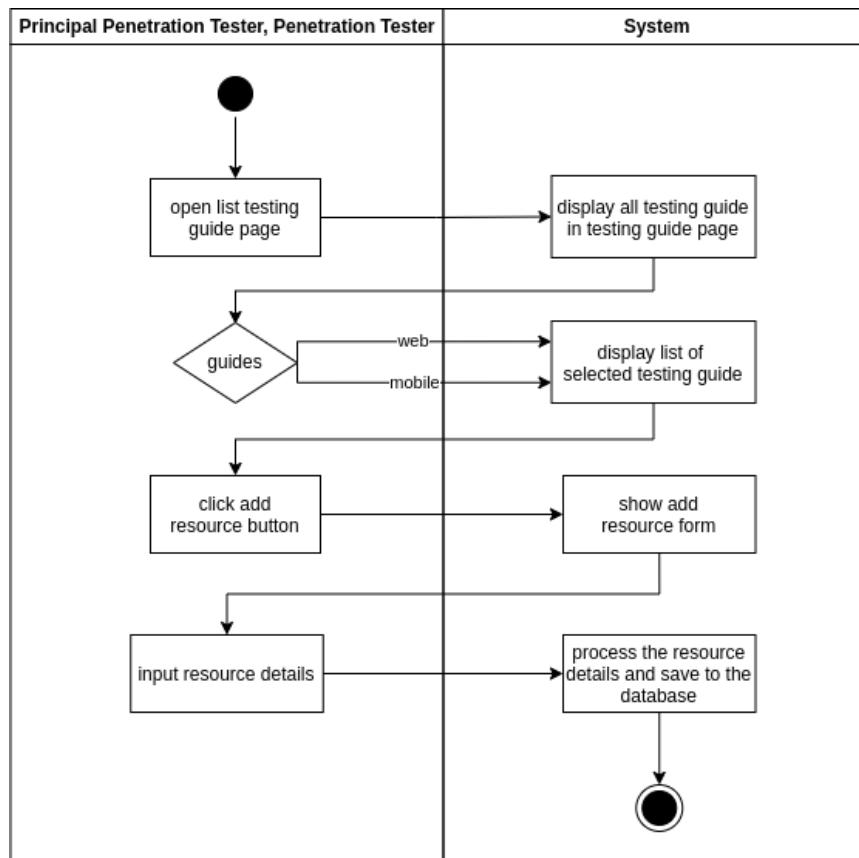
Fitur *Generate Report* ini berguna bagi *User* untuk melihat laporan hasil dari suatu *project*. Pada fitur *Generate Report* ini, *actor* membuka halaman

*project list* yang nantinya sistem akan menampilkan semua *project* yang ada pada halaman *project list*. Lalu *actor* menekan tiga titik menu dan memilih *generate report*, setelah itu sistem akan memproses dan menampilkan hasil laporan dari *project* yang dipilih.



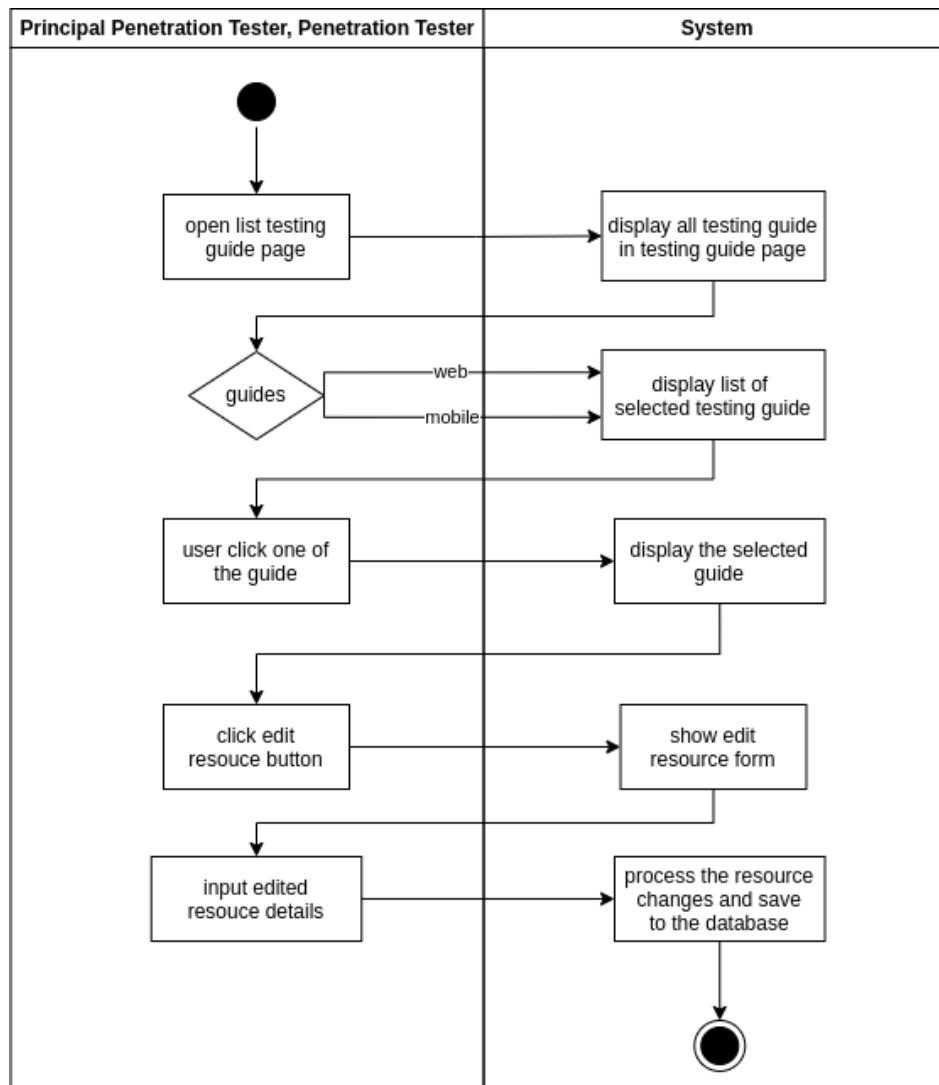
**Gambar 3.25 Activity Diagram Manage Testing Guide**

Fitur *Manage Testing Guide* ini berguna bagi *Principal Penetration Tester* dan *Penetration Tester* untuk mengelola *testing guide* yang ada. Pada fitur *Manage Testing Guide* ini, *actor* membuka halaman *list testing guide* yang nantinya sistem akan menampilkan semua *testing guide* yang ada pada halaman *testing guide*. Lalu *actor* akan memilih salah satu *testing guide* yang ada dan sistem akan menampilkan detail dari *testing guide*. Kemudian *actor* dapat memilih beberapa aksi, yang pertama adalah *add resource* yang akan menampilkan *add resource form* untuk menambahkan *testing guide*, kedua *edit resource* yang akan menampilkan *edit resource form* untuk mengubah *testing guide* dan yang ketiga adalah *delete resource* untuk menghapus *testing guide*.



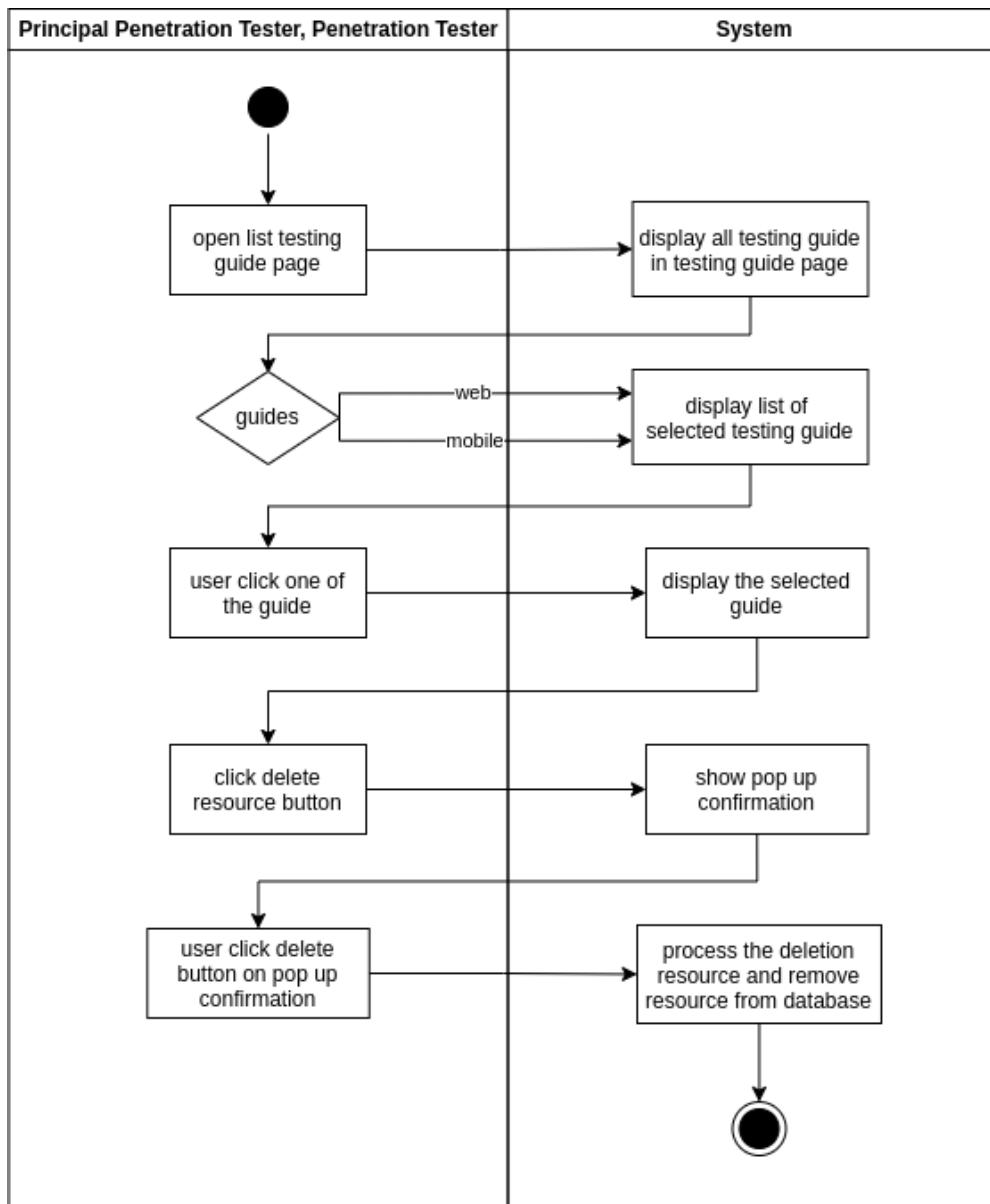
**Gambar 3.26 Activity Diagram Add Testing Guide**

Fitur *Add Testing Guide* ini berguna bagi *Principal Penetration Tester* dan *Penetration Tester* untuk menambahkan *testing guide*. Pada fitur *Add Testing Guide* ini, *actor* membuka halaman *list testing guide* yang nantinya sistem akan menampilkan semua *testing guide* yang ada pada halaman *testing guide*. Lalu *actor* akan memilih salah satu *testing guide* antara *web* atau *mobile*, setelah itu sistem akan menampilkan detail dari *testing guide* yang dipilih. Kemudian *actor* dapat menekan tombol *add resource* yang nantinya sistem akan menampilkan *add resource form*. *Actor* menginputkan data *testing guide* yang ingin ditambahkan, lalu sistem akan memproses dan menyimpan data *testing guide* tersebut ke *database*.



**Gambar 3.27 Activity Diagram Edit Testing Guide**

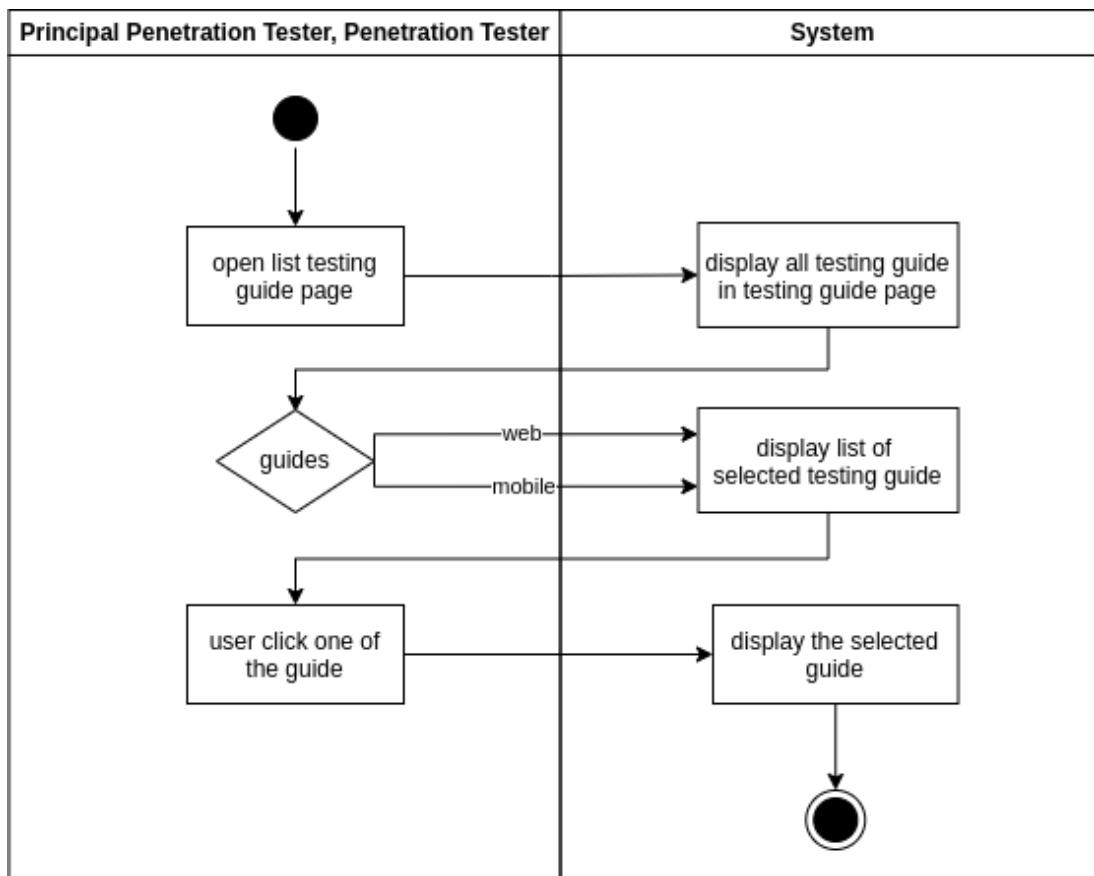
Fitur *Edit Testing Guide* ini berguna bagi *Principal Penetration Tester* dan *Penetration Tester* untuk mengubah data *testing guide*. Pada fitur *Edit Testing Guide* ini, *actor* membuka halaman *list testing guide* yang nantinya sistem akan menampilkan semua *testing guide* yang ada pada halaman *testing guide*. Lalu *actor* akan memilih salah satu *testing guide* antara *web* atau *mobile*, setelah itu sistem akan menampilkan detail dari *testing guide* yang dipilih. Kemudian *actor* memilih kembali salah satu materi *testing guide* yang tersedia dan sistem akan menampilkan *testing guide* tersebut. Setelah itu *actor* dapat menekan tombol *edit resource* yang nantinya sistem akan menampilkan *edit resource form*. *Actor* menginputkan data *testing guide* yang sudah diubah, lalu sistem akan memproses dan menyimpan perubahan data *testing guide* tersebut ke *database*.



**Gambar 3.28 Activity Diagram Delete Testing Guide**

Fitur *Delete Testing Guide* ini berguna bagi *Principal Penetration Tester* dan *Penetration Tester* untuk menghapus *testing guide*. Pada fitur *Delete Testing Guide* ini, *actor* membuka halaman *list testing guide* yang nantinya sistem akan menampilkan semua *testing guide* yang ada pada halaman *testing guide*. Lalu *actor* akan memilih salah satu *testing guide* antara *web* atau *mobile*, setelah itu sistem akan menampilkan detail dari *testing guide* yang dipilih. Kemudian *actor* memilih kembali salah satu materi *testing guide* yang tersedia dan sistem akan menampilkan *testing guide* tersebut. Setelah itu *actor* dapat menekan tombol *delete resource* yang nantinya sistem akan menampilkan *pop-up* konfirmasi

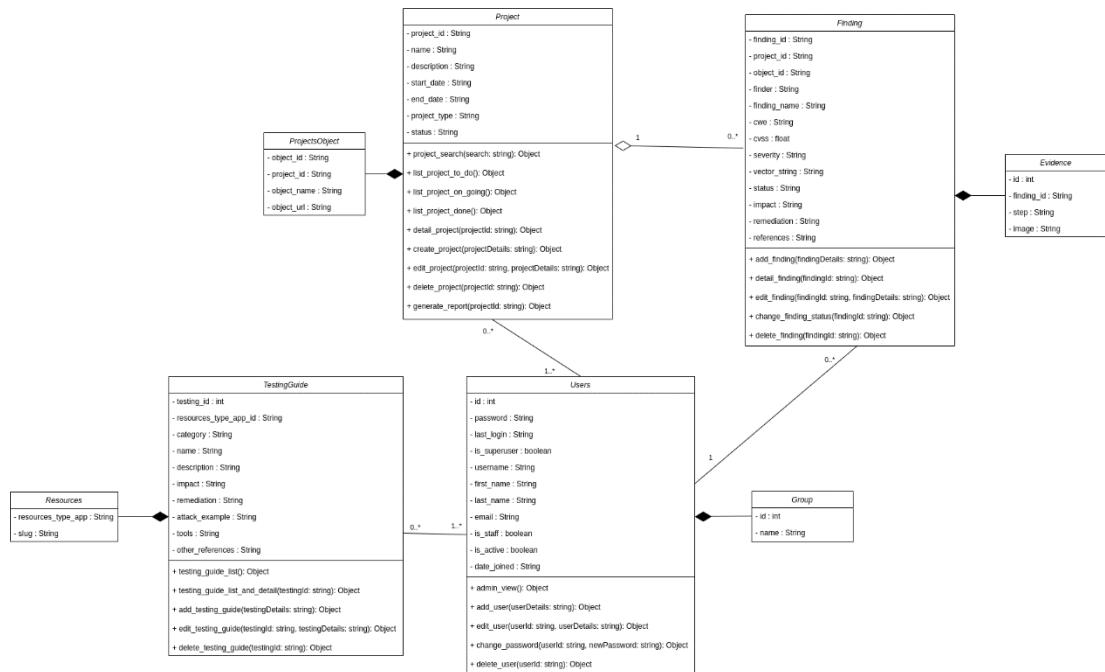
penghapusan. Setelah itu *actor* menekan tombol *delete*, lalu sistem akan memproses dan menghapus *testing guide* tersebut dari *database*.



**Gambar 3.29 Activity Diagram View Testing Guide**

Fitur *View Testing Guide* ini berguna bagi *Principal Penetration Tester* dan *Penetration Tester* untuk melihat *testing guide*. Pada fitur *View Testing Guide* ini, *actor* membuka halaman *list testing guide* yang nantinya sistem akan menampilkan semua *testing guide* yang ada pada halaman *testing guide*. Lalu *actor* akan memilih salah satu *testing guide* antara *web* atau *mobile*, setelah itu sistem akan menampilkan detail dari *testing guide* yang dipilih. Kemudian *actor* memilih kembali salah satu materi *testing guide* yang tersedia dan sistem akan menampilkan *testing guide* tersebut.

### 3.3.4 Class Diagram



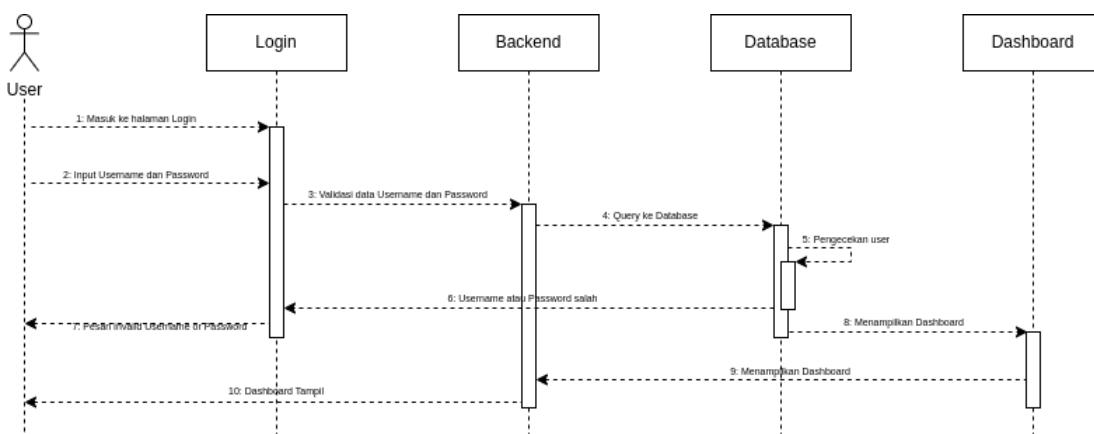
Gambar 3.30 Class Diagram

Dari *class diagram* diatas aplikasi *vulnerability management* memiliki delapan kelas yaitu kelas *Group* yang merupakan bagian dari kelas *Users*, kelas *Resources* yang merupakan bagian dari kelas *TestingGuide*, kelas *ProjectObject* yang merupakan bagian dari kelas *Project*, dan kelas *Evidence* yang merupakan bagian dari kelas *Finding*. Kelas *Users* dan *Group* adalah kelas yang menangani profil dan identifikasi dari pengguna seperti nama, *username*, *password* dan grup. Kelas ini juga mengelola aksi seperti melihat, menambah, mengubah atau menghapus pengguna. Untuk kelas *Resources* dan *TestingGuide*, kelas ini menangani fitur *Testing Guide* pada aplikasi seperti nama, deskripsi, *impact* dan *remediation*. Kelas ini juga mengelola aksi seperti melihat, menambah, mengubah atau menghapus *Testing Guide*. Kelas *ProjectObject* dan *Project* adalah kelas yang menangani *project* pada aplikasi seperti nama, deskripsi, tanggal mulai, tanggal selesai dan status *project*. Kelas ini juga mengelola aksi seperti melihat, menambah, mengubah, menghapus dan melakukan automasi pembuatan laporan pada *project* yang ada. Kelas *Evidence* dan *Finding* adalah kelas yang menangani temuan *vulnerability* pada suatu *project* seperti nama, penemu, kategori, *severity*, status dan juga bukti beserta gambar. Kelas ini juga mengelola aksi seperti

melihat, menambah, mengubah dan menghapus *finding* yang sudah ditemukan. Kelas *Users* mempunyai relasi dengan kelas *TestingGuide*, *Project* dan *Finding*, sedangkan untuk kelas *Project* memiliki relasi *aggregation* yang berarti mempunyai kelas *Finding*.

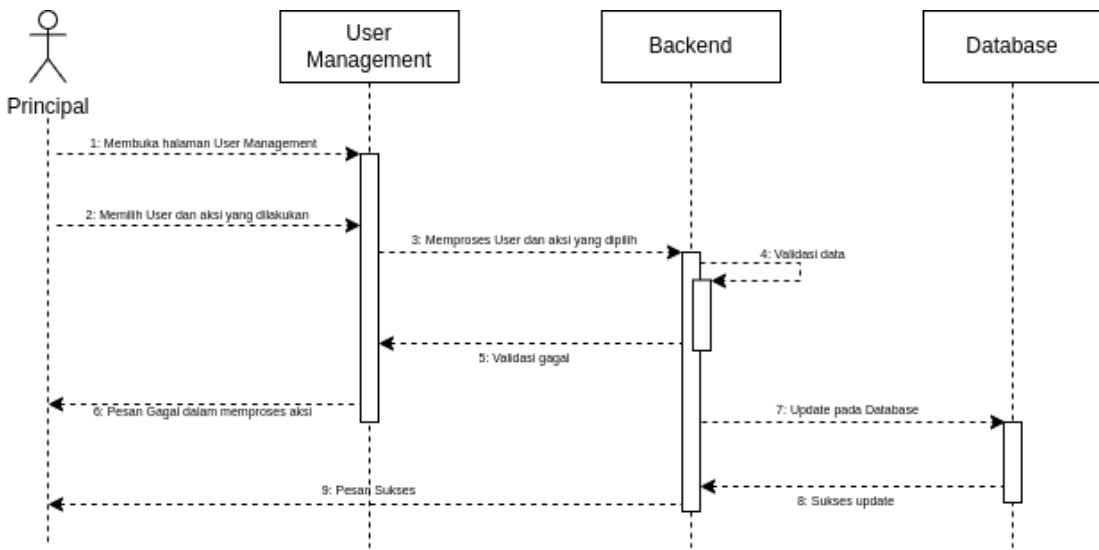
### 3.3.5 Sequence Diagram

Penulis juga membuat *sequence diagram* agar dapat menampilkan interaksi antar objek dalam dua dimensi:



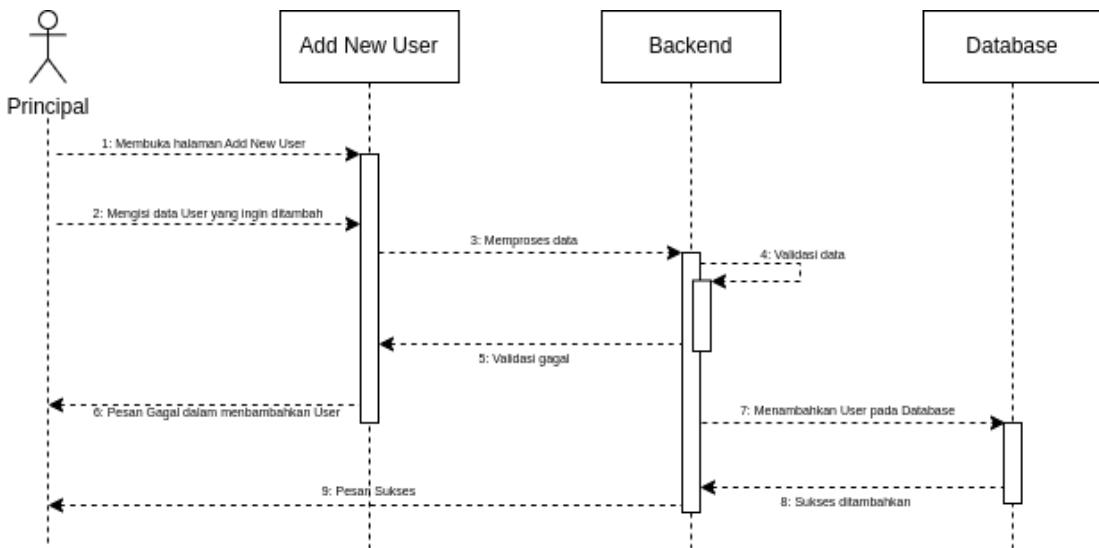
**Gambar 3.31 Sequence Diagram Login**

Gambar tersebut adalah *sequence diagram* pada fitur login sampai muncul dashboard dengan menampilkan interaksi antara *user*, *database*, autentikasi yang ada, dan dashboard *controller*. *User* mengisi *username* dan *password* mereka pada *login* yang akan mengirim *request* autentikasi melalui *backend* untuk verifikasi *credential* dari *user* dari *database*. Jika sukses, sistem akan menampilkan dashboard, namun jika gagal akan menampilkan pesan ke *user*.



**Gambar 3.32 Sequence Diagram Manage User**

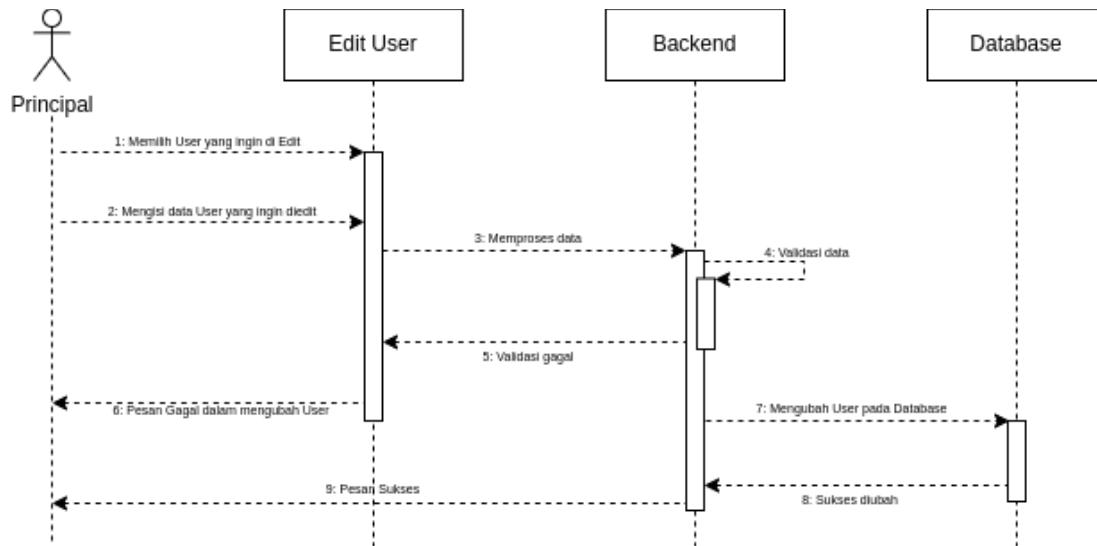
Gambar diatas menunjukkan urutan dalam mengelola *user* sampai menampilkan pesan dari aksi yang dilakukan. Sistem menampilkan halaman *user management*, *user* kemudian memilih aksi yang akan dilakukan pada salah satu *user*. *Backend* akan melakukan validasi dari aksi yang akan dilakukan, jika gagal akan menampilkan pesan gagal dalam memproses aksi, jika verifikasi berhasil aksi yang dilakukan akan diupdate pada *database* kemudian akan ditampilkan pesan sukses.



**Gambar 3.33 Sequence Diagram Add User**

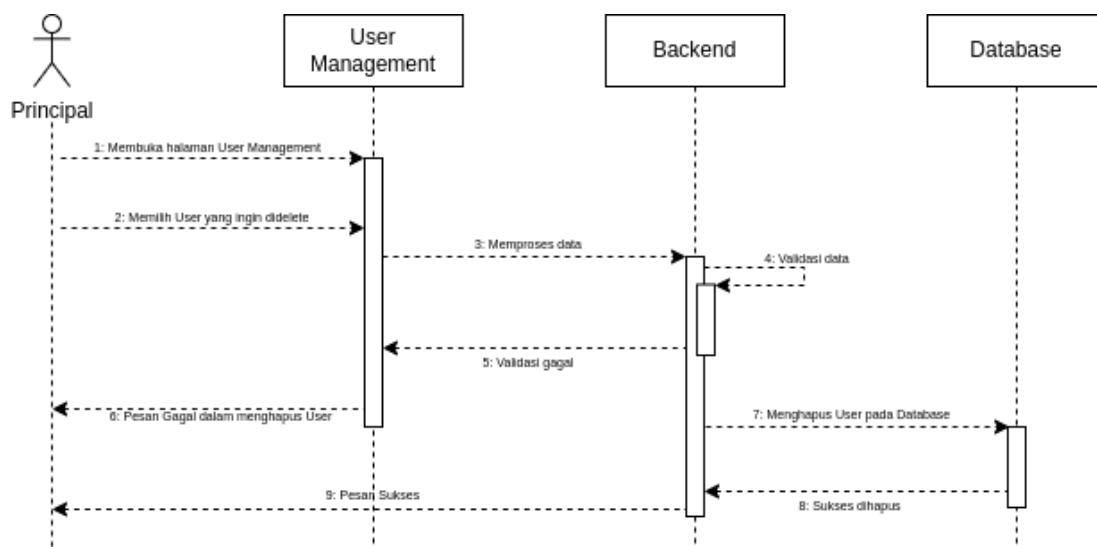
Gambar diatas menunjukkan urutan proses pada fitur *Add User*. Dimulai dari sistem menampilkan halaman *Add New User* kemudian mengisi detail mengenai *user* yang akan ditambahkan. Sistem akan melakukan validasi data *user*. Jika verifikasi

gagal, sistem akan menampilkan pesan gagal, jika validasi berhasil, sistem akan menambahkan data *user* ke *database* kemudian menampilkan pesan jika *user* telah ditambahkan.



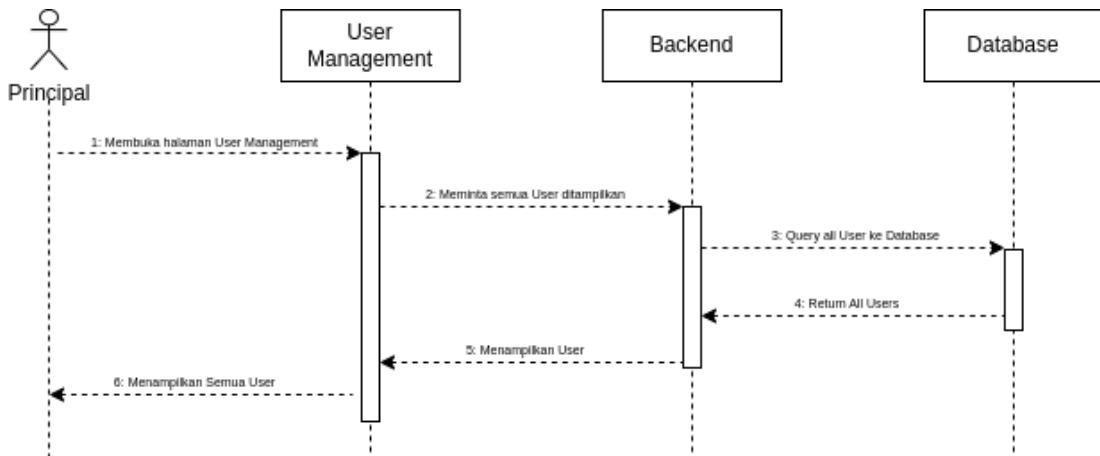
**Gambar 3.34 Sequence Diagram Edit User**

Gambar diatas menunjukkan urutan proses pada fitur *Edit User*. Dimulai dari Principal memilih *user* yang akan di-edit, memilih aksi *Edit User*, dan mengisi detail yang ingin diganti. Sistem akan memproses dan melakukan validasi data yang diberikan, jika verifikasi gagal akan menampilkan pesan gagal, namun jika berhasil sistem akan melakukan *update* ke *database* dan menampilkan pesan sukses.



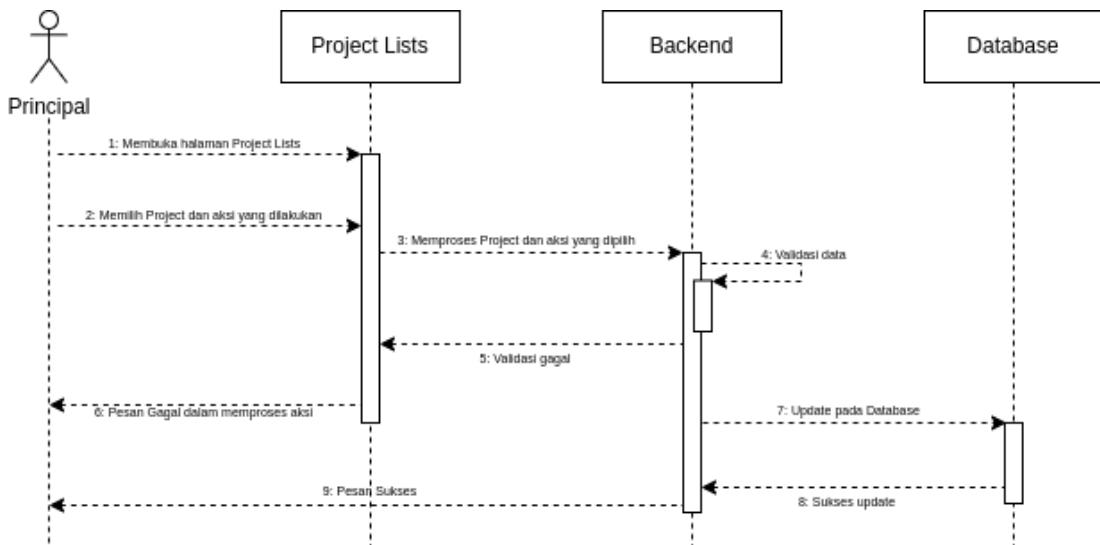
**Gambar 3.35 Sequence Diagram Delete User**

Gambar diatas menunjukkan urutan proses pada fitur *Delete User*. Dimulai dari Principal memilih *user* yang akan dihapus dan memilih aksi *Delete User*. Sistem akan memproses dan melakukan validasi data yang diberikan, jika verifikasi gagal akan menampilkan pesan gagal, namun jika berhasil sistem akan menghapus data *user* dari *database* dan menampilkan pesan sukses.



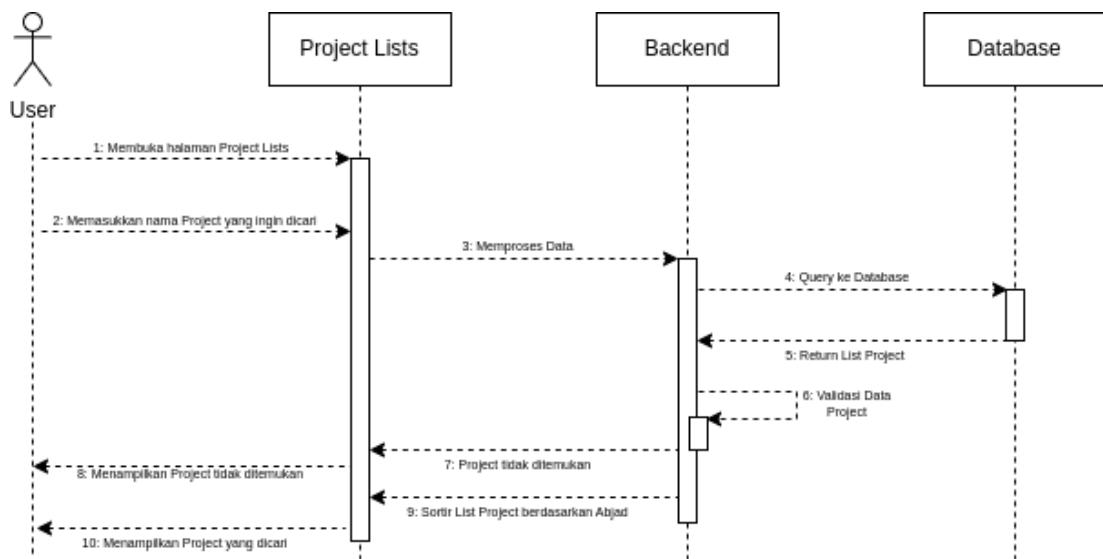
**Gambar 3.36 Sequence Diagram View User**

Gambar diatas menunjukkan urutan proses dalam *View User*. Dengan membuka halaman *User Management* kemudian *request* untuk menampilkan seluruh *user*. Sistem akan melakukan *query* ke *database* untuk menampilkan seluruh *user* dan akan ditampilkan pada halaman *user management*.



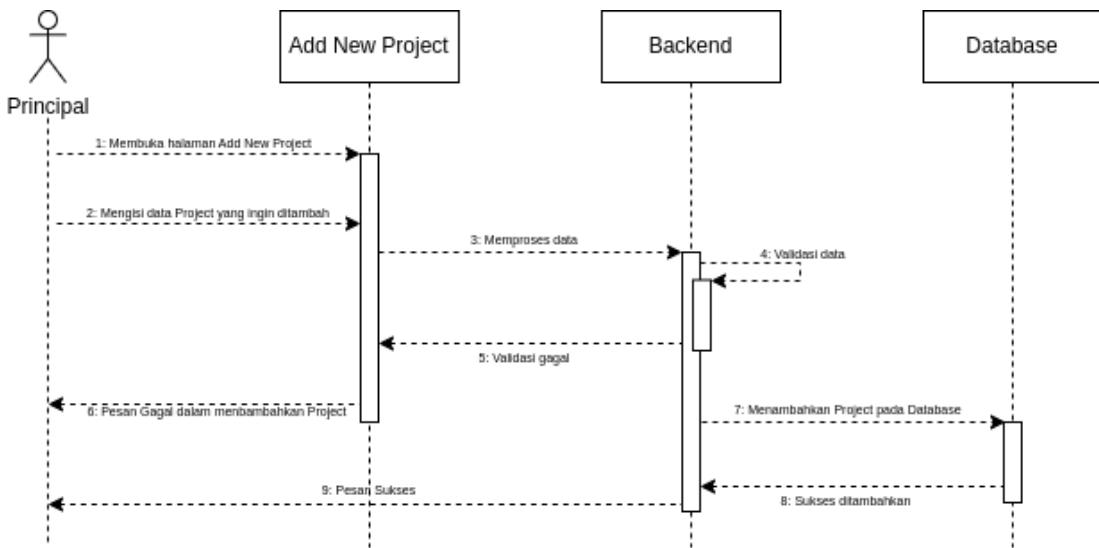
**Gambar 3.37 Sequence Diagram Manage Project**

Gambar diatas menunjukkan urutan proses dalam mengelola *project* sampai menampilkan pesan dari aksi yang dilakukan. Sistem menampilkan halaman *project lists*, principal kemudian memilih aksi yang akan dilakukan pada salah satu *project*. Sistem akan melakukan validasi dari aksi yang dilakukan, jika gagal akan menampilkan pesan gagal dalam memproses aksi, jika verifikasi berhasil aksi yang dilakukan akan diupdate pada *database* kemudian akan ditampilkan pesan sukses.



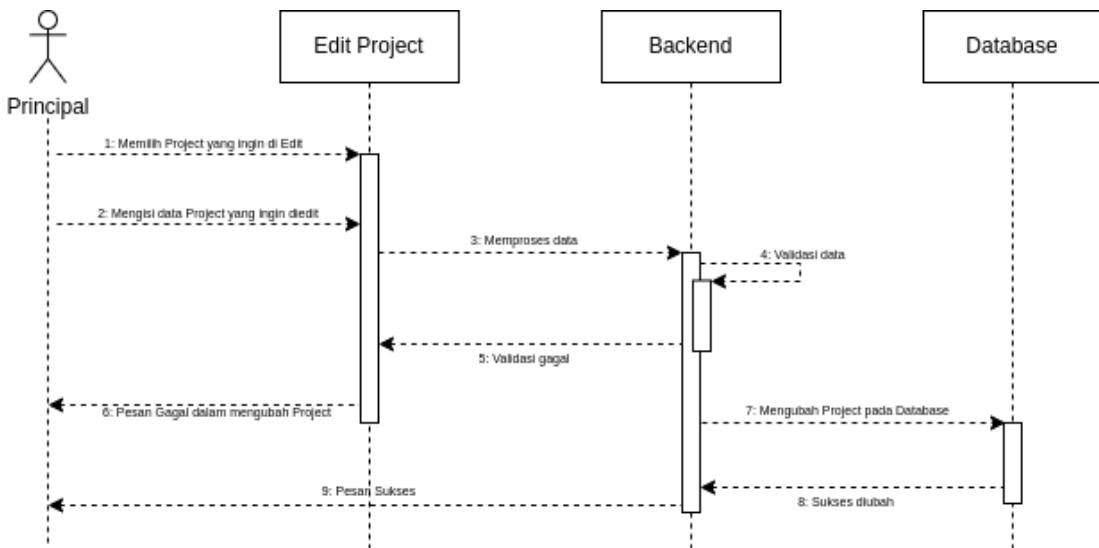
**Gambar 3.38 Sequence Diagram Search Project**

Gambar diatas menunjukkan urutan proses dalam mencari *project*. Dimulai dari membuka halaman *project lists*, *user* kemudian memasukkan nama *project* yang akan dicari, sistem kemudian akan memproses data dan melakukan *query* ke *database* kemudian mengembalikan hasilnya. Selanjutnya ada validasi data dari *project*, jika tidak ditemukan maka akan ditampilkan pesan *project* tidak ditemukan, jika ditemukan maka akan menampilkan daftar *project* berdasarkan abjad.



**Gambar 3.39 Sequence Diagram Add Project**

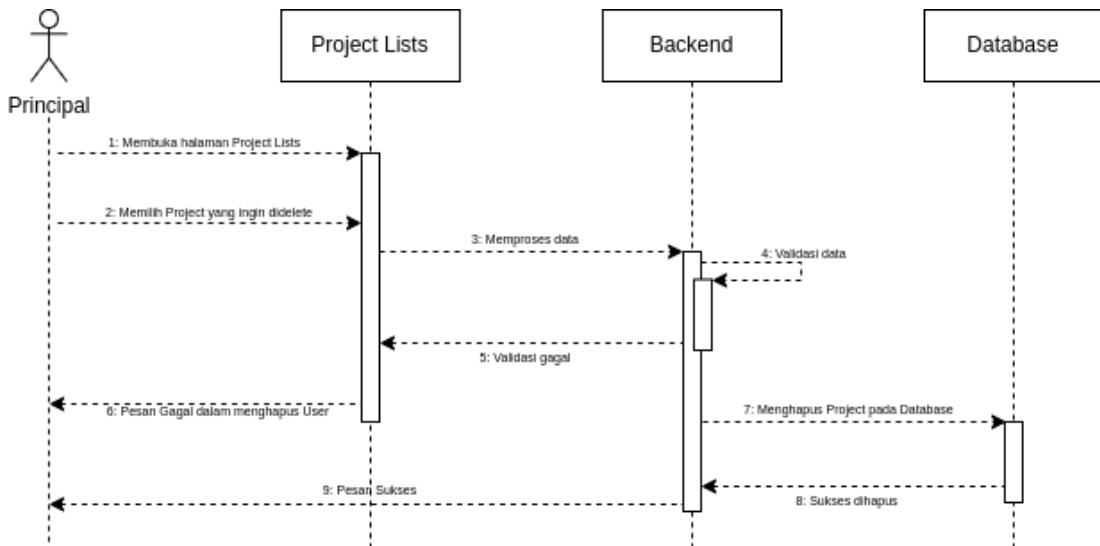
Gambar diatas menunjukkan urutan proses pada fitur *Add Project*. Dimulai dari sistem menampilkan halaman *Add New Project* kemudian mengisi detail mengenai *project* yang akan ditambahkan. Sistem akan melakukan validasi data *project*. Jika verifikasi gagal, sistem akan menampilkan pesan gagal, jika validasi berhasil, sistem akan menambahkan data *project* ke database kemudian menampilkan pesan jika *project* telah ditambahkan.



**Gambar 3.40 Sequence Diagram Edit Project**

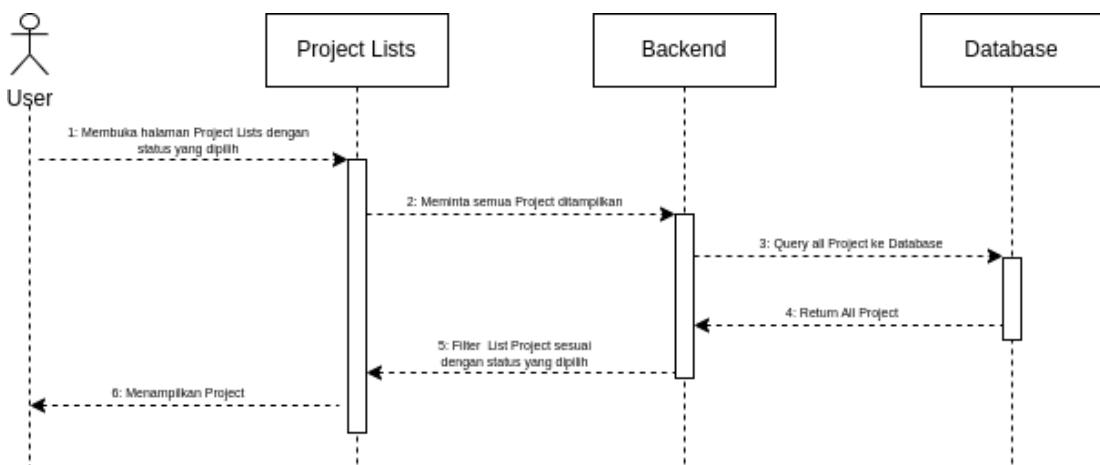
Gambar diatas menunjukkan urutan proses pada fitur *Edit Project*. Dimulai dari Principal memilih *project* yang akan di-edit, memilih aksi *Edit Project*, dan mengisi detail yang ingin diganti. Sistem akan memproses dan melakukan validasi data

yang diberikan, jika verifikasi gagal akan menampilkan pesan gagal, namun jika berhasil sistem akan melakukan *update* ke *database* dan menampilkan pesan sukses.



**Gambar 3.41 Sequence Diagram Delete Project**

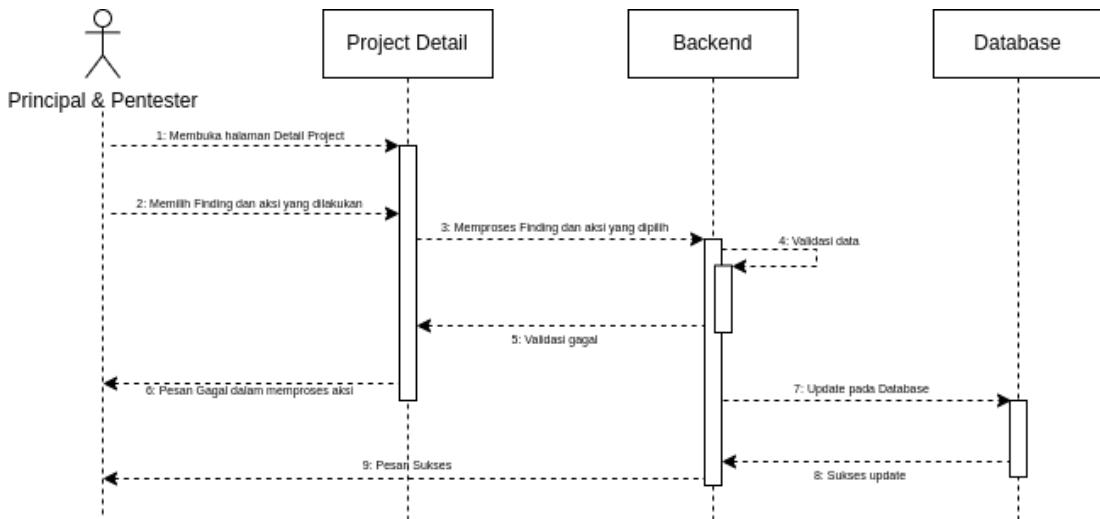
Gambar diatas menunjukkan urutan proses pada fitur *Delete Project*. Dimulai dari Principal memilih *project* yang akan dihapus dan memilih aksi *Delete Project*. Sistem akan memproses dan melakukan validasi data yang diberikan, jika verifikasi gagal akan menampilkan pesan gagal, namun jika berhasil sistem akan menghapus data *project* dari *database* dan menampilkan pesan sukses.



**Gambar 3.42 Sequence Diagram View Project**

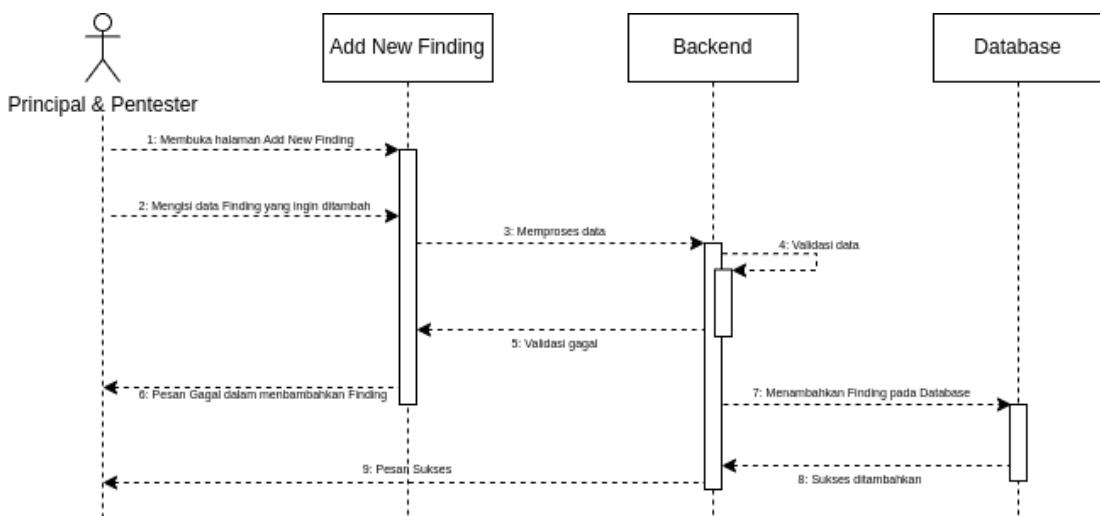
Gambar diatas menunjukkan urutan proses dalam *View Project*. Dengan membuka halaman *Project Lists* dan memilih status *project* kemudian *request* untuk menampilkan seluruh *project* dari status tersebut. Sistem akan melakukan *query* ke

*database* untuk mengembalikan seluruh *project* sesuai status dan akan ditampilkan pada halaman status *Project Lists*.



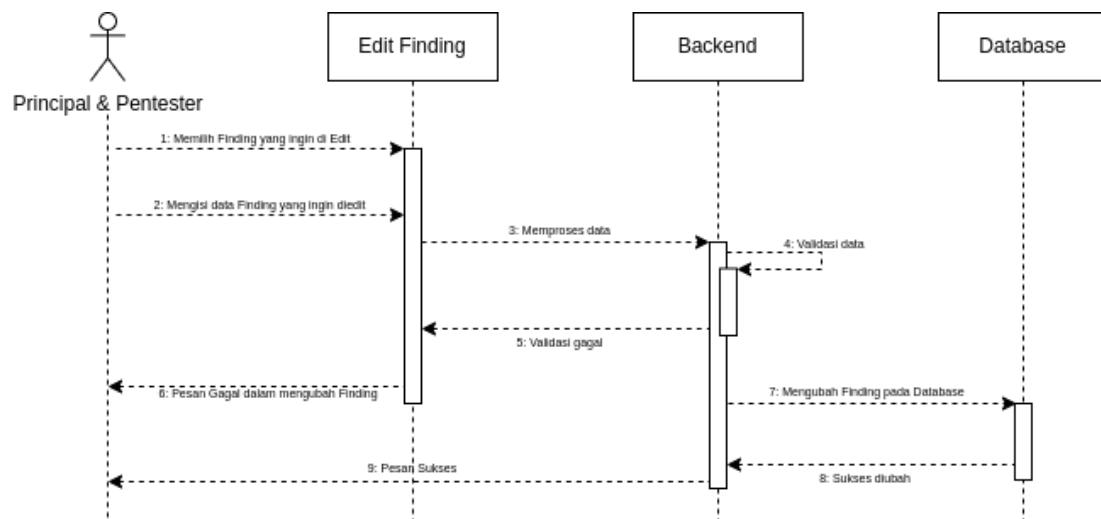
**Gambar 3.43 Sequence Diagram Manage Finding**

Gambar diatas menunjukkan urutan proses dalam mengelola *finfing* sampai menampilkan pesan dari aksi yang dilakukan. Sistem menampilkan halaman *detail project*, principal kemudian memilih aksi yang akan dilakukan pada salah satu *finding*. Sistem akan melakukan validasi dari aksi yang dilakukan, jika gagal akan menampilkan pesan gagal dalam memproses aksi, jika verifikasi berhasil aksi yang dilakukan akan diupdate pada *database* kemudian akan ditampilkan pesan sukses.



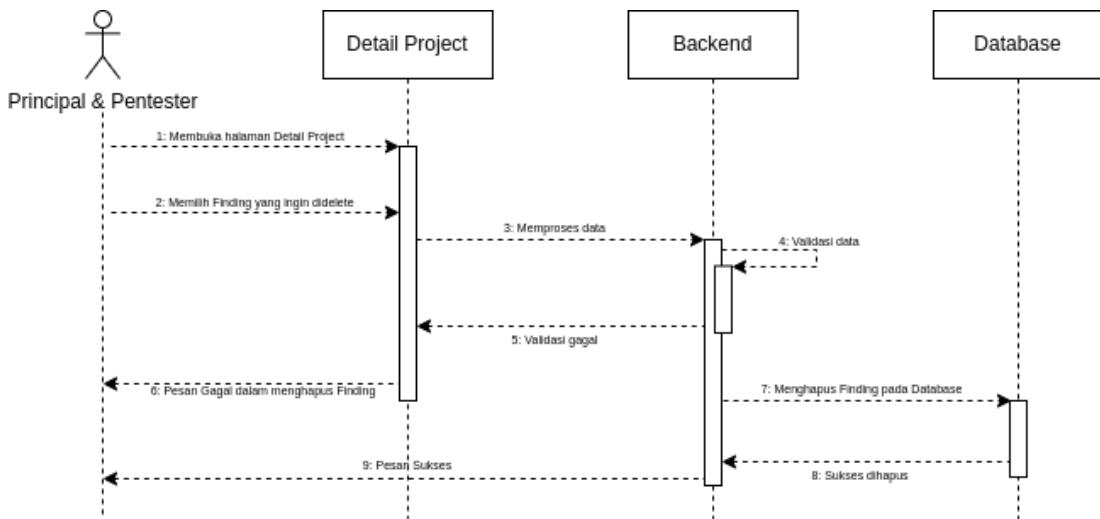
**Gambar 3.44 Sequence Diagram Add Finding**

Gambar diatas menunjukkan urutan proses pada fitur *Add Finding*. Dimulai dari sistem menampilkan halaman *Add New Finding* kemudian mengisi detail mengenai *finding* yang akan ditambahkan. Sistem akan melakukan validasi data *finding*. Jika verifikasi gagal, sistem akan menampilkan pesan gagal, jika validasi berhasil, sistem akan menambahkan data *finding* ke *database* kemudian menampilkan pesan jika *finding* telah ditambahkan.



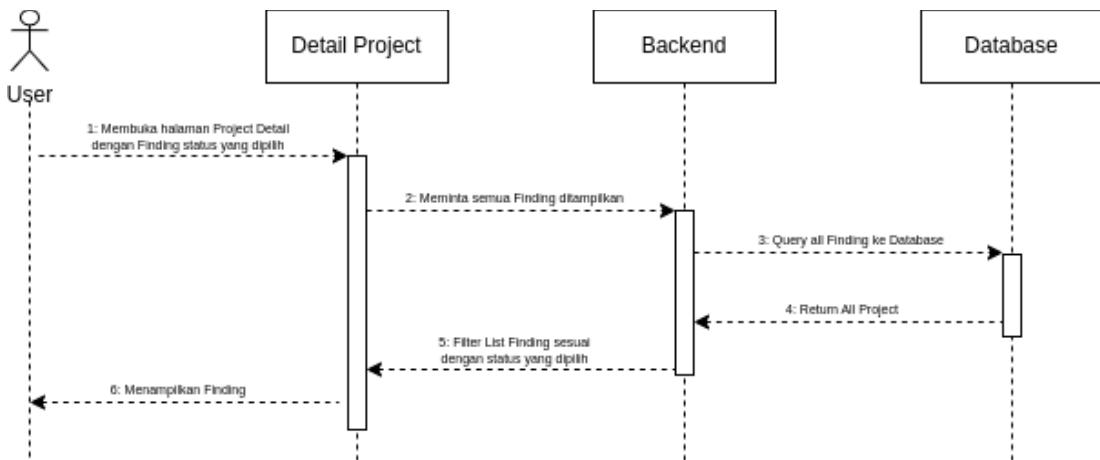
**Gambar 3.45 Sequence Diagram Edit Finding**

Gambar diatas menunjukkan urutan proses pada fitur *Edit Finding*. Dimulai dari sistem menampilkan halaman *detail project*, *principal* memilih *finding* yang akan di-edit, memilih aksi *Edit Finding*, dan mengisi detail yang ingin diganti. Sistem akan memproses dan melakukan validasi data yang diberikan, jika verifikasi gagal akan menampilkan pesan gagal, namun jika berhasil sistem akan melakukan *update* ke *database* dan menampilkan pesan sukses.



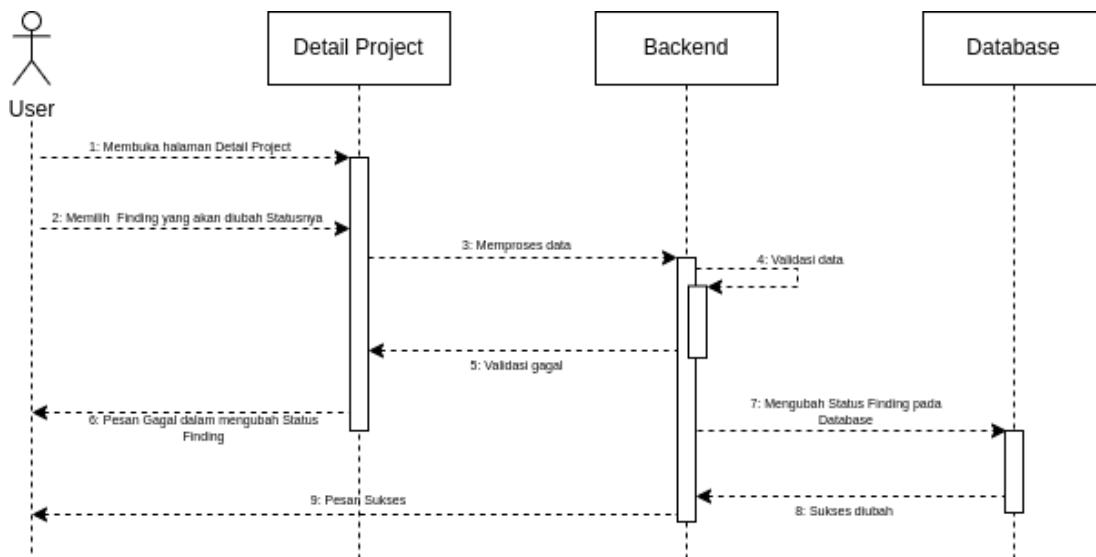
**Gambar 3.46 Sequence Diagram Delete Finding**

Gambar diatas menunjukkan urutan proses pada fitur Delete User. Dimulai dari sistem menampilkan halaman *detail project*, principal memilih user yang akan dihapus dan memilih aksi Delete User. Sistem akan memproses dan melakukan validasi data yang diberikan, jika verifikasi gagal akan menampilkan pesan gagal, namun jika berhasil sistem akan menghapus data user dari database dan menampilkan pesan sukses.



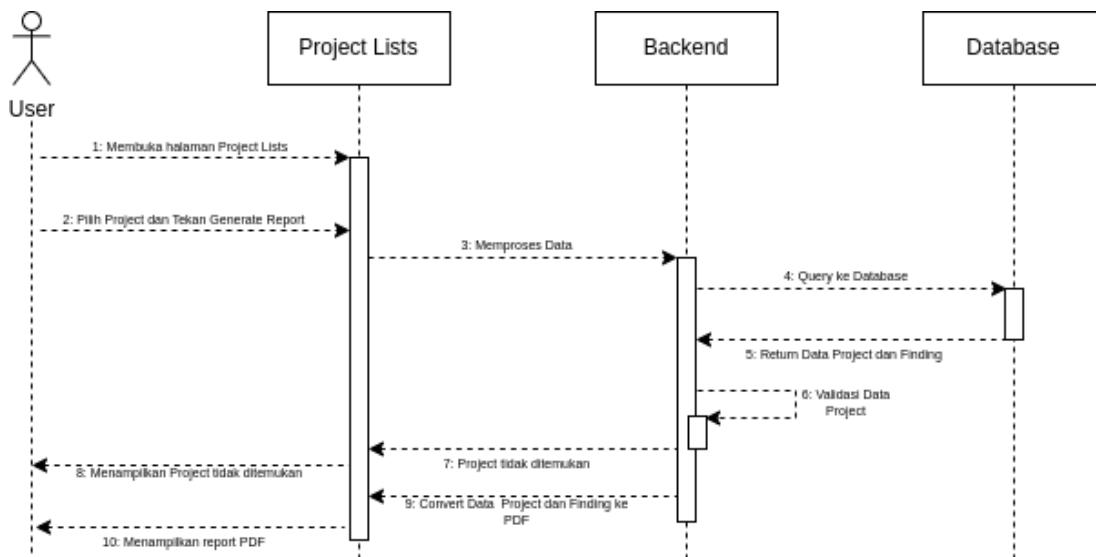
**Gambar 3.47 Sequence Diagram View Finding**

Gambar diatas menunjukkan urutan proses dalam *View Finding*. Dengan membuka halaman *Detail Project* dan memilih status *finding* kemudian *request* untuk menampilkan seluruh *finding* dari status tersebut. Sistem akan melakukan *query* ke *database* untuk mengembalikan seluruh *finding* sesuai status dan akan ditampilkan pada halaman status *Detail Project*.



**Gambar 3.48 Sequence Diagram Change Finding Status**

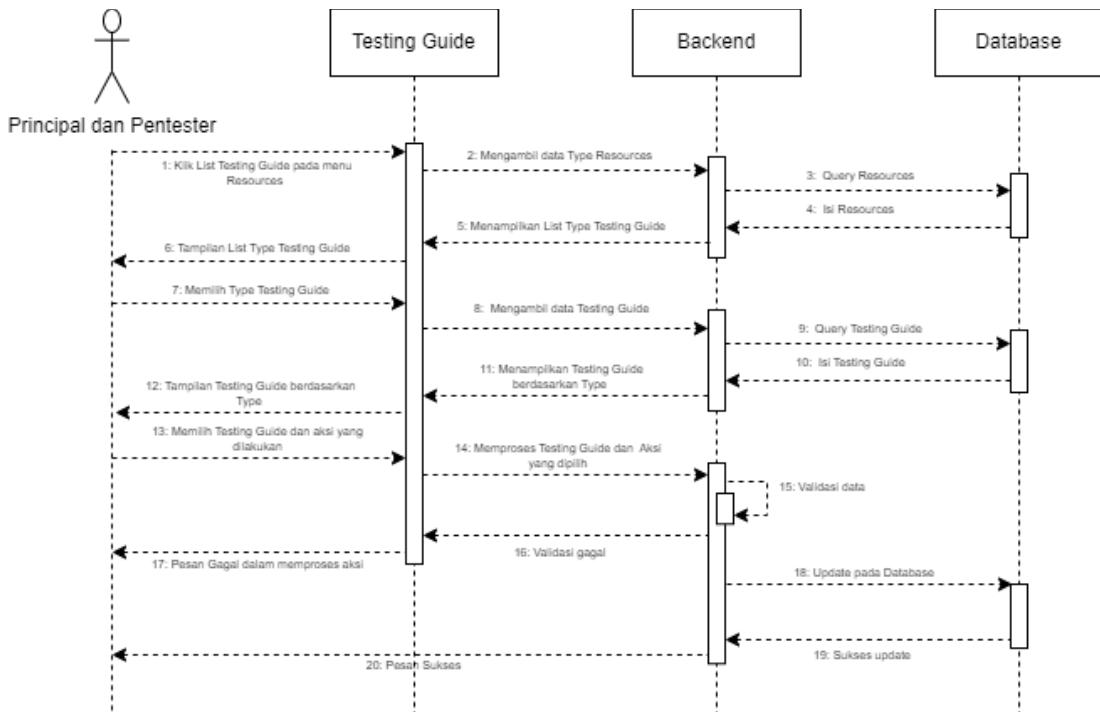
Gambar diatas menunjukkan urutan proses dalam mengganti status *finding*. Dengan membuka halaman *Detail Project* dan memilih status *finding* kemudian memilih *finding* yang akan diganti status. Sistem akan memproses dan memvalidasi data, jika validasi gagal akan menampilkan pesan gagal, namun jika berhasil akan di *update* ke *database* dan menampilkan pesan jika status *finding* berhasil diubah.



**Gambar 3.49 Sequence Diagram Generate Report**

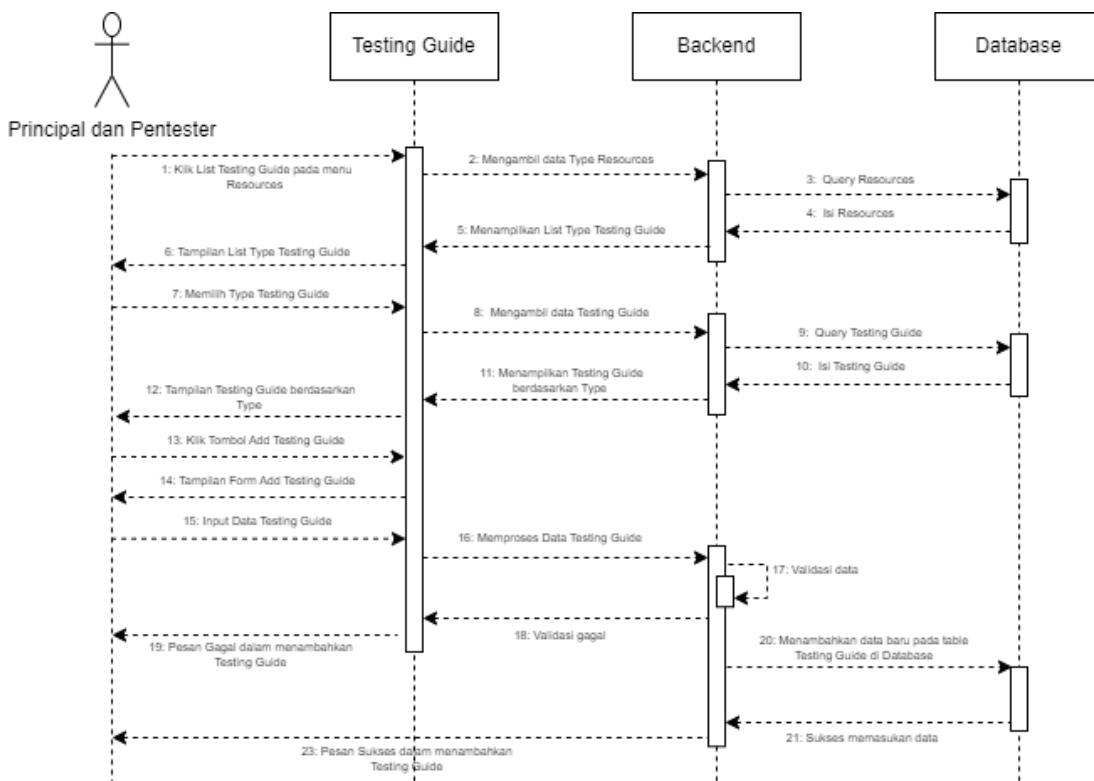
Gambar diatas menunjukkan urutan proses pada fitur *Generate Report*. Dengan membuka halaman *Project Lists* kemudian memilih *project* dan aksi *generate report*. Sistem akan memproses data dan melakukan *query* ke *database*, jika data *project* tidak

ditemukan maka akan menampilkan pesan gagal, namun jika ditemukan akan dilakukan validasi data kemudian di *generate* ke dalam file PDF yang akan otomatis ditampilkan.



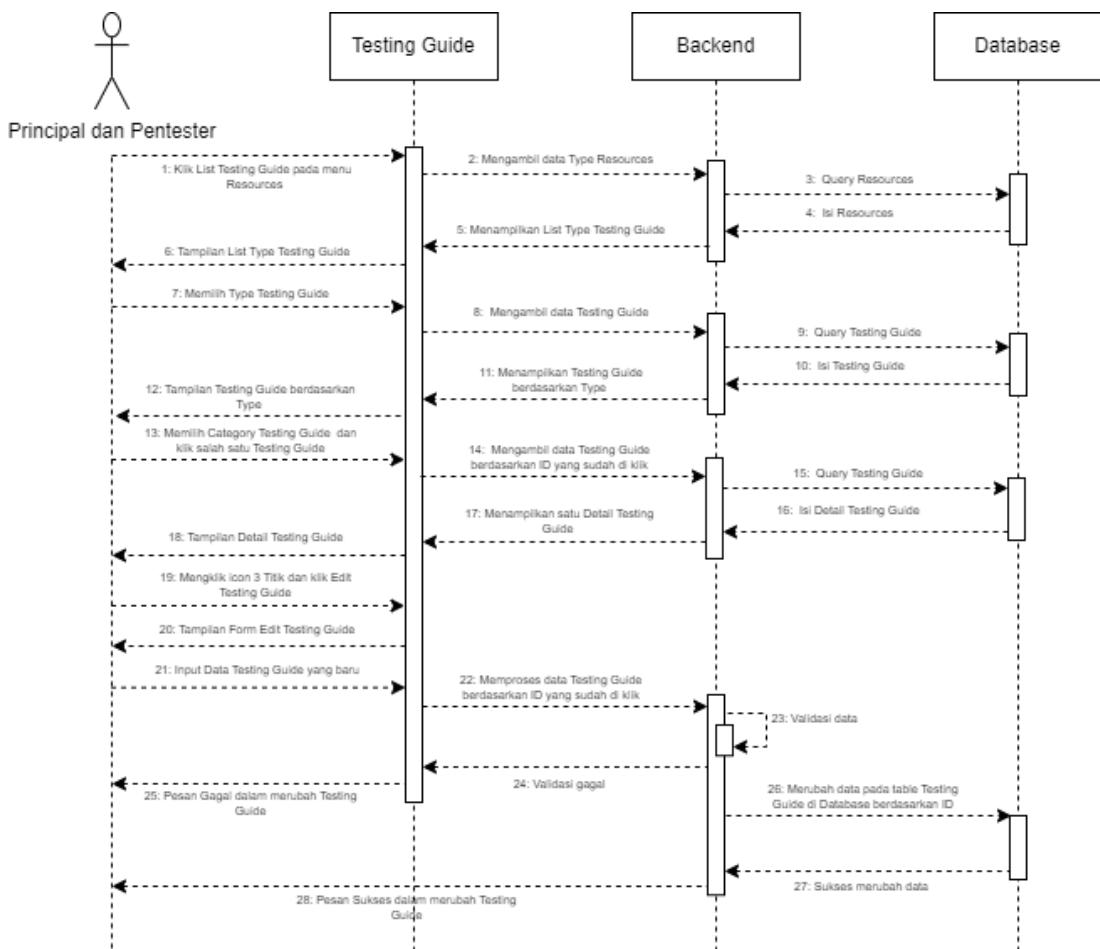
**Gambar 3.50 Sequence Diagram Manage Testing Guide**

Gambar diatas menunjukkan urutan proses dalam mengelola *Testing Guide*. Dengan membuka halaman *Resources* kemudian memilih tipe *resources*, sistem akan memproses dan melakukan *query* ke *database* dan menampilkan tipe *testing guide* kemudian memilih tipe *testing guide*, sistem akan memproses dan melakukan *query* ke *database* dan menampilkan daftar *testing guide*. *Principal* dan *pentester* kemudian memilih aksi yang akan dilakukan pada salah satu *testing guide*. Sistem akan melakukan validasi dari aksi yang dilakukan, jika gagal akan menampilkan pesan gagal dalam memproses aksi, jika berhasil aksi yang dilakukan akan diupdate pada *database* kemudian akan ditampilkan pesan sukses.



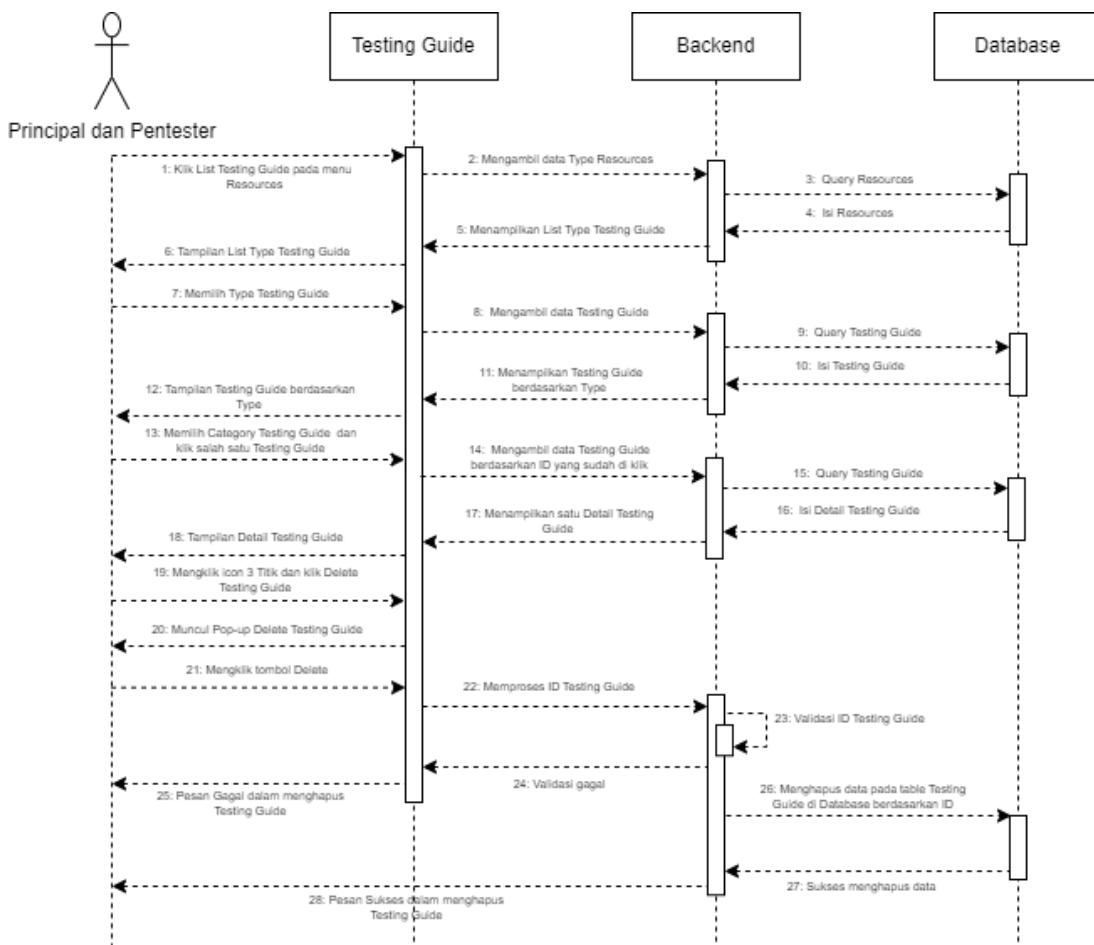
**Gambar 3.51 Sequence Diagram Add Testing Guide**

Gambar diatas menunjukkan urutan proses dalam menambahkan *Testing Guide*. Dengan membuka halaman *Resources* kemudian memilih tipe *resources*, sistem akan memproses dan melakukan *query* ke *database* dan menampilkan tipe *testing guide* kemudian memilih tipe *testing guide*, sistem akan memproses dan melakukan *query* ke *database* dan menampilkan daftar *testing guide*. *Principal* dan *pentester* kemudian menekan tombol *Add Testing Guide* dan sistem akan menampilkan *form Testing Guide*. Setelah itu *principal* dan *pentester* menginputkan data *testing guide* yang ingin ditambahkan. Sistem akan melakukan validasi dari data yang dimasukkan, jika gagal akan menampilkan pesan gagal dalam menambahkan *testing guide*, jika validasi berhasil yang dilakukan selanjutnya adalah penambahan *testing guide* pada *database* kemudian akan ditampilkan pesan sukses.



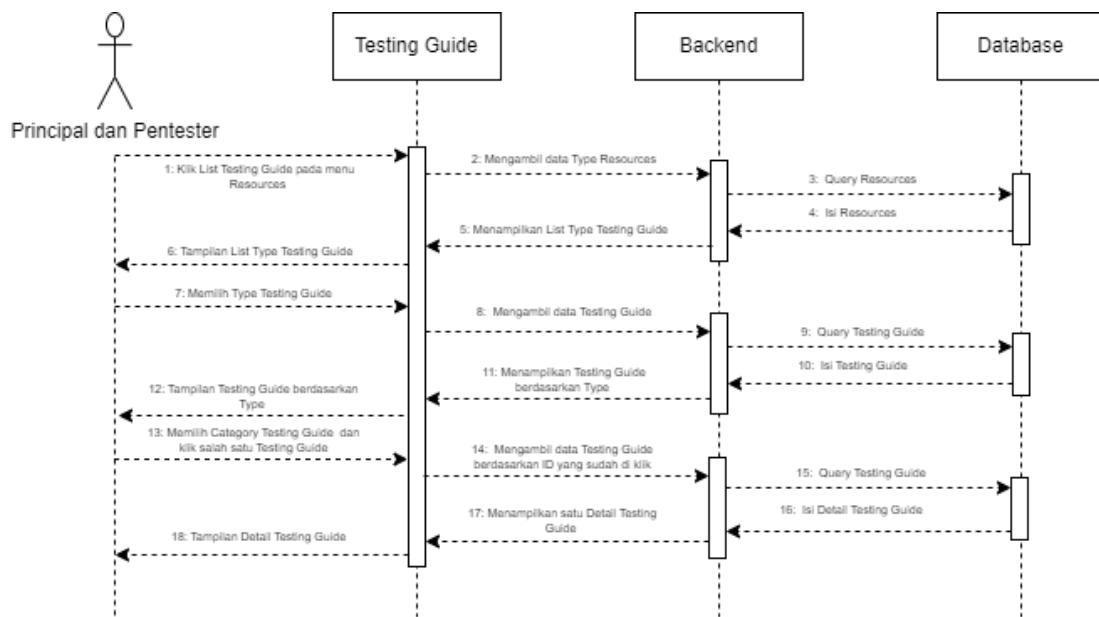
**Gambar 3.52 Sequence Diagram Edit Testing Guide**

Gambar diatas menunjukkan urutan proses dalam mengubah sebuah *Testing Guide*. Dengan membuka halaman *Resources* kemudian memilih tipe *resources*, sistem akan memproses dan melakukan *query* ke *database* dan menampilkan tipe *testing guide* kemudian memilih tipe *testing guide*, sistem akan memproses dan melakukan *query* ke *database* dan menampilkan daftar *testing guide*. *Principal* dan *pentester* kemudian dapat memilih salah satu *testing guide* dan menekan tombol *icon* titik tiga lalu klik *Edit Testing Guide*, maka sistem akan menampilkan *form* *Edit Testing Guide*. Setelah itu *principal* dan *pentester* menginputkan data *testing guide* yang ingin diubah. Sistem akan melakukan validasi dari data yang dimasukkan, jika gagal akan menampilkan pesan gagal dalam mengubah *testing guide*, jika validasi berhasil yang dilakukan selanjutnya adalah perubahan data *testing guide* pada *database* kemudian akan ditampilkan pesan sukses.



**Gambar 3.53 Sequence Diagram Delete Testing Guide**

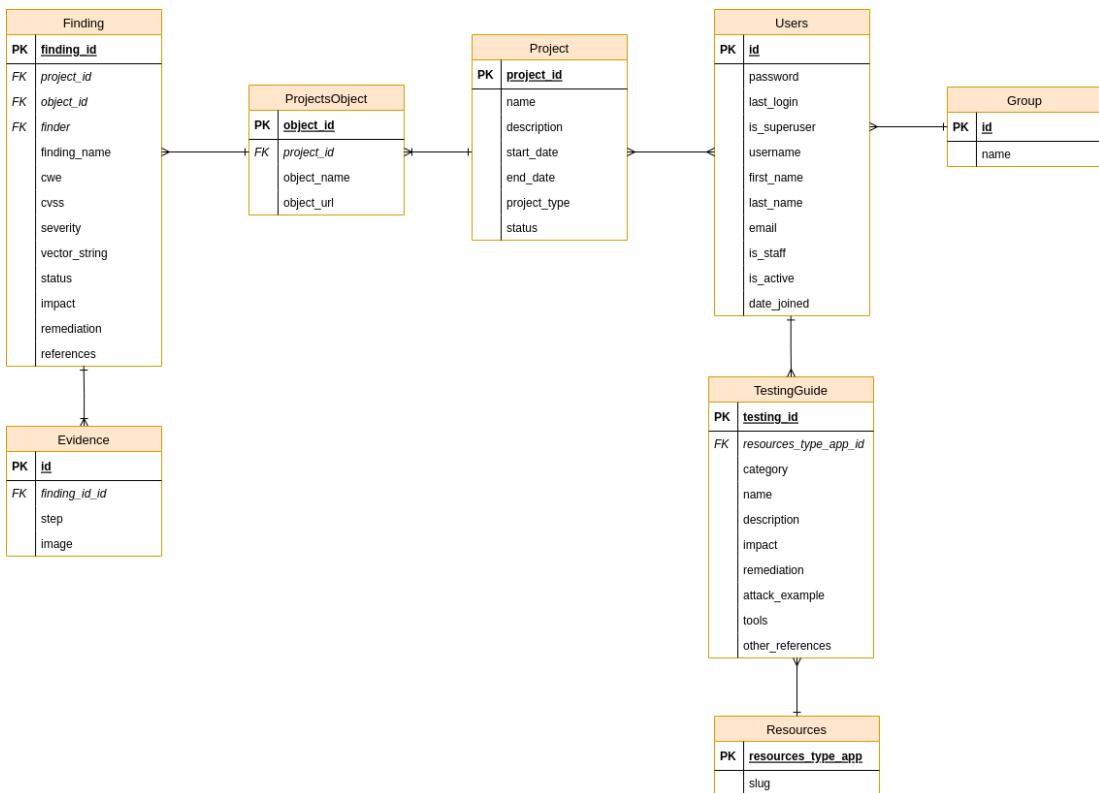
Gambar diatas menunjukkan urutan proses dalam mengubah sebuah *Testing Guide*. Dengan membuka halaman *Resources* kemudian memilih tipe *resources*, sistem akan memproses dan melakukan *query* ke *database* dan menampilkan tipe *testing guide* kemudian memilih tipe *testing guide*, sistem akan memproses dan melakukan *query* ke *database* dan menampilkan daftar *testing guide*. *Principal* dan *pentester* kemudian dapat memilih salah satu *testing guide* dan menekan tombol *icon titik tiga* lalu klik *Delete*, maka sistem akan memproses *id testing guide* yang dipilih. Sistem akan melakukan validasi dari data yang dimasukkan, jika gagal akan menampilkan pesan gagal dalam menghapus *testing guide*, jika validasi berhasil yang dilakukan selanjutnya adalah penghapusan *testing guide* pada *database* kemudian akan ditampilkan pesan sukses.



**Gambar 3.54 Sequence Diagram View Testing Guide**

Gambar diatas menunjukkan urutan proses dalam mengubah sebuah *Testing Guide*. Dengan membuka halaman *Resources* kemudian memilih tipe *resources*, sistem akan memproses dan melakukan *query* ke *database* dan menampilkan tipe *testing guide* kemudian memilih tipe *testing guide*, sistem akan memproses dan melakukan *query* ke *database* dan menampilkan daftar *testing guide*. *Principal* dan *pentester* kemudian dapat memilih salah satu *testing guide* untuk melihat detail dari *testing guide*.

### 3.3.6 Entity Relationship Diagram



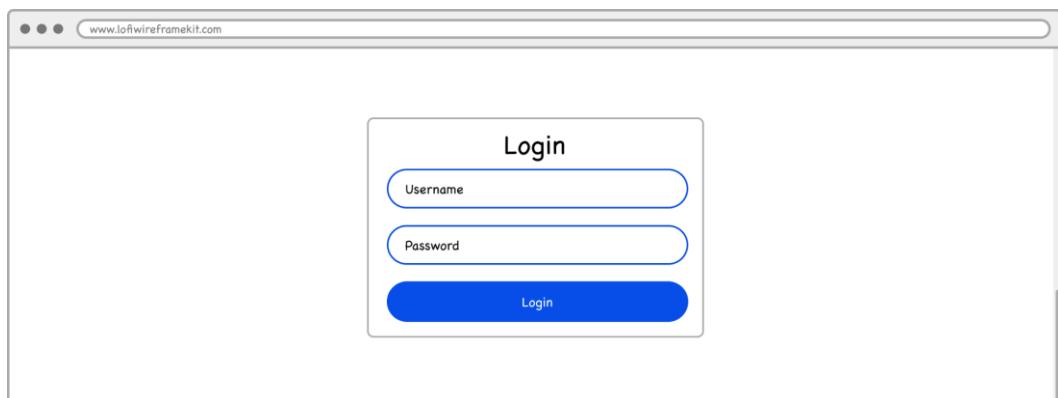
**Gambar 3.55 Entity Relationship Diagram**

Dari *entity relationship diagram* diatas aplikasi *vulnerability management* memiliki delapan *entity* yang direpresentasikan sebagai *table* pada *database*. *Entity Users* memiliki relasi *many to one* dengan *entity Group* yang berarti setiap pengguna hanya bisa memiliki satu grup atau *role* dan beberapa pengguna bisa memiliki grup atau *role* yang sama. *Entity Users* juga berelasi *one to many* dengan *entity TestingGuide* yang berarti setiap pengguna bisa memiliki banyak *testing guide*. *Entity TestingGuide* memiliki relasi *many to one* dengan *entity Resources* yang merupakan tipe dari *testing guide*. *Entity Users* juga memiliki relasi *many to many* dengan *entity Project* yang berarti setiap pengguna bisa mengikuti banyak *project* dan banyak pengguna bisa tergabung dalam *project* yang sama. *Entity Project* memiliki relasi *one to one or many* dengan *entity ProjectsObject* yang berarti setiap *project* bisa memiliki banyak *object penetration testing*. *Entity ProjectsObject* memiliki relasi *one to many* dengan *entity Finding* yang berarti setiap *object penetration testing* memiliki banyak *finding* atau temuan *vulnerability*. *Entity Finding* mempunyai relasi *one to one or many* dengan *entity Evidence*.

*Evidence* yang berarti setiap *finding* bisa memiliki satu atau banyak dari bukti yang disertakan.

### 3.3.7 User Interface

Perancangan *User interface* pada aplikasi Vulnerability Management menggunakan *High Fidelity Wireframe*. Penulis menggunakan komponen Lo-fi Wireframe Kit dibuat oleh Dave Whitley dari Community Figma. Semua rancangan gambar akan ditampilkan berdasarkan *role* masing-masing. Untuk tampilan hasil *Generate Report*, tidak ada perancangan gambar karena akan langsung ditampilkan dalam bentuk PDF sesuai dengan *web browser* yang dipakai oleh *user*.



**Gambar 3.56 Login Page**

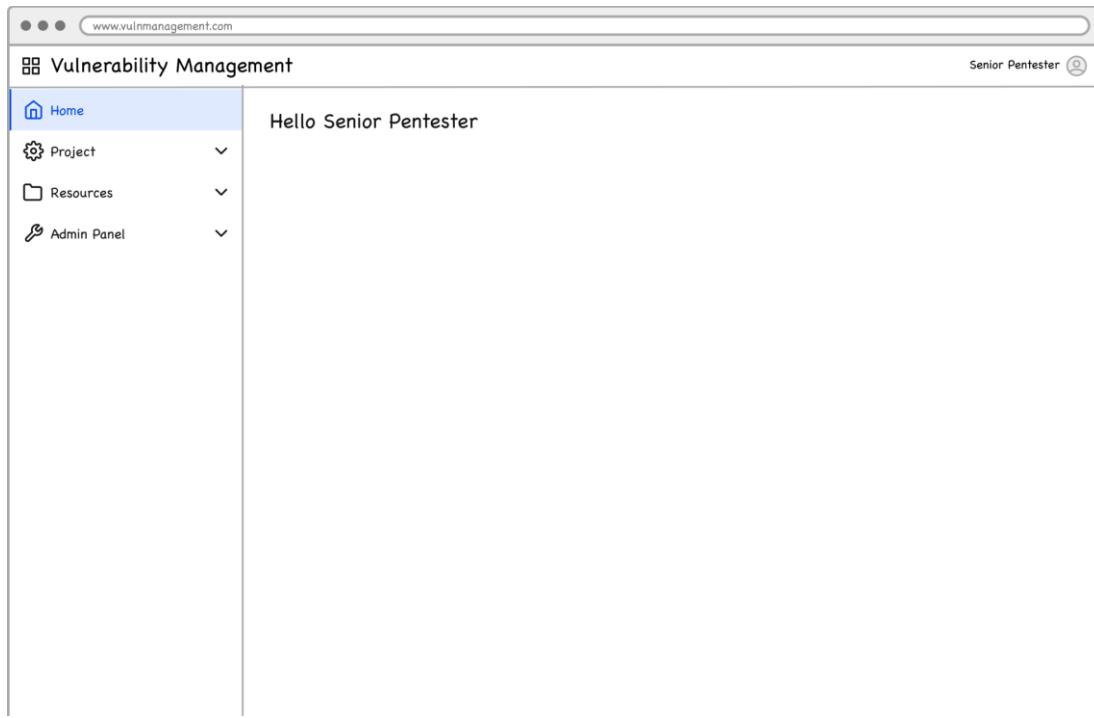
Halaman pertama yang akan ditampilkan adalah *Login Page*, *user* dapat memasukan *username* dan *password* untuk menentukan *role user*.

### 3.3.6.1 Home Page

Tampilan *Home Page* untuk *role senior penetration tester, junior penetration tester*, dan *client*. Aplikasi akan menampilkan *Home Page* setelah pengguna sudah sukses *login* ke dalam aplikasi menggunakan *username* dan *password* yang benar. *Home Page* hanya akan menampilkan ucapan selamat datang kepada user yang sudah berhasil login. Ada beberapa bagian yang akan selalu tetap yaitu bagian atas (*top bar*) dan bagian kiri (*side bar*), bagian atas akan menampilkan nama aplikasi, *search project* setelah klik *project lists*, dan *profile user*, bagian kiri akan menampilkan semua fitur yang

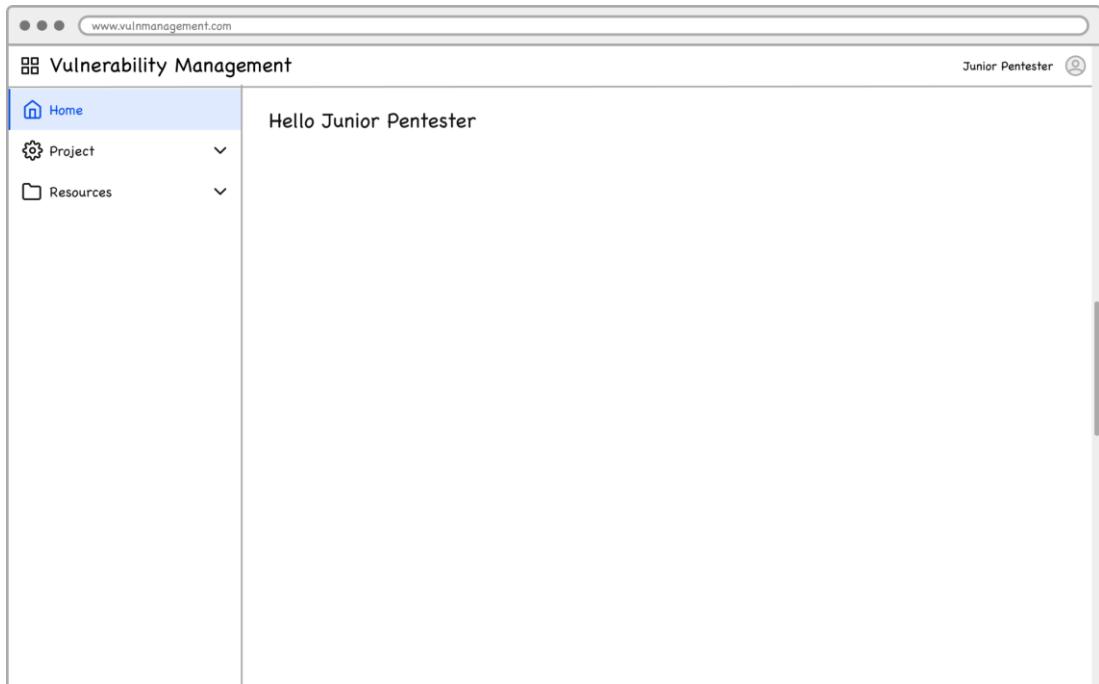
dapat digunakan dari *role user* tersebut. Dari ketiga *role* tersebut memiliki perbedaan tampilan di bagian kiri.

Tampilan *Home Page* untuk *senior penetration tester*, pada bagian kiri terdapat fitur *Home*, *Project*, *Resources*, dan *Admin Panel*.



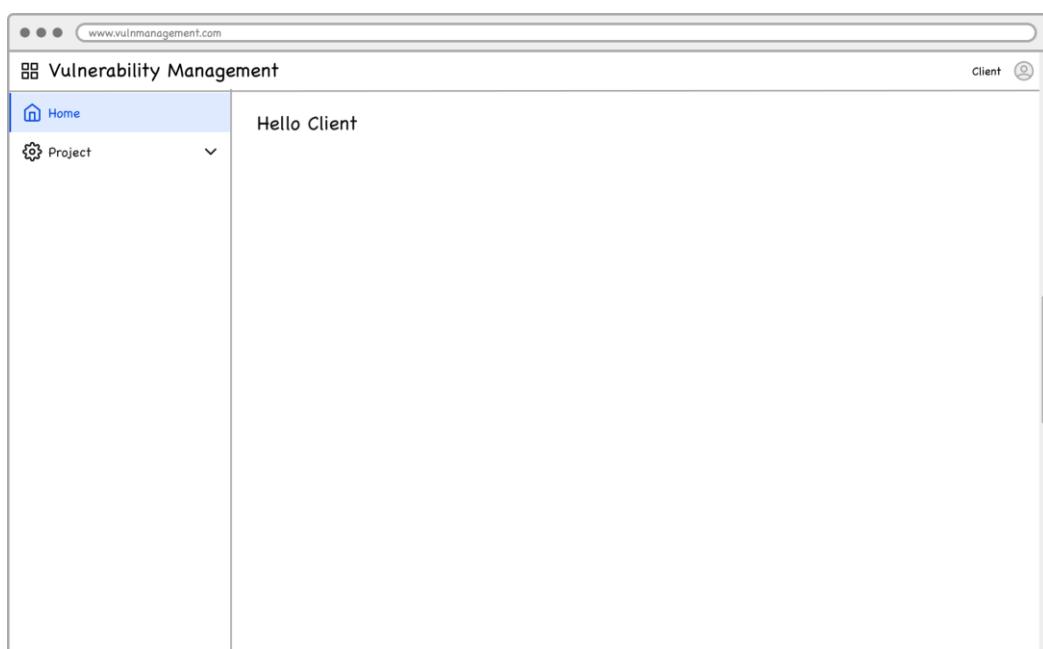
**Gambar 3.57 Home Page ( senior penetration tester )**

Tampilan *Home Page* untuk *junior penetration tester*, pada bagian kiri terdapat fitur *Home*, *Project*, dan *Resources*.



**Gambar 3.58 Home Page ( junior penetration tester )**

Tampilan *Home Page* untuk *client*, pada bagian kiri terdapat fitur *Home* dan *Project*.



**Gambar 3.59 Home Page ( client )**

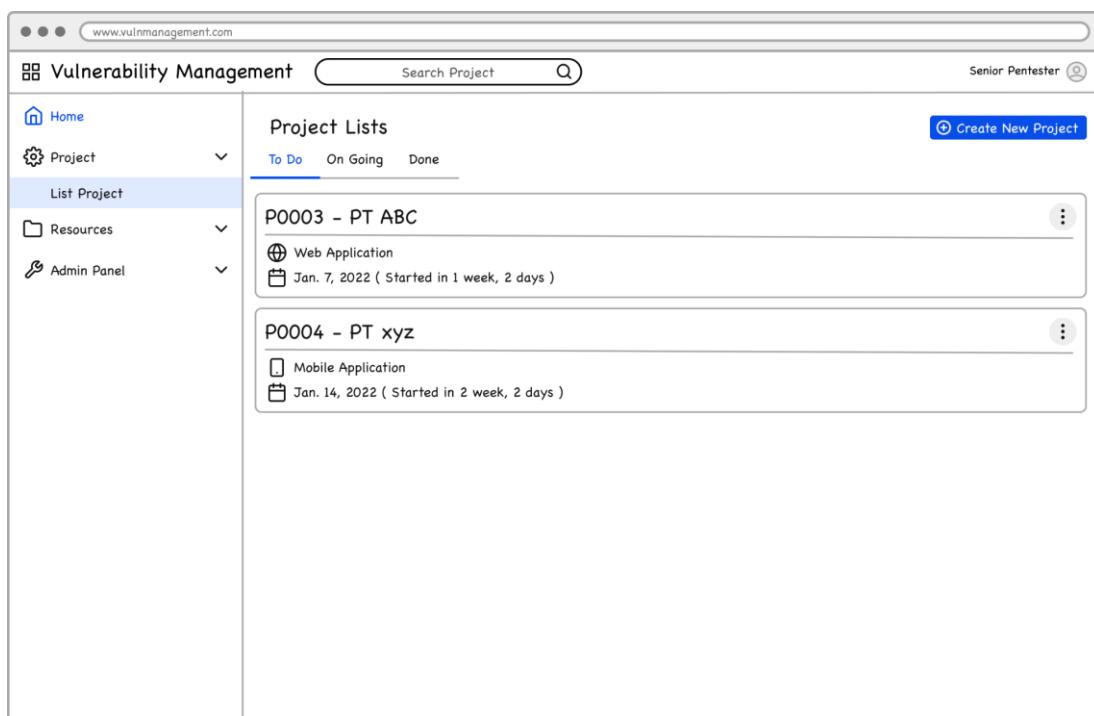
### 3.3.6.2 Project

*Project Lists* memiliki hak akses untuk *role senior penetration tester*, *junior penetration tester*, dan *client*. *Project Lists* terdapat 3 status

yang dapat ditampilkan yaitu *Project Lists To Do*, *Project Lists On Going*, dan *Project Lists Done*.

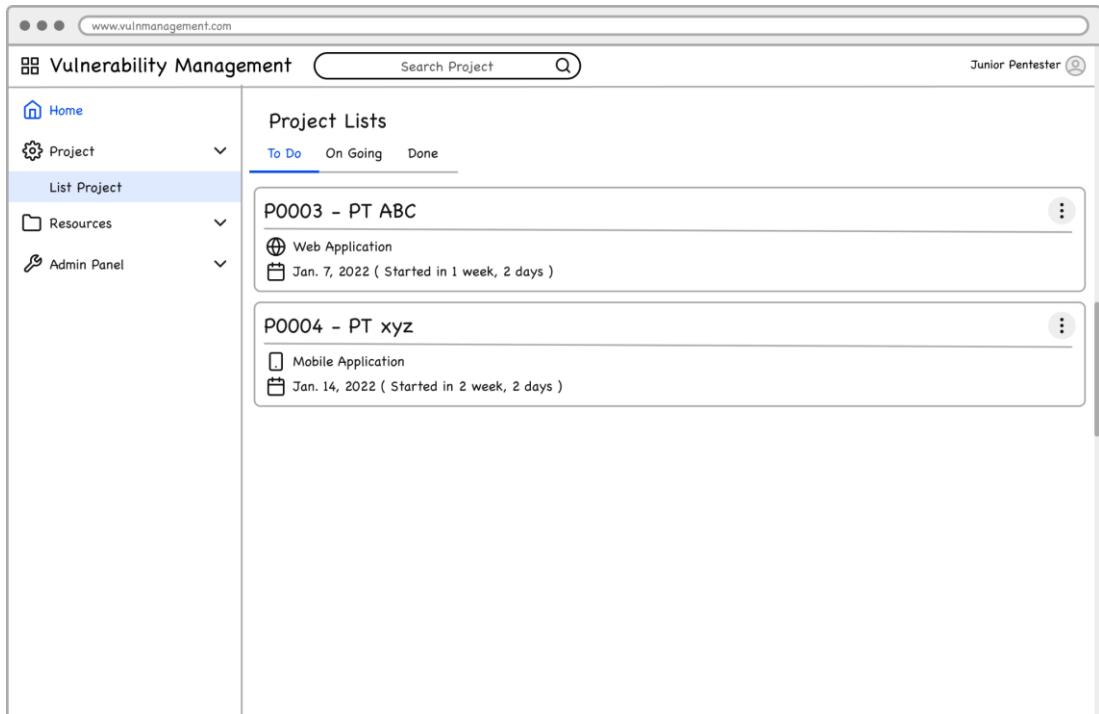
*Project Lists To Do* akan menampilkan semua *project penetration testing* yang belum berjalan dan *user* masuk ke dalam *project* tersebut . Aplikasi akan menampilkan *Project Lists To Do* setelah pengguna mengklik *Project* setelah itu *List Project* atau mengklik *To Do* pada barisan tombol status dibawah judul *Project Lists*.

Tampilan *Project Lists To Do* untuk *senior penetration tester*, aplikasi akan menampilkan fitur *Search Project*, judul *Project Lists*, barisan tombol status dengan tanda di *To Do*, tombol *create new project*, dan *project* yang belum berjalan.



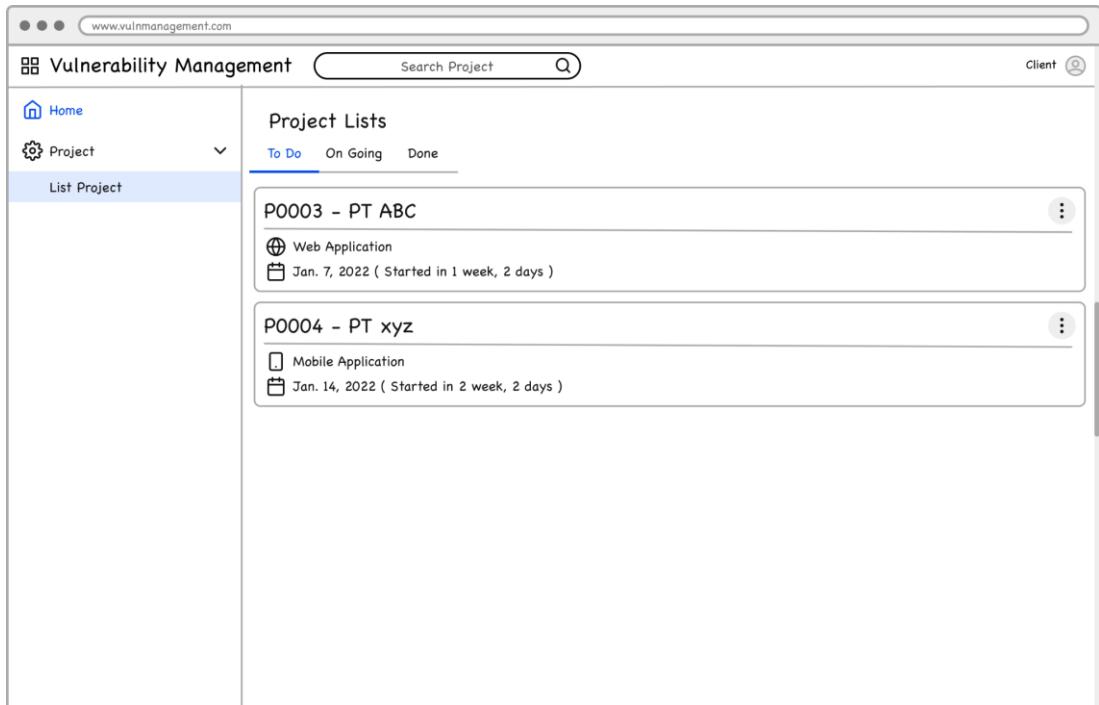
**Gambar 3.60 *Project Lists To Do* ( senior penetration tester )**

Tampilan *Project Lists To Do* untuk *junior penetration tester*, aplikasi akan menampilkan fitur *Search Project*, judul *Project Lists*, barisan tombol status dengan tanda di *To Do*, dan *project* yang belum berjalan.



**Gambar 3.61 Project Lists To Do ( junior penetration tester )**

Tampilan *Project Lists To Do* untuk *client*, aplikasi akan menampilkan fitur *Search Project*, judul *Project Lists*, barisan tombol status dengan tanda di *To Do*, dan *project* yang belum berjalan.

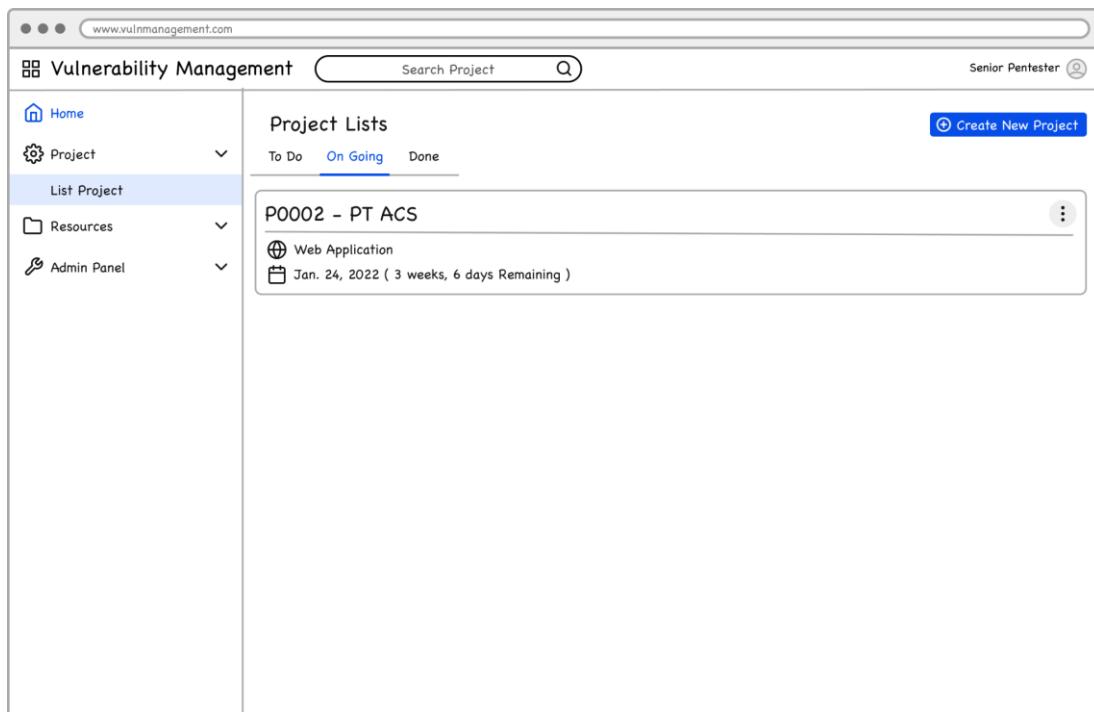


**Gambar 3.62 Project Lists To Do ( client )**

*Project Lists On Going* akan menampilkan semua *project penetration testing* yang sedang berjalan dan *user* masuk ke dalam *project*

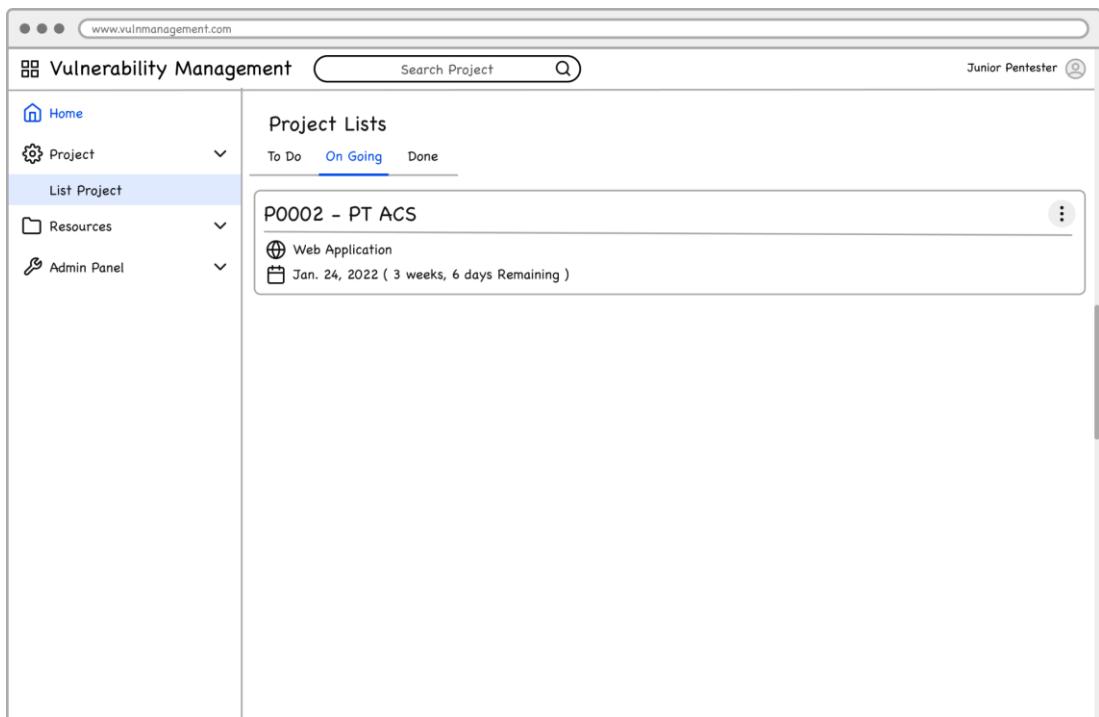
tersebut. Aplikasi akan menampilkan *Project Lists On Going* setelah pengguna mengklik *On Going* pada barisan tombol status dibawah judul *Project Lists*.

Tampilan *Project Lists On Going* untuk *senior penetration tester*, aplikasi akan menampilkan fitur *Search Project*, judul *Project Lists*, barisan tombol status dengan tanda di *On Going*, tombol *create new project*, dan *project* yang sedang berjalan.



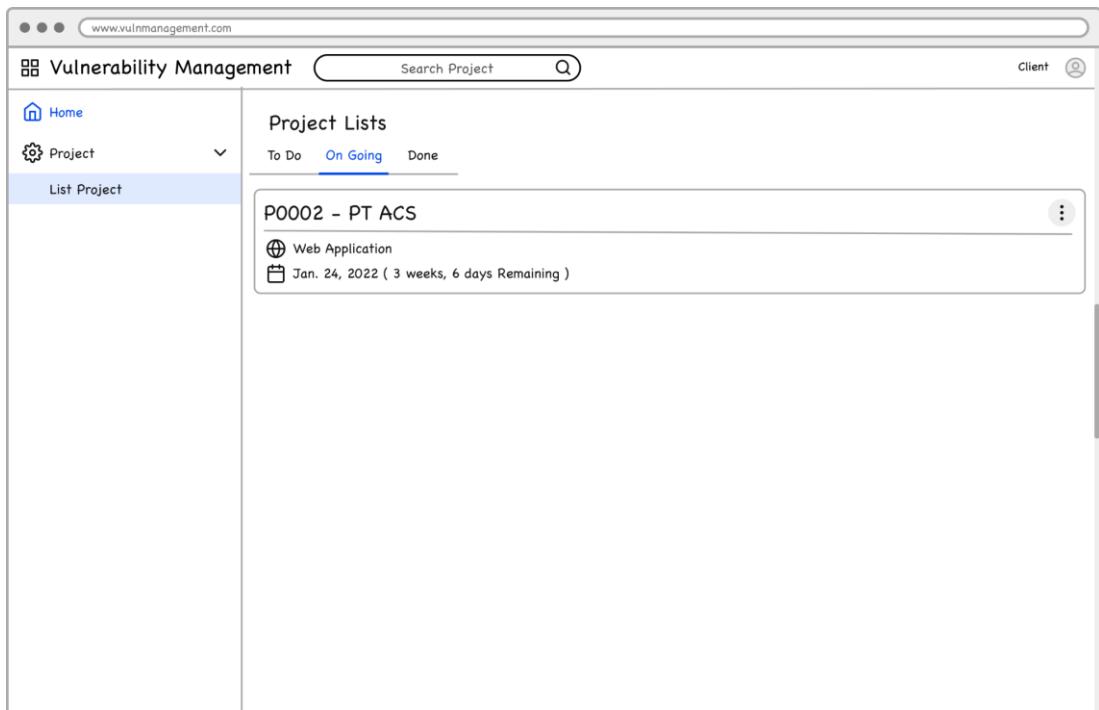
**Gambar 3.63** *Project Lists On Going ( senior penetration tester )*

Tampilan *Project Lists On Going* untuk *junior penetration tester*, aplikasi akan menampilkan fitur *Search Project*, judul *Project Lists*, barisan tombol status dengan tanda di *On Going*, dan *project* yang sedang berjalan.



**Gambar 3.64 Project Lists On Going ( junior penetration tester )**

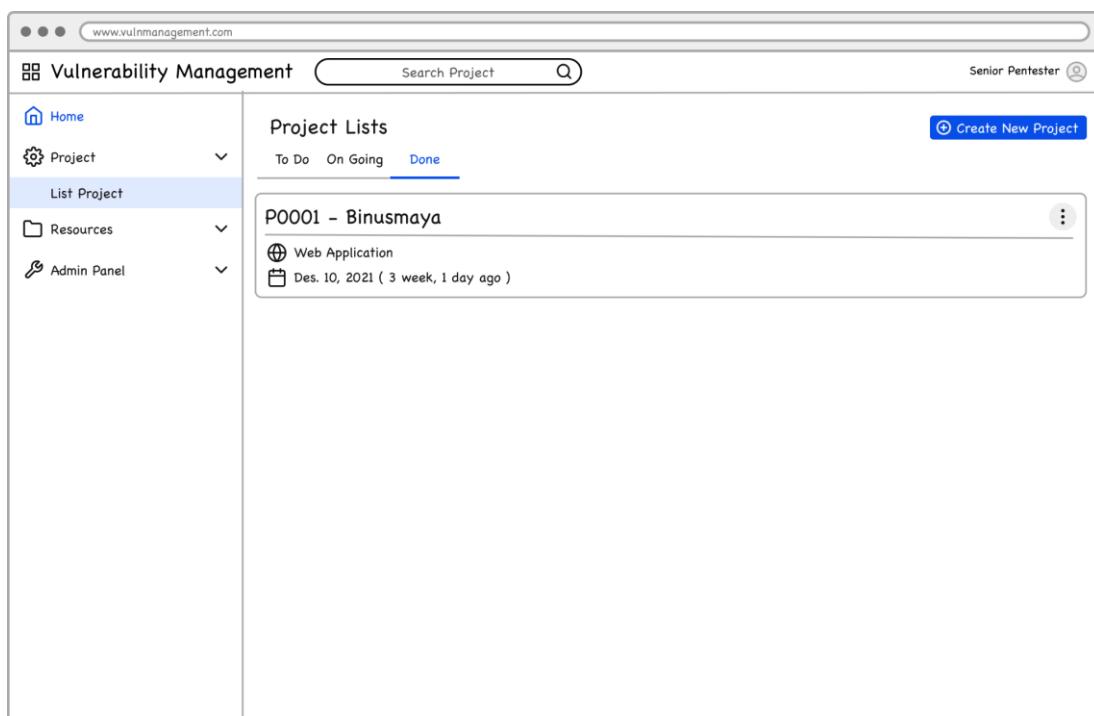
Tampilan *Project Lists On Going* untuk *client*, aplikasi akan menampilkan fitur *Search Project*, judul *Project Lists*, barisan tombol status dengan tanda di *On Going*, dan *project* yang sedang berjalan.



**Gambar 3.65 Project Lists On Going ( client )**

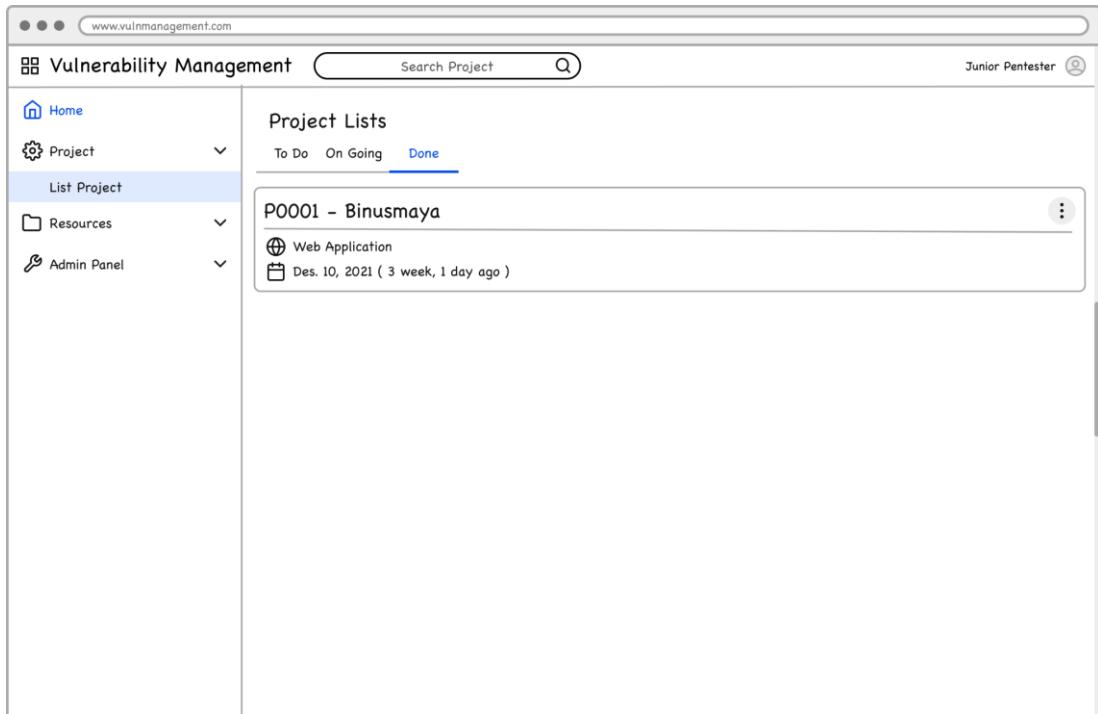
*Project Lists Done* akan menampilkan semua *project penetration testing* yang sudah selesai dan *user* masuk ke dalam *project* tersebut. Aplikasi akan menampilkan *Project Lists Done* setelah pengguna mengklik *Done* pada barisan tombol status dibawah judul *Project Lists*.

Tampilan *Project Lists Done* untuk *senior penetration tester*, aplikasi akan menampilkan fitur *Search Project*, judul *Project Lists*, barisan tombol status dengan tanda di *Done*, tombol *create new project*, dan *project* yang sudah selesai.



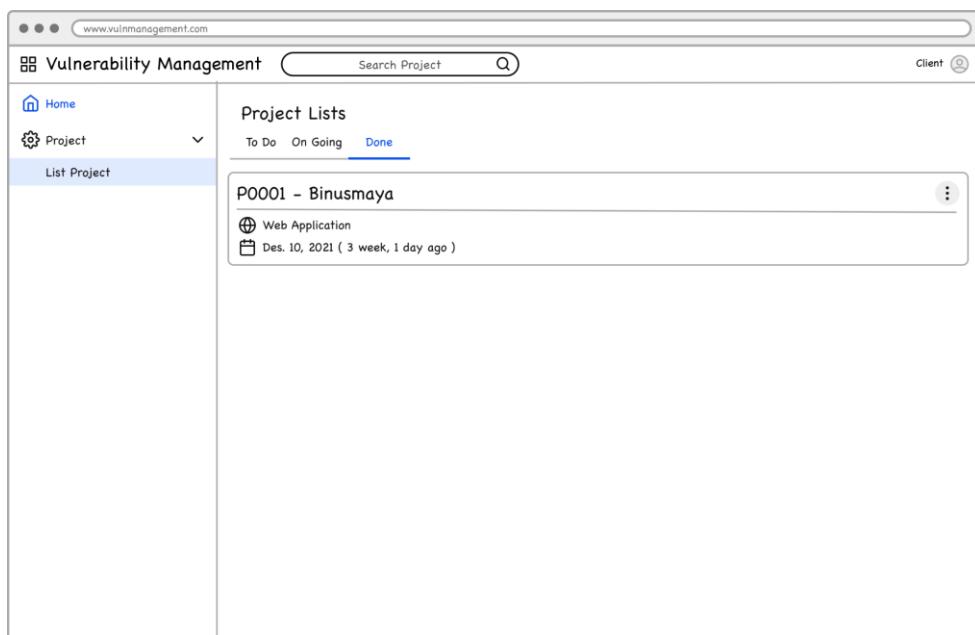
**Gambar 3.66 *Project Lists Done (junior penetration tester)***

Tampilan *Project Lists Done* untuk *junior penetration tester*, aplikasi akan menampilkan fitur *Search Project*, judul *Project Lists*, barisan tombol status dengan tanda di *Done*, dan *project* yang sudah selesai.



**Gambar 3.67 Project Lists Done**

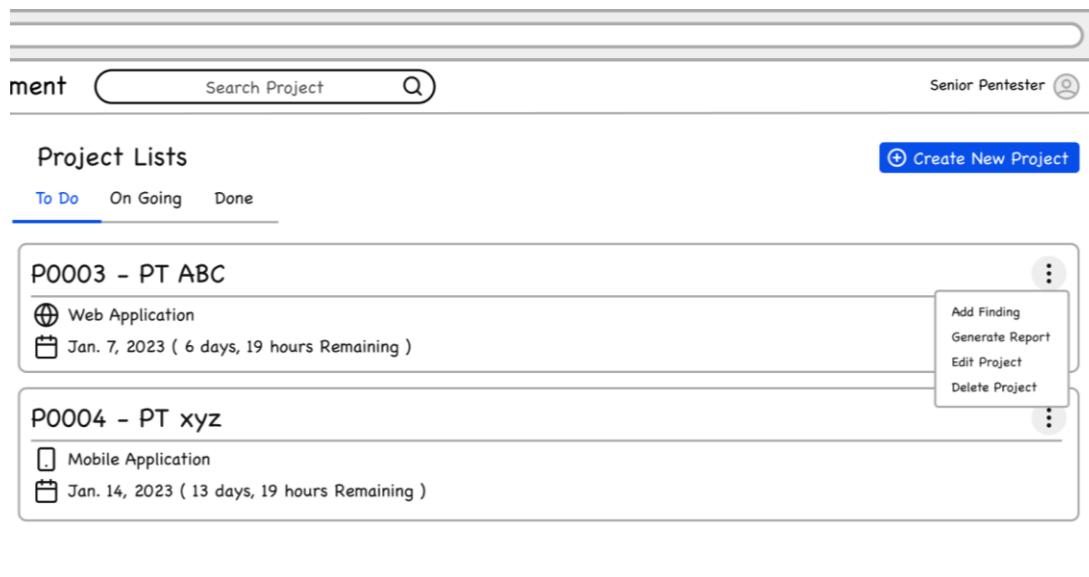
Tampilan *Project Lists Done* untuk *client*, aplikasi akan menampilkan fitur *Search Project*, judul *Project Lists*, barisan tombol status dengan tanda di *Done*, dan *project* yang sudah selesai.



**Gambar 3.68 Project Lists Done**

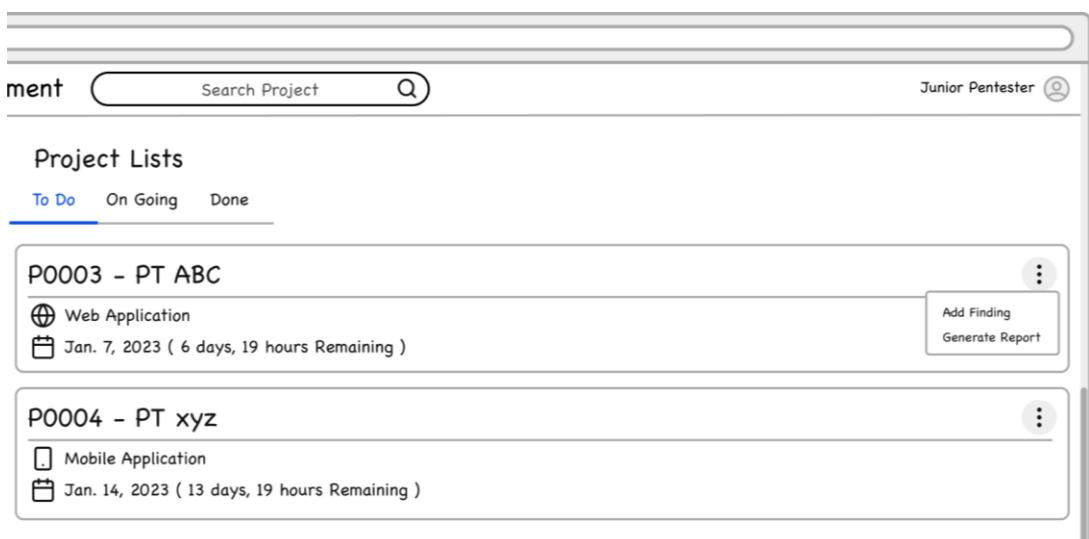
Setiap *project* memiliki *menu action* untuk mengelola *project* tersebut. Aplikasi akan menampilkan *menu action project* setelah pengguna mengklik icon 3 titik paling kanan di dalam kotak *project*.

Tampilan menu action project untuk senior penetration tester, aplikasi akan menampilkan *Add Finding*, *Generate Report*, *Edit Project*, dan *Delete Project*.



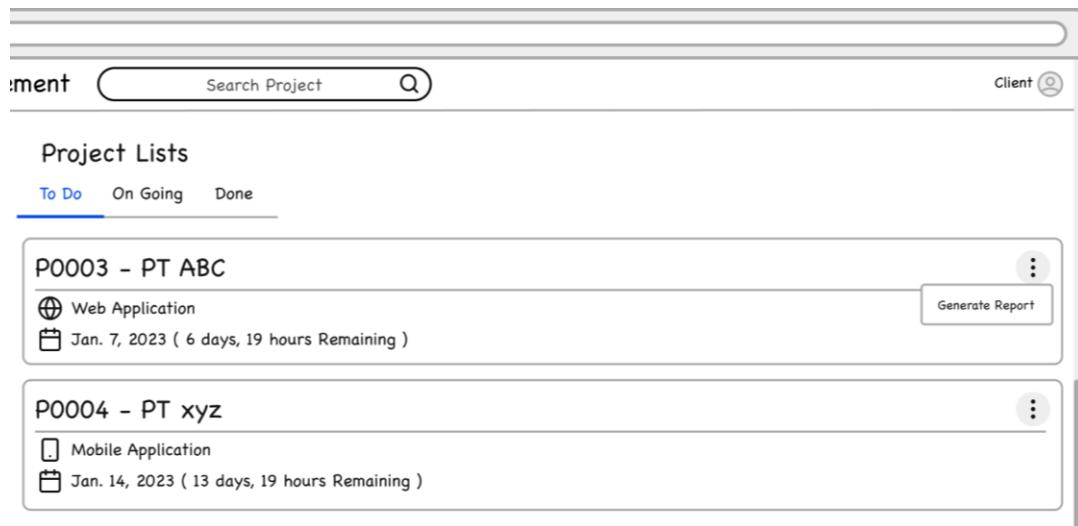
**Gambar 3.69 Project setelah klik icon 3 titik ( senior penetration tester )**

Tampilan menu action project untuk junior penetration tester, aplikasi akan menampilkan *Add Finding* dan *Generate Report*.



**Gambar 3.70 Project setelah klik icon 3 titik ( junior penetration tester )**

Tampilan menu action project untuk client, aplikasi akan menampilkan *Generate Report*.



**Gambar 3.71 Project setelah klik icon 3 titik ( client )**

Form *Create New Project* akan menampilkan *form* yang harus diisi untuk membuat *project* baru. Aplikasi akan menampilkan *Form Create New Project* setelah pengguna mengklik *button Create New Project* di halaman *Project Lists*. *Button Create New Project* di halaman *Project Lists* hanya akan tampil untuk *role senior penetration tester*.

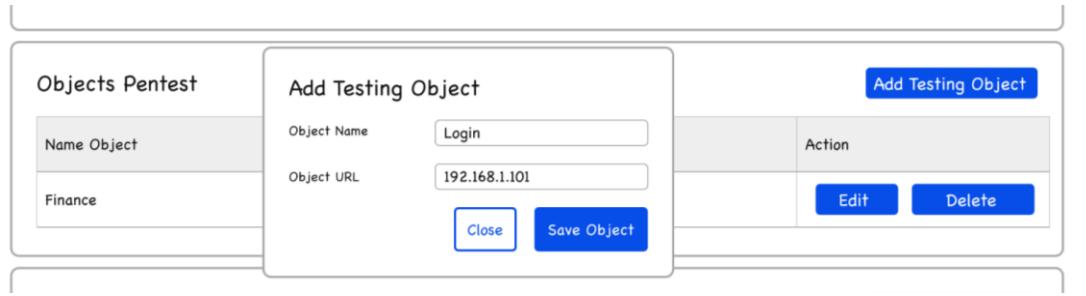
Tampilan *Form Create New Project* untuk *senior penetration tester*, aplikasi akan menampilkan tempat untuk mengisi *Project Name*, *Project Type*, *Start Date*, *End Date*, *Description*, mengelola *Objects Pentest*, mengelola *Members*, mengelola *Clients*, dan *button Add New Project*.

The screenshot shows a web-based vulnerability management application interface. On the left, there's a sidebar with navigation links: Home, Project (selected), List Project, Resources, and Admin Panel. The main content area is titled 'Create New Project'. It contains several input fields: 'Project Name' (PT angkasa), 'Project Type' (Web Application), 'Start Date' (dd/mm/yyyy), 'End Date' (dd/mm/yyyy), and a 'Description' text area (Project angkasa adalah website financial). Below this is a section titled 'Objects Pentest' with a table showing one entry: 'Finance' with URL/IP Address '204.168.90.24'. There are 'Edit' and 'Delete' buttons for this entry. Further down are sections for 'Members' (with a table for Stanley) and 'Client' (with a table for budi). At the bottom left is a blue 'Add New Project' button.

**Gambar 3.72 Form Create New Project ( senior penetration tester )**

*Modal Add Testing Object* akan menampilkan jendela kecil untuk menambahkan *objects pentest* yang baru. Aplikasi akan menampilkan *Modal Add Testing Object* setelah pengguna mengklik *button Add Testing Object* di halaman *Create New Project* dan *Edit Project*. Tampilan *Modal Add Testing Object* akan sama di *Form Create New Project* dan *Form Edit Project*.

Tampilan *Modal Add Testing Object* di *Form Create New Project* dan *Form Edit Project*, aplikasi akan menampilkan jendela kecil untuk menambahkan *Objects Pentest* dengan memasukan *Object Name* dan *Object URL*, setelah itu klik *button save object*, atau pengguna dapat klik *button close* untuk tidak jadi menambahkan *Objects Pentest*.



**Gambar 3.73 Modal Add Testing Object di Form New Create Project dan Form Edit Project**

Modal *Edit Testing Object* akan menampilkan jendela kecil untuk merubah *objects pentest*. Aplikasi akan menampilkan *Modal Edit Testing Object* setelah pengguna mengklik *button Edit* di dalam kotak *Objects Pentest* pada halaman *Create New Project* dan *Edit Project*. Tampilan *Modal Edit Testing Object* akan sama di *Form Create New Project* dan *Form Edit Project*.

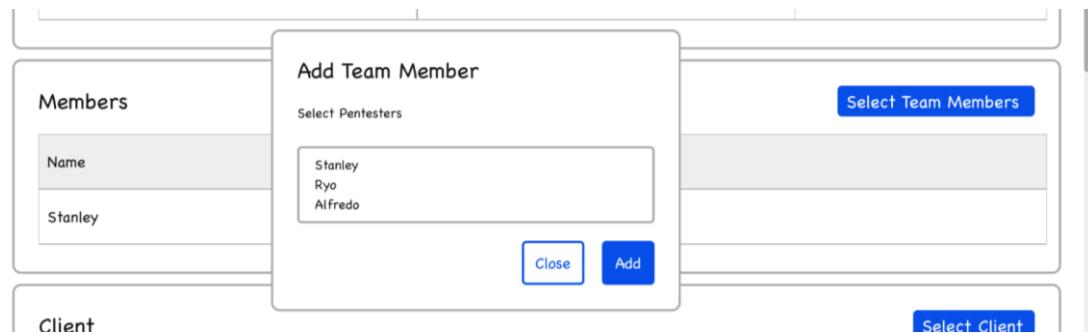
Tampilan *Modal Edit Testing Object* di *Form Create New Project* dan *Form Edit Project*, aplikasi akan menampilkan jendela kecil untuk merubah *Objects Pentest* dengan mengganti *Object Name* dan *Object URL*, setelah itu klik *button save object*, atau pengguna dapat klik *button close* untuk tidak jadi merubah *Objects Pentest*.



**Gambar 3.74 Modal Edit Testing Object di Form New Create Project dan Form Edit Project**

Modal *Select Team Member* akan menampilkan jendela kecil untuk memilih *penetration tester*. Aplikasi akan menampilkan *Modal Select Team Member* setelah pengguna mengklik *button Select Team Members* di dalam kotak *Members* pada halaman *Create New Project* dan *Edit Project*. Tampilan *Modal Select Team Member* akan sama di *Form Create New Project* dan *Form Edit Project*.

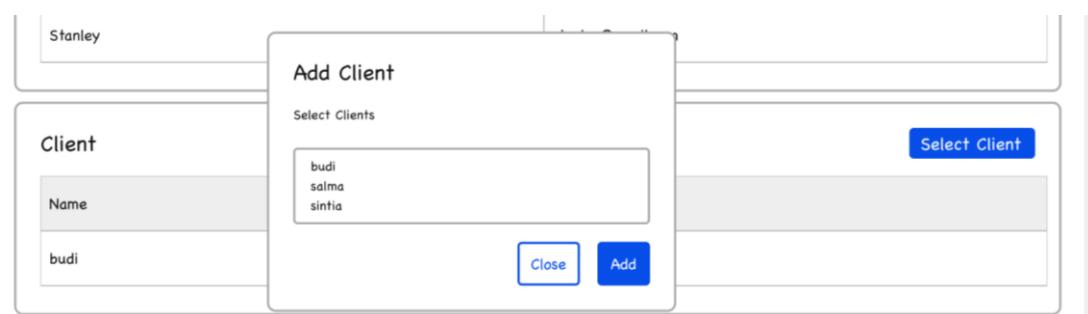
Tampilan *Modal Select Team Member* di *Form Create New Project* dan *Form Edit Project*, aplikasi akan menampilkan jendela kecil dengan *list penetration tester* yang dapat dipilih setelah itu klik *button add*, atau pengguna dapat klik *button close* untuk tidak jadi memilih *penetration tester*.



**Gambar 3.75 Modal Select Team Member di Form New Create Project dan Form Edit Project**

*Modal Select Client* akan menampilkan jendela kecil untuk memilih *client*. Aplikasi akan menampilkan *Modal Select Client* setelah pengguna mengklik *button Select Client* di dalam kotak *Client* pada halaman *Create New Project* dan *Edit Project*. Tampilan *Modal Select Client* akan sama di *Form Create New Project* dan *Form Edit Project*.

Tampilan *Modal Select Client* di *Form Create New Project* dan *Form Edit Project*, aplikasi akan menampilkan jendela kecil dengan *list client* yang dapat dipilih setelah itu klik *button add*, atau pengguna dapat klik *button close* untuk tidak jadi memilih *client*.



**Gambar 3.76 Modal Select Client di Form Create Project dan Form Edit Project**

*Form Edit Project* akan menampilkan *form* yang harus diisi untuk merubah *project*. Aplikasi akan menampilkan *Form Edit Project* setelah pengguna mengklik *button Edit Project* dari *icon 3 titik Project* pada halaman *Project Lists*. *Button Edit Project* hanya akan tampil untuk *role senior penetration tester*.

Tampilan *Form Edit Project* untuk *senior penetration tester*, aplikasi akan menampilkan tempat untuk merubah *Project Name*, *Project Type*, *Start Date*, *End Date*, *Description*, mengelola *Objects Pentest*, mengelola *Members*, mengelola *Clients*, dan button *Add New Project*.

Name Object	URL/IP Address	Action
Finance	204.168.90.24	<button>Edit</button> <button>Delete</button>

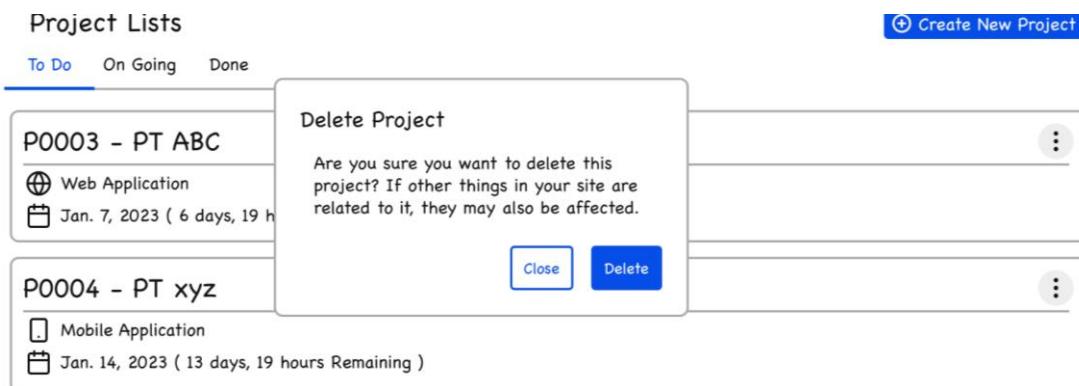
Name	Email
Stanley	stanley@gmail.com

Name	Email
budi	budi@gmail.com

**Gambar 3.77 Form Edit Project**

*Modal Delete Project* akan menampilkan jendela kecil untuk memastikan bahwa pengguna ingin menghapus *project* itu. Aplikasi akan menampilkan *Modal Delete Project* setelah pengguna mengklik button *Delete Project* dari icon 3 titik *Project* pada halaman *Project Lists*. Button *Delete Project* hanya akan tampil untuk *role senior penetration tester*.

Tampilan *Modal Delete Client*, aplikasi akan menampilkan jendela kecil dengan memberikan pesan untuk memastikan bahwa pengguna sudah yakin untuk menghapus *project* tersebut, ada button *close* untuk membatalkan hapus *project*, ada button *Delete* untuk menghapus *project*.



**Gambar 3.78 Modal Delete Project**

*View Detail Project with Not Fixed Finding Lists* akan menampilkan informasi *detail* sebuah *project* dengan *checklist* dan *finding list* yang statusnya *Not Fixed*. Aplikasi akan menampilkan *View Detail Project with Not Fixed Finding Lists* setelah pengguna mengklik salah satu kotak *project* pada halaman *Project Lists* atau mengklik *Not Fixed* di bawah tulisan *Finding Lists* pada halaman *Detail Project*. Tampilan *View Detail Project with Not Fixed Finding Lists* untuk *role senior penetration tester* dengan *junior penetration tester* sama, tetapi untuk *role client* berbeda.

Tampilan *View Detail Project with Not Fixed Finding Lists* untuk *senior penetration tester* dengan *junior penetration tester*, aplikasi akan menampilkan informasi *project* yaitu *name project*, *type project*, jarak *date start* sampai *date end*, *description project*, semua *client* yang terlibat dalam *project*, semua *member* yang terlibat dalam *project*, dan semua *object pentest* dengan *link checklists* dan tempat untuk mengelola *finding lists* yang statusnya *Not Fixed*, didalam *finding lists* akan menampilkan setiap *finding* dengan informasi *Finding Name*, *Target Object*, *CVSS Score*, *Severity*, *Finder*, *Status*, dan *Action*, ditambah ada *button Add New Finding*.

The screenshot displays a web-based application interface for managing projects. At the top, there is a navigation bar with a search bar labeled "Search Project" and a user profile icon for "Senior Pentester". Below the navigation bar, the title "Detail Project" is visible, followed by the project identifier "P0005 - PT angkasa".

**Project Information:**

- Type: Web Application
- Date Range: Tue, 3 Jan 2023 - Thu, 26 Jan 2023 (3 weeks, 2 days)

**Description:**

Project angkasa adalah website financial

**Client Details:**

Name	Email
budi	budi@gmail.com

**Members:**

Name	Email
Stanley	stanley@gmail.com

**Objects Pentest:**

Name Object	URL/IP Address
Finance	204.168.90.24

**WSTG Checklist:**

To open the checklist, please click this link <https://docs.google.com/spreadsheets/d/>

**Findings Lists:**

Not Fixed      Fixed      Add New Finding

Finding Name	Target Object	CVSS Score	Saverity	Finder	Status	Action
Reflected XSS	Finance	6.1	Medium	stanley	Not Fixed	Action ▾

**Gambar 3.79 View Detail Project with Not Fixed Finding Lists ( senior penetration tester dan junior penetration tester )**

Tampilan *View Detail Project with Not Fixed Finding Lists* untuk *client*, aplikasi akan menampilkan informasi *project* yaitu *name project*, *type project*, jarak *date start* sampai *date end*, *description project*, semua *client* yang terlibat dalam *project*, semua *member* yang terlibat dalam *project*, dan semua *object pentest* dengan *link checklists* dan tempat untuk mengelola *finding lists* yang statusnya *Not Fixed*, didalam *finding lists* akan menampilkan setiap *finding* dengan informasi *Finding Name*, *Target Object*,

*CVSS Score, Severity, Finder, dan Status. Client tidak memiliki tombol Add New Finding dan Action.*

Finding Name	Target Object	CVSS Score	Saverity	Finder	Status
Reflected XSS	Finance	6.1	Medium	stanley	Not Fixed

**Gambar 3.80 View Detail Project with Not Fixed Finding Lists ( client )**

Setiap *finding* memiliki *menu action* untuk mengelola *finding* tersebut. Aplikasi akan menampilkan *menu action* setelah pengguna mengklik *button Action* pada salah satu *finding*.

Tampilan *menu action finding not fixed* untuk *senior penetration tester*, aplikasi akan menampilkan *Resolve To Fixed*, *Edit Finding*, dan *Delete Finding*.

**Findings Lists**

Finding Name	Target Object	CVSS Score	Severity	Finder	Status	Action
Reflected XSS	Finance	6.1	Medium	stanley	Not Fixed	<a href="#">Resolve To Fixed</a> <a href="#">Edit Finding</a> <a href="#">Delete Finding</a>

**Gambar 3.81 View Detail Project with Not Fixed Finding Lists setelah klik button action ( senior penetration tester )**

Tampilan *menu action finding not fixed* untuk *junior penetration tester*, aplikasi akan menampilkan *Edit Finding*, dan *Delete Finding*.

**Findings Lists**

Finding Name	Target Object	CVSS Score	Severity	Finder	Status	Action
Reflected XSS	Finance	6.1	Medium	stanley	Not Fixed	<a href="#">Add New Finding</a> <a href="#">Edit Finding</a> <a href="#">Delete Finding</a>

**Gambar 3.82 View Detail Project with Not Fixed Finding Lists setelah klik button action ( junior penetration tester )**

*View Detail Project with Fixed Finding Lists* akan menampilkan informasi *detail* sebuah *project* dengan *checklist* dan *finding list* yang statusnya *Fixed*. Aplikasi akan menampilkan *View Detail Project with Fixed Finding Lists* setelah pengguna mengklik salah satu kotak *project* pada halaman *Project Lists* atau mengklik *Fixed* di bawah tulisan *Finding Lists* pada halaman *Detail Project*. Tampilan *View Detail Project with Not Fixed Finding Lists* untuk *role senior penetration tester* dengan *junior penetration tester* sama, tetapi untuk *role client* berbeda.

Tampilan *View Detail Project with Fixed Finding Lists* untuk *senior penetration tester* dengan *junior penetration tester*, aplikasi akan menampilkan tampilan yang hampir sama dengan *View Detail Project with Not Fixed Finding List* tetapi status *Finding* yang ditampilkan adalah *Fixed*, ditambah tanda biru akan berpindah ke *Fixed*.

Findings Lists						
Not Fixed	Fixed					
Finding Name	Target Object	CVSS Score	Saverity	Finder	Status	Action
Reflected XSS	Finance	6.1	Medium	stanley	Fixed	<button>Action</button>

**Gambar 3.83 View Detail Project with Fixed Finding Lists ( senior penetration tester dan junior penetration tester )**

Tampilan *View Detail Project with Fixed Finding Lists* untuk *client*, aplikasi akan menampilkan tampilan yang hampir sama dengan *View Detail Project with Not Fixed Finding List* tetapi status *Finding* yang ditampilkan adalah *Fixed*, ditambah tanda biru akan berpindah ke *Fixed*.

Findings Lists					
Not Fixed	Fixed				
Finding Name	Target Object	CVSS Score	Saverity	Finder	Status
Reflected XSS	Finance	6.1	Medium	stanley	Fixed

**Gambar 3.84 View Detail Project with Fixed Finding Lists ( Client )**

Setiap *finding* memiliki *menu action* untuk mengelola *finding* tersebut. Aplikasi akan menampilkan *menu action* setelah pengguna mengklik *button Action* pada salah satu *finding*.

Tampilan *menu action finding fixed* untuk *senior penetration tester*, aplikasi akan menampilkan *Re-open To Not Fixed*, *Edit Finding*, dan *Delete Finding*.

Findings Lists						
Not Fixed	Fixed					
Finding Name	Target Object	CVSS Score	Saverity	Finder	Status	
Reflected XSS	Finance	6.1	Medium	stanley	Fixed	<button>Action</button>

**Gambar 3.85 View Detail Project with Not Fixed Finding Lists setelah klik button action ( senior penetration tester )**

Tampilan *menu action finding not fixed* untuk *junior penetration tester*, aplikasi akan menampilkan *Edit Finding*, dan *Delete Finding*.

Findings Lists						<a href="#">Add New Finding</a>
Not Fixed	Fixed					
Finding Name	Target Object	CVSS Score	Saverity	Finder	Status	
Reflected XSS	Finance	6.1	Medium	stanley	Fixed	<a href="#">Edit Finding</a> <a href="#">Delete Finding</a>

**Gambar 3.86 View Detail Project with Not Fixed Finding Lists setelah klik button action ( junior penetration tester )**

*Form Add New Finding* akan menampilkan form yang harus diisi untuk menambahkan *finding*. Aplikasi akan menampilkan *Form Add New Finding* setelah pengguna mengklik button *Add New Finding* pada halaman *Detail Project*. Button *Add New Finding* hanya akan tampil untuk *role senior penetration tester* dan *junior penetration tester*.

Tampilan *Form Add New Finding* untuk *senior penetration tester* dan *junior penetration tester*, aplikasi akan menampilkan tempat untuk mengisi *Object Pentest*, *Finding Name*, *Weakness*, *Severity*, *Description*, *Impact*, mengelola *Evidences*, *Remediation*, dan *References*. Tampilan *Severity* akan diambil langsung dari website asli *CVSS*.

The screenshot shows a web-based application for vulnerability management. The URL is [www.vulnmanagement.com](http://www.vulnmanagement.com). The page title is "Vulnerability Management". A search bar at the top right contains the text "Search Project". On the far right, there is a "Senior Pentester" user profile icon.

The left sidebar has a navigation menu with the following items:

- Home**
- Project** (selected)
- List Project**
- Resources**
- Admin Panel**

The main content area is titled "Add New Finding" and shows the following details:

- Object Project:** Financial - 204.168.90.24
- Finding Name:** [Empty input field]
- Weakness:** Relative Path Traversal (CWE-23)
- Severity:** Base Score (6.5) (Medium)
- CVSS 3.1 Score Breakdown:**
  - Attack Vector (AV):** Network (N)
  - Attack Complexity (AC):** Low (L)
  - Privileges Required (PR):** None (N)
  - User Interaction (UI):** None (N)
  - Scope (S):** Unchanged (U)
  - Confidentiality (C):** Low (L)
  - Integrity (I):** Low (L)
  - Availability (A):** Low (L)
- Vector String:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L
- Description:** [Empty input field]
- Impact:** [Empty input field]
- Evidences:**

Description	Images	Action
Berikut hasil dari menjalankan payload		<b>Edit</b> <b>Delete</b>
- Remediation:** [Empty input field]
- References:** [Empty input field]

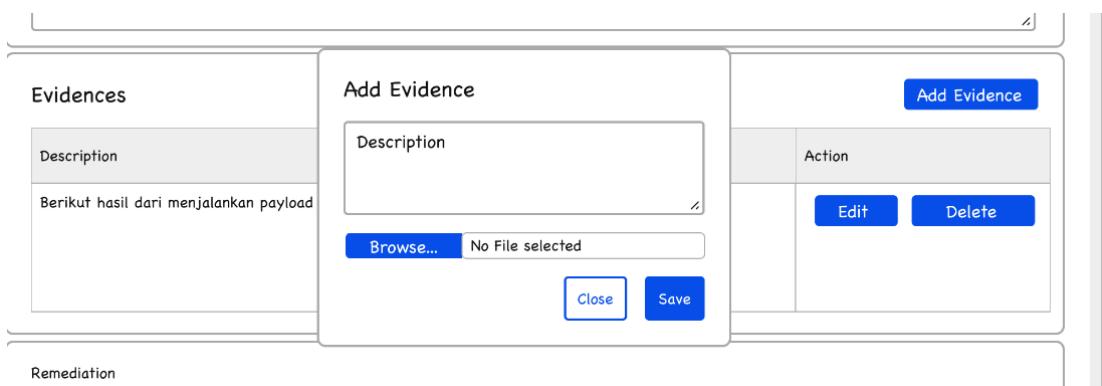
A blue "Add Finding" button is located at the bottom left of the main form area.

**Gambar 3.87 Form Add New Finding ( senior penetration tester dan junior penetration tester )**

Modal Add Evidence akan menampilkan jendela kecil untuk menambahkan Evidence. Aplikasi akan menampilkan Modal Add Evidence

setelah pengguna mengklik *button Add Evidence* pada halaman *Add New Finding* atau *Edit Finding*. Tampilan *Modal Add Evidence* akan sama di *Form Add New Finding* dan *Form Edit Finding*.

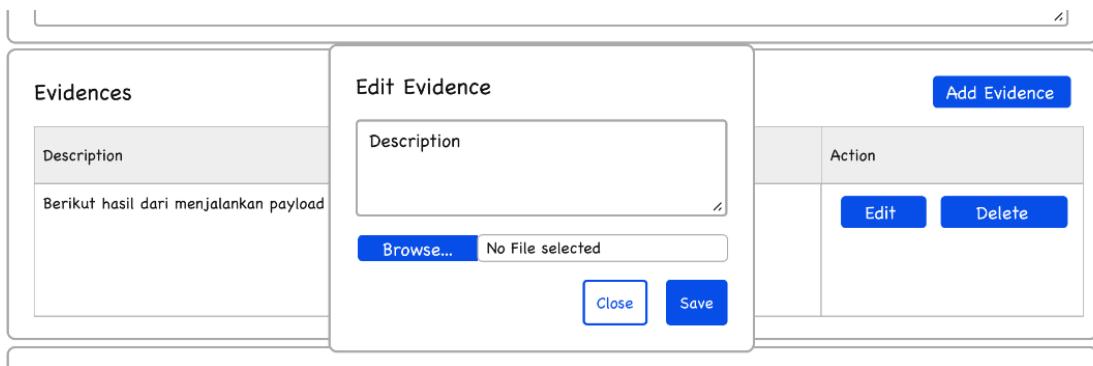
Tampilan *Modal Add Evidence* di *Form Add New Finding* dan *Form Edit Finding* untuk *senior penetration tester* dan *junior penetration tester*. aplikasi akan menampilkan jendela kecil dengan tempat untuk mengisi *Description* dan *upload* gambar setelah itu dapat klik *button save*, atau pengguna dapat klik *button close* untuk membatalkan tambah *Evidence*.



**Gambar 3.88 Modal Add Evidence ( senior penetration tester dan junior penetration tester )**

*Modal Edit Evidence* akan menampilkan jendela kecil untuk merubah *Evidence*. Aplikasi akan menampilkan *Modal Edit Evidence* setelah pengguna mengklik *button Edit* di salah satu *Evidence* pada halaman *Add New Finding* atau *Edit Finding*. Tampilan *Modal Edit Evidence* akan sama di *Form Add New Finding* dan *Form Edit Finding*.

Tampilan *Modal Edit Evidence* di *Form Add New Finding* dan *Form Edit Finding* untuk *senior penetration tester* dan *junior penetration tester*. aplikasi akan menampilkan jendela kecil dengan tempat untuk merubah *Description* dan *upload* gambar setelah itu dapat klik *button save*, atau pengguna dapat klik *button close* untuk membatalkan ubah *Evidence*.



**Gambar 3.89 Modal Edit Evidence ( senior penetration tester dan junior penetration tester )**

Form *Edit Finding* akan menampilkan form untuk merubah *finding*. Aplikasi akan menampilkan *Form Edit Finding* setelah pengguna mengklik button *Edit Finding* di salah satu *action finding* pada halaman *Detail Project*.

Tampilan *Form Edit Finding* untuk *senior penetration tester* dan *junior penetration tester*, aplikasi akan menampilkan tempat untuk merubah *Object Pentest*, *Finding Name*, *Weakness*, *Severity*, *Description*, *Impact*, mengelola *Evidences*, *Remediation*, dan *References*. Tampilan *Severity* akan diambil langsung dari website asli *CVSS*.

ment Search Project  Senior Pentester

### Edit Finding

My Project / P0005 - PT Angkasa / Reflected XSS / Edit Finding

Object Project	Financial - 204.168.90.24
Finding Name	Reflected XSS
Weakness	Improper Neutralization of Input During Web Page Generation (Cross-site Scripting) ( CWE-79 )

Severity 6.5 (Medium)

**Base Score**

Attack Vector (AV)	Network (N) <input checked="" type="checkbox"/> Adjacent (A) <input type="checkbox"/> Local (L) <input type="checkbox"/> Physical (P) <input type="checkbox"/>	Scope (S)	Unchanged (U) <input checked="" type="checkbox"/> Changed (C) <input type="checkbox"/>
Attack Complexity (AC)	Low (L) <input checked="" type="checkbox"/> High (H) <input type="checkbox"/>	Confidentiality (C)	None (N) <input type="checkbox"/> Low (L) <input checked="" type="checkbox"/> High (H) <input type="checkbox"/>
Privileges Required (PR)	None (N) <input checked="" type="checkbox"/> Low (L) <input type="checkbox"/> High (H) <input type="checkbox"/>	Integrity (I)	None (N) <input checked="" type="checkbox"/> Low (L) <input type="checkbox"/> High (H) <input type="checkbox"/>
User Interaction (UI)	None (N) <input checked="" type="checkbox"/> Required (R) <input type="checkbox"/>	Availability (A)	None (N) <input type="checkbox"/> Low (L) <input checked="" type="checkbox"/> High (H) <input type="checkbox"/>

Vector String -  
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N

Description

Impact

Evidences Add Evidence

Description	Images	Action
Berikut hasil dari menjalankan payload		<span style="border: 1px solid blue; padding: 2px;">Edit</span> <span style="border: 1px solid blue; padding: 2px;">Delete</span>

Remediation

References

Save Finding

**Gambar 3.90 Form Edit Finding ( senior penetration tester dan junior penetration tester )**

*View Detail Finding* akan menampilkan informasi *detail* sebuah *finding*. Aplikasi akan menampilkan *View Detail Finding* setelah pengguna mengklik salah satu nama *finding* pada halaman *Detail Project*.

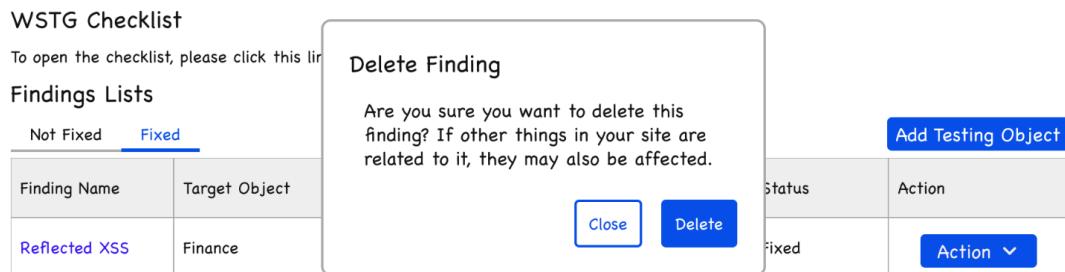
Tampilan *View Detail Project with Not Fixed Finding Lists* untuk semua *role*, aplikasi akan menampilkan informasi *finding* yaitu *Object Pentest*, *Finding Name*, *Weakness*, *Severity*, *CVSS*, *Description*, *Impact*, semua *Evidences*, *Remediation*, dan *References*.



**Gambar 3.91 View Detail Finding**

*Modal Delete Finding* akan menampilkan jendela kecil untuk memastikan bahwa pengguna ingin menghapus *finding* itu. Aplikasi akan menampilkan *Modal Delete Finding* setelah pengguna mengklik *button Delete Finding* di salah satu *action finding* pada halaman *Detail Project*.

Tampilan *Modal Delete Client* untuk *senior penetration tester* dan *junior penetration tester*, aplikasi akan menampilkan jendela kecil dengan memberikan pesan untuk memastikan bahwa pengguna sudah yakin untuk menghapus *finding* tersebut, ada *button close* untuk membatalkan hapus *finding*, ada *button Delete* untuk menghapus *finding*.



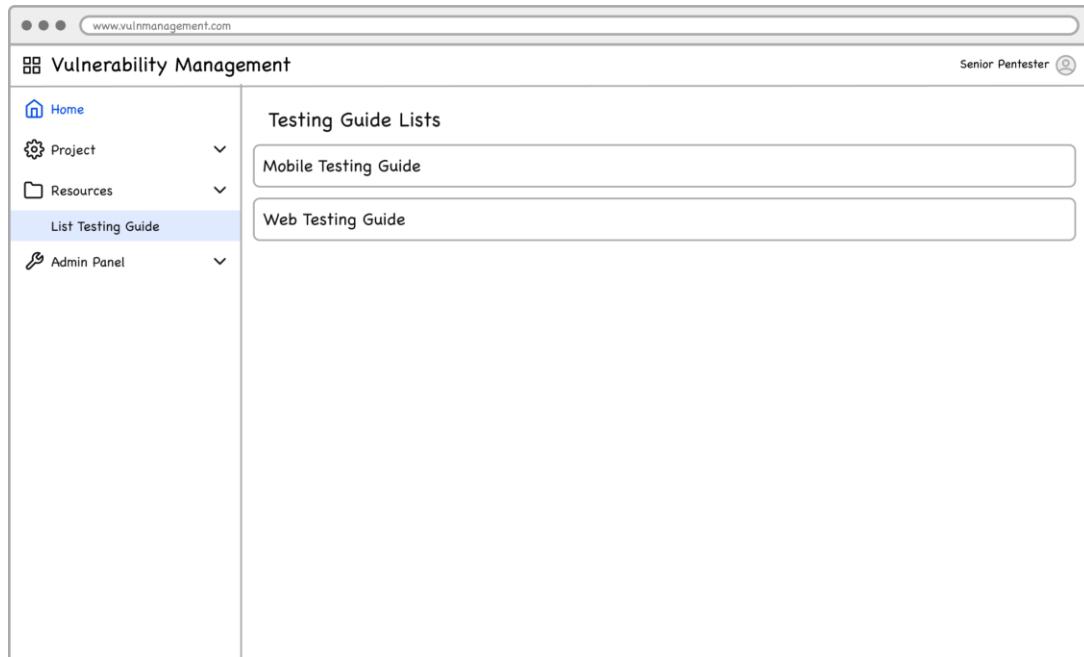
**Gambar 3.92 Modal Delete Finding**

### 3.3.6.3 Resources

*Resources* memiliki hak akses untuk *role senior penetration tester* dan *junior penetration tester*. *Resources* memiliki *Testing Guide Lists* yang berisi informasi cara *testing security* pada sebuah aplikasi. Semua tampilan untuk *role senior penetration tester* dan *junior penetration tester* sama.

*Testing Guide Lists* akan menampilkan semua *type Testing Guide*. Aplikasi akan menampilkan *Testing Guide Lists* setelah pengguna mengklik *Resources* setelah itu *List Testing Guide*. Saat ini hanya tersedia *type web* dan *mobile*.

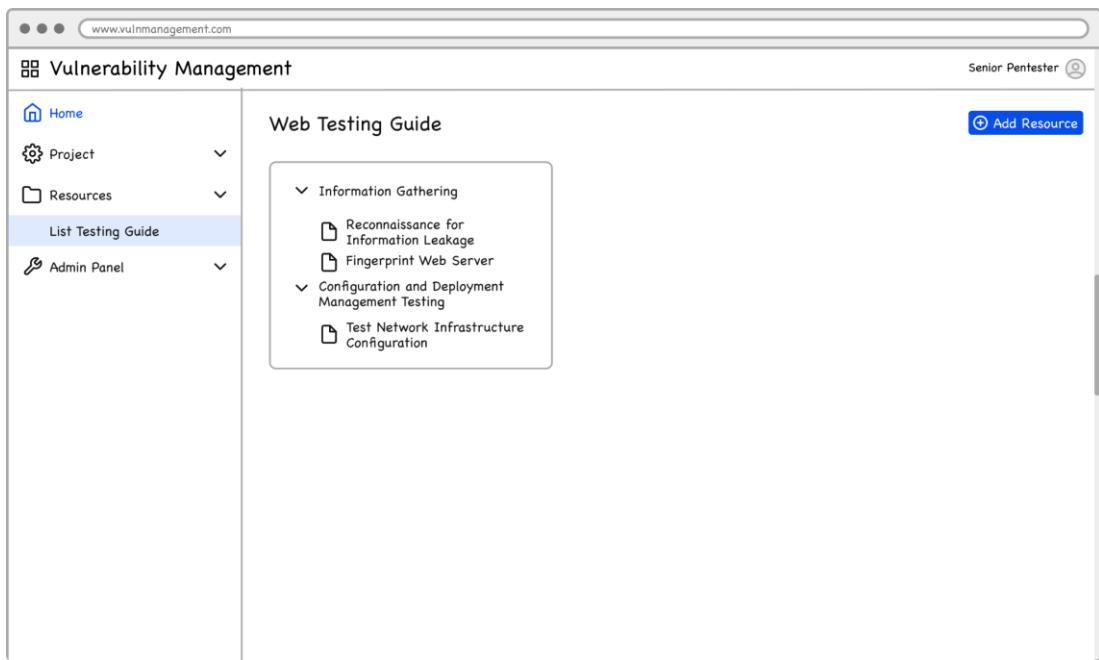
Tampilan *Testing Guide Lists*, aplikasi akan menampilkan *type Testing Guide* yang tersedia, saat ini menampilkan *Mobile Testing Guide* dan *Web Testing Guide*.



**Gambar 3.93 Testing Guide Lists**

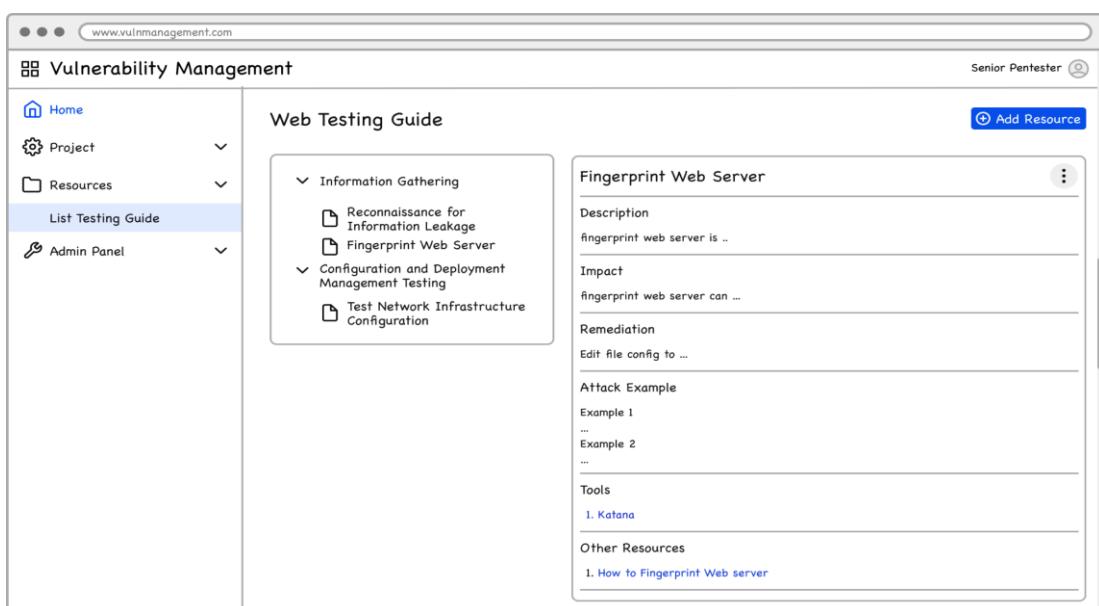
*Detail Testing Guide* akan menampilkan *list category* beserta *testing* sesuai dengan tipe *Testing Guide* yang telah dipilih. Aplikasi akan menampilkan *Detail Testing Guide* setelah pengguna mengklik salah satu type *Testing Guide* pada halaman *Testing Guide Lists*.

Tampilan *Detail Testing Guide* untuk *Web*, aplikasi akan menampilkan semua *testing* setiap *category* dan *button Add Resources*, pada dibawah ini akan menampilkan 2 kategori yaitu *Information Gathering* dan *Configuration and Deployment Management Testing*, *Information Gathering* memiliki *testing* yaitu *Reconnaissance for Information Leakage* dan *Fingerprint Web Server*, *Configuration and Deployment Management Testing* memiliki *testing* yaitu *Test Network Infrastructure Configuration*.



**Gambar 3.94 Detail Testing Guide**

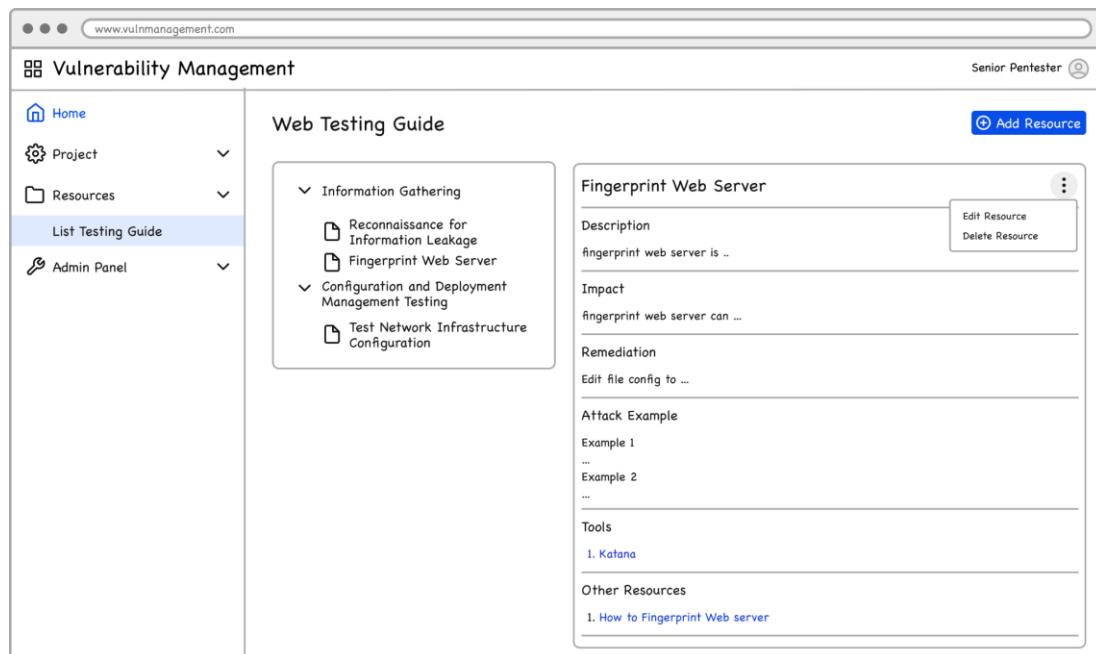
*Detail Testing* akan menampilkan informasi detail terhadap *testing*. Aplikasi akan menampilkan *Detail Testing* setelah pengguna mengklik salah satu *testing* pada halaman *Testing Guide*. Tampilan *Detail Testing*, aplikasi akan menampilkan *Name Testing*, *Description*, *Impact*, *Remediation*, *Attack Example*, *Tools*, dan *Other Resources* pada bagian sebelah kanan.



**Gambar 3.95 Detail Testing**

Setiap *testing* memiliki *menu action* untuk mengelola *testing* tersebut. Aplikasi akan menampilkan *menu action testing* setelah pengguna mengklik icon 3 titik paling kanan pada *Detail Testing*.

Tampilan *menu action testing*, aplikasi akan menampilkan *Edit Resources*, dan *Delete Resources*.



**Gambar 3.96 Detail Testing setelah klik icon 3 titik**

*Form Add New Testing Guide* akan menampilkan *form* yang harus diisi untuk membuat *testing* baru. Aplikasi akan menampilkan *Form Add New Testing Guide* setelah pengguna mengklik *button Add Resources* di halaman *Detail Testing Guide*.

Tampilan *Form Add New Testing Guide*, aplikasi akan menampilkan tempat untuk mengisi *Category*, *Name Testing*, *Description*, *Impact*, *Remediation*, *Attack Example*, *Tools*, dan *Other Resources*, ada *button Add Testing Guide* untuk menambahkan *testing*.

The screenshot shows a web-based vulnerability management application. The URL in the address bar is www.vulnmanagement.com. The title of the page is "Vulnerability Management". On the left side, there is a sidebar with the following navigation items: Home, Project, Resources, List Testing Guide (which is currently selected), and Admin Panel. The main content area is titled "Add New Testing Guide". It contains several input fields: "Category" (with a dropdown arrow), "Name Testing Guide" (text input), "Description" (text area), "Impact" (text area), "Remediation" (text area), and three separate "Attack Example" sections (each with a text area). At the bottom of the form is a blue "Add Testing Guide" button.

**Gambar 3.97 Form Add Testing Guide**

Form Edit Testing Guide akan menampilkan form yang harus diisi untuk merubah testing. Aplikasi akan menampilkan Form Edit Testing Guide setelah pengguna mengklik button Edit Resources di menu action testing pada halaman Detail Testing.

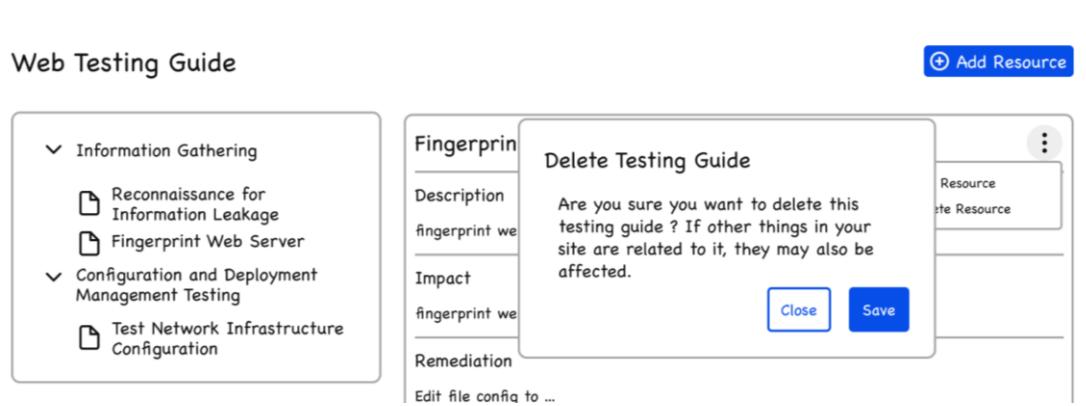
Tampilan Form Add New Testing Guide, aplikasi akan menampilkan tempat untuk merubah Category, Name Testing, Description, Impact, Remediation, Attack Example, Tools, dan Other Resources, ada button Edit Testing Guide untuk menyimpan perubahan testing.

The screenshot shows a web-based application interface for 'Vulnerability Management'. The main title bar says 'www.vulnmanagement.com' and 'Vulnerability Management'. On the right, it says 'Senior Pentester'. The left sidebar has a 'Project' section with a dropdown arrow, a 'Resources' section with a dropdown arrow, and a 'List Testing Guide' section which is highlighted with a blue background and a dropdown arrow. Below these are 'Admin Panel' and another dropdown arrow. The main content area is titled 'Edit Testing Guide'. It contains several input fields: 'Category' (with a dropdown arrow), 'Name Testing Guide' (empty), 'Description' (empty), 'Impact' (empty), 'Remediation' (empty), and three 'Attack Example' sections, each with an empty text area. At the bottom is a blue 'Edit Testing Guide' button.

**Gambar 3.98 Form Edit Testing Guide**

*Modal Delete Testing Guide* akan menampilkan jendela kecil untuk memastikan bahwa pengguna ingin menghapus *testing guide* itu. Aplikasi akan menampilkan *Modal Delete Testing Guide* setelah pengguna mengklik *button Delete Resources* di salah satu *Testing Guide* pada halaman *Detail Testing Guide*.

Tampilan *Modal Delete Testing Guide* untuk *senior penetration tester* dan *junior penetration tester*, aplikasi akan menampilkan jendela kecil dengan memberikan pesan untuk memastikan bahwa pengguna sudah yakin untuk menghapus *finding* tersebut, ada *button close* untuk membatalkan hapus *finding*, ada *button Delete* untuk menghapus *finding*.



**Gambar 3.99 Modal Delete Testing Guide**

### 3.3.6.4 User Management

*User Management* memiliki hak akses hanya untuk *role senior penetration tester*. *User Management* akan menampilkan tempat untuk mengelola *user*. Aplikasi akan menampilkan *User Management* setelah pengguna mengklik *Admin Panel* setelah itu *User Management*.

Tampilan *User Management*, aplikasi akan menampilkan *button Add New User*, dan semua user dengan informasi *Name*, *Username*, *Email*, dan *Role*, ditambah dengan *Action* untuk mengelola *user*.

The screenshot shows a sidebar with 'Home', 'Project', 'Resources', 'Admin Panel' (selected), and 'Users Management'. The main area is titled 'Users Management' and shows a table of users:

Name	Username	Email	Role	Action
Stanley	stan	stanley@gmail.com	Penetration Tester	Action ▾
Ryo Chris	ryo	ryo@gmail.com	Penetration Tester	Action ▾
Alfredo	alfre	alfredo@gmail.com	Penetration Tester	Action ▾
Budi	budi	budi@gmail.com	Client	Action ▾
Min	min	min@gmail.com	Principal Penetration Tester	Action ▾

**Gambar 3.100 Users Management**

Setiap *user* memiliki *menu action* untuk mengelola *user* tersebut. Aplikasi akan menampilkan *menu action user* setelah pengguna mengklik *button action* di salah satu *user* pada halaman *Users Management*.

Tampilan *menu action user*, aplikasi akan menampilkan *Edit User*, *Change Password*, dan *Delete User*.



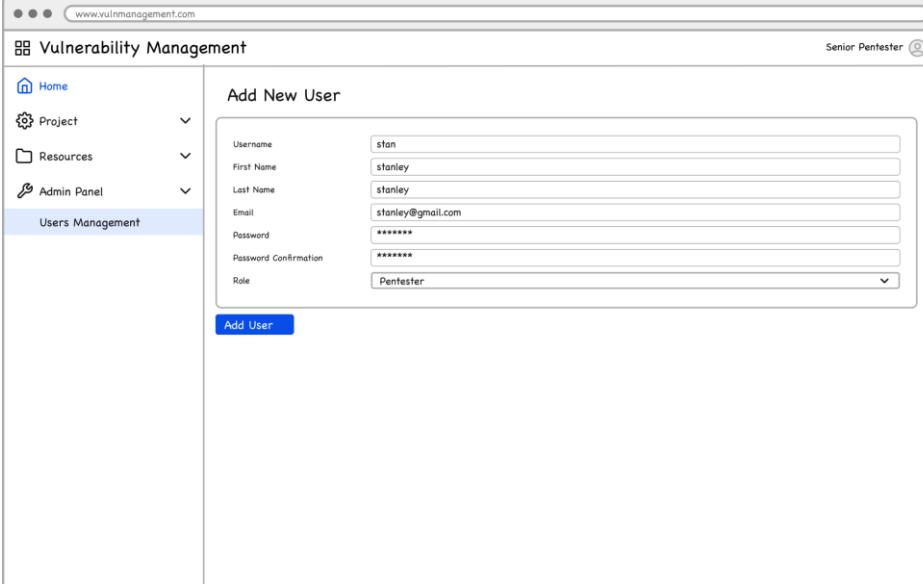
The screenshot shows a table titled 'Users Management' with columns: Name, Username, Email, Role, and Action. There are four rows of data. In the 'Action' column for the fourth row (Alfredo), a context menu is displayed with options: 'Edit User', 'Change Password', and 'Delete User'. Each option has a small downward arrow indicating it's a dropdown menu.

Name	Username	Email	Role	Action
Stanley	stan	stanley@gmail.com	Penetration Tester	<span>Action ▾</span>
Ryo Chris	ryo	ryo@gmail.com	Penetration Tester	<span>Action ▾</span>
Alfredo	alfre	alfredo@gmail.com	Penetration Tester	<span>Action ▾</span>

Gambar 3.101 *User Management* setelah klik *button Action*

*Form Add New User* akan menampilkan *form* yang harus diisi untuk membuat *user* baru. Aplikasi akan menampilkan *Form Add New User* setelah pengguna mengklik *button Add New User* pada halaman *Users Management*.

Tampilan *Form Add New User*, aplikasi akan menampilkan tempat untuk mengisi *username*, *First Name*, *Last Name*, *Email*, *Password*, *Password Confirmation*, dan *Role*, ada button *Add User* untuk membuat *user* baru.

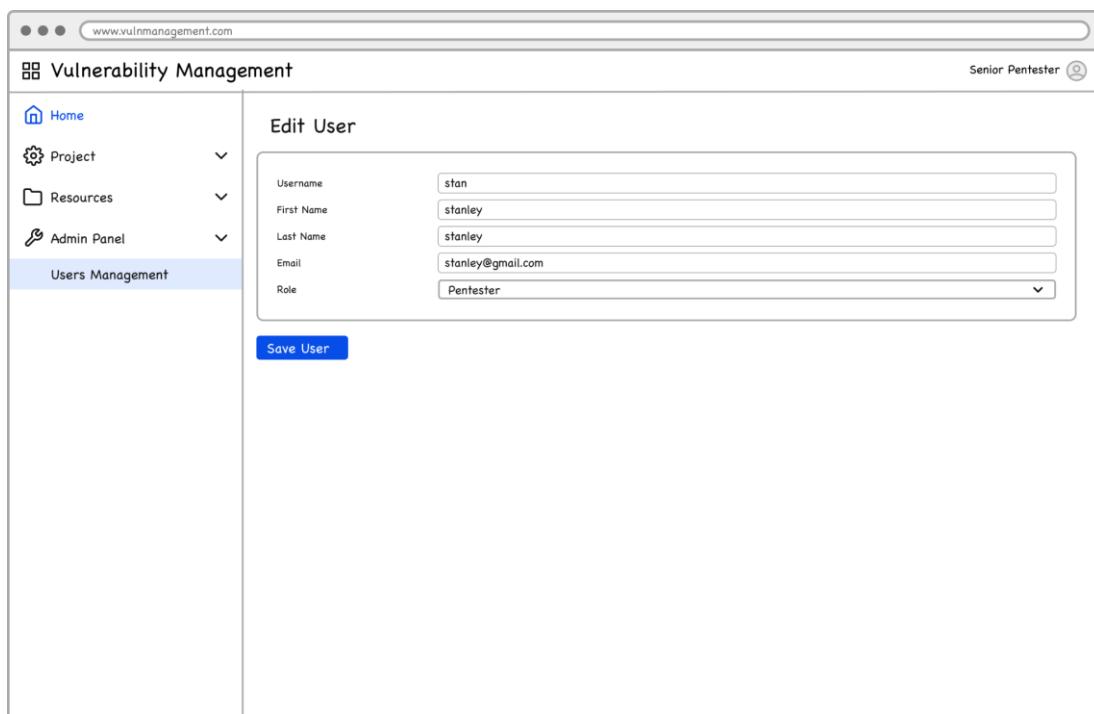


The screenshot shows the 'Add New User' form within the 'Vulnerability Management' application. The left sidebar has a navigation menu with 'Home', 'Project', 'Resources', 'Admin Panel', and 'Users Management' selected. The main area is titled 'Add New User' and contains fields for 'Username' (stan), 'First Name' (stanley), 'Last Name' (stanley), 'Email' (stanley@gmail.com), 'Password' (\*\*\*\*\*), 'Password Confirmation' (\*\*\*\*\*), and 'Role' (Penetration Tester). At the bottom of the form is a blue 'Add User' button.

Gambar 3.102 *Form Add New User*

*Form Edit User* akan menampilkan *form* yang harus diisi untuk membuat *user* baru. Aplikasi akan menampilkan *Form Edit User* setelah pengguna mengklik *button Edit User* di salah satu *action user* pada halaman *Users Management*.

Tampilan *Form Edit User*, aplikasi akan menampilkan tempat untuk merubah *username*, *First Name*, *Last Name*, *Email*, dan *Role*, ada *button Save User* untuk merubah informasi *user*.



**Gambar 3.103 Form Edit User**

*Modal Change Password User* akan menampilkan jendela kecil untuk merubah *password user*. Aplikasi akan menampilkan *Modal Change Password User* setelah pengguna mengklik *button Delete User* di salah satu *action user* pada halaman *User Management*.

Tampilan *Modal Change Password User*, aplikasi akan menampilkan jendela kecil dengan memberikan *input password*, ada *button close* untuk membatalkan ubah *password user*, ada *button Submit* untuk merubah *password user*.

The screenshot shows a table of users with columns for Name, Username, Email, Role, and Action. A modal window titled 'Change Password' is overlaid on the table. The modal contains a 'Password' input field, a 'Close' button, and a 'Submit' button.

Name	Username	Email	Role	Action
Stanley	stan		Penetration Tester	Action ▾
Ryo Chris	ryo		Penetration Tester	Action ▾
Alfredo	alfre		Penetration Tester	Action ▾
Budi	budi	budi@gmail.com	Client	Action ▾
Min	min	min@gmail.com	Principal Penetration Tester	Action ▾

**Gambar 3.104 Modal Change Password User**

*Modal Delete User* akan menampilkan jendela kecil untuk memastikan bahwa pengguna ingin menghapus *user*. Aplikasi akan menampilkan *Modal Delete User* setelah pengguna mengklik *button Delete User* di salah satu action *user* pada halaman *User Management*.

Tampilan *Modal Delete User*, aplikasi akan menampilkan jendela kecil dengan memberikan pesan untuk memastikan bahwa pengguna sudah yakin untuk menghapus *user* tersebut, ada *button close* untuk membatalkan hapus *finding*, ada *button Delete* untuk menghapus *user*.

The screenshot shows a table of users with columns for Name, Username, Email, Role, and Action. A modal window titled 'Delete User' is overlaid on the table. The modal contains a message asking if the user is sure they want to delete the user, and two buttons: 'Close' and 'Delete'.

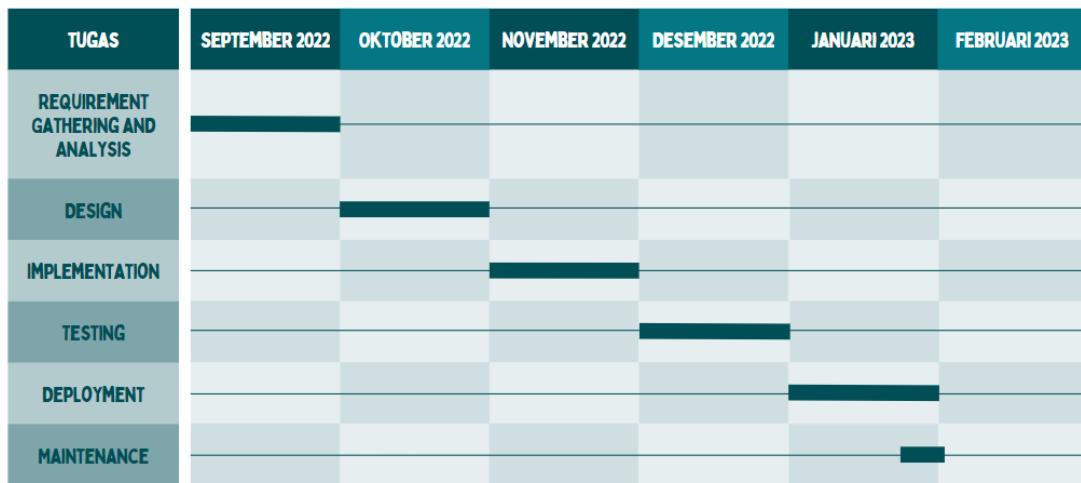
Name	Username	Email	Role	Action
Stanley	stan	stanley@gmail.com	Penetration Tester	Action ▾
Ryo Chris	ryo		Penetration Tester	Action ▾
Alfredo	alfre		Penetration Tester	Action ▾
Budi	budi		Client	Action ▾
Min	min		Principal Penetration Tester	Action ▾

**Gambar 3.105 Modal Delete User**

### 3.4 Gantt Chart

*Gantt chart* di bawah ini mengilustrasikan jadwal dalam penggerjaan aplikasi *vulnerability management* berdasarkan langkah-langkah SDLC *Waterfall*. Pada bulan September 2022 melakukan *requirement gathering and analysis*. Pada bulan Oktober 2022 melakukan *design*. Pada bulan November 2022 melakukan *implementation*.

Pada bulan Desember 2022 melakukan *testing*. Pada bulan Januari 2022 melakukan *deployment* beserta *maintenance* di satu minggu terakhir bulan Januari 2022. Pada bulan Februari 2022 melakukan *maintenance* selama 3 hari.



**Gambar 3.82 Gantt Chart**

### 3.5 Teknologi yang digunakan

Dalam pengembangan *back end* aplikasi, penulis menggunakan bahasa pemrograman *Python* dengan versi 3 dengan bantuan *framework Django*. Alasan penulis menggunakan bahasa pemrograman *Python* adalah karena *Python* bersifat *open source* sehingga mudah untuk dipahami dan diimplementasikan, *Python* juga sudah memberikan kemudahan integrasi ke beberapa platform lain seperti *website*, *mobile* dan *desktop*. Dengan bantuan *framework Django* ini memudahkan penulis untuk membangun sebuah *back end website* menggunakan bahasa pemrograman *Python*. *Framework Django* dipilih untuk mengembangkan *back end* dari aplikasi karena *Django* merupakan salah satu *framework* untuk *website* yang sangat populer pada bahasa pemrograman *Python*. Pada repositori mereka, lebih dari 2000 orang sudah berkontribusi mengembangkan *framework Django* itu sendiri. *Framework Django* juga memiliki dokumentasi yang lengkap dan jelas serta komunitas yang besar sehingga memudahkan penulis dalam mengembangkan aplikasi.

Dalam pengembangan *front end* aplikasi, penulis menggunakan *framework CSS Bootstrap* dengan bantuan tema *SB Admin 2*. Alasan penulis menggunakan *framework CSS Bootstrap* karena *framework CSS Bootstrap* memiliki dokumentasi yang lengkap beserta dengan contoh yang diberikan, ini memudahkan penulis dalam mengimplementasikan *user interface* yang sudah dirancang sebelumnya. *framework*

*CSS Bootstrap* juga membawa *library Javascript jQuery* untuk menjadikan aplikasi kami menjadi lebih interaktif untuk penggunaan pengguna.

Sebagai penyimpanan dan manajemen data, penulis menggunakan *MySQL database*. Alasan penulis menggunakan *MySQL database* karena memiliki kecepatan yang cepat dalam menampilkan, menyimpan dan juga mengolah data. *MySQL database* juga bersifat gratis dan mempunyai komunitas yang besar sehingga dapat dan mudah dipakai oleh semua orang.

Untuk mengemas semuanya, baik dari *back end*, *front end* dan *database*, kami menggunakan *Docker* yang dapat menjalankan aplikasi kami di *environment* yang terisolasi. *Docker* juga memudahkan penulis untuk mengatur jenis dan spesifikasi yang diperlukan agar aplikasi kami dapat berjalan normal. Berikut ruang lingkup dari pembuatan aplikasi *vulnerability management*:

1. *Operating System* yang digunakan adalah *Linux*.
2. Bahasa Pemrograman yang digunakan adalah *Python* versi 3.11.1.
3. *Framework Django* yang digunakan adalah *Django* versi 3.2.16.
4. *Framework CSS* yang digunakan adalah *Bootstrap* versi 4.6.0 dengan tema *SB Admin 2*.
5. *Library Javascript* yang digunakan adalah *jQuery* dengan versi 3.6.0.
6. *Database Management System* yang digunakan adalah *MySQL* versi 8.
7. *Deployment platform* yang digunakan adalah *Docker* versi 20.10.22 dengan *Docker Compose* versi 2.14.2.
8. Perhitungan *severity* yang digunakan adalah CVSS dengan versi 3.1

Berikut *Python package* yang perlu diinstal dalam pengembangan aplikasi *vulnerability management*:

1. *mysqlclient* dengan versi 2.1.1.
2. *martor* dengan versi 1.6.13.
3. *cwe* dengan versi 1.6.
4. *WeasyPrint* dengan versi 57.1.
5. *pygsheets* dengan versi 2.0.6.



## BAB 4

### HASIL DAN PEMBAHASAN

#### 4.1 Spesifikasi Sistem

Dari penelitian yang telah dibuat penulis, berikut spesifikasi dari aplikasi *vulnerability management* :

**Tabel 4.1 Spesifikasi Aplikasi *vulnerability management***

<i>File Size</i>	11 MB
<i>Category</i>	<i>Security Management Tool</i>
<i>Requirement</i>	<i>Docker and Docker compose</i>
<i>Languages</i>	<i>English</i>
<i>Download source</i>	<i>Github</i>

Untuk melakukan penelitian serta pembuatan aplikasi *vulnerability management* ini dibutuhkan perangkat komputer dengan spesifikasi minimum:

**Tabel 4.2 Spesifikasi server Aplikasi *vulnerability management***

<i>Operating System</i>	<i>Kali Linux 2021</i>
<i>Processor</i>	Intel(R) Core(TM) i3-8145u CPU @ 2.10GHz (4 CPUs), ~2.3GHz 64-bit processor
<i>RAM</i>	8 GB
<i>Storage</i>	1 TB
<i>Graphic</i>	All graphic card
<i>Requirement Software</i>	<i>Docker and Docker compose</i>

Spesifikasi minimum pada perangkat komputer server tidak menjadi masalah utama karena aplikasi aplikasi tetap dapat dijalankan selama memiliki *Memory RAM* 4GB, *free storage* 5 GB, dan *requirement software*.

**Tabel 4.3 Spesifikasi *user* Aplikasi *vulnerability management***

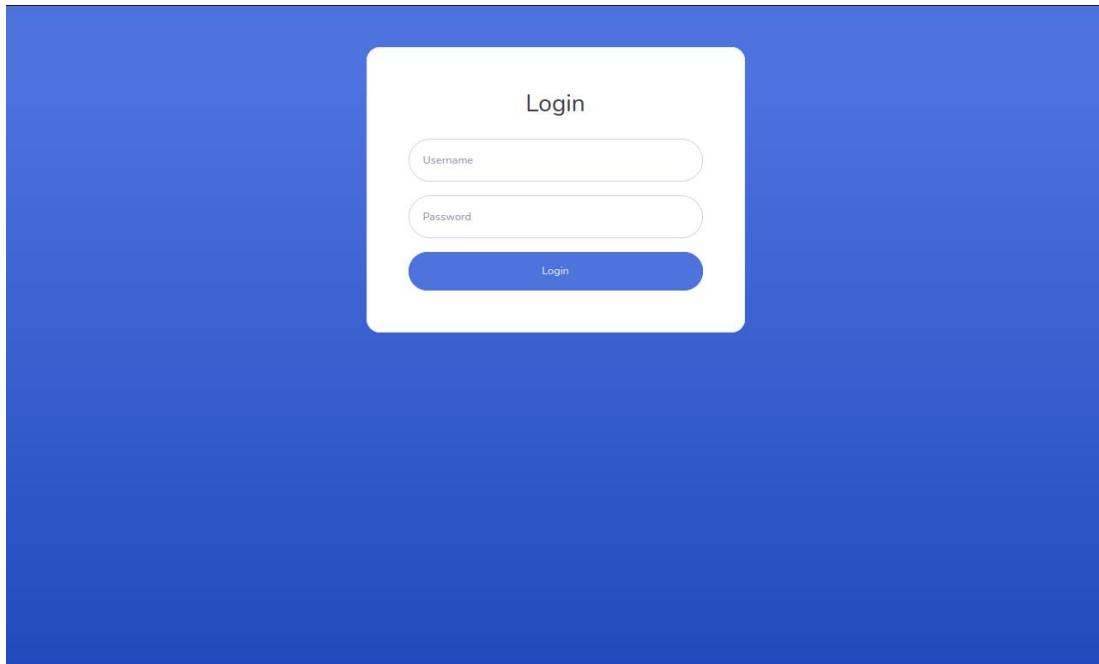
<i>Operating System</i>	<i>Windows 10</i>
<i>Processor</i>	Intel(R) Core(TM) i3-8145u CPU @ 2.10GHz (4 CPUs), ~2.3GHz 64-bit processor
<i>RAM</i>	8 GB
<i>Storage</i>	1 TB
<i>Graphic</i>	All graphic card
<i>Requirement Software</i>	<i>Google Chrome</i>

Spesifikasi minimum pada perangkat komputer *user* tidak menjadi masalah utama karena aplikasi tetap dapat dibuka selama memiliki *Memory RAM* 4GB, dan memiliki *software web browser*.

## 4.2 Hasil Perancangan

### 4.2.1 Login

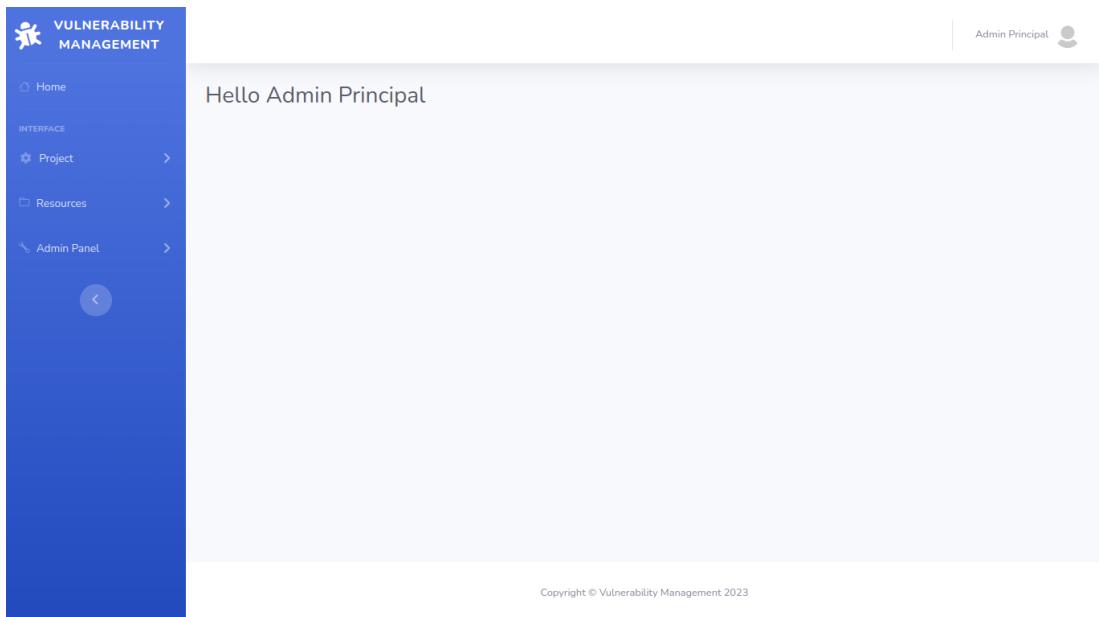
Gambar di bawah merupakan tampilan awal dari aplikasi *vulnerability management* yaitu *Login page* yang pertama kali ditampilkan ketika pengguna membuka aplikasi ini. Terdapat dua *form* yaitu *username* dan *password* serta tombol *submit* sebagai autentikasi untuk pengguna yang terdaftar pada aplikasi.



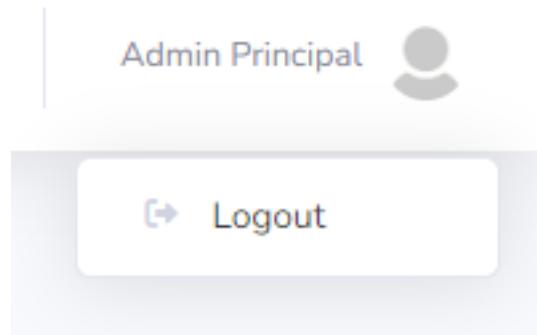
**Gambar 4.1 Tampilan halaman *Login***

#### 4.2.2 *Home Page*

Gambar di bawah merupakan tampilan *Home Page* setelah pengguna sudah sukses *login* ke dalam aplikasi menggunakan *username* dan *password* yang benar. Pada pojok kanan atas dari aplikasi, terdapat informasi pengguna dan juga memunculkan menu *logout* jika ditekan, bisa dilihat pada Gambar 4.3.



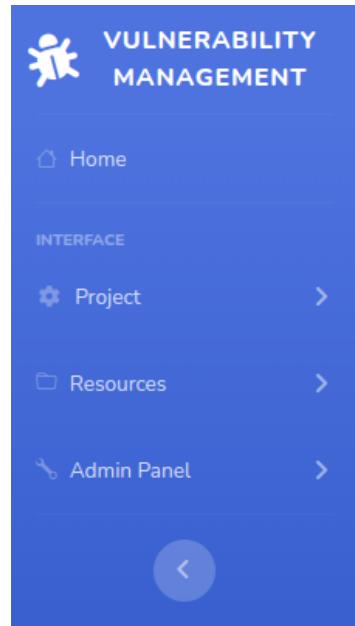
**Gambar 4.2 Tampilan *Home Page***



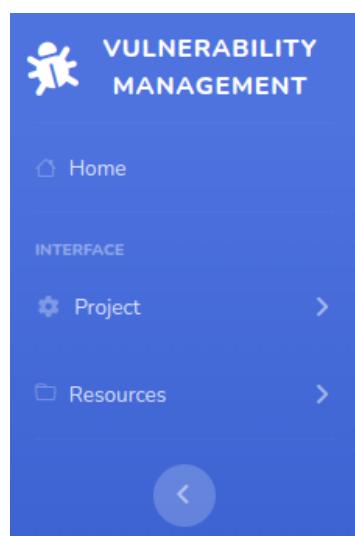
**Gambar 4.3 Tampilan Menu *Logout***

#### **4.2.2.1 Side Navigation Bar**

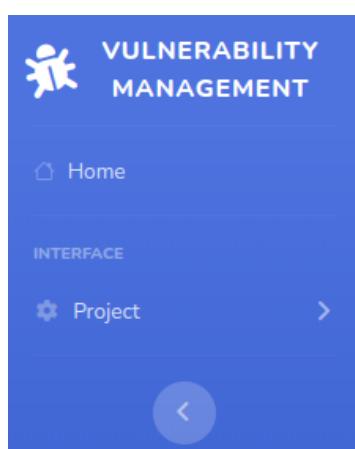
Pada *side navigation bar* terdapat beberapa menu yang menjadi fitur utama pada aplikasi. Tampilan *side navigation bar* untuk setiap *role* berbeda-beda, untuk *role Principal Penetration Tester* memiliki menu fitur *Project*, *Resources* dan *Admin Panel*, bisa dilihat pada Gambar 4.4. Sedangkan untuk *role Penetration Tester* memiliki menu fitur *Project* dan *Resources*, bisa dilihat pada Gambar 4.5. Untuk *role Client* hanya memiliki menu fitur *Project*, bisa dilihat pada Gambar 4.6. Gambar di bawah menunjukkan perbedaan tampilan *side navigation bar* di beberapa *role* pengguna.



Gambar 4.4 Tampilan Side Navigation Bar Principal Penetration Tester



Gambar 4.5 Tampilan Side Navigation Bar Penetration Tester



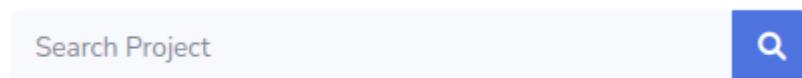
Gambar 4.6 Tampilan Side Navigation Bar Client

### 4.2.3 Project

Fitur *Project* merupakan fitur yang berfungsi sebagai manajemen *project penetration testing*. Pada fitur ini hanya pengguna dengan *role Principal Penetration Tester* yang mempunyai aksi seperti menambahkan, mengubah dan menghapus suatu *project*, sedangkan untuk *role Penetration Tester* dan *Client* hanya bisa melihat *project* beserta detailnya.

#### 4.2.3.1 Search Project

Gambar di bawah merupakan tampilan menu *Search Project*. Menu ini terdapat pada bagian atas dari halaman yang berfungsi mencari dan menampilkan semua *project* sesuai dengan kata kunci yang dicari.



**Gambar 4.7 Tampilan Search Project**

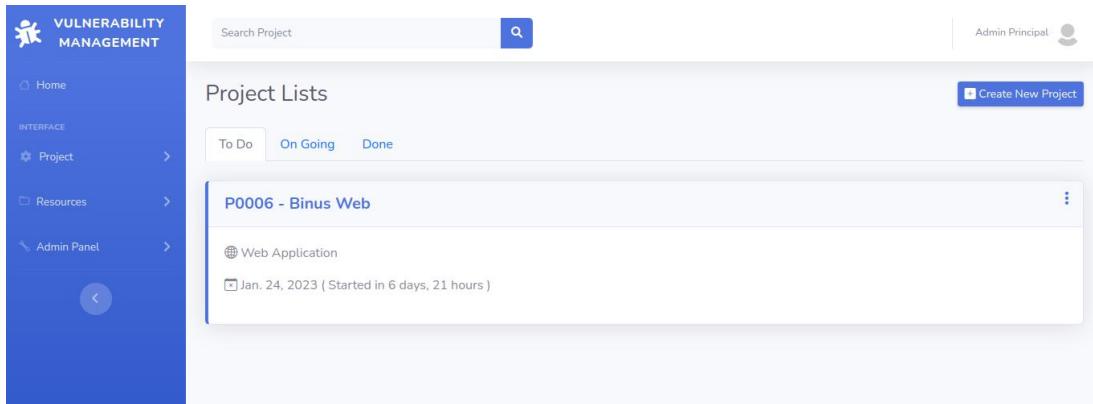
 A screenshot of the 'Search Results' page from a web application. The sidebar on the left shows navigation links for Home, Project, Resources, and Admin Panel. The main area displays three project entries in a list format:
 

- P0001 - Binusmaya
  - Web Application
  - Dec. 25, 2022 (1 month, 2 weeks)
- P0002 - Binus Mobile
  - Mobile Application
  - Oct. 20, 2022 (1 week, 5 days)
- P0003 - New Binusmaya
  - Web Application
  - Oct. 8, 2022 (0 minutes)

**Gambar 4.8 Tampilan Hasil Pencarian dari Search Project**

#### 4.2.3.2 Project Lists (To Do)

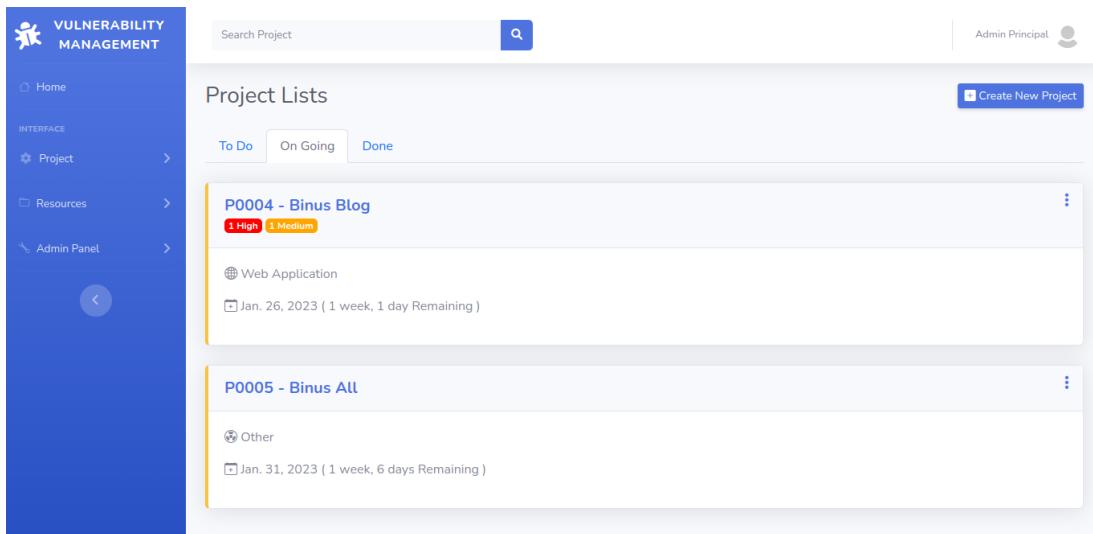
Gambar di bawah merupakan tampilan menu *Project Lists* yang berstatus *To Do*. Menu ini menampilkan semua *project* yang diikuti oleh pengguna dengan proses penggerjaan yang akan dimulai.



**Gambar 4.9 Tampilan *Project List* dengan status *To Do***

#### 4.2.3.3 *Project Lists (On Going)*

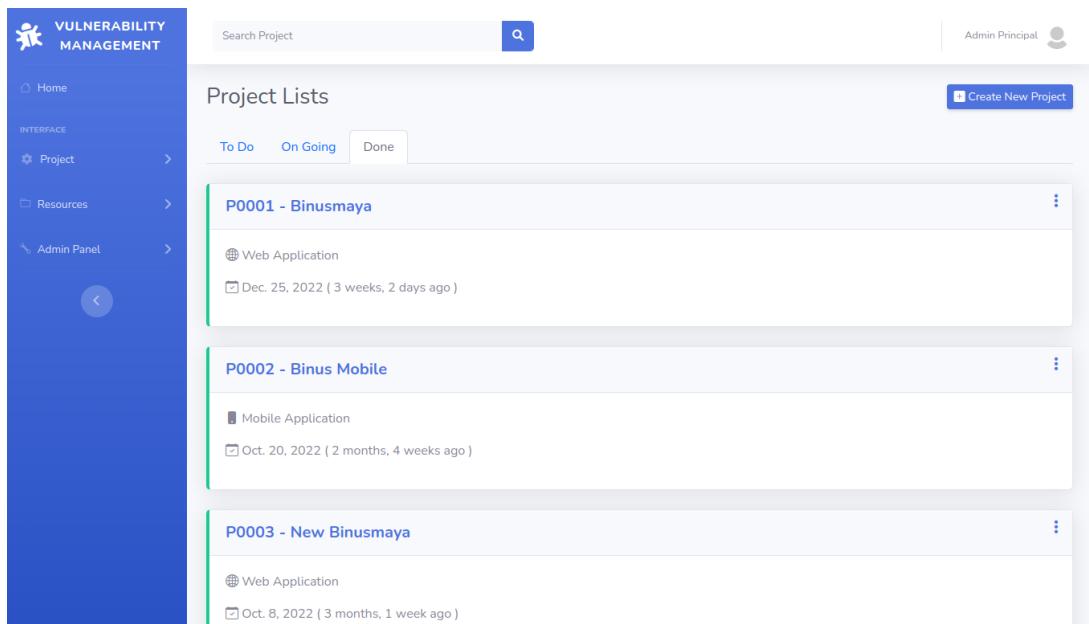
Gambar di bawah merupakan tampilan menu *Project Lists* yang berstatus *On Going*. Menu ini menampilkan semua *project* yang diikuti oleh pengguna dengan proses penggerjaan yang sedang berjalan.



**Gambar 4.10 Tampilan *Project List* dengan status *On Going***

#### 4.2.3.4 *Project Lists (Done)*

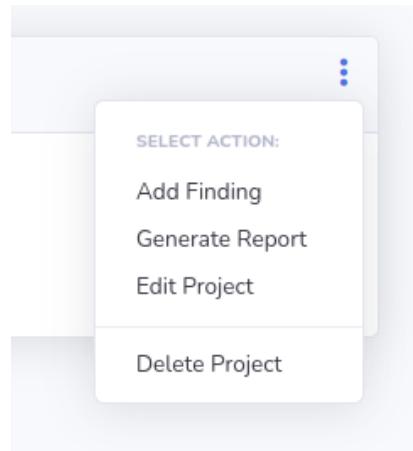
Gambar di bawah merupakan tampilan menu *Project Lists* yang berstatus *Done*. Menu ini menampilkan semua *project* yang diikuti oleh pengguna dengan proses penggerjaan sudah selesai dikerjakan.



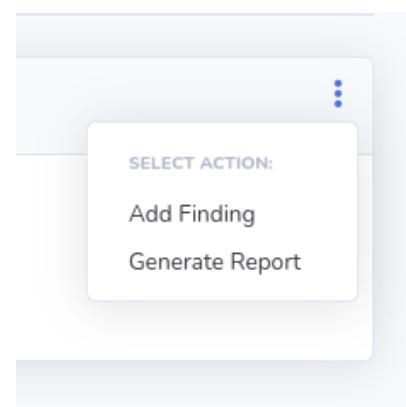
**Gambar 4.11 Tampilan *Project List* dengan status *Done***

#### 4.2.3.5 *Icon Titik Tiga Project*

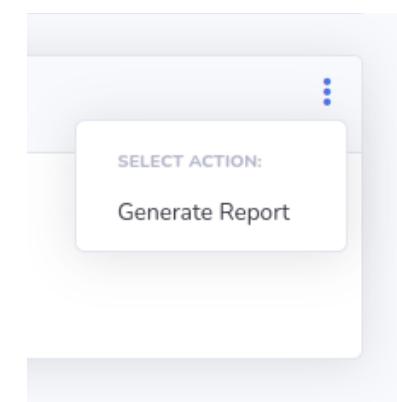
*Icon Titik Tiga Project* merupakan menu navigasi untuk memilih aksi yang ingin dilakukan pada suatu *project*. Tampilan dari aksi *Icon Titik Tiga Project* untuk setiap *role* berbeda-beda, untuk *role Principal Penetration Tester* memiliki aksi *Add Finding*, *Generate Report*, *Edit Project* dan *Delete Project*, bisa dilihat pada Gambar 4.12. Sedangkan untuk *role Penetration Tester* memiliki aksi *Add Finding* dan *Generate Report*, bisa dilihat pada Gambar 4.13. Untuk *role Client* hanya memiliki aksi *Generate Report*, bisa dilihat pada Gambar 4.14. Gambar di bawah menunjukkan perbedaan tampilan *icon titik tiga project* di beberapa *role* pengguna.



**Gambar 4.12 Tampilan Icon Titik Tiga Project di role Principal Penetration Tester**



**Gambar 4.13 Tampilan Icon Titik Tiga Project di role Penetration Tester**



**Gambar 4.14 Tampilan Icon Titik Tiga Project di role Client**

#### **4.2.3.6 Create New Project**

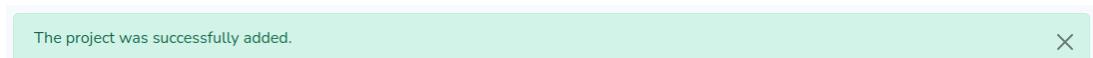
Gambar di bawah merupakan tampilan menu *Create New Project*, menu ini akan tampil setelah pengguna menekan tombol *Create New Project* pada menu *Project Lists*. Menu ini berfungsi bagi pengguna dengan *role*

*Principal Penetration Tester* untuk menambahkan suatu *project penetration testing* pada aplikasi. Pada menu ini ditampilkan *form* yang dapat diisi oleh pengguna mengenai detail dari *project* seperti *Project Name, Project Type, Start Date, End Date, Description, Object Pentest, Members, Client*.

The screenshot shows the 'Create New Project' interface. On the left sidebar, under 'INTERFACE', 'Project' is selected. The main area has a search bar at the top right. Below it, the 'Create New Project' section contains fields for 'Project Name', 'Project Type' (dropdown), 'Start Date' (date input), 'End Date' (date input), and 'Description'. Below this is a 'Objects Pentest' section with a table header ('Name Object', 'URL/IP Address', 'Action') and a message 'No Testing Objects Added'. There is also a 'Select Team Members' button. The 'Members' section has a similar table structure with a header ('Name', 'Email') and a message 'No Members Selected', with a 'Select Client' button. The 'Client' section also has a table structure with a header ('Name', 'Email') and a message 'No Clients Selected', with a 'Select Client' button. At the bottom left is an 'Add New Project' button, and at the bottom center is a copyright notice 'Copyright © Vulnerability Management 2023'.

**Gambar 4.15 Tampilan *Create New Project***

Setelah pengguna mengisi *form* dan menekan tombol *Add New Project* pada bagian bawah, data akan diproses untuk disimpan ke *database*. Jika proses penambahan *project* berhasil, sistem akan menampilkan *alert* sukses, bisa dilihat pada Gambar 4.16. Sedangkan jika penambahan *project* gagal, sistem akan menampilkan *alert* gagal, bisa dilihat pada Gambar 4.17.



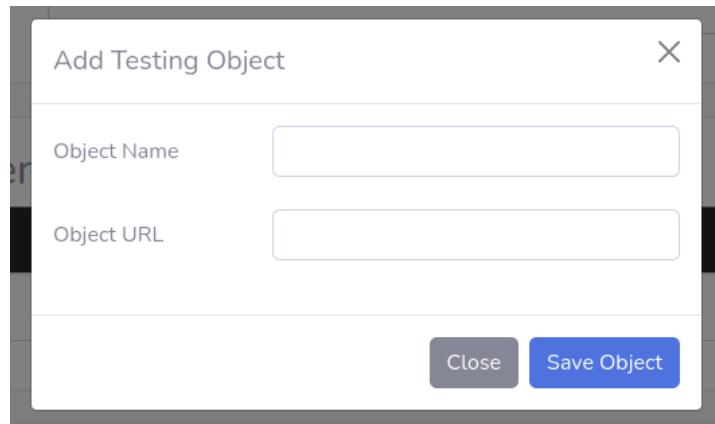
**Gambar 4.16 Tampilan Alert Sukses Create New Project**



**Gambar 4.17 Tampilan Alert Gagal Create New Project**

#### 4.2.3.7 Add Testing Object

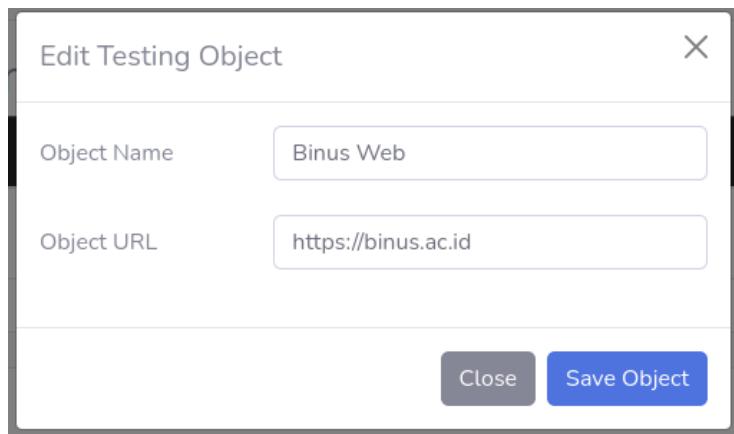
Gambar di bawah merupakan tampilan *modal Add Testing Object*, *modal* ini akan tampil setelah pengguna menekan tombol *Add Testing Object* pada bagian *Object Pentest*. Menu ini berfungsi untuk menambahkan objek *testing* dari *project penetration testing*. Pada menu ini ditampilkan *form* yang dapat diisi oleh pengguna guna memperjelas objek *testing* seperti *Object Name* dan *Object URL*.



**Gambar 4.18 Tampilan Add Testing Object pada Create New Project dan Edit Project**

#### 4.2.3.8 Edit Testing Object

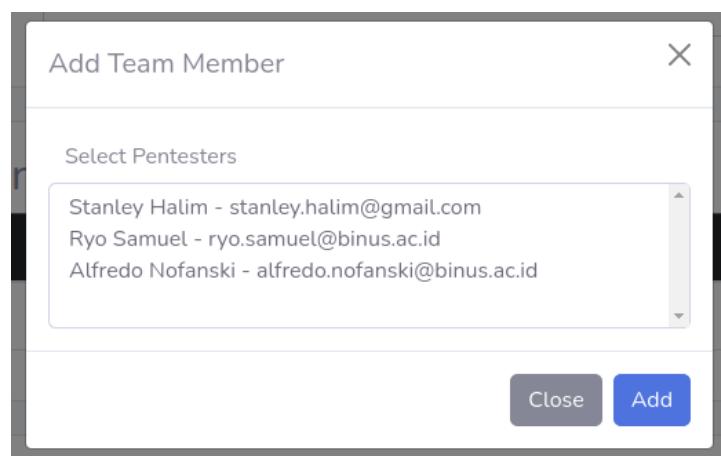
Gambar di bawah merupakan tampilan *modal Edit Testing Object*, *modal* ini akan tampil setelah pengguna menekan tombol *Edit* pada bagian *Object Pentest* yang dipilih. Menu ini berfungsi untuk untuk mengubah detail dari suatu *object penetration testing* yang dipilih. Pada menu ini ditampilkan *form* yang sudah terisi informasi *object penetration testing* yang sudah tersimpan sebelumnya dan dapat diubah oleh pengguna.



**Gambar 4.19 Tampilan *Edit Testing Object* pada *Create New Project* dan *Edit Project***

#### 4.2.3.9 Add Team Member

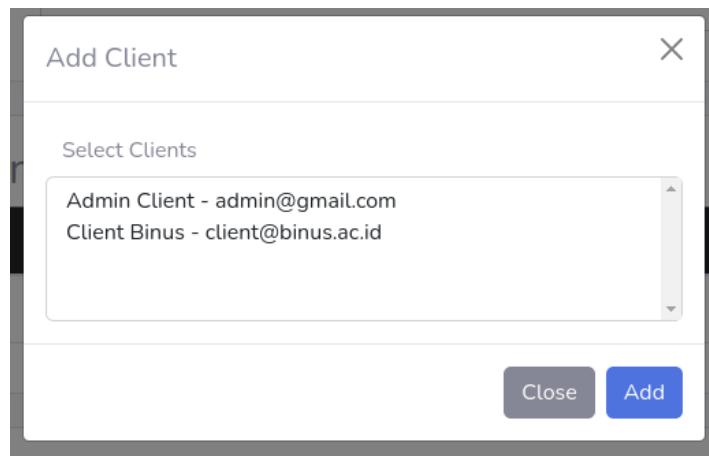
Gambar di bawah merupakan tampilan *modal Add Team Member*, *modal* ini akan tampil setelah pengguna menekan tombol *Select Team Member* pada bagian *Members*. Menu ini berfungsi untuk menambahkan pengguna lain dalam pengerjaan *project penetration testing* sebagai *penetration tester*. Pada menu ini ditampilkan daftar pengguna yang memiliki *role Penetration Tester* dan dapat memilih satu atau lebih *penetration tester* yang terlibat dalam *project* tersebut.



**Gambar 4.20 Tampilan *Add Team Member* pada *Create New Project* dan *Edit Project***

#### 4.2.3.10 Add Client

Gambar di bawah merupakan tampilan *modal Add Client*, *modal* ini akan tampil setelah pengguna menekan tombol *Select Client* pada bagian *Client*. Menu ini berfungsi untuk menambahkan pengguna lain dalam pengerjaan *project penetration testing* sebagai *client*. Pada menu ini ditampilkan daftar pengguna yang memiliki *role Client* dan dapat memilih satu atau lebih *client* yang terlibat dalam *project* tersebut.



**Gambar 4.21 Tampilan Add Client pada Create New Project dan Edit Project**

#### 4.2.3.11 Edit Project

Gambar di bawah merupakan tampilan menu *Edit Project*, menu ini akan tampil setelah pengguna menekan tombol *Edit Project* pada menu navigasi *Icon Titik Tiga Project* di *Project Lists*. Menu ini berfungsi bagi pengguna dengan *role Principal Penetration Tester* untuk mengubah detail dari suatu *project penetration testing* yang dipilih. Pada menu ini ditampilkan *form* yang sudah terisi informasi *project* yang sudah tersimpan sebelumnya dan dapat diubah oleh pengguna.

Search Project

Admin Principal

**Edit Project**

Project Name	Binus Blog
Project Type	Web Application
Start Date	01/09/2023
End Date	01/26/2023
Description	Deskripsi Pentest Binus Blog

**Objects Pentest**

Name Object	URL/IP Address	Action
Blog	http://blog.binus.ac.id	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

**Members**

Name	Email
Stanley Halim	stanley.halim@gmail.com
Ryo Samuel	ryo.samuel@binus.ac.id

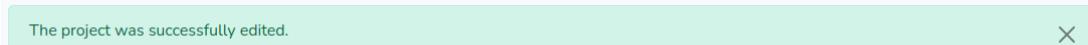
**Client**

Name	Email
Admin Client	admin@gmail.com

Copyright © Vulnerability Management 2023

**Gambar 4.22 Tampilan *Edit Project***

Setelah pengguna mengisi *form* dan menekan tombol *Edit Project* pada bagian bawah, data akan diproses untuk disimpan ke *database*. Jika proses perubahan *project* berhasil, sistem akan menampilkan *alert* sukses, bisa dilihat pada Gambar 4.23. Sedangkan jika pengubahan *project* gagal, sistem akan menampilkan *alert* gagal, bisa dilihat pada Gambar 4.24.



The project was successfully edited.

**Gambar 4.23 Tampilan Alert Sukses Edit Project**

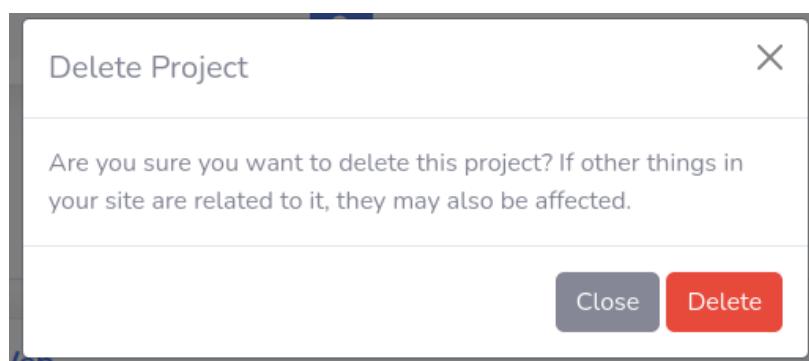


Unable to edit the project.

**Gambar 4.24 Tampilan Alert Gagal Edit Project**

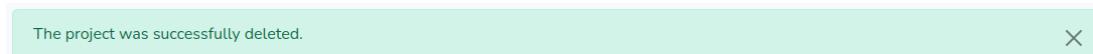
#### 4.2.3.12 Delete Project

Gambar di bawah merupakan tampilan *modal* konfirmasi *Delete Project*, *modal* ini akan tampil setelah pengguna menekan tombol *Delete Project* pada menu navigasi *Icon Titik Tiga Project* di *Project Lists*. Menu ini berfungsi bagi pengguna dengan *role Principal Penetration Tester* untuk menghapus suatu *project penetration testing* yang dipilih. Pada menu ini ditampilkan konfirmasi penghapusan, jika user menekan tombol *Delete* penghapusan akan diproses sedangkan jika user menekan tombol *Close* penghapusan akan dibatalkan.



**Gambar 4.25 Tampilan Konfirmasi Delete Project**

Setelah pengguna menekan tombol *Delete*, *project* akan diproses untuk dihapus dari *database*. Jika proses penghapusan *project* berhasil, sistem akan menampilkan *alert* sukses, bisa dilihat pada Gambar 4.26. Sedangkan jika penghapusan *project* gagal, sistem akan menampilkan *alert* gagal, bisa dilihat pada Gambar 4.27.



**Gambar 4.26 Tampilan Alert Sukses Delete Project**



**Gambar 4.27 Tampilan Alert Gagal Delete Project**

#### **4.2.3.13 View Detail Project**

Gambar di bawah merupakan tampilan menu *View Detail Project*, menu ini akan tampil setelah pengguna menekan salah satu *project* di *Project Lists*. Menu ini berfungsi untuk menampilkan detail dari suatu *project penetration testing* yang dipilih seperti *Project Name*, *Project Type*, *Description*, *Start Date*, *End Date*, *Client Details*, *Pentester Details*, *Object Details*, *Checklist* dan *Finding Lists*.

The screenshot shows the 'VULNERABILITY MANAGEMENT' application interface. On the left, a sidebar lists 'INTERFACE' options: Home, Project, Resources, and Admin Panel. The main content area is titled 'Detail Project' for 'P0006 - Binus Web'. It displays project details like 'Web Application', creation date ('Tue, 24 Jan 2023 - Wed, 29 Mar 2023 ( 2 months )'), and a description ('Binus Web merupakan web application binus pada https://binus.ac.id/. Dibuat dari Wordpress'). Below this are sections for 'Client Details' (with one entry: Name Client: Admin Client, Email Client: admin@gmail.com), 'Pentesters Details' (listing Stanley, Ryo, Admin, and Alfredo with their respective emails), 'Objects Details' (listing Binus Web with URL https://binus.ac.id), and 'WSTG Checklist' (with a link to a Google Sheets document). At the bottom, there's a 'Findings Lists' section with tabs for 'Not Fixed' (selected) and 'Fixed', and a table showing a single finding: Reflected XSS in Binus Web with CVSS Score 7.1, Severity High, Finder principal, Status Not Fixed, and an Action button.

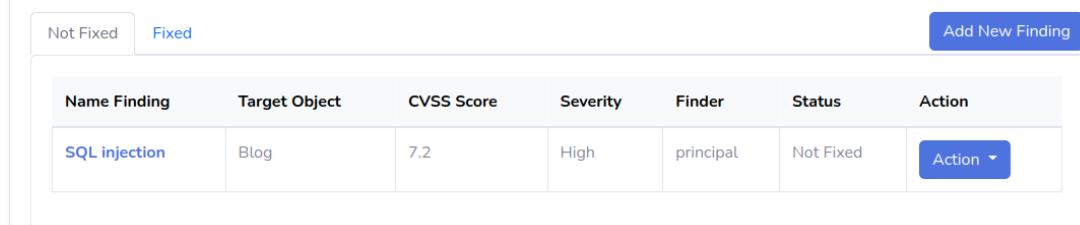
**Gambar 4.28 Tampilan View Detail Project**

#### 4.2.3.14 Not Fixed Finding Lists

Gambar di bawah merupakan tampilan menu *Not Fixed Finding Lists*, menu ini akan tergabung pada menu *View Detail Project* pada bagian *Finding Lists*. Menu ini akan tampil setelah pengguna membuka *View Detail Project* atau menekan tab *Not Fixed* pada bagian *Finding Lists*. Menu ini

berfungsi untuk menampilkan daftar *finding* yang belum diperbaiki oleh *client* dari *project* yang dipilih.

#### Findings Lists



The screenshot shows a table titled 'Findings Lists' with a tab for 'Not Fixed'. A single row is displayed with the following data:

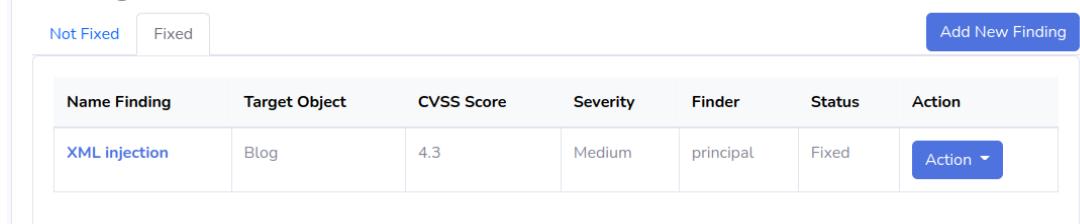
Name Finding	Target Object	CVSS Score	Severity	Finder	Status	Action
SQL injection	Blog	7.2	High	principal	Not Fixed	<button>Action</button>

Gambar 4.29 Tampilan *Not Fixed Finding Lists*

#### 4.2.3.15 Fixed Finding Lists

Gambar di bawah merupakan tampilan menu *Fixed Finding Lists*, menu ini akan tergabung pada menu *View Detail Project* pada bagian *Finding Lists*. Menu ini akan tampil setelah pengguna membuka *View Detail Project* atau menekan tab *Fixed* pada bagian *Finding Lists*. Menu ini berfungsi untuk menampilkan daftar *finding* yang sudah diperbaiki oleh *client* dari *project* yang dipilih.

#### Findings Lists



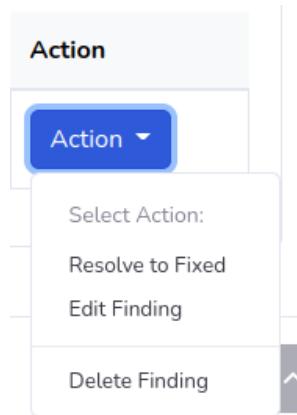
The screenshot shows a table titled 'Findings Lists' with a tab for 'Fixed'. A single row is displayed with the following data:

Name Finding	Target Object	CVSS Score	Severity	Finder	Status	Action
XML injection	Blog	4.3	Medium	principal	Fixed	<button>Action</button>

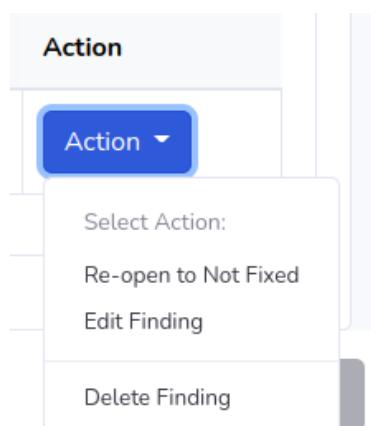
Gambar 4.30 Tampilan *Fixed Finding Lists*

#### 4.2.3.16 Action Finding

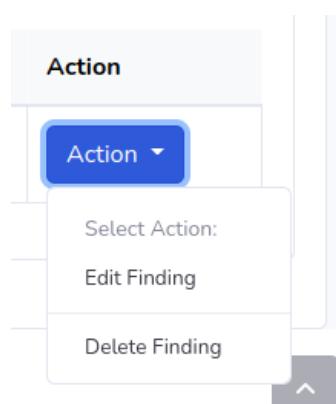
*Action Finding* merupakan menu navigasi untuk memilih aksi yang ingin dilakukan pada suatu *finding* di *project* yang dipilih. Tampilan dari aksi *Action Finding* untuk hanya ditampilkan pada *role Principal Penetration Tester* dan *Penetration Tester*. Untuk *role Principal Penetration Tester* memiliki aksi *Resolve to Fixed* atau *Re-open to Not Fixed*, *Edit Finding* dan *Delete Finding*, bisa dilihat pada Gambar 4.31. Sedangkan untuk *role Penetration Tester* memiliki aksi *Edit Finding* dan *Delete Finding*, bisa dilihat pada Gambar 4.32. Gambar di bawah menunjukkan perbedaan tampilan *icon* titik tiga *finding* di *role* tersebut.



Gambar 4.31 Tampilan *Action Finding* dengan status *Not Fixed* di role *Principal Penetration Tester*



Gambar 4.32 Tampilan *Action Finding* dengan status *Fixed* di role *Principal Penetration Tester*



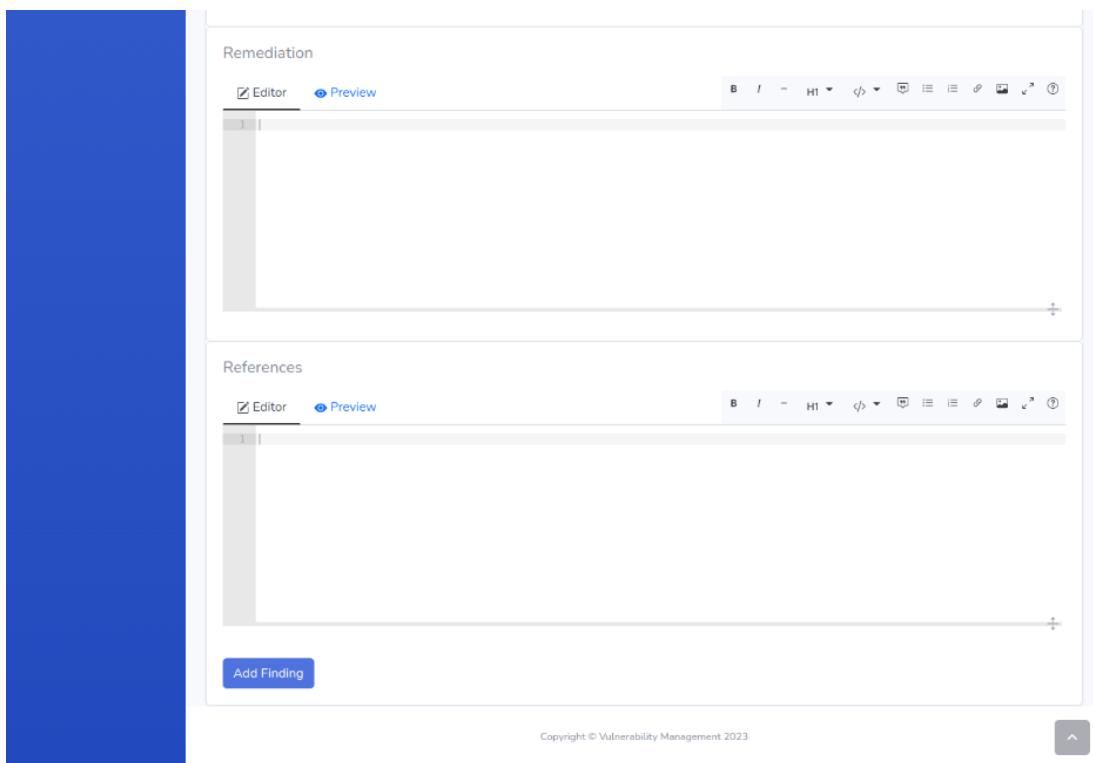
Gambar 4.33 Tampilan *Action Finding* pada semua status di role *Principal Penetration Tester*

#### **4.2.3.17 Add New Finding**

Gambar di bawah merupakan tampilan menu *Add New Finding*, menu ini akan tampil setelah pengguna menekan tombol *Add Finding* pada menu navigasi *Icon Titik Tiga Project* di *Project Lists* atau menekan tombol *Add New Finding* pada menu navigasi *Action Finding* di *View Detail Project*. Menu ini berfungsi bagi pengguna dengan *role Principal Penetration Tester* dan *Penetration Tester* untuk menambahkan suatu temuan *vulnerability* di *project penetration testing* yang dipilih. Pada menu ini ditampilkan *form* yang dapat diisi oleh pengguna mengenai detail dari *finding* seperti *Object Project*, *Finding Name*, *Weakness*, *Severity*, *Description*, *Impact*, *Evidences*, *Remediation* dan juga *References*.

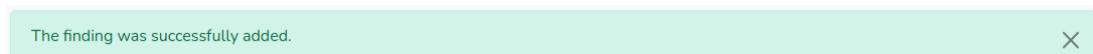
The screenshot shows the 'Add New Finding' page within a 'VULNERABILITY MANAGEMENT' application. The left sidebar includes navigation links for Home, Project (selected), Resources, and Admin Panel. The main content area has a search bar and a user profile for 'Admin Principal'. The 'Add New Finding' form is displayed, showing the project path 'My Project / P0006 - Binus Web ( Todo ) / Add New Finding'. The form fields include 'Object Project' (dropdown), 'Finding Name' (text input), 'Weakness' (dropdown set to 'External Control of System or Configuration Setting (CWE-15)'), 'Severity' (dropdown), and a 'Base Score' section with various metrics like Attack Vector (AV) and Scope (S). A note says 'Select values for all base metrics to generate score'. Below this is a green button for 'Vector String - select values for all base metrics to generate a vector'. The 'Description' and 'Impact' sections each have an 'Editor' and 'Preview' tab, both currently showing a blank rich text area. The 'Evidences' section has a table with columns 'Description', 'Images', and 'Action', and a note 'No Evidence Added'.

**Gambar 4.34 Tampilan *Add New Finding***



**Gambar 4.35 Tampilan *Add New Finding* lanjutan**

Setelah pengguna mengisi *form* dan menekan tombol *Add Finding* pada bagian bawah, data akan diproses untuk disimpan ke *database*. Jika proses penambahan *finding* berhasil, sistem akan menampilkan *alert* sukses, bisa dilihat pada Gambar 4.36. Sedangkan jika penambahan *finding* gagal, sistem akan menampilkan *alert* gagal, bisa dilihat pada Gambar 4.37.



**Gambar 4.36 Tampilan *Alert* Sukses *Add New Finding***



**Gambar 4.37 Tampilan *Alert* Gagal *Add New Finding***

#### 4.2.3.18 *Add Evidence*

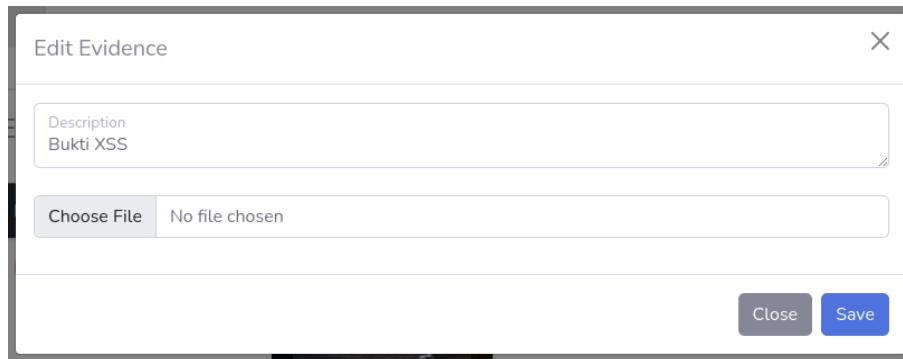
Gambar di bawah merupakan tampilan *modal Add Evidence*, *modal* ini akan tampil setelah pengguna menekan tombol *Add Evidence* pada bagian *Evidences*. Menu ini berfungsi untuk menambahkan bukti berupa penjelasan dan juga gambar dari temuan suatu *vulnerability*.



**Gambar 4.38 Tampilan Add Evidence**

#### 4.2.3.19 Edit Evidence

Gambar di bawah merupakan tampilan *modal Edit Evidence*, *modal* ini akan tampil setelah pengguna menekan tombol *Edit* pada bagian *Evidences* yang dipilih. Menu ini berfungsi untuk mengubah detail dari suatu *evidence* yang dipilih. Pada menu ini ditampilkan *form* yang sudah terisi informasi bukti dari suatu *finding* yang sudah tersimpan sebelumnya dan dapat diubah oleh pengguna.



**Gambar 4.39 Tampilan Edit Evidence**

#### 4.2.3.20 Edit Finding

Gambar di bawah merupakan tampilan menu *Edit Finding*, menu ini akan tampil setelah pengguna menekan tombol *Edit Project* pada menu navigasi *Action Finding* di *View Detail Project*. Menu ini berfungsi bagi pengguna dengan *role Principal Penetration Tester* dan *Penetration Tester* untuk mengubah detail dari suatu *finding* di *project penetration testing* yang dipilih. Pada menu ini ditampilkan *form* yang sudah terisi informasi *finding* yang sudah tersimpan sebelumnya dan dapat diubah oleh pengguna.

The screenshot shows the 'Edit Finding' page of a vulnerability management tool. The left sidebar has a blue header 'VULNERABILITY MANAGEMENT' with icons for Home, Project, Resources, and Admin Panel. The main area has a search bar 'Search Project' and a user icon 'Admin Principal'. The title 'Edit Finding' is at the top, followed by the breadcrumb 'My Project / P0006 - Binus Web (Todo) / Reflected XSS / Edit Finding'. The 'Object Project' is set to 'Binus Web - https://binus.ac.id'. The 'Finding Name' is 'Reflected XSS'. The 'Weakness' is 'Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS) (CWE-80)'. The 'Severity' section shows a base score of 7.1 (High). The 'Base Score' is calculated based on various factors: Attack Vector (AV) [Network (N)], Attack Complexity (AC) [Low (L), High (H)], Privileges Required (PR) [None (N), Low (L), High (H)], User Interaction (UI) [None (N), Required (R)], Scope (S) [Unchanged (U), Changed (C)], Confidentiality (C) [None (N), Low (L), High (H)], Integrity (I) [None (N), Low (L), High (H)], and Availability (A) [None (N), Low (L), High (H)].

Gambar 4.40 Tampilan *Edit Finding*

Vector String -  
CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:L

Description

Editor  Preview

1 XSS Ditemukan pada Endpoint <https://binus.ac.id>, menggunakan payload <script>alert(1)</script>

Impact

Editor  Preview

1 Cookie Stealing, Account Takeover

Evidences

Add Evidence

Description	Images	Action
Bukti XSS		<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Remediation

Editor  Preview

1 Sanitasi dan Encoding

References

Editor  Preview

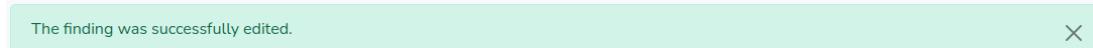
1 OWASP

Save Finding

Copyright © Vulnerability Management 2023

Gambar 4.41 Tampilan *Edit Finding* lanjutan

Setelah pengguna mengisi *form* dan menekan tombol *Save Finding* pada bagian bawah, data akan diproses untuk disimpan ke *database*. Jika proses perubahan *finding* berhasil, sistem akan menampilkan *alert* sukses, bisa dilihat pada Gambar 4.42. Sedangkan jika perubahan *finding* gagal, sistem akan menampilkan *alert* gagal, bisa dilihat pada Gambar 4.43.



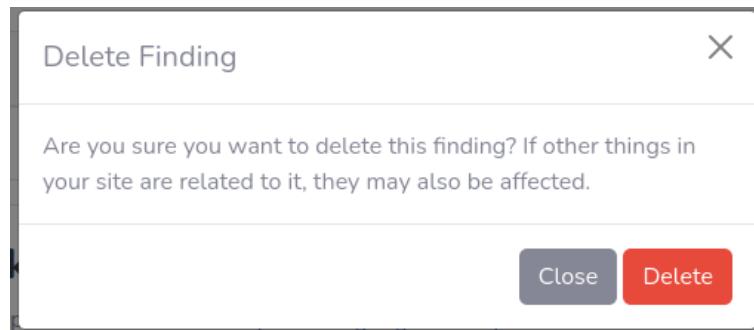
**Gambar 4.42 Tampilan Alert Sukses Edit Finding**



**Gambar 4.43 Tampilan Alert Gagal Edit Finding**

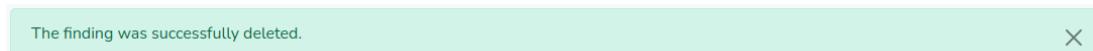
#### 4.2.3.21 Delete Finding

Gambar di bawah merupakan tampilan *modal* konfirmasi *Delete Finding*, *modal* ini akan tampil setelah pengguna menekan tombol *Delete Finding* pada menu navigasi *Action Finding* di *View Detail Project*. Menu ini berfungsi bagi pengguna dengan *role Principal Penetration Tester* dan *Penetration Tester* untuk menghapus suatu *finding* di *project* yang dipilih. Pada menu ini ditampilkan konfirmasi penghapusan, jika user menekan tombol *Delete* penghapusan akan diproses sedangkan jika user menekan tombol *Close* penghapusan akan dibatalkan.



**Gambar 4.44 Tampilan Konfirmasi Delete Finding**

Setelah pengguna menekan tombol *Delete*, *finding* akan diproses untuk dihapus dari *database*. Jika proses penghapusan *finding* berhasil, sistem akan menampilkan *alert* sukses, bisa dilihat pada Gambar 4.45. Sedangkan jika penghapusan *finding* gagal, sistem akan menampilkan *alert* gagal, bisa dilihat pada Gambar 4.46.



**Gambar 4.45 Tampilan Alert Sukses Delete Finding**



**Gambar 4.46 Tampilan Alert Gagal Delete Finding**

#### **4.2.3.22 View Detail Finding**

Gambar di bawah merupakan tampilan menu *View Detail Finding*, menu ini akan tampil setelah pengguna menekan salah satu *finding* di *View Detail Project* bagian *Finding Lists*. Menu ini berfungsi untuk menampilkan detail dari suatu *finding* yang ditemukan dari *project penetration testing* yang dipilih seperti *Object Testing, Finding Name, Weakness, Severity - CVSS, Description, Impact, Evidences, Remediation, References*.

The screenshot shows the 'Detail Finding' page of a vulnerability management system. The left sidebar has a blue header 'VULNERABILITY MANAGEMENT' with icons for Home, Project, Resources, and Admin Panel. The main content area has a search bar at the top. The title 'Detail Finding' is followed by the breadcrumb 'My Project / P0006 - Binus Web ( Todo ) / Reflected XSS'. The finding details are listed in rows:

- Object Testing:** Binus Web - <https://binus.ac.id>
- Finding Name:** Reflected XSS
- Weakness:** Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS)
- Severity - CVSS:** High ( 7.1 ) - CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:L
- Description:** XSS Ditemukan pada Endpoint <https://binus.ac.id>, menggunakan payload <script>alert(1)</script>
- Impact:** Cookie Stealing, Account Takeover

Below this is a section titled 'Evidences' with a table:

Description	Images
Bukti XSS	

Other sections include 'Remediation' (Sanitasi dan Encoding) and 'References' (OWASP).

**Gambar 4.47 Tampilan View Detail Finding**

#### 4.2.3.23 Change Finding Status

*Change Finding Status* terdapat pada *Action Finding*, terdapat dua pilihan status dari *finding* yaitu *Resolve to Fixed* atau *Re-open to Not Fixed*. Untuk gambar bisa mengacu pada gambar 4.29 untuk pilihan *Resolve to Fixed* dan 4.30 untuk pilihan *Re-open to Not Fixed*. Menu ini berfungsi bagi pengguna dengan role *Principal Penetration Tester* untuk mengganti status dari suatu finding.

Setelah pengguna menekan dan memilih salah satu dari pilihan *status*, sistem akan memproses untuk melakukan perubahan pada *database*.

Jika proses perubahan status *finding* berhasil, sistem akan menampilkan *alert* sukses, bisa dilihat pada Gambar 4.48. Sedangkan jika perubahan status *finding* gagal, sistem akan menampilkan *alert* gagal, bisa dilihat pada Gambar 4.49.



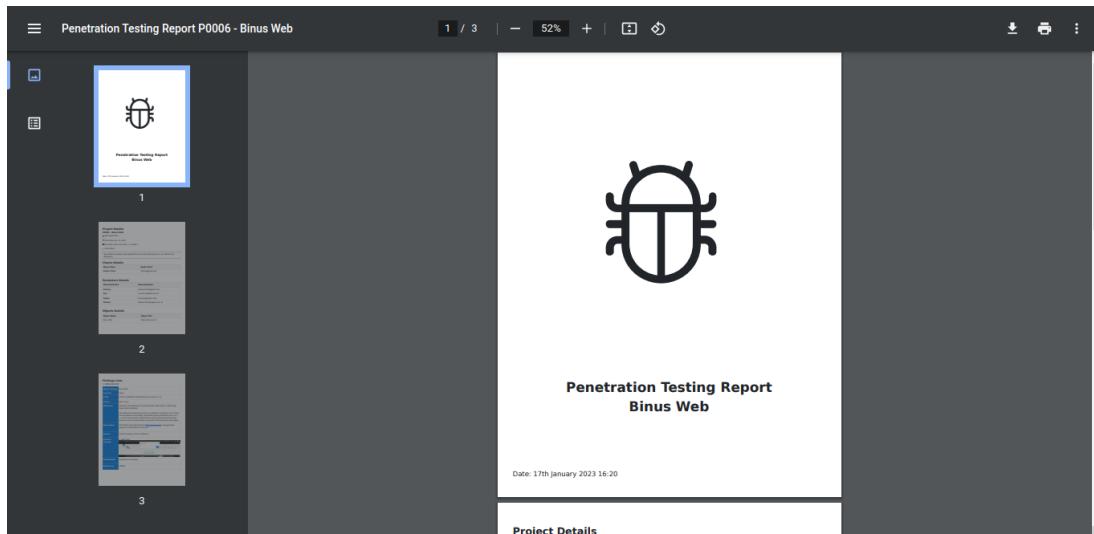
**Gambar 4.48 Tampilan Alert Sukses Change Finding Status**



**Gambar 4.49 Tampilan Alert Gagal Change Finding Status**

#### 4.2.3.24 Generate Report

Gambar di bawah merupakan tampilan menu *Generate Report*, menu ini akan tampil setelah pengguna menekan tombol *Generate Report* pada menu navigasi *Action Finding* di *View Detail Project*. Menu ini berfungsi bagi pengguna untuk menghasilkan laporan dari *project penetration testing* dalam bentuk PDF. Pada menu ini ditampilkan detail dari *project* beserta daftar *finding* dan detailnya.



**Gambar 4.50 Tampilan Generate Report**

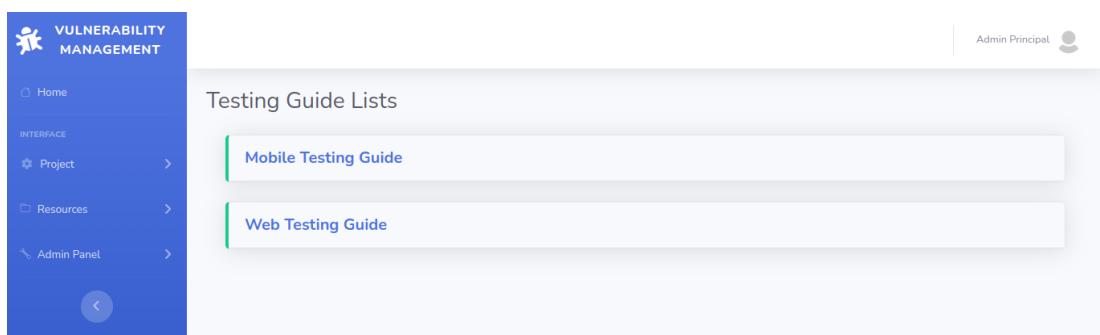
#### 4.2.4 Resources

Fitur *Resources* merupakan fitur yang berfungsi sebagai manajemen *Testing Guide* pada aplikasi. Pada fitur ini berguna untuk pengguna dengan *role*

*Principal Penetration Tester* dan *Penetration Tester* untuk membaca, menambahkan, mengubah dan menghapus panduan pengetesan.

#### 4.2.4.1 *Testing Guide Lists*

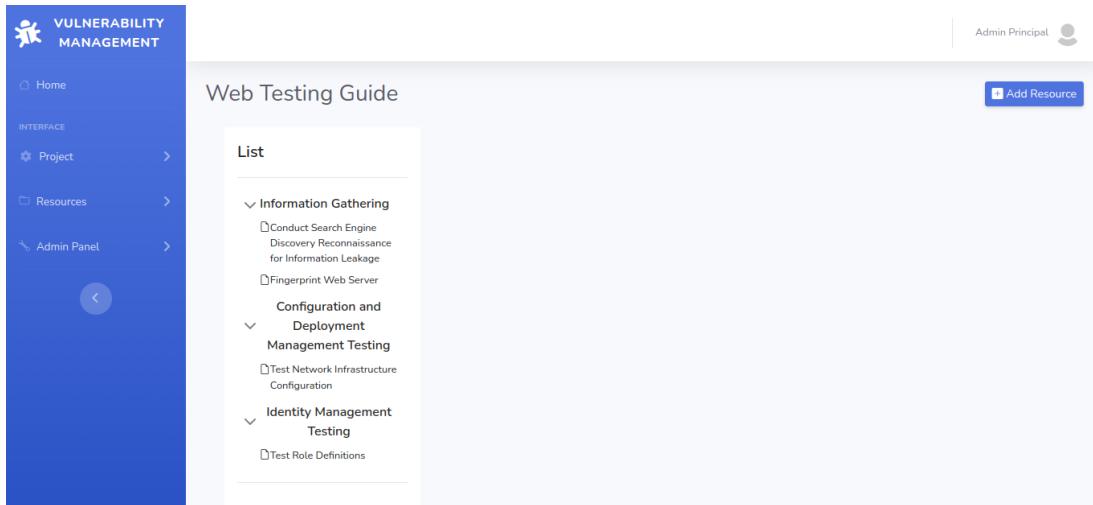
Gambar di bawah merupakan tampilan menu *Testing Guide Lists*, menu ini akan tampil setelah pengguna menekan tombol *List Testing Guide* pada menu *Resources*. Menu ini berfungsi bagi pengguna dengan *role Principal Penetration Tester* dan *Penetration Tester* untuk memilih jenis *testing guide* yang akan dibaca, diubah atau dihapus. Pada menu ini terdapat 2 jenis *Testing Guide* yaitu *Mobile Testing Guide* dan *Web Testing Guide*.



**Gambar 4.51 Tampilan *Testing Guide List***

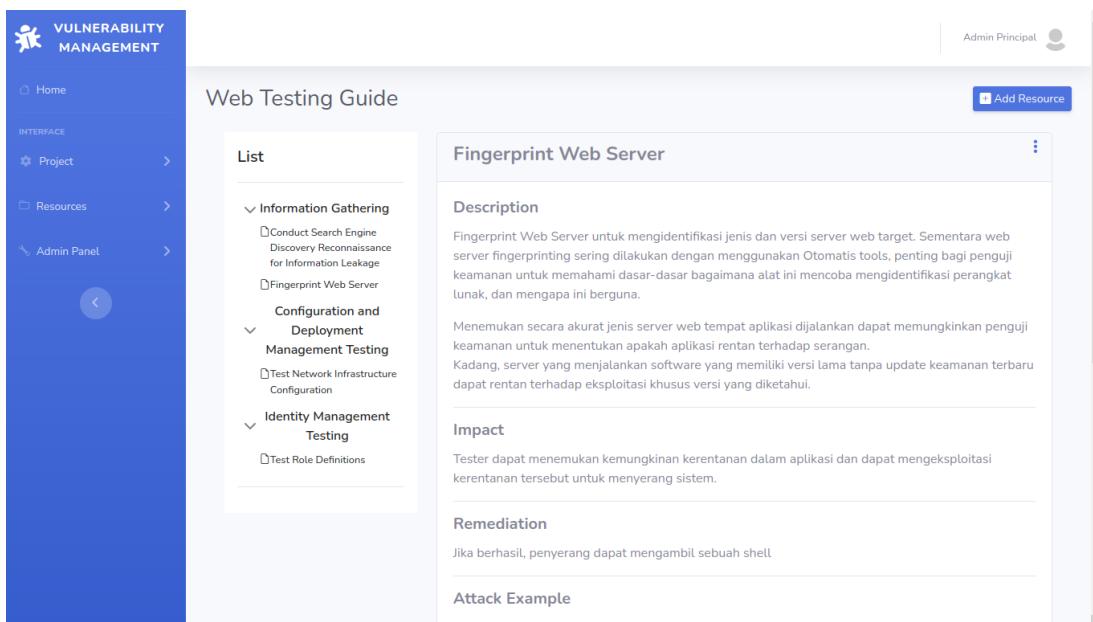
#### 4.2.4.2 *View Detail Testing Guide*

Gambar di bawah merupakan tampilan menu *View Detail Testing Guide*, menu ini akan tampil setelah pengguna memilih salah satu *Testing Guide* pada menu *Testing Guide Lists*. Menu ini berfungsi bagi pengguna dengan *role Principal Penetration Tester* dan *Penetration Tester* untuk melihat detail dari *testing guide* yang dipilih.



**Gambar 4.52 Tampilan View Detail Testing Guide**

Pengguna dapat kembali memilih salah satu dari daftar *Testing Guide* yang ditampilkan. Gambar di bawah merupakan tampilan saat pengguna sudah memilih kembali *testing guide* yang tersedia.

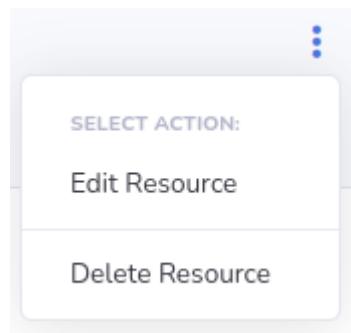


**Gambar 4.53 Tampilan Salah Satu Testing Guide**

#### 4.2.4.3 Icon Titik Tiga Testing Guide

*Icon Titik Tiga Testing Guide* merupakan menu navigasi untuk memilih aksi yang ingin dilakukan pada suatu *testing guide*. Tampilan dari aksi *Icon Titik Tiga Testing Guide* memiliki aksi *Edit Resource* dan *Delete*

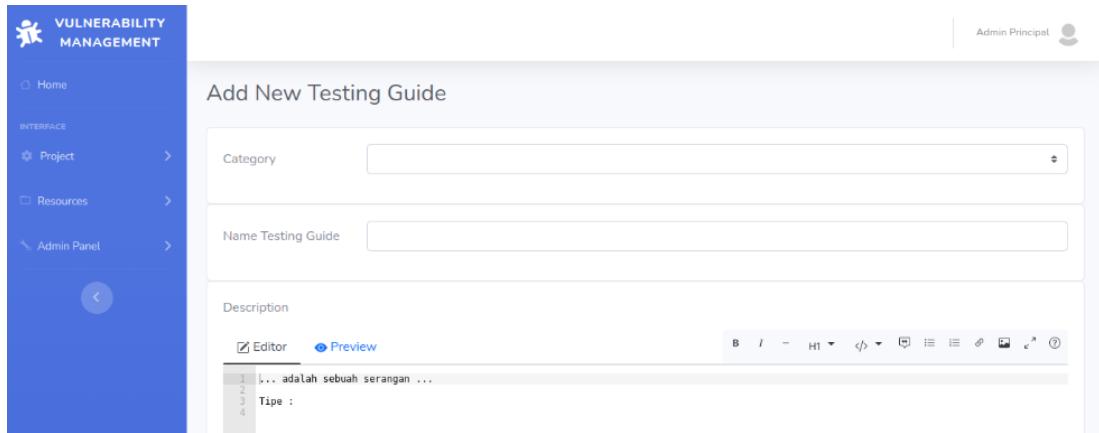
*Resource.* Gambar di bawah menunjukkan tampilan *icon titik tiga testing guide*.



**Gambar 4.54 Tampilan *Icon Titik Tiga Testing Guide***

#### **4.2.4.4 *Add Testing Guide***

Gambar di bawah merupakan tampilan menu *Add Testing Guide*, menu ini akan tampil setelah pengguna menekan tombol *Add Resource* pada menu *View Detail Testing Guide*. Menu ini berfungsi bagi pengguna dengan *role Principal Penetration Tester* dan *Penetration Tester* untuk menambahkan suatu panduan pengetesan. Pada menu ini ditampilkan *form* yang dapat diisi oleh pengguna mengenai detail dari *testing guide* seperti *Category*, *Name Testing Guide*, *Description*, *Impact*, *Remediation*, *Attack Example*, *Tools* dan *Other*.



The screenshot shows a user interface for creating a testing guide. It consists of five horizontal tabs, each with an 'Editor' and 'Preview' tab:

- Impact:**  
Editor tab content: 1. Dika berhasil, penyerang dapat ...  
Preview tab content: (empty)
- Remediation:**  
Editor tab content: 1. Pentester menyarankan untuk menggunakan ...  
Preview tab content: (empty)
- Attack Example:**  
Editor tab content:

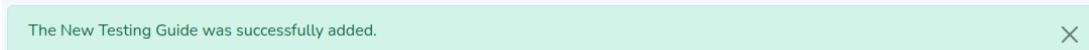
```
1. ##### Example 1
2. [Public report](https://hackerone.com)
3.
4. ##### Example 2
5. Telah ditemukan kelamahan ... pada path ... dengan payload ...
6.
7. ##### Example 3
8. Kelamahan - [File report](https://google.com)
```

Preview tab content: (empty)
- Tools:**  
Editor tab content: 1. [Katana](https://github.com/)  
Preview tab content: (empty)
- Other:**  
Editor tab content: 1. [Burp suite XSS](https://portswigger.net/)  
Preview tab content: (empty)

At the bottom left is a blue button labeled "Add Testing Guide". At the bottom right is a small upward-pointing arrow icon.

Gambar 4.56 Tampilan *Add Testing Guide* lanjutan

Setelah pengguna mengisi *form* dan menekan tombol *Add Testing Guide* pada bagian bawah, data akan diproses untuk disimpan ke *database*. Jika proses penambahan *testing guide* berhasil, sistem akan menampilkan *alert* sukses, bisa dilihat pada Gambar 4.57. Sedangkan jika penambahan *testing guide* gagal, sistem akan menampilkan *alert* gagal, bisa dilihat pada Gambar 4.58.



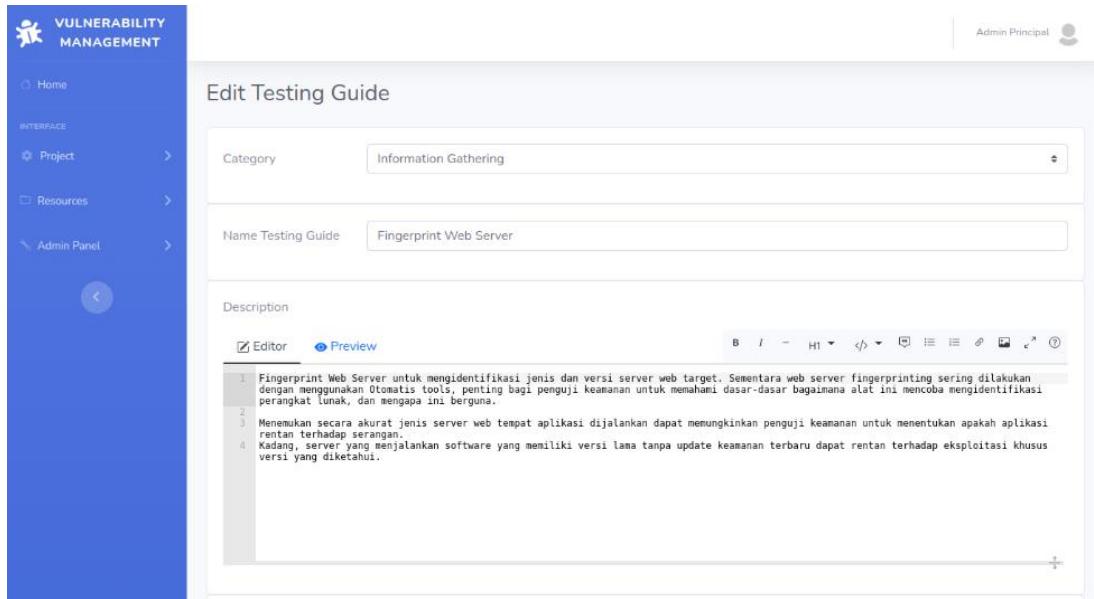
**Gambar 4.57 Tampilan Alert Sukses Add Testing Guide**



**Gambar 4.58 Tampilan Alert Gagal Add Testing Guide**

#### 4.2.4.5 *Edit Testing Guide*

Gambar di bawah merupakan tampilan menu *Edit Testing Guide*, menu ini akan tampil setelah pengguna menekan tombol *Edit Resource* pada menu navigasi *Icon Titik Tiga Testing Guide* di *View Detail Testing Guide*. Menu ini berfungsi bagi pengguna dengan *role Principal Penetration Tester* dan *Penetration Tester* untuk mengubah detail dari suatu *testing guide* yang dipilih. Pada menu ini ditampilkan *form* yang sudah terisi informasi *testing guide* yang sudah tersimpan sebelumnya dan dapat diubah oleh pengguna.



Gambar 4.59 Tampilan *Edit Testing Guide*

The screenshot shows a software interface for editing a testing guide. The interface is divided into several sections:

- Impact:** Contains a preview pane with the text: "1 Tester dapat menemukan kemungkinan kerentanan dalam aplikasi dan dapat mengeksloitasi kerentanan tersebut untuk menyerang sistem."
- Remediation:** Contains a preview pane with the same text as the Impact section.
- Attack Example:** Contains a preview pane with the following code examples:
 

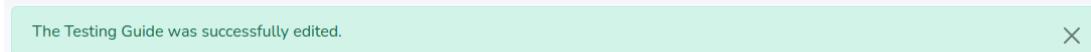
```

1. ## Example 1
2. [Public report](https://hackerone.com)
3.
4. ## Example 2 - Banner Grabbing
5.
6. ## Example 3
7. Ketahuan - [File report](https://google.com)
8.
      
```
- Tools:** Contains a preview pane with the following list:
  1. [\[Netcraft\]](https://toolbar.netcraft.com/site_report)(https://toolbar.netcraft.com/site\_report), an online tool that scans websites for information, including the web server.
  2. [\[Nikto\]](https://github.com/sullo/nikto)(https://github.com/sullo/nikto), an Open Source command-line scanning tool.
  3. [\[Nmap\]](https://nmap.org)(https://nmap.org), an Open Source command-line tool that also has a GUI, Zenmap.
- Other:** Contains a preview pane with the following links:
  1. [\[Fingerprinting Web Server - Beagle Security\]](https://beaglesecurity.com/blog/vulnerability/fingerprinting-web-server.html)(https://beaglesecurity.com/blog/vulnerability/fingerprinting-web-server.html)
  2. [\[Fingerprinting Web Server - OWASP\]](https://www-project-web-security-testing-guide.v42/4-Web_Application_Security_Testing/01-Information_Gathering/02-Fingerprint_Web_Server)(https://www-project-web-security-testing-guide.v42/4-Web\_Application\_Security\_Testing/01-Information\_Gathering/02-Fingerprint\_Web\_Server)

At the bottom left is a blue button labeled "Edit Testing Guide". At the bottom right is a small upward-pointing arrow icon. The footer of the interface includes the text "Copyright © Vulnerability Management 2023".

Gambar 4.60 Tampilan *Edit Testing Guide* lanjutan

Setelah pengguna mengisi *form* dan menekan tombol *Edit Testing Guide* pada bagian bawah, data akan diproses untuk disimpan ke *database*. Jika proses perubahan *testing guide* berhasil, sistem akan menampilkan *alert* sukses, bisa dilihat pada Gambar 4.61. Sedangkan jika perubahan *testing guide* gagal, sistem akan menampilkan *alert* gagal, bisa dilihat pada Gambar 4.62.



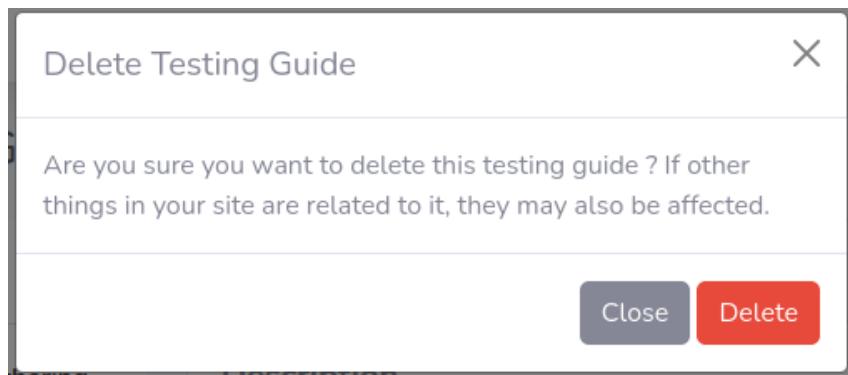
**Gambar 4.61 Tampilan Alert Sukses Edit Testing Guide**



**Gambar 4.62 Tampilan Alert Gagal Edit Testing Guide**

#### 4.2.4.6 Delete Testing Guide

Gambar di bawah merupakan tampilan *modal* konfirmasi *Delete Testing Guide*, *modal* ini akan tampil setelah pengguna menekan tombol *Delete Resource* pada menu navigasi *Icon Titik Tiga Testing Guide* di *View Detail Testing Guide*. Menu ini berfungsi bagi pengguna dengan *role Principal Penetration Tester* dan *Penetration Tester* untuk menghapus suatu *testing* yang dipilih. Pada menu ini ditampilkan konfirmasi penghapusan, jika user menekan tombol *Delete* penghapusan akan diproses sedangkan jika user menekan tombol *Close* penghapusan akan dibatalkan.



**Gambar 4.63 Tampilan Delete Testing Guide**

Setelah pengguna menekan tombol *Delete*, *testing guide* akan diproses untuk dihapus dari *database*. Jika proses penghapusan *testing guide*

berhasil, sistem akan menampilkan *alert* sukses, bisa dilihat pada Gambar 4.64. Sedangkan jika penghapusan *testing guide* gagal, sistem akan menampilkan *alert* gagal, bisa dilihat pada Gambar 4.65.



**Gambar 4.64 Tampilan Alert Sukses Delete Testing Guide**



**Gambar 4.65 Tampilan Alert Gagal Delete Testing Guide**

#### 4.2.5 Admin Panel

Fitur *Admin Panel* merupakan fitur yang berfungsi sebagai manajemen pengguna pada aplikasi. Pada fitur ini berguna untuk pengguna dengan *role Principal Penetration Tester* untuk melakukan aksi seperti menambahkan, mengubah dan menghapus pengguna.

##### 4.2.5.1 User Lists

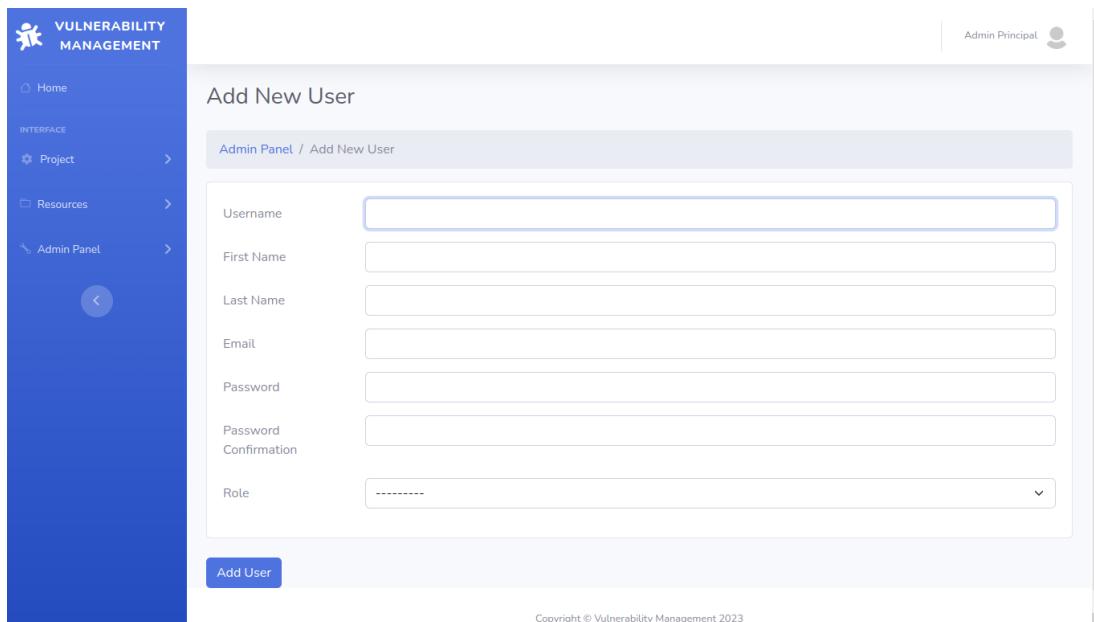
Gambar di bawah merupakan tampilan menu *Users Lists*, menu ini akan tampil setelah pengguna menekan tombol *Users Management* pada menu *Admin Panel*. Menu ini berfungsi bagi pengguna dengan *role Principal Penetration Tester* untuk melihat daftar pengguna yang terdaftar pada aplikasi. Pada menu ini ditampilkan informasi dari semua pengguna seperti *Name, Username, Email* dan *Role*.

Name	Username	Email	Role	Action
Admin Client	admin	admin@gmail.com	Client	Action
Admin Principal	principal	principal@admin.com	Principal Penetration Tester	Action
Alfredo Nofanski	alfredo	alfredo.nofanski@binus.ac.id	Penetration Tester	Action
Client Binus	client	client@binus.ac.id	Client	Action
Ryo Samuel	ryo	ryo.samuel@binus.ac.id	Penetration Tester	Action
Stanley Halim	stanley	stanley.halim@gmail.com	Penetration Tester	Action

**Gambar 4.66 Tampilan *User Lists***

#### 4.2.5.2 Add New User

Gambar di bawah merupakan tampilan menu *Add New User*, menu ini akan tampil setelah pengguna menekan tombol *Add New User* pada menu *Users Management*. Menu ini berfungsi bagi pengguna dengan *role Principal Penetration Tester* untuk menambahkan pengguna pada aplikasi. Pada menu ini ditampilkan *form* yang dapat diisi oleh pengguna mengenai detail dari pengguna seperti *Username*, *First Name*, *Last Name*, *Email*, *Password*, *Password Confirmation* dan *Role*.



**Gambar 4.67 Tampilan Add New User**

Setelah pengguna mengisi *form* dan menekan tombol *Add User* pada bagian bawah, data akan diproses untuk disimpan ke *database*. Jika proses penambahan *user* berhasil, sistem akan menampilkan *alert* sukses, bisa dilihat pada Gambar 4.68. Sedangkan jika penambahan *user* gagal, sistem akan menampilkan *alert* gagal, bisa dilihat pada Gambar 4.69.



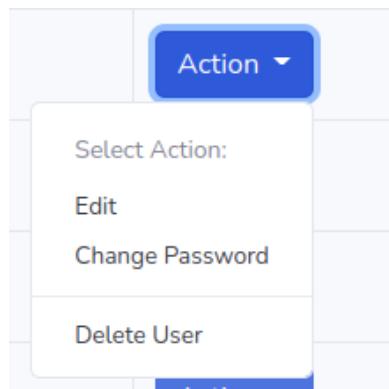
**Gambar 4.68 Tampilan Alert Sukses Add User**



**Gambar 4.69 Tampilan Alert Gagal Add User**

#### 4.2.5.3 Action User

*Action User* merupakan menu navigasi untuk memilih aksi yang ingin dilakukan pada suatu pengguna yang dipilih. Gambar dibawah menampilkan menu *Action User* yang memiliki beberapa aksi seperti *Edit*, *Change Password* dan *Delete User*.



**Gambar 4.70 Tampilan Action User**

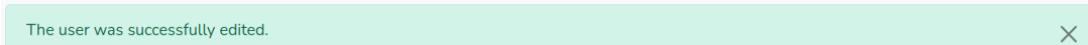
#### 4.2.5.4 Edit User

Gambar di bawah merupakan tampilan menu *Edit User*, menu ini akan tampil setelah pengguna menekan tombol *Edit* pada menu navigasi *Action User*. Menu ini berfungsi bagi pengguna dengan *role Principal Penetration Tester* untuk mengubah detail dari pengguna yang dipilih. Pada menu ini ditampilkan *form* yang sudah terisi informasi pengguna yang sudah tersimpan sebelumnya dan dapat diubah oleh pengguna.

Username	admin
First Name	Admin
Last Name	Client
Email	admin@gmail.com
Role	Client

**Gambar 4.71 Tampilan Edit User**

Setelah pengguna mengisi *form* dan menekan tombol *Save User* pada bagian bawah, data akan diproses untuk disimpan ke *database*. Jika proses perubahan *user* berhasil, sistem akan menampilkan *alert* sukses, bisa dilihat pada Gambar 4.72. Sedangkan jika perubahan *user* gagal, sistem akan menampilkan *alert* gagal, bisa dilihat pada Gambar 4.73.



The user was successfully edited.

X

**Gambar 4.72 Tampilan Alert Sukses Edit User**



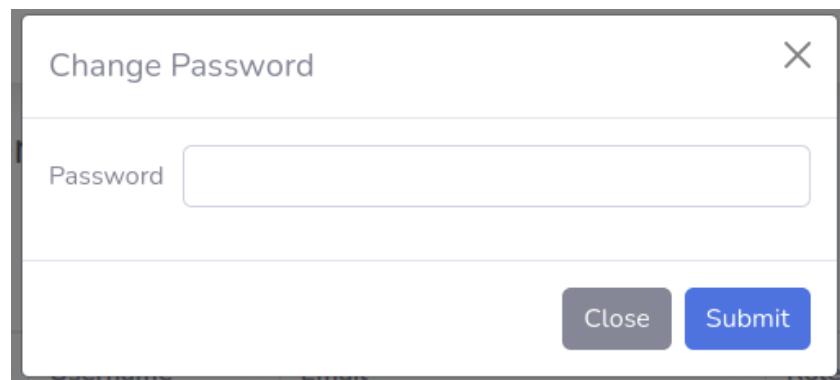
Unable to edit the user.

X

**Gambar 4.73 Tampilan Alert Gagal Edit User**

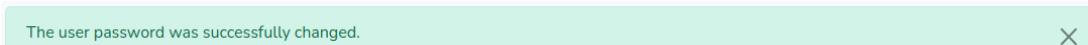
#### 4.2.5.5 *Change Password*

Gambar di bawah merupakan tampilan *modal Change Password*, *modal* ini akan tampil setelah pengguna menekan tombol *Change Password* pada menu navigasi *Action User*. Menu ini berfungsi bagi pengguna dengan *role Principal Penetration Tester* untuk menghapus mengganti kata sandi atau *password* dari pengguna yang dipilih. Pada menu ini pengguna dapat memasukkan *Password* yang akan diubah.



**Gambar 4.74 Tampilan Change Password**

Setelah pengguna mengisi *form* dan menekan tombol *Submit*, *password* yang dimasukkan akan diproses untuk memperbarui *password user* di *database*. Jika proses perubahan *password* berhasil, sistem akan menampilkan *alert* sukses, bisa dilihat pada Gambar 4.75. Sedangkan jika perubahan *password* gagal, sistem akan menampilkan *alert* gagal, bisa dilihat pada Gambar 4.76.



The user password was successfully changed.

X

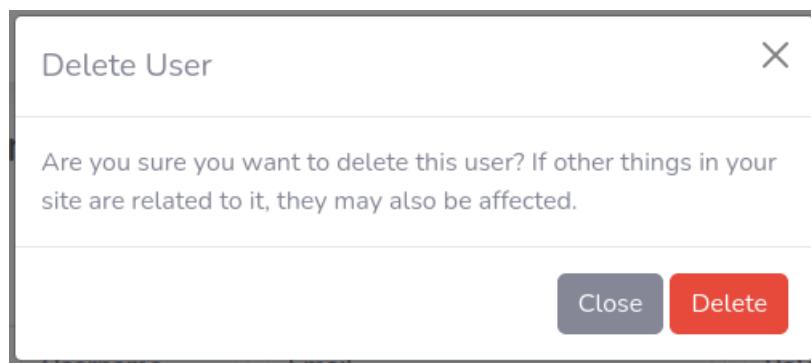
**Gambar 4.75 Tampilan Alert Sukses Change Password**



**Gambar 4.76 Tampilan Alert Gagal Edit Change Password**

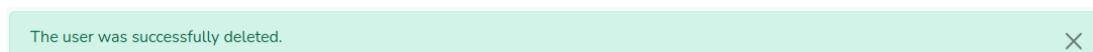
#### 4.2.5.6 Delete User

Gambar di bawah merupakan tampilan *modal* konfirmasi *Delete User*, *modal* ini akan tampil setelah pengguna menekan tombol *Delete Finding* pada menu navigasi *Action User*. Menu ini berfungsi bagi pengguna dengan *role Principal Penetration Tester* untuk menghapus suatu pengguna yang dipilih. Pada menu ini ditampilkan konfirmasi penghapusan, jika user menekan tombol *Delete* penghapusan akan diproses sedangkan jika user menekan tombol *Close* penghapusan akan dibatalkan.



**Gambar 4.77 Tampilan Konfirmasi Delete User**

Setelah pengguna menekan tombol *Delete*, *user* akan diproses untuk dihapus dari *database*. Jika proses penghapusan *user* berhasil, sistem akan menampilkan *alert* sukses, bisa dilihat pada Gambar 4.78. sedangkan jika penghapusan *user* gagal, sistem akan menampilkan *alert* gagal, bisa dilihat pada Gambar 4.79.



**Gambar 4.78 Tampilan Alert Sukses Delete User**



**Gambar 4.79 Tampilan Alert Gagal Delete User**

### 4.3 Black Box Testing

**Tabel 4.4 Evaluasi Black Box Testing**

No.	Scenario	Expected Result	Output
<i>Login</i>			
1.	User mengisi form <i>username</i> dan <i>password</i> dan menekan tombol <i>Login</i> untuk autentikasi.	User terautentikasi dan menampilkan halaman <i>home page</i> .	Sukses
<i>Project Lists</i>			
2.	User memilih menu <i>Project</i> , menekan menu <i>List Projects</i> , memilih tab <i>To Do</i> .	Aplikasi akan menampilkan daftar <i>project</i> yang berstatus <i>To Do</i> .	Sukses
3.	User memilih menu <i>Project</i> , menekan menu <i>List Projects</i> , memilih tab <i>On Going</i> .	Aplikasi akan menampilkan daftar <i>project</i> yang berstatus <i>On Going</i> .	Sukses
4.	User memilih menu <i>Project</i> , menekan menu <i>List Projects</i> , memilih tab <i>Done</i> .	Aplikasi akan menampilkan daftar <i>project</i> yang berstatus <i>Done</i> .	Sukses
<i>Add Project</i>			
5.	User memilih menu <i>Project</i> , menekan menu <i>List Projects</i> , menekan tombol <i>Create New Project</i> , mengisi data <i>Project</i> ,	Aplikasi akan menampilkan halaman <i>Create New Project</i> berisi <i>form</i> dan <i>option</i> , menyimpan data yang telah dipilih dan diisi, dan menampilkan <i>alert</i> bahwa	Sukses

	<i>project</i> dan menekan tombol <i>Add New Project</i> .	<i>project</i> berhasil ditambahkan.	
<i>View Project</i>			
6.	<i>User</i> memilih menu <i>Project</i> , menekan menu <i>List Projects</i> , menekan salah satu nama <i>project</i> yang ingin dilihat.	Aplikasi akan menampilkan detail <i>project</i> yang dipilih.	Sukses
<i>Edit Project</i>			
7.	<i>User</i> memilih menu <i>Project</i> , menekan menu <i>List Projects</i> , memilih salah satu <i>project</i> , menekan icon titik tiga <i>project</i> dan menekan opsi <i>Edit Project</i> , mengganti data <i>project</i> , dan menekan tombol <i>Edit Project</i> .	Aplikasi akan menampilkan halaman <i>Edit Project</i> berisi form dan <i>option</i> dan menampilkan <i>alert</i> bahwa data <i>project</i> telah diedit.	Sukses
<i>Delete Project</i>			
8.	<i>User</i> memilih menu <i>Project</i> , menekan menu <i>List Projects</i> , memilih salah satu <i>project</i> , menekan icon titik tiga <i>project</i> , menekan opsi <i>Delete Project</i> , dan menekan tombol <i>Delete</i> pada <i>pop-up</i> konfirmasi.	Aplikasi akan menampilkan <i>pop-up alert</i> konfirmasi penghapusan dan menampilkan <i>alert</i> bahwa <i>project</i> telah dihapus.	Sukses
<i>Search Project</i>			

9.	<p>User memilih menu <i>Project</i>, menekan menu <i>List Projects</i>, mengisi <i>search bar</i> dengan kata kunci yang ingin dicari, menekan <i>icon</i> kaca pembesar atau menekan tombol <i>Enter</i>.</p>	Aplikasi menampilkan daftar <i>project</i> yang terdapat kata kunci yang dicari.	Sukses
<i>Finding Lists</i>			
10.	<p>User memilih menu <i>Project</i>, menekan menu <i>List Projects</i>, memilih dan menekan salah satu nama <i>project</i>, menekan tab <i>Not Fixed</i>.</p>	Aplikasi menampilkan daftar <i>finding</i> berstatus <i>Not Fixed</i> .	Sukses
11.	<p>User memilih menu <i>Project</i>, menekan menu <i>List Projects</i>, memilih dan menekan salah satu nama <i>project</i>, menekan tab <i>Fixed</i>.</p>	Aplikasi menampilkan daftar <i>finding</i> berstatus <i>Fixed</i> .	Sukses
12.	<p>User memilih menu <i>Project</i>, menekan menu <i>List Projects</i>, memilih dan menekan salah satu nama <i>project</i>, menekan tab <i>Not Fixed</i>, memilih salah satu <i>finding</i>, menekan <i>icon</i> titik tiga <i>finding</i> tersebut, dan menekan opsi <i>Resolve to Fixed</i>.</p>	Aplikasi menampilkan <i>alert</i> bahwa status <i>finding</i> telah diubah dan memindahkan <i>finding</i> yang dipilih ke tab <i>Fixed</i> .	Sukses

13.	<i>User memilih menu Project, menekan menu List Projects, memilih dan menekan salah satu nama project, menekan tab Fixed, memilih salah satu finding, menekan icon titik tiga finding tersebut, dan menekan opsi Re-open to Not Fixed.</i>	Aplikasi menampilkan alert bahwa status <i>finding</i> telah diubah dan memindahkan <i>finding</i> yang dipilih ke tab <i>Not Fixed</i> .	Sukses
<i>Add Finding</i>			
14.	<i>User memilih menu Project, menekan menu List Projects, memilih dan menekan salah satu nama project, menekan tombol Add New Finding, mengisi data mengenai finding, dan menekan tombol Add Finding.</i>	Aplikasi akan menampilkan halaman <i>Add New Finding</i> berisi <i>form</i> dan <i>option</i> , menyimpan data yang telah dipilih dan diisi, dan menampilkan alert bahwa <i>finding</i> berhasil ditambahkan.	Sukses
<i>View Finding</i>			
15.	<i>User memilih menu Project, menekan menu List Projects, memilih dan menekan salah satu nama project, memilih dan menekan salah satu nama finding.</i>	Aplikasi akan menampilkan detail mengenai <i>finding</i> tersebut.	Sukses
<i>Edit Finding</i>			

16.	<i>User memilih menu Project, menekan menu List Projects, menekan tombol Action pada salah satu finding, menekan opsi Edit Finding, mengganti data finding, dan menekan tombol Save Finding.</i>	Aplikasi akan menampilkan halaman Detail Finding berisi form dan option dan menampilkan alert bahwa data finding telah diedit.	Sukses
<i>Delete Finding</i>			
17.	<i>User memilih menu Project, menekan menu List Projects, menekan tombol Action pada salah satu finding, menekan opsi Delete Finding, dan menekan tombol Delete pada pop-up konfirmasi.</i>	Aplikasi akan menampilkan pop-up alert konfirmasi penghapusan dan menampilkan alert bahwa finding telah dihapus.	Sukses
<i>Generate Report</i>			
18.	<i>User memilih menu Project, menekan menu List Projects, menekan icon titik tiga pada salah satu project, dan menekan opsi Generate Report.</i>	Aplikasi akan mengirimkan file untuk di download oleh user.	Sukses
<i>View User</i>			
19.	<i>User memilih menu Admin Panel, menekan menu Users Management,</i>	Aplikasi akan menampilkan daftar user yang terdaftar pada aplikasi.	Sukses

<i>Add User</i>				
20.	<i>User</i> memilih menu <i>Admin Panel</i> , menekan menu <i>Users Management</i> , menekan tombol <i>Add New User</i> , mengisi data <i>user</i> , dan menekan tombol <i>Add User</i> .	Aplikasi akan menampilkan halaman <i>Add New User</i> berisi <i>form</i> dan <i>option</i> , menyimpan data yang telah dipilih dan diisi, dan menampilkan <i>alert</i> bahwa <i>user</i> berhasil ditambahkan.	Sukses	
<i>Edit User</i>				
21.	<i>User</i> memilih menu <i>Admin Panel</i> , menekan menu <i>Users Management</i> , menekan tombol <i>Action</i> pada salah satu <i>user</i> , menekan opsi <i>Edit</i> , mengganti data <i>user</i> , dan menekan tombol <i>Save User</i> .	Aplikasi akan menampilkan halaman <i>Edit User</i> berisi <i>form</i> dan <i>option</i> dan menampilkan <i>alert</i> bahwa data <i>finding</i> telah diedit.	Sukses	
<i>Delete User</i>				
22.	<i>User</i> memilih menu <i>Admin Panel</i> , menekan menu <i>Users Management</i> , menekan tombol <i>Action</i> pada salah satu <i>user</i> , menekan opsi <i>Delete User</i> , dan menekan tombol <i>Delete</i> pada <i>pop-up</i> konfirmasi.	Aplikasi akan menampilkan <i>pop-up alert</i> konfirmasi penghapusan dan menampilkan <i>alert</i> bahwa <i>user</i> telah dihapus.	Sukses	
<i>Change Password</i>				

23.	<p>User memilih menu <i>Admin Panel</i>, menekan menu <i>Users Management</i>, menekan tombol <i>Action</i> pada salah satu <i>user</i>, menekan opsi <i>Change Password</i>, mengisi <i>password</i> baru <i>user</i>, dan menekan tombol <i>Submit</i>.</p>	<p>Aplikasi akan menampilkan <i>pop-up alert form</i> <i>password</i> baru dan menampilkan <i>alert</i> bahwa <i>password user</i> telah diubah.</p>	Sukses
<i>View Testing Guide</i>			
24.	<p>User memilih menu <i>Resources</i>, menekan menu <i>List Testing Guide</i>, memilih dan menekan salah satu <i>testing guide</i>, memilih dan menekan salah satu daftar <i>guide</i>.</p>	<p>Aplikasi akan menampilkan halaman <i>Testing Guide Lists</i> dan menampilkan detail dari salah satu <i>guide</i> yang dipilih.</p>	Sukses
<i>Add Testing Guide</i>			
25.	<p>User memilih menu <i>Resources</i>, menekan menu <i>List Testing Guide</i>, menekan tombol <i>Add Resource</i>, mengisi data mengenai <i>resource</i>, dan menekan tombol <i>Add Testing Guide</i>.</p>	<p>Aplikasi akan menampilkan halaman <i>Add New Testing Guide</i> dan menampilkan <i>alert</i> bahwa <i>guide</i> telah ditambahkan.</p>	Sukses
<i>Edit Testing Guide</i>			
26.	<p>User memilih menu <i>Resources</i>, menekan menu <i>List Testing Guide</i>,</p>	<p>Aplikasi akan menampilkan halaman <i>Edit Testing Guide</i> berisi</p>	Sukses

	memilih dan menekan salah satu <i>testing guide</i> , menekan icon titik tiga dan menekan opsi <i>Edit Resource</i> , mengganti data <i>guide</i> , dan menekan tombol <i>Edit Testing Guide</i> .	<i>form</i> dan <i>option</i> dan menampilkan <i>alert</i> bahwa data <i>guide</i> telah diedit.	
<i>Delete Testing Guide</i>			
27.	User memilih menu <i>Resources</i> , menekan menu <i>List Testing Guide</i> , memilih dan menekan salah satu <i>testing guide</i> , menekan icon titik tiga dan menekan opsi <i>Delete Resource</i> , dan menekan tombol <i>Delete</i> pada <i>pop-up</i> konfirmasi..	Aplikasi akan menampilkan <i>pop-up alert</i> konfirmasi penghapusan dan menampilkan <i>alert</i> bahwa <i>guide</i> telah dihapus.	Sukses

#### 4.4 Evaluasi Aplikasi berdasarkan *user feedback*

Penulis akan mempresentasikan Aplikasi *Vulnerability Management* kepada *user* (*junior penetration tester*, *senior penetration tester* dan *client*), *user* juga diminta untuk mencoba secara langsung Aplikasi *Vulnerability Management*, dan terakhir *user* diminta untuk menjawab pertanyaan yang telah diberikan oleh penulis. Ada beberapa pertanyaan khusus hanya untuk *penetration tester* (*senior penetration tester* dan *junior penetration tester*) dan *client*. Hasilnya akan digunakan untuk mengetahui pendapat *User* mengenai Aplikasi *Vulnerability Management* dan dapat melakukan evaluasi terhadap Aplikasi *Vulnerability Management*.

Rangkuman dari hasil yang didapat dari *user feedback* adalah:

1. Dari sisi tampilan, Aplikasi *Vulnerability Management* mudah dipahami. Penjelasan tambahan dari responden adalah *User Interface* simpel, minimalis, dan konsisten
2. Dari sisi penggunaan fitur, fitur Aplikasi *Vulnerability Management* mudah digunakan, mudah dipahami, dan berjalan dengan baik . Meskipun demikian, ada masukan dari responden adalah menambahkan *Frequently Asked Question (FAQ)* atau dokumentasi setiap fitur.
3. Dari Fitur *Project*, dapat membantu atau mempermudah kolaborasi antar *penetration tester* dengan *client* dan *penetration tester* dengan *penetration tester* lain. Penjelasan tambahan dari responden adalah dapat mempermudah dalam *monitoring project*, *management project*, dan pemaparan *finding*, dapat juga membantu *client* untuk melihat progress *pentest* secara keseluruhan
4. Dari Fitur *Generate Report*, dapat membantu untuk mempercepat proses *pentest*. Penjelasan tambahan dari responden adalah dapat membantu *penetration tester* untuk mempercepat proses reporting dikarenakan *penetration tester* tidak perlu membuat kerangka laporan dari awal secara manual. Meskipun demikian, ada masukan dari responden adalah membuat *logging* untuk mengetahui *user* mana yang sudah mengakses atau mendownload report tersebut.
5. Dari Fitur *Resources*, dapat membantu untuk memberikan pengetahuan *pentest* ke *penetration tester* lain. Penjelasan tambahan dari responden adalah dapat membantu *penetration tester* untuk mencari informasi mengenai berbagai *pentest*, dan membantu *penetration tester* menjelaskan *finding* yang telah ditemukan ke *penetration tester* lain. Meskipun demikian, ada masukan dari responden adalah harus diperbanyak default dari developer dengan memasukan standar yang sudah ada sehingga nantinya user tinggal menambahkan teknik baru dan tak perlu mengulang lagi untuk mengisi secara manual.
6. Secara Keseluruhan, aplikasi *Vulnerability Management* dapat membantu untuk mempercepat pekerjaan *penetration tester* dan *client*. Penjelasan tambahan dari responden adalah fitur-fitur yang diberikan dapat membantu *penetration tester* untuk mempercepat proses *pentest* seperti fitur *listing*

*finding, checklist* setiap *project* dan *generate report*, yang akan sangat mengikis waktu dan juga meningkatkan efisiensi dalam perihal komunikasi.

Pada pertanyaan pertama dan kedua untuk *penetration tester* dan *client* digunakan untuk mengetahui nama dan pekerjaan sekarang. Hasilnya penulis mendapatkan Ming Hong dan Christoval Leaved sebagai junior *penetration tester*, Rafi Ramzy dan Henky Tornado sebagai senior *penetration tester*, dan Ferry Alif sebagai *client*.

1. Siapa nama anda?
2. Apa pekerjaan anda sekarang?

Junior *penetration tester* ke-1 : Ming Hong - Security Consultant

Junior *penetration tester* ke-2 : Christoval Leaved - Security Consultant

Senior *penetration tester* ke-1 : Rafi Ramzy - Senior Penetration Tester Consultant

Senior *penetration tester* ke-2 : Henky Tornado - Senior Penetration Tester Consultant & Team Lead

*Client* : Ferry Alif - Cyber Security Engineer

Pada pertanyaan ketiga untuk *penetration tester* dan *client* digunakan untuk mengetahui *User Interface* yang dimiliki oleh Aplikasi *Vulnerability Management* apakah mudah dipahami dalam penggunaannya. Hasilnya adalah semua *user* mudah memahami *User Interface* yang dimiliki oleh Aplikasi *Vulnerability Management*

3. Apakah *User Interface* mudah dipahami?

Junior *penetration tester* ke-1 : Yes, secara keseluruhan User Interface mudah dipahami. Terlihat simple dan lumayan lengkap untuk Vulnerable Management.

Junior *penetration tester* ke-2 : Iya, dikarenakan penggunaan desain yang minimalis dan konsisten, sehingga fitur dapat diakses lebih mudah.

Senior *penetration tester* ke-1 : Ya sangat mudah, sistemnya juga ok

Senior *penetration tester* ke-2 : iya

*Client* : yes

Pada pertanyaan keempat untuk *penetration tester* dan *client* digunakan untuk mengetahui fitur yang dimiliki oleh Aplikasi *Vulnerability Management* apakah mudah dipahami dalam penggunaannya.

4. Apakah seluruh fitur mudah digunakan dan dipahami?

Junior *penetration tester* ke-1 : Yes, secara keseluruhan fitur sudah mencakup kebutuhan *penetration tester* untuk melakukan reporting.

Junior *penetration tester* ke-2 : Iya, dikarenakan UI & UX yang diterapkan sangat baik, sehingga fitur juga mudah dipahami

Senior *penetration tester* ke-1 : Mudah dipahami penggunaannya, tapi alangkah baiknya dibuat faq or docs

Senior *penetration tester* ke-2 : iya

*Client* : yes

Pada pertanyaan kelima untuk *penetration tester* dan *client* digunakan untuk mengetahui fitur yang dimiliki oleh Aplikasi *Vulnerability Management* apakah berjalan dengan baik.

5. Apakah seluruh fitur berjalan dengan baik?

Junior *penetration tester* ke-1 : Yes, semua fitur berjalan dengan baik sesuai penjelasan sebelumnya.

Junior *penetration tester* ke-2 : Iya, setelah dilakukan pengujian, fitur dinilai dapat berjalan dengan baik

Senior *penetration tester* ke-1 : So far oke berjalan dengan baik

Senior *penetration tester* ke-2 : iya sudah berjalan dengan sesuai dengan yang sudah dijelaskan

*Client* : yes

Pada pertanyaan keenam untuk *penetration tester* digunakan untuk mengetahui fitur *project* yang dimiliki oleh Aplikasi *Vulnerability Management* apakah dapat membantu *penetration tester* melakukan kolaborasi dengan *penetration tester* lain atau *client* untuk mempercepat proses pentest.

6. Apakah fitur project dapat membantu atau mempermudah anda dalam kolaborasi dengan pentester lain dan client?

Junior *penetration tester* ke-1 : Yes, fitur tersebut dapat membantu *client* untuk melihat Junior progress pentest secara keseluruhan.

Junior *penetration tester* ke-2 : Iya, fitur pada website dapat mempermudah pemaparan finding

Senior *penetration tester* ke-1 : ya sangat mudah

Senior *penetration tester* ke-2 : iya dapat mempermudah dalam konteks monitoring dan juga management project

Pada pertanyaan keenam untuk *client* digunakan untuk mengetahui fitur *project* yang dimiliki oleh Aplikasi *Vulnerability Management* apakah membantu *client* untuk kolaborasi dengan *penetration tester* dengan bisa melihat proses pentest setiap waktu.

6. Apakah fitur *project* dapat membantu atau mempermudah anda dalam kolaborasi dalam *project* pentest?

*Client* : yes

Pada pertanyaan ketujuh untuk *penetration tester* digunakan untuk mengetahui fitur *Generate Report* yang dimiliki oleh Aplikasi *Vulnerability Management* apakah dapat mempercepat proses pentest dalam penulisan *report*.

7. Apakah fitur Generate Report dapat mempercepat proses pentest?

Junior *penetration tester* ke-1 : Yes, fitur tersebut membantu pentester untuk mempercepat proses reporting dikarenakan pentester tidak membutuhkan templating dalam reporting.

Junior *penetration tester* ke-2 : Iya, dikarenakan pentester tidak perlu membuat kerangka laporan dari awal secara manual

Senior *penetration tester* ke-1 : ya, mempercepat pentest result delivery

Senior *penetration tester* ke-2 : iya dapat mempermudah, tapi harus dibuat logging, sehingga kita dapat tahu siapa saja yang sudah pernah mendownload report tersebut, karena jika terjadi misunderstanding akan lebih mudah di validasi siapa saja yang sudah pernah mengakses report tersebut

Pada pertanyaan ketujuh untuk *client* digunakan untuk mengetahui fitur *Generate Report* yang dimiliki oleh Aplikasi *Vulnerability Management* apakah dapat mempermudah *client* untuk mendapatkan *file report* setiap waktu dengan proses pentest yang terbaru.

7. Apakah fitur Generate Report dapat mempermudah dalam mendapatkan Laporan Pentest setiap waktu dengan Kondisi Pentest yang terbaru?

*Client* : yes

Pada pertanyaan kedelapan untuk *penetration tester* digunakan untuk mengetahui fitur *Resources* yang dimiliki oleh Aplikasi *Vulnerability Management* apakah dapat membantu *penetration tester* untuk memberikan pengetahuan pentest ke *penetration tester* lain sehingga dapat memudahkan proses pentest.

8. Apakah fitur *Resources* dapat membantu atau mempermudah anda saat melakukan pentest?

Junior *penetration tester* ke-1 : Yes, fitur tersebut dapat membantu pentester untuk mencari info mengenai pentest.

Junior *penetration tester* ke-2 : Iya, dikarenakan fitur tersebut dapat membantu pentester menjelaskan finding ke pentester lain, sehingga mempermudah proses kolaborasi

Senior *penetration tester* ke-1 : sepertinya lumayan membantu

Senior *penetration tester* ke-2 : dapat mempermudah tapi harus diperbanyak default dari developer dengan memasukan standar yang sudah ada sehingga nantinya user tinggal menambahkan teknik baru dan tak perlu mengulang lagi untuk mengisi secara manual.

Pada pertanyaan kesembilan untuk *penetration tester* dan kedelapan untuk *client* digunakan untuk mengetahui Apakah Aplikasi *Vulnerability Management* secara keseluruhan dapat membantu pekerjaan *penetration tester* dan *client*.

9. Apakah Aplikasi *Vulnerability Management* ini dapat membantu atau mempermudah pekerjaan anda?

Junior *penetration tester* ke-1 : Yes, secara keseluruhan fitur2 yang ada di aplikasi ini dapat membantu pentester untuk mempercepat proses pentest.

Junior *penetration tester* ke-2 : Iya, dikarenakan fitur-fiturnya yang dapat mempermudah proses listing finding dan generate report

Senior *penetration tester* ke-1 : ya, dapat dengan mudah karena tidak perlu adanya pembuatan report, dan checklist sudah tersedia setiap memulai project baru

Senior *penetration tester* ke-2 : iya dapat membantu karena akan sangat mengikis waktu dan juga meningkatkan efisiensi dalam perihal komunikasi

*Client* : yes

Pada pertanyaan kesepuluh untuk pentester dan kedelapan untuk *client* digunakan untuk mengetahui saran dari *user* yang bisa membantu pengembangan Aplikasi *Vulnerability Management*.

10. Apakah ada saran terhadap Aplikasi Vulnerability Management ini?

Junior *penetration tester* ke-1 : Sebaiknya ditambahkan dokumentasi pada setiap fitur agar pentester yang masih pemula dapat lebih memahami fitur lebih baik

Junior *penetration tester* ke-2 : Yes, mungkin untuk enhancement bisa dibuat database finding yang sudah memuat deskripsi seperti background, impact, cvss, & dll sehingga ketika input finding sudah tinggal dipilih.

Senior *penetration tester* ke-1 : pada aplikasi tersebut terdapat input untuk poc, tetapi gambar terlihat terpisah dengan deskripsi. sebaiknya disatukan karena deskripsi dan evidence merupakan sebuah rangkaian kesatuan

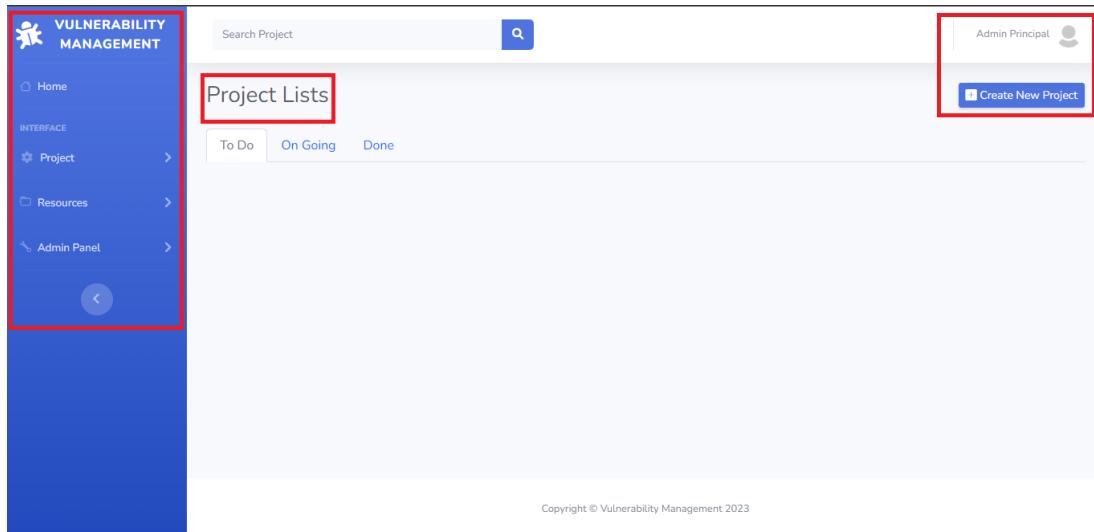
Senior *penetration tester* ke-2 : tolong jangan menciptakan fitur yang dapat menjadi musibah untuk kita semua seperti change password, karena harusnya password bersifat pribadi dan hanya bisa diganti oleh orang tersebut karena password secara konseptual merupakan salah satu faktor yang digunakan dalam konsep authentikasi dimana itu menjadi sebuah validasi bahwa orang tersebut memang real, dikarenakan tersebut seharusnya penggantian password hanya dapat dilakukan oleh user itu sendiri, seandainya tetap diperlukan maka hanya bisa dibentuk menjadi reset password dimana itu tidak dapat mengganti password dari orang lain

*Client* : Ditambahkan view page yang menampilkan sisa mandays dari masing-masing client, Ditambahkan status risk accepted untuk masing-masing temuan, dan client page diberikan hak akses untuk memasukkan komen untuk temuan yang risk accepted, dan Reportnya pentestnya diberikan password.

## 4.5 Evaluasi Aplikasi berdasarkan *Eight Golden Rules*

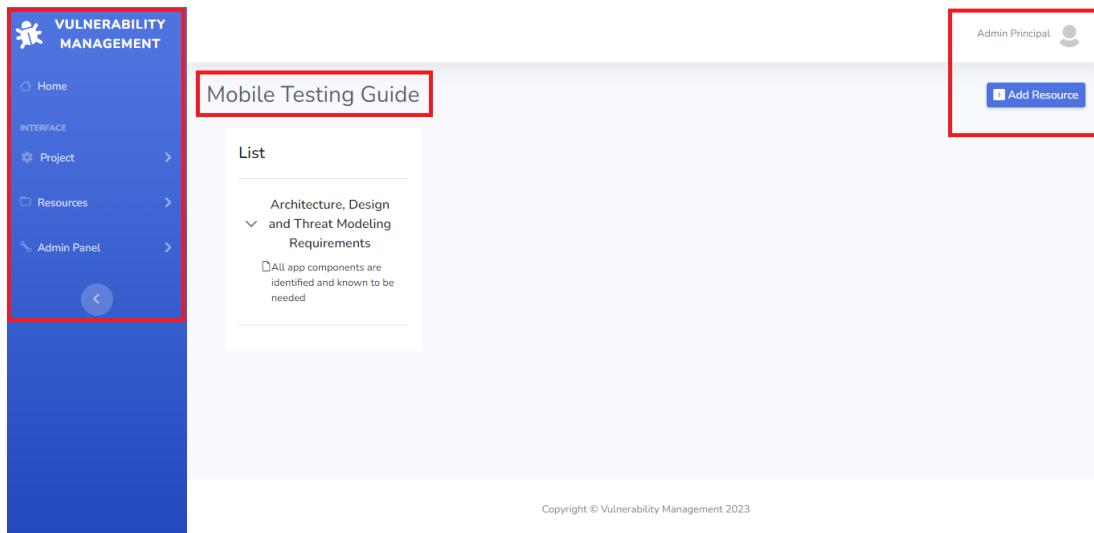
### 4.5.1 Strive for Consistency

Evaluasi ini mengarahkan untuk konsisten menggunakan pola desain yang sama dan urutan tindakan yang sama pada situasi yang sama pula. Pola desain yang dimaksud contohnya adalah setiap halaman mempunyai letak menu, judul serta bentuk tampilan sistem itu sendiri. Konsistensi ini dapat kita lihat dari menu *Project Lists, Resources* dan *Admin Panel*.



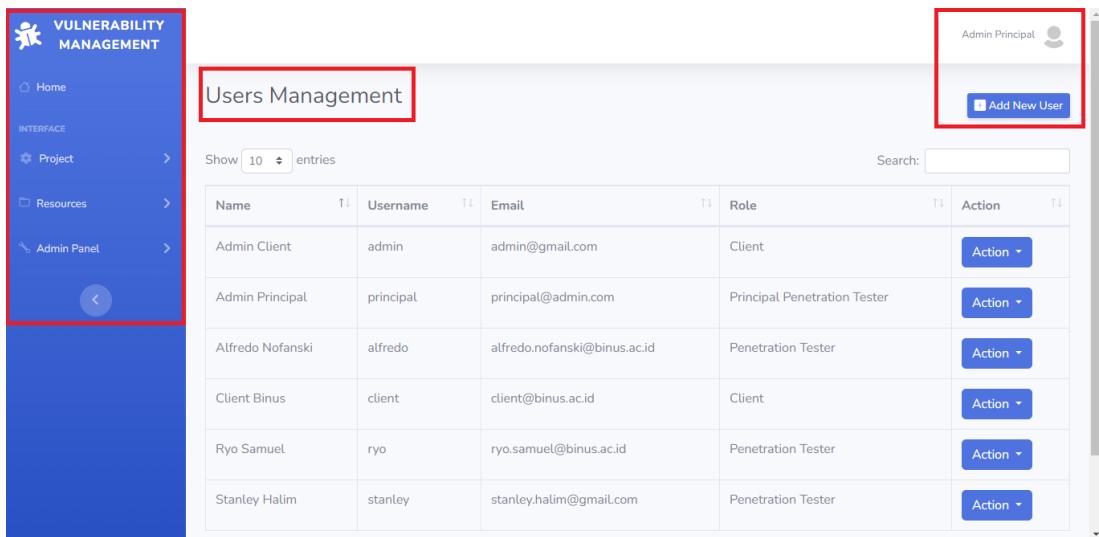
**Gambar 4.80 Tampilan Konsistensi Letak Menu, Judul, dan Tombol di *Project Lists***

Gambar diatas ada di halaman *Project Lists* yang menunjukan pola desain yang sama yaitu navigasi atau *menu* di bagian kiri, Judul *Project Lists* ada di bagian paling atas dalam *content*, tombol untuk menambah *Project* ada di bagian paling kanan judul, dan *Profile* di bagian pojok kanan atas.



**Gambar 4.81 Tampilan Konsistensi Letak Menu, Judul, dan Tombol di *Resources***

Gambar diatas ada di halaman *Testing Guide* yang menunjukan pola desain yang sama yaitu navigasi atau *menu* di bagian kiri, Judul *Testing Guide* ada di bagian paling atas dalam *content*, tombol untuk menambah *Testing Guide* ada di bagian paling kanan judul, dan *Profile* di bagian pojok kanan atas.

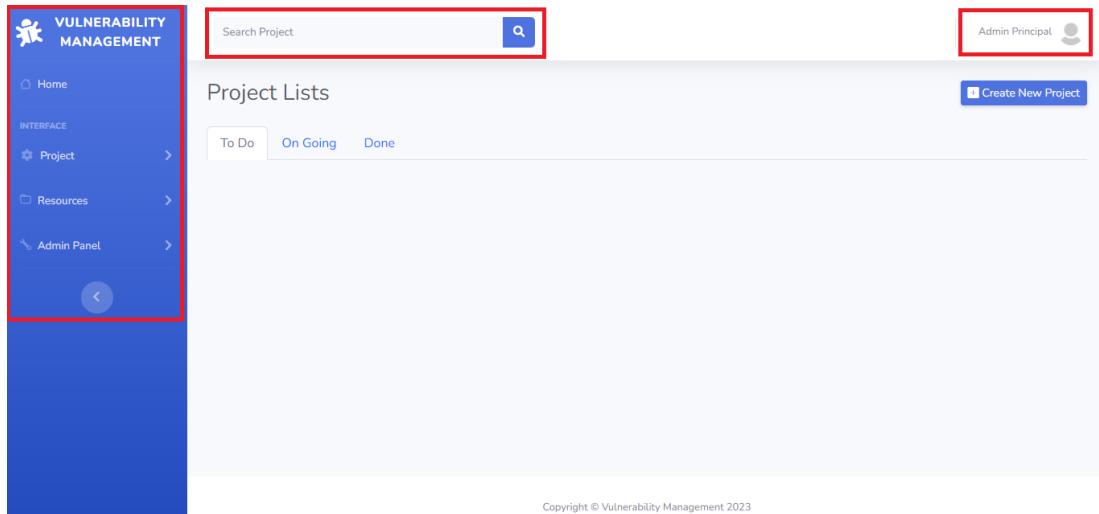


**Gambar 4.82 Tampilan Konsistensi Letak Menu, Judul, dan Tombol di Admin Panel**

Gambar diatas ada di halaman *User Management* yang menunjukan pola desain yang sama yaitu navigasi atau *menu* di bagian kiri, Judul *User Management* ada di bagian paling atas dalam *content*, tombol untuk menambah *user* ada di bagian paling kanan judul, dan *Profile* di bagian pojok kanan atas.

#### 4.5.2 Cater to Universal Usability

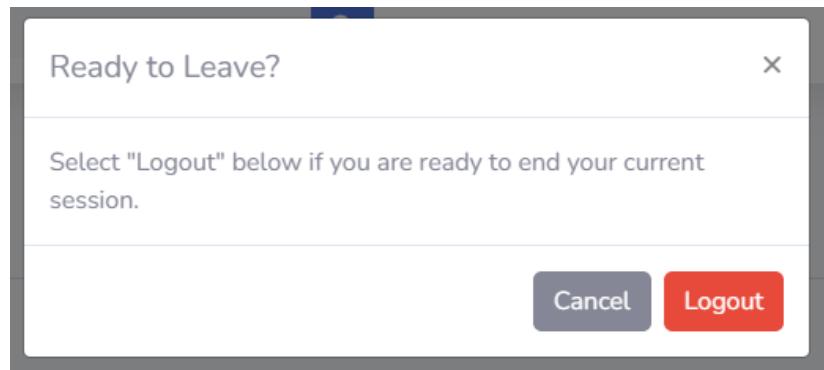
Evaluasi ini mengarahkan untuk menggunakan pola desain universal atau sudah sering digunakan pada aplikasi lainnya. Seperti menu navigasi yang terletak pada bagian kiri dari halaman, tombol Profile yang pada dasarnya terletak di pojok kanan atas halaman dan *Search box* yang berada di atas halaman.



**Gambar 4.83 Tampilan Letak Menu Navigasi, tombol *Profile* dan *Search box***

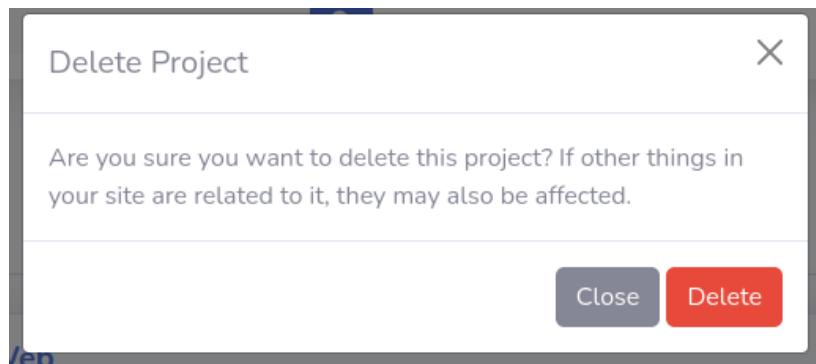
#### 4.5.3 *Offer Informative Feedback*

Evaluasi ini mengarahkan untuk memiliki *feedback* pada setiap proses yang akan dijalankan, seperti ketika akan *logout* akan muncul *feedback* seperti memberikan konfirmasi apakah *user* ingin mengakhiri sesi yang berjalan. Contoh lainnya seperti konfirmasi *delete project, finding, resources* ataupun *user*.



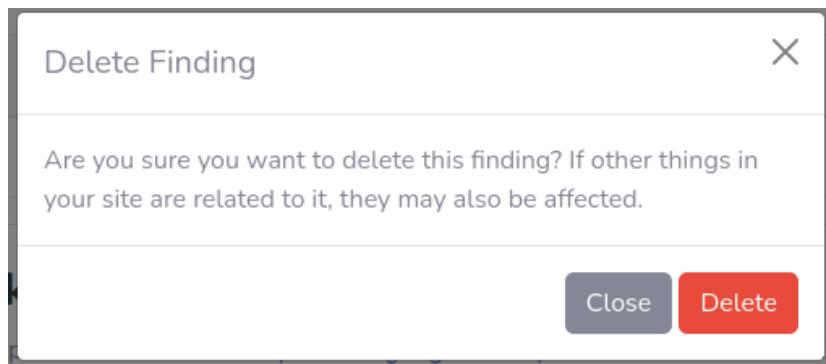
**Gambar 4.84 Tampilan Konfirmasi *Logout***

Gambar diatas muncul ketika *user* mengklik tombol *Logout*, jendela kecil yang muncul digunakan untuk memastikan lagi *user* ingin keluar dari *account* yang sedang digunakan.



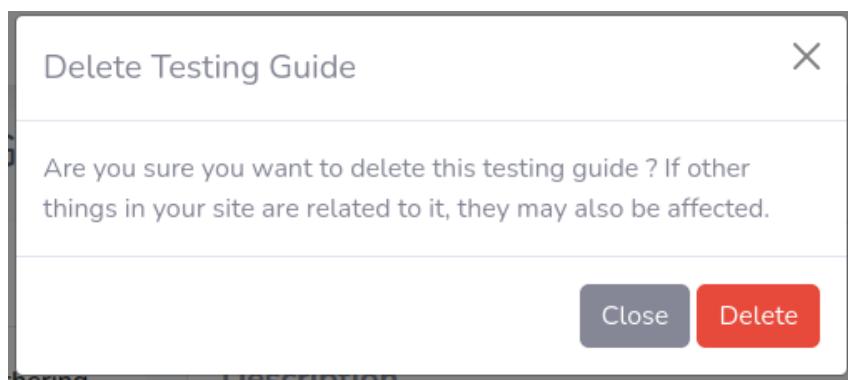
**Gambar 4.85 Tampilan Konfirmasi *Delete Project***

Gambar diatas muncul ketika *user* mengklik tombol *Delete Project*, jendela kecil yang muncul digunakan untuk memastikan lagi *user* ingin menghapus *project* tersebut.



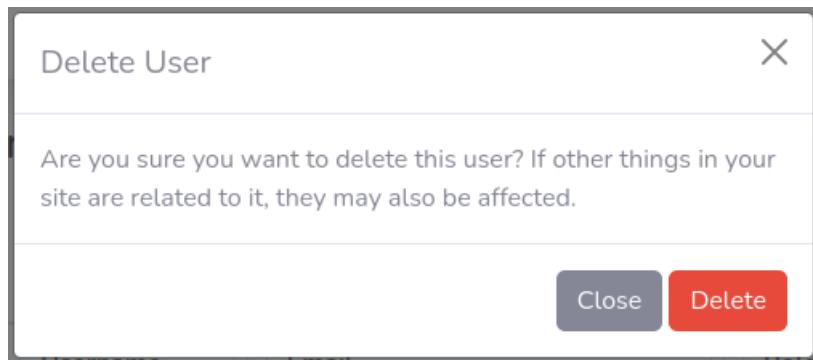
**Gambar 4.86 Tampilan Konfirmasi *Delete Finding***

Gambar diatas muncul ketika *user* mengklik tombol *Delete Finding*, jendela kecil yang muncul digunakan untuk memastikan lagi *user* ingin menghapus *finding* tersebut.



**Gambar 4.87 Tampilan Konfirmasi *Delete Resources***

Gambar diatas muncul ketika *user* mengklik tombol *Delete Resources*, jendela kecil yang muncul digunakan untuk memastikan lagi *user* ingin menghapus *testing guide* tersebut.

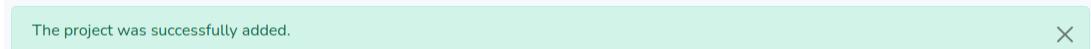


**Gambar 4.88 Tampilan Konfirmasi *Delete User***

Gambar diatas muncul ketika *user* mengklik tombol *Delete User*, jendela kecil yang muncul digunakan untuk memastikan lagi *user* ingin menghapus *account user* tersebut.

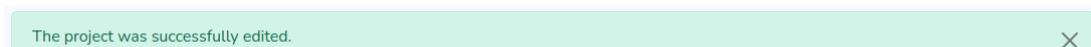
#### 4.5.4 *Design Dialogs to Yield Closure*

Evaluasi ini mengarahkan untuk memberikan respon kepada user ketika proses yang dijalankan sudah berhasil dan selesai. Seperti contohnya pada fitur *Project* yang menampilkan pesan sukses jika melakukan aksi seperti *add*, *edit* atau *delete*.



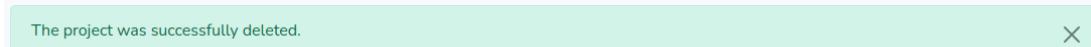
**Gambar 4.89 Tampilan Sukses *Add Project***

Gambar diatas ketika sistem berhasil melakukan *add project*, sistem akan menampilkan sebuah kotak berwarna hijau dengan tulisan *The project was successfully added*.



**Gambar 4.90 Tampilan Sukses *Edit Project***

Gambar diatas ketika sistem berhasil melakukan *edit project*, sistem akan menampilkan sebuah kotak berwarna hijau dengan tulisan *The project was successfully edited.*

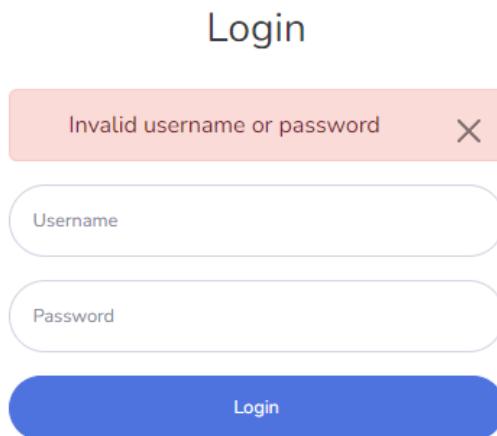


**Gambar 4.91 Tampilan Sukses *Delete Project***

Gambar diatas ketika sistem berhasil melakukan *delete project*, sistem akan menampilkan sebuah kotak berwarna hijau dengan tulisan *The project was successfully deleted.*

#### 4.5.5 *Prevent Errors*

Evaluasi ini mengarahkan untuk memberikan pencegahan sederhana terhadap *error*. Contohnya ketika *user* salah memasukkan *username* dan *password* pada saat melakukan *login*, sistem akan memberikan tampilan *error* seperti gambar dibawah ini.

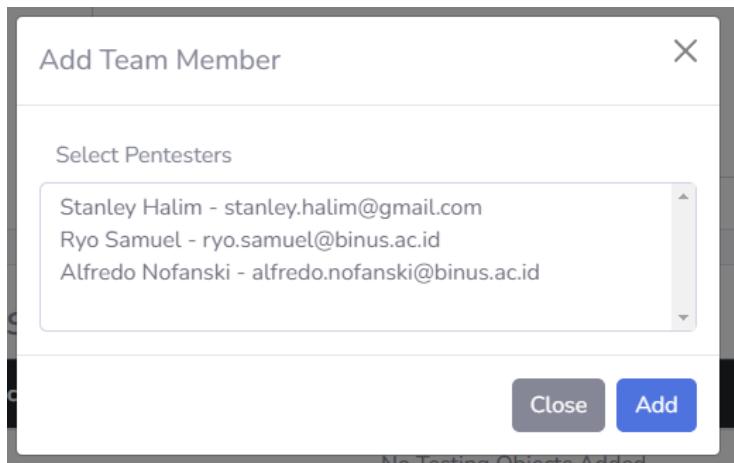


**Gambar 4.92 Tampilan *User* Salah Memasukkan *Username* dan *Password***

#### 4.5.6 *Permit Easy Reversal of Actions*

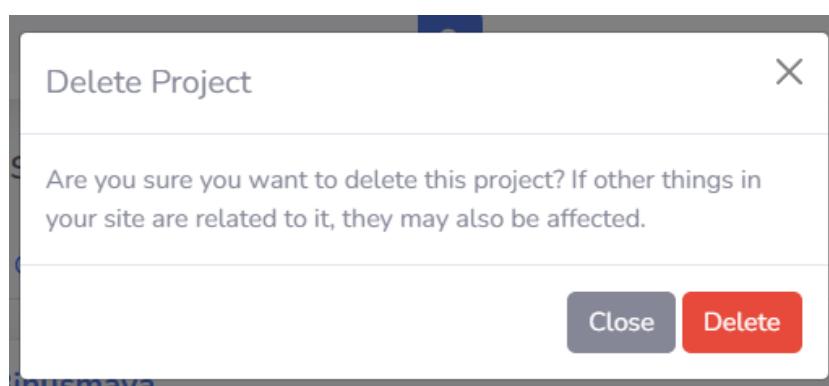
Evaluasi ini mengarahkan untuk memberikan kesempatan *user* untuk membatalkan aksi yang sudah dilakukan Seperti contoh pada fitur *Project* dan *Resources* yang memiliki beberapa tombol yang dapat memunculkan jendela

kecil. Ketika jendela kecil muncul, *user* akan diberikan sebuah tombol *Close* yang berfungsi untuk membatalkan aksi yang sudah dilakukan.



**Gambar 4.93 Tampilan user setelah klik tombol Add Team Member**

Gambar diatas muncul ketika user mengklik tombol *Add Team Member* saat di halaman *Form Edit Project* dan *Form Add New Project*, pada bagian paling bawah jendela kecil terdapat tombol *Close* yang di sebelah kiri tombol *Add*, tombol tersebut dapat digunakan *user* untuk membatalkan aksi menambahkan *penetration tester*.

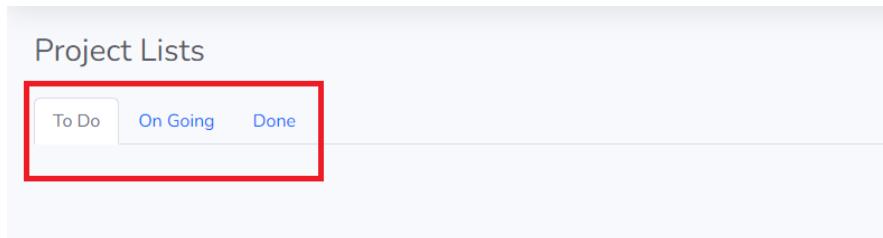


**Gambar 4.94 Tampilan user setelah klik tombol Delete Project di salah satu action project**

Gambar diatas muncul ketika user mengklik tombol *Delete Project* saat di halaman *Project Lists*, pada bagian paling bawah jendela kecil terdapat tombol *Close* yang di sebelah kiri tombol *Delete*, tombol tersebut dapat digunakan *user* untuk membatalkan aksi menghapus *project*.

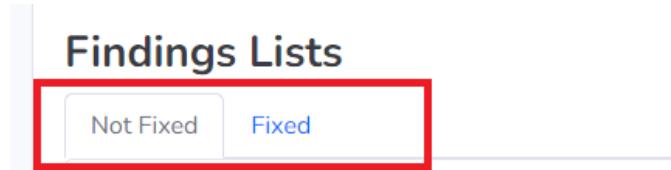
#### 4.5.7 Support Internal Locus of Control

Evaluasi ini mengarahkan agar *user* dapat mengkostumisasi tampilan pada aplikasi. Contohnya terdapat pada penggunaan *project* status fitur *Project Lists*. *User* dibebaskan untuk memilih satu dari tiga status *project* (*To Do*, *Ongoing*, *Done*) untuk melihat daftar *project* yang mempunyai status yang dipilih.



Gambar 4.95 Tampilan Pilihan Status pada *Project Lists*

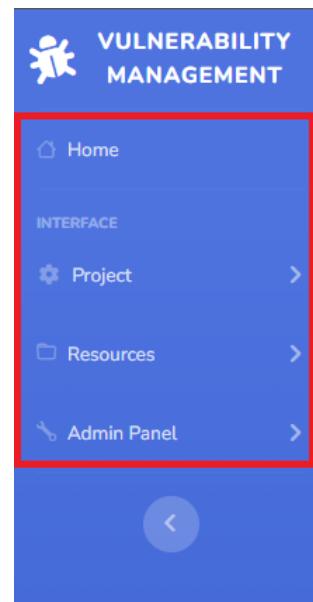
Contoh lainnya adalah pada fitur *Finding Lists* yang terdapat pilihan status *Not Fixed* dan *Fixed*.



Gambar 4.96 Tampilan Pilihan Status pada *Finding Lists*

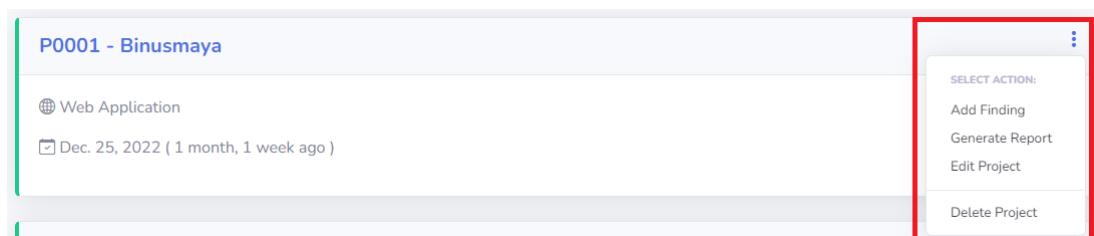
#### 4.5.8 Reduce Short-Term Memory Load

Evaluasi ini mengarahkan untuk menyediakan informasi yang dapat diingat dalam waktu jangka pendek. Contohnya seperti pada fitur menu navigasi. Dengan bantuan *icon* dan tulisan pada fitur ini, dapat memudahkan *user* mengingat terhadap fitur atau aksi tertentu.



**Gambar 4.97 Tampilan Daftar Fitur pada Menu Navigasi**

Contoh lainnya adalah pada menu *icon* titik tiga pada *project* yang memberikan informasi fitur-fitur.



**Gambar 4.98 Tampilan *Icon* Titik Tiga pada *Project***

## 4.6 Perbandingan Aplikasi Sejenis

**Tabel 4.5 Spesifikasi aplikasi *vulnerability management***

Perbandingan	Confluence	Vulnerability Management
Fungsi	Platform kolaborasi dan manajemen untuk berbagi, mengatur segala sesuatu	Platform kolaborasi dan manajemen untuk mengatur suatu <i>project penetration testing</i>

Kolaborasi	<i>User</i> dapat berkolaborasi dalam suatu <i>page</i>	<i>User</i> dapat berkolaborasi pada suatu <i>project</i>
<i>Checklist</i>	Tidak memiliki fitur <i>Checklist</i>	Memiliki fitur <i>Checklist</i> dengan acuan OWASP MSTG dan OWASP WSTG
<i>CVSS Calculator</i>	Tidak memiliki fitur <i>CVSS Calculator</i>	Memiliki fitur CVSS <i>Calculator</i> untuk mempermudah penghitungan <i>severity</i>
<i>Common Weakness Enumeration</i>	Tidak memiliki fitur <i>Common Weakness Enumeration</i>	Memiliki fitur <i>Common Weakness Enumeration</i> untuk mempermudah mengkategorikan <i>finding</i>
<i>Generate Report</i>	Memiliki fitur <i>Generate Report</i> dengan pilihan <i>extension</i> yang beragam	Memiliki fitur <i>Generate Report</i> dalam <i>extension PDF</i>
<i>Testing Guide</i>	Tidak memiliki fitur <i>Testing Guide</i> sebagai panduan pengetesan untuk <i>penetration tester</i>	Memiliki fitur <i>Testing Guide</i> sebagai panduan pengetesan untuk <i>penetration tester</i>

## **BAB 5**

### **SIMPULAN DAN SARAN**

#### **5.1 Simpulan**

Berdasarkan hasil pembuatan, *testing* yang telah dilakukan dan evaluasi dari *user feedback* terhadap Aplikasi *Vulnerability Management*, penulis memperoleh kesimpulan sebagai berikut:

1. Aplikasi *Vulnerability Management* memiliki tampilan yang mudah dipahami dan fitur yang mudah digunakan serta setiap fungsi di dalam aplikasi sudah berjalan dengan baik. Hal ini didasari oleh *testing* terhadap fungsi-fungsi aplikasi dan juga hasil *user feedback* yang telah didapatkan dari *penetration tester* dan *client*.
2. Aplikasi *Vulnerability Management* telah mempermudah *penetration tester* dan *client* dalam melakukan kolaborasi setiap *project penetration testing*. Hal ini didasari oleh hasil *user feedback* yang telah didapatkan dari *penetration tester* dan *client*.
3. Aplikasi *Vulnerability Management* dapat membuat report *penetration testing* untuk mempermudah proses *penetration testing*. Hal ini didasari oleh hasil *user feedback* yang telah didapatkan dari *penetration tester* dan *client*.
4. Aplikasi *Vulnerability Management* telah membantu *junior penetration tester* dan *senior penetration tester* dalam saling memberikan pengetahuan tentang *penetration testing*. Hal ini didasari oleh hasil *user feedback* yang telah didapatkan dari *penetration tester*.

#### **5.2 Saran**

Dari evaluasi yang telah dilakukan berdasarkan *feedback* dari para *pentester* dan *client*, ada beberapa saran yang diberikan pengguna untuk pengembangan Aplikasi *Vulnerability Management* yaitu:

1. Aplikasi *Vulnerability Management* dapat menambahkan *Frequently Asked Question (FAQ)* atau dokumentasi setiap fitur bagi pemula.
2. Aplikasi *Vulnerability Management* dapat menambahkan fitur *logging* untuk mengetahui *user* yang pernah mengakses atau mengunduh *file report* serta diberikan proteksi *password* untuk melihat isi *file report*.
3. Aplikasi *Vulnerability Management* dapat membuat database *finding* yang sudah memuat deskripsi seperti *background*, *impact*, *CVSS*, dan lain-lain sehingga ketika memasukan *finding* sudah tinggal dipilih.
4. Aplikasi *Vulnerability Management* dapat menambahkan *view page* yang menampilkan sisa *mandays* dari masing-masing *client*.
5. Aplikasi *Vulnerability Management* dapat menambahkan status *risk accepted* untuk masing-masing *finding* dan *client* dapat memberikan komentar untuk *finding* yang *risk accepted*.
6. Aplikasi *Vulnerability Management* dapat mengganti *change password* di *user management* menjadi *reset password* agar hanya *user* itu yang mengetahui *password*-nya.

## REFERENSI

- Ardhana, Y. M. K., & Kusuma, S. T. (2012). PHP Menyelesaikan website 30 juta. Yogyakarta: Jasakom.
- Badan Siber dan Sandi Negara (BSSN). (2022). Laporan Tahunan Monitoring Keamanan Siber 2021. <https://cloud.bssn.go.id/s/Lyw8E4LxwNiJoNw>
- Bahaa, A., Kamal, A., & Ghoneim, A. (n.d.). A systematic literature review on software vulnerability detection. Retrieved July 6, 2022, from [https://fcihib.journals.ekb.eg/article\\_214737\\_2f77698c247daf7e39f8760babfd1f60.pdf](https://fcihib.journals.ekb.eg/article_214737_2f77698c247daf7e39f8760babfd1f60.pdf)
- Bashari Rad, B., Bhatti, H. J., & Ahmadi, M. (2017). An Introduction to Docker and Analysis of its Performance. International Journal of Computer Science and Network Security, 17(3), 27-32. Retrieved from [http://paper.ijcsns.org/07\\_book/201703/20170327.pdf](http://paper.ijcsns.org/07_book/201703/20170327.pdf)
- Beller, M. (Ed.). (2021). Software testing: Principles and practices (2nd ed.). John Wiley & Sons.
- Carroll, J. M. (2016). Making use: Scenario-based design of human-computer interactions (2nd ed.). The MIT Press.
- Connolly, T., & Begg, C. (2015). Database Systems: A Practical Approach to Design, Implementation, and Management (6th ed., Global ed.). Harlow, England: Pearson Education Limited.
- Docker, Inc. (n.d.). Docker Compose documentation. Retrieved from <https://docs.docker.com/compose/>
- Elmasri, S., & Navathe, B. (2019). An Introduction to Entity-Relationship Modeling. ACM Computing Surveys, 51(4), 1-31. <https://doi.org/10.1145/3317251>
- Fowler, M. (2015). Refactoring: Improving the design of existing code. Addison-Wesley.

- Greenberg, S. (2018). Designing interfaces (4th ed.). O'Reilly Media.
- Hejlsberg, A., & al. (2016). ECMAScript 2016 Language Specification. ECMA International.
- Herman Tolle (penulis); Aryo Pinandito (penulis); Agi Putra Kharisma (penulis); Ratih Kartika Dewi (penulis); Tim UB Press (editor); Tim UB Press (penyunting). (2017). Pengembangan aplikasi perangkat bergerak (konsep & implementasi)/ Dr. Eng., Herman Tolle, ST., MT, Aryo Pinandito, ST., M.MT., Agi Putra Kharisma, ST., MT, Ratih Kartika Dewi, ST., M.Kom; editor, Tim UB Press; penyunting, Tim UB Press. Malang:: UB Press.,
- Irfan, A. (2020). Implementasi Teknologi Cloud Computing Menggunakan Amazon Web Service Sebagai Media Pendukung Alur Deploying Di PT Ninety Nine Dot Co. Other thesis, Universitas Komputer Indonesia.
- Jorgensen, P. (2019). Software testing: A guide to the TMap approach (3rd ed.). Van Haren Publishing.
- K, N., A, A., Ravichandran, C., Varshini, B., & P, C. (2019). Web Application Penetration Testing. VOLUME-8 ISSUE-10, AUGUST 2019, REGULAR ISSUE, 8(10), 1029-1035. <https://doi.org/10.35940/ijitee.j9173.0881019>.
- Kemp, S. (2021, November 4). Digital in Indonesia: All the statistics you need in 2021 - DataReportal – global digital insights. DataReportal. Retrieved July 6, 2022, from <https://datareportal.com/reports/digital-2021-indonesia>.
- Kolko, J. (2018). Beyond the golden rule: A new model of usability. UX Magazine, 17(1), 56-63.
- Kristanto, E. B., Andrayana, S., & Benramhman, B. (2020). Application of Waterfall SDLC Method in Designing Student's Web Blog Information System at the National University: Application of Waterfall SDLC Method in Designing Student's Web Blog Information System at the National University. Jurnal Mantik, 4(1), 472-482.

- Larman, C. (2018). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Prentice Hall
- Mamun, M. A., Rahman, M. M., & Islam, M. M. (2021). A Systematic Literature Review on Project Management Techniques for Software Development. IEEE Access, 9, 99952-99970
- McKinnel, D. R., Dargahi, T., Dehghanianha, A., & Choo, K. K. R. (2019). A systematic literature review and meta-analysis on artificial intelligence in penetration testing and vulnerability assessment. Computers & Electrical Engineering, 75, 175-188.
- MITRE. CWE - common weakness enumeration. (n.d.). Retrieved July 6, 2022, from [https://cwe.mitre.org/data/published/cwe\\_latest.pdf](https://cwe.mitre.org/data/published/cwe_latest.pdf)
- Mell, P., & Gueye, A. (2020). A suite of metrics for calculating the most significant security relevant software flaw types. 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC). <https://doi.org/10.1109/compsac48688.2020.0-201>.
- Nagpure, S., & Kurkure, S. (2017, August). Vulnerability assessment and penetration testing of Web application. In 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA) (pp. 1-6). IEEE.
- Nanda, S., & Ghugar, U. (2017). Approach to an efficient vulnerability management program. International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), 5(6).
- Nielsen, J., & Molich, R. (2014). Heuristic evaluation. In J. Jacko (Ed.), Human-computer interaction: Development process (pp. 359-380). Routledge.
- Palo Alto Networks. (2020). The State of Cybersecurity in ASEAN 2020. Palo Alto Networks. <https://www.paloaltonetworks.sg/resources/whitepapers/the-state-of-cybersecurity-in-asean-2020>
- Rahman, S. F. (2014). Jump Start Bootstrap. SitePoint.

- Saad, E., & Mitchell, R. (2020). OWASP Web Security Testing Guide v4.2. OWASP.
- Saputra, D. (2018). Analisis Perbandingan Performa Web Service Rest Menggunakan Framework Laravel, Django Dan Ruby On Rails Untuk Akses Data Dengan. Jurnal Bangkit Indonesia, 7(2), 17-17.
- Satzinger, J., Jackson, R., & Burd, S. (2015). Systems analysis and design in a changing world (6th ed.). Boston, MA: Cengage Learning.
- Shneiderman, B. (2020). The eight golden rules of interface design. ACM Transactions on Computer-Human Interaction, 27(5), 1-28.
- Sianipar, (2015). HTML 5 dan CSS 3: Belajar dari kasus. Jakarta: Elex Media Komputindo.
- Sommerville, I. (2019). Software Engineering. Addison-Wesley.
- Srinath, K. R. (2017). Python—the fastest growing programming language. International Research Journal of Engineering and Technology, 4(12), 354-357.
- Start Bootstrap. (2022). SB Admin 2 [Web template, version 3.5]. Retrieved from <https://startbootstrap.com/theme/sb-admin-2>
- Velu, V. K. (2016). Mobile application penetration testing: Explore real-world threat scenarios, attacks on mobile applications, and ways to counter them. Packt Publishing.

## LAMPIRAN

### **LAMPIRAN 1 - Transkrip Wawancara mengenai *User Requirement***

#### ***Transkrip Wawancara dengan Henky Tornado***

Penulis : Terima kasih sudah meluangkan waktunya untuk bersedia di interview, sebelum itu boleh memperkenalkan diri terlebih dahulu dan posisi bekerja saat ini?

Narasumber : Selamat malam semua, perkenalkan saya Henky Tornado, sekarang posisi saya sebagai Senior Pentester Consultant di ITSEC Asia dan Team Leader dari 21 orang pentester.

Penulis : Apa tahapan saat anda mengerjakan project penetration testing?

Narasumber : Kalau dari sisi manajemen, Biasanya pertama ada meeting namanya prerequisite meeting, jadi kita ngobrol sama klien tentang service yang kita tawarkan, misal kita punya monitoring service atau kita ada jasa penesting supaya kita bisa simulasi ketika terjadi serangan atau insiden di aplikasinya itu reworknya gimana. Pertama, kita ada meeting prerequisite, jadi kita dari vendor bakal jelaskan produk kita dan klien jelaskan dari mereka harapannya apa dan detail pentest yang bakal dilakuin kayak apa. Setelah itu bakal ditunjuk dari klien vendor mana yang bakal dipilih (tender), jadi misal kita menang tender baru masuk fase scoping dimana yang menjelaskan lebih detail aplikasi yang mau di-pentest mulai dari fitur, tipe pentest, dll, terus dari vendor ngitung main basenya, jadi disitu misal ada 1 web yang punya 20-25 fitur dan 40 pages, dari kita ngitung berapa lama penggerjaan dan berapa banyak pentester yang terlibat. Setelah dari scoping dan udah jelas, terus ada kick-off meeting dengan developernya (sebelumnya meeting sama bagian it securitynya) dan ngobrol tentang detail pentestnya, setelah itu baru mulai pentestnya. Setelah proses pentest selesai, nanti meeting lagi presentasi tentang finding yang ditemui waktu pentest dan jelasin juga sebagai konsultan best practice yang bisa dilakuin. Setelah itu baru dibikin final reportnya dari proses pentestnya.

- Penulis : Apakah ada pekerjaan yang berulang dalam penetration testing dan merasa perlu ada tools?
- Narasumber : Banyak sih, contohnya misal kalo dari sisi teknis misal kita disuruh ngecek injection, kan gak mungkin kita ngetes payload ini di parameter ini di request ini, nah kalo ada 100 parameter kan gak mungkin dilakuin manual. Kalo dari manajemen, yang berulah tuh follow-up projek, jadi kayak misal project udah selesai tapi ada miskom ditanyain “ini udah selesai?”, “udah sampai mana?”. Jadi dari alur komunikasinya agak susah, jadi dari klien ngomong ke PM mereka, dan kita ngomong ke PM kita, jadi komunikasinya antar PM, jadi sering miskom tentang pentestnya.
- Penulis : Apa kendala yang dihadapi dalam melakukan pentest sebagai pentester senior?
- Narasumber : Kendala sih dari gua pribadi time management, jadi susah bagi waktu antara jadi team lead dan senior pentester consultant dimana projek masih numpuk dan susah tapi harus memimpin tim yang isinya banyak orang mulai ngajarin, update, ngeliatin mereka. Kalo dari sisi teknis biasanya banyak klien perusahaan bilangnya menjajal tinggi “kita mendukung cyber security untuk dikembangkan” tapi dibelakang omongannya gak sesuai sama action mereka. Misalkan ada finding IDOR, di fixnya bukan object-access control atau ditambah autentikasi khusus tapi malah ngomongnya fixnya requestnya di-encrypt. Mereka lebih milih quick fixnya. Dan kalo ngejelasin finding di Indonesia susah karena kurangnya cyber security awareness di tingkat manajemen atau developer, kayak mereka gak peduli resiko aplikasi yang mereka bikin, misal ada finding root detection, nah dari developernya “kita gak bisa memperbaiki kalo misal root detection salah user kenapa pake hp yang rooted”, padahal kan harusnya edukasi dari pembuat aplikasi kita ga boleh jalan di root karena data2nya bisa diekstrak dan diinfesksi malware. Karena kurangnya awareness dan responsibility tersebut terciptalah indonesia sekarang.
- Penulis : Apa kendala pentester junior yang diketahui saat melakukan penetration testing sebagai pentester senior?

Narasumber : Di junior gua masih belum ada rasa harus protect mereka gitu, pas jadi junior cuma kerja kelar dapet finding bagus selesai gitu. Dan yang menghambat pas jadi junior itu transisi dari kuliah ke dunia kerja, kuliah itu kerja tapi kayak dummy system dari kerja dibebasin gagal sebanyak-banyaknya tapi pas masuk dunia kerja disitu lu harus membuktikan apa yang lu dapet pas kuliah. Jadi kayak pas gua dapet finding tapi gak bisa jelaskan karena itu gak ada di OWASP jadi bingung nyari justifikasi findingnya. Dan masalah lainnya ya time management pas itu masih semester 4 jadi ketua CSC dan tim petir, jadi ya agak susah ngatur waktu. Kalo yang junior sekarang itu gampang nyerah ketika ketemu sesuatu yang susah. Misal ada sesuatu yang bisa dieksplloit, mereka berhenti ketika dapet indikasinya gak dilanjutin lagi. Kalo mau lanjutin harus ganti payload, itukan cukup susah dan mereka terkadang engsi nanya atau nyari di google gak ketemu jadi kayak tertekan pas kerja. Rata-rata masalah junior sekarang ada di komunikasi dan gampang nyerah gitu sih.

Penulis : Bagaimana cara anda melakukan kolaborasi dengan pentester lain di project tersebut?

Narasumber : Biasa sih pas pentest sama junior gini, gua ngecek finding2 yang aneh, business logic dll yang perlu analisis lebih, dan yang junior biasanya gua suruh bebas dan ngecek checklist WSTG atau MSTG. Gua ngerjain yang lebih advanced poc nya dari logic businessnya. Nah, untuk komunikasi gitu ga ada cuma ngobrol-ngobrol aja, kalo udah selesai baru digabung. Setiap 2 atau 3 hari sekali kita ngobrol apa yang ditemuin, tuker pikiran tentang masalah yang ada. Kalau Platformnya yang biasa digunakan adalah Confluence.

Penulis : Apa kendala yang anda hadapi saat melakukan kolaborasi dengan pentester lain?

Narasumber : Sama sih, komunikasi, kayak yang tadi, kadang kalo ketemu orang kayak gua yang lagi fokus kan diem, kita cuma ngomong pas pentest udah selesai, jadinya gak efektif kayak kita gatau apa yang dicek masing2. Makanya kadang ada finding yang berulang, itu karena komunikasi. Itu sering terjadi karena lagi males ngomong atau lagi concurrent project atau lu tipikal orangnya kayak gitu, tapi itu semua

bisa di solve pake sistem misal pake excel isinya data pas pentest biar gak bentrok.

Penulis : Apa kegiatan yang dilakukan saat berhubungan dengan Client?

Narasumber : Ada, itu nanti pas selesai pentest ada final presentation atau pemaparan dari finding kita, dijadwalin sama PM, terus ngeliat tanggal yang bisa, baru meeting untuk ngejelasin finding dan rekomendasinya. Bedanya misal kalo final report belum selesai, kita pake raw report dari confluence.

Penulis : Apa kendala yang dihadapi saat berhubungan dengan klien yang anda ketahui?

Narasumber : Klien bebal, kurang awareness, paling pas sesi setelah pentest, ada klien yang gak suka dikasih rekomendasi konsep. Jadi kayak ngasih rekomendasi sesuai owasp atau blog tertentu gak mau, maunya dikasih langsung ini codingannya begini.

Penulis : Baik, terima kasih waktunya ko Henky sudah bersedia kami interview.

Narasumber : Sama-sama.

### ***Transkrip Wawancara dengan Rafi Ramzy***

- Penulis : Terima kasih sudah meluangkan waktunya untuk bersedia di interview, sebelum itu boleh memperkenalkan diri terlebih dahulu dan posisi bekerja saat ini?
- Narasumber : Nama saya Rafi Ramzy, posisi sebagai consultant pentester di ITSEC Asia.
- Penulis : Apa tahapan saat anda mengerjakan project penetration testing?
- Narasumber : Secara manajemen yang pertama meeting dengan klien, sebelumnya mungkin ada scoping dulu meeting dengan klien, kemudian ada proses pentest, terus reporting, kemudian pemaparan finding biasanya kalo misal klien ada yang bingung, kemudian phase retest misal finding sudah diperbaiki. Kemudian kalo udah selesai semua, proses pentest selesai, tidak menyisakan finding yang open mau itu findingnya di fix atau di close sama aja.
- Penulis : Apakah ada pekerjaan yang berulang dalam penetration testing dan merasa perlu ada tools?
- Narasumber : Sebenarnya dulu gua pernah mau bikin kayak gini tapi untuk gua sendiri, jadi gua nyimpen list vulnerability tapi gua hanya ingin menyimpan rekomendasinya, scriptnya gimana, risknya tapi dengan seiring waktu sudah tidak diperlukan tapi itu hal yang berulang gitu kayak kita nyari rekomendasinya seperti apa atau treatmentnya harus seperti apa. Hampir sama seperti reporting referensi, jadi kita kan harus nyari referensi yang sesuai kebutuhan klien, misal kita pentest framework A, jadi kita nyari referensinya dari framework A.
- Penulis : Apa kendala yang dihadapi dalam melakukan pentest sebagai pentester senior?
- Narasumber : Biasanya penentuan risk, karena tergantung dari sisi bisnis dan teknikal bisa sejauh apa, kedua dari segi fundamental lebih penting menurut gua, karena mungkin ada hal yang kalian gak tau ini boleh dilakukan atau nggak, itu yang penting sih dan kadang masih ditemukan sama gua. Kalau dari sisi manajemen, managemennya udah dibilang rapi, karena kita ada technical writer disitu, jadi kita cuma ngepost aja dan mereka yang ngurus reportnya sampai dirilis ke klien.

Jadi, kalau di manajemen belum ada kendala. Mungkin kalau di vendor lain, mungkin ada kendala disitu karena pentester sendiri yang buat reportnya.

- Penulis : Apa kendala pentester junior yang diketahui saat melakukan penetration testing sebagai pentester senior?
- Narasumber : Udah pasti dari sisi teknikal, cara kita baca logic business, ngasih rekomendasi juga, kemudian saat kita adu argumen sama klien, tapi kita ngasih tau kira-kira kalau mereka nanya kita bantuin sih.
- Penulis : Bagaimana cara anda melakukan kolaborasi dengan pentester lain di project tersebut?
- Narasumber : Kolaborasi banyak cara, kalau gua menerapkan misalkan kita kolaborasi ber-4, kalau misalkan udah sesama pentester, konsultan dengan konsultan, junior dengan junior, mungkin misalkan kalau masih magang dikasih task tertentu kayak tolong ini atau cek ini. Cuma kalau udah konsultan, ada yang ngecek keseluruhan, misal ada 2 orang, gua sama alfredo misalkan ngecek 1 web full atau ada yang bagi tugas alfredo fitur a-b, gua fitur c-d. Tapi tergantung platform yang di pentest jadi opsional. Platform yang dipakai oleh itsec semacam cms gitu, kita ngepost kesana isinya deskripsi, poc, threat risk, rekomendasi, dan ada juga list retest yang sudah dilakukan dan pocnya, jadi sesama pentester tinggal ngeliat misal gua ngerjain projek A, nah misal ryo mencari project A, jadi ryo bisa ngeliat semua finding di project A itu. Kalau misal ryo nyari rekomendasi finding idor on x, bisa dicari lewat situ. Jadi kita bisa ngeliat kerjaan orang lain dan bisa ngedit kalau misal ada retest atau misal kita bisa pake sistem draft, jadi misalkan kita belum post, masih draft, disitu ada link drafnya, jadi bisa di edit barengan.
- Penulis : Apa kendala yang anda hadapi saat melakukan kolaborasi dengan pentester lain?
- Narasumber : Masalah ya, kalau kita ngomongin subjektif, kalo gua bisa bilang pentest itu subjektif, karena pentest itu hasilnya bisa jadi beda, misal gua sama ryo pentest, hasil ryo sama gua bisa beda 1 atau bisa aja ada yang miss. Untuk itu, komunikasi itu penting dalam kolaborasi. Kalo menurut gua di platform itsec ini, dia kurang cuma 1 entah itu komen

atau apa yang digunakan misal oh ini udah selesai, oh ini belum selesai di test. Kalau dari sisi teknikal, kadang misal kayak tadi alfredo di a-b, gua c-d, gua gak suka kayak gitu karena biasanya pentester suka miss, mending berdua ngerjain full untuk menghindari ada yang terlewat.

Penulis : Apa kegiatan yang dilakukan saat berhubungan dengan Client?

Narasumber : Singkat sih, kita cuma ngasih tau finding apa aja yang ditemukan, risk sampai mana, kerugian bisa sejauh apa, terus rekomendasinya kayak gimana biar developer cepat benerin atau gak pusing cara fixnya. Paling juga misal kalau pas proses pentest ada error di produknya, kita interaksi sama klien biar dibenerin atau minta data yang kurang. Misal kalau pentest pdam, ada fitur bayar pdam, tapi gak ada nomor pdamnya pasti bingung ngetestnya gimana jadi butuh datanya.

Penulis : Apa kendala yang dihadapi saat berhubungan dengan klien yang anda ketahui?

Narasumber : Paling kalau sama klien tergantung orangnya, kalau orangnya batu, beda cara nanggepin orangnya. Kalau mereka kurang terima gitu, coba sebisa mungkin menjelaskan kenapa kita laporin finding itu, itu sebenarnya butuh pengalaman juga, itu yang sejauh ini menjadi poin penting kenapa finding ditolak, nah itu butuh pengalaman sehingga kita bisa menjelaskan dengan baik biar gak ditolak dan jelaskan risk betulnya (real world scenario) kalau misal finding ini valid, apa yang bisa kena impact sebenarnya. Ketika klien bantah “oh, ini dibenerin kok” “oh, nanti di production gak kayak gini kok” nah itu butuh pertimbangan lagi, gak bisa kita langsung bilang oh boleh atau oh ya jangan.

Penulis : Baik, terima kasih waktunya ko Ramzy sudah bersedia kami interview.

Narasumber : Sama-sama.

***Transkrip Wawancara dengan Christoval Leaved***

- Penulis : Terima kasih sudah meluangkan waktunya untuk bersedia di interview, sebelum itu boleh memperkenalkan diri terlebih dahulu dan posisi bekerja saat ini?
- Narasumber : Halo, nama gua Christoval Leaved, gua Associate Security Consultant di Vantage Point Security dan ini rekan saya Ming, angkatan B22 juga.
- Penulis : Apa tahapan saat anda mengerjakan project penetration testing?
- Narasumber : Kalau di profesional, biasanya dari klien ada RFP (Request For Proposal), kemudian menentukan scope fitur apa saja yang mau dites, setelah itu consultant yang akan mengerjakan mulai dari identifikasi, eksplorasi, dan membuat report dari hasil pentest. Setelah itu bakal ada regression testing untuk retest.
- Penulis : Apakah ada pekerjaan yang berulang dalam penetration testing dan merasa perlu ada tools?
- Narasumber : Kita biasa pake tools-tools kayak burp, nessus, dll. Cuma mungkin yang berulang saat reporting kalau dapat finding yang sama dan harus didokumentasi itu biasanya yang berulang.
- Penulis : Apa kendala yang dihadapi dalam melakukan pentest sebagai pentester junior?
- Narasumber : Kendala yang sering terjadi biasanya bingung cara testing WSTG, jadi memerlukan referensi, dan termasuk penulisan reportnya.
- Penulis : Bagaimana cara anda melakukan kolaborasi antara pentester?
- Narasumber : Biasa kita komunikasinya lewat Teams, misalnya Pentester lain dapat SQLi di fitur A pake payload X, nah nanti payload X ini bakal dicoba juga sama yang lain di fitur yang mereka pentest.
- Penulis : Apakah ada kendala dalam melakukan kolaborasi antara pentester?
- Narasumber : Karena WFH, kita nggak tau pentester lain itu lagi kerja atau diluar, jadi kalau kita lagi butuh cepat dia gak respon.
- Penulis : Apa kegiatan yang dilakukan saat berhubungan dengan Client?
- Narasumber : Setelah kita selesai pentest, reporting, terus kita presentasi findingnya ke klien. Terus ada juga retest, nah pas retest gak selalu semua findingnya solve langsung, nah biasanya itu kita komunikasikan lagi

ke klien kalo ini masih vuln. akhir-akhir ini dapat vuln nya di logicnya, kalaupun difix, kliennya suka nanya ini harus diapain? Atau ini cara fixnya gimana? Dari kitanya ngasih opsi-opsi doang.

Penulis : Apa kendala yang dihadapi saat berhubungan dengan klien yang anda ketahui?

Narasumber : Kadang klien juga minta ini itu, benerin report lah, grammarnya salah, kurang lah. Sama client juga sering minta file report dan lihat progress sekarang.

Penulis : Baik, terima kasih waktunya ko Christoval sudah bersedia kami interview.

Narasumber : Sama-sama.

***Transkrip Wawancara dengan Ming Hong***

- Penulis : Terima kasih sudah meluangkan waktunya untuk bersedia di interview, sebelum itu boleh memperkenalkan diri terlebih dahulu dan posisi bekerja saat ini?
- Narasumber : Halo semuanya, nama saya Ming Hong dari binusian 22 sekarang bekerja sebagai Security Consultant di Vantage Point.
- Penulis : Apa tahapan saat anda mengerjakan project penetration testing?
- Narasumber : Kalau di profesional, biasanya dari klien ada RFP (Request For Proposal), kemudian menentukan scope fitur apa saja yang mau dites, setelah itu consultant yang akan mengerjakan mulai dari identifikasi, eksplorasi, dan membuat report dari hasil pentest. Setelah itu bakal ada regression testing untuk retest.
- Penulis : Apakah ada pekerjaan yang berulang dalam penetration testing dan merasa perlu ada tools?
- Narasumber : Ada, lumayan banyak. Karena di vantage point kebanyakan melakukan pentest secara manual, paling pas static analysis menggunakan tool otomatis, tapi kalau dynamic analysis belum ada. Tulis finding atau reporting atau dokumentasi yang biasanya berulang.
- Penulis : Apa kendala yang dihadapi dalam melakukan pentest sebagai pentester junior?
- Narasumber : Karena kita mengerjakannya masih manual, mungkin ada checklist (OWASP dsb) yang terlewat atau lupa test. Reportnya juga masih manual.
- Penulis : Bagaimana cara anda melakukan kolaborasi antara pentester?
- Narasumber : Biasa kita komunikasinya lewat Teams, kan ada daftar fitur yang mau di pentest, nah kita bagi tugas aja. Pentester A pentest fitur A dan seterusnya.
- Penulis : Apakah ada kendala dalam melakukan kolaborasi antara pentester?
- Narasumber : Bingung dengan pentester lain lagi kerjain yang mana, nanti takutnya kita lagi kerjain pada bagian yang sama dan akhirnya menghabiskan waktu yang sia sia.
- Penulis : Apa kegiatan yang dilakukan saat berhubungan dengan Client?

Narasumber : Setelah kita selesai pentest, reporting, terus kita presentasi findingnya ke klien. Terus ada juga retest, nah pas retest gak selalu semua findingnya solve langsung, nah biasanya itu kita komunikasikan lagi ke klien kalo ini masih vuln.

Penulis : Apa kendala yang dihadapi saat berhubungan dengan klien yang anda ketahui?

Narasumber : Kadang bahasa kita terlalu tinggi, makanya klien gak ngerti, kita jelaskan apa klien nggak paham gitu jadi dianggap gak vuln dan gak nerima findingnya, dan ada juga client yang sering minta file report dan progress pentest sekarang.

Penulis : Baik, terima kasih waktunya ko Ming Hong sudah bersedia kami interview.

Narasumber : Sama-sama.

*Transkrip Wawancara dengan Ferry Alifani*

Penulis : Terima kasih sudah meluangkan waktunya untuk bersedia di interview, sebelum itu boleh memperkenalkan diri terlebih dahulu dan posisi bekerja saat ini?

Narasumber : Namaku Ferry Alifani, sekarang aku di blibli sebagai red team atau offensive security engineer.

Penulis : Bagaimana tahapan saat menggunakan jasa pentest pada produk anda?

Narasumber : Ini pengalamanku ya, pertama kita open tender, bahwa ada projek pentest blibli, terus kita ceritakan scopenya apa aja, entah itu blibli.com atau sellernya. Terus kita undang kandidat vendornya, misalkan MII, horangi, ethic.ninja, atau yang lainnya. Terus kita kasih scope yang akan di pentest, masing-masing kandidat akan review scopenya seluas apa atau sekompelks apa pentestnya. Lalu mereka akan memasang harga masing-masing vendor berdasarkan scope yang ada kalau mau menggunakan jasa mereka. Dari sisi klien akan memberi maksimum budget yang dimiliki, lalu klien memilih dengan budget sekian, scope sekian memilih vendor mana yang dipilih berdasarkan track record, history, harga, dll. Setelah itu akan ada diskusi internal klien, misal kita pilih horangi, yaudah deal lalu ngomong lebih lanjut mengenai scope, waktu, siapa yang terlibat, waktu pentest, dan sampai jadi reportnya. Dan kira kira berapa mandays yang mereka kalkulasikan. Kalau saat masa pentest Kan kalo scoping tadi udah dan fix vendornya, mereka langsung eksekusi, kadang mereka tanya memastikan metode pentestnya. Setelah itu kalo black box langsung pentest, kalo white box mungkin kita kasih creds atau token vpn. Agar kita dapat update finding, kita dibagi akses ke spreadsheet, kita kan kalo nunggu sampai selesai report kan lama, takutnya ada sesuatu yang ditunggu entah itu compliance atau apapun itu jadi dishare finding dalam bentuk spreadsheet isinya nomor, nama finding, dan severitynya. Dan kalau sudah selesai pentest dan reportnya terus dikasih ke kita.

Penulis : Apakah ada kendala dari anda sebagai client ke pentester pada saat proses jasa pentest tersebut?

Narasumber : Kendala pernah ada, jadi waktu itu scope yang di pentest butuh VPN, itukan kita memberi fasilitas dalam bentuk token, nah tokennya itu nggak berfungsi jadi mereka gak bisa akses. Itu aja sih kendalanya. Kalau blackbox kadang mereka tanya fitur ini kesini ya maksudnya. Kalau komunikasi so far aman. Kalau dari finding, So far pernah ada kendala misalkan mereka ngasih summary finding nya terlalu sedikit, kayak nomor, nama finding, dan severity. Dari kita kan gak ngerti maksudnya findingnya kemana nih, di module apa, di fitur apa, impactnya apa, dan kalau valid seperti apa mitigasinya. Dari mereka terlalu singkat informasi findingnya, dari sisi klien kan kan juga perlu ngasih tau ke tim devnya. Jadi jangan terlalu singkat diberikan impactnya apa dan yang lain-lain.

Penulis : Bagaimana cara anda dan pentester melakukan diskusi selama masa penetration testing?

Narasumber : So far ada 3 platform, whatsapp, telegram, atau email.

Penulis : Apakah ada hal yang ingin dipermudah dalam kolaborasi dengan pentester?

Narasumber : Mungkin tadi sih terkait penyampaian finding, mungkin lebih diperjelas. Walaupun bersifat sementara, paling tidak apa yang ditemukan, impactnya apa, seperti apa mitigasinya. Jadi nggak nama dan severitynya, jadi kita bingung.

Penulis : Apakah ada informasi yang dirasa kurang dari pihak pentester?

Narasumber : Sebenarnya bukan kurang sih, jadi ketika final report kadang masih ada beberapa finding yang kurang jelas step by stepnya. Yang nemuin kan orang lain ya, kita kadang ada beberapa finding yang ini darimana dulu ya, kok dapet ini sih? Nah kalau itu terjadi, kita bakal hubungin vendornya.

Penulis : Baik, terima kasih waktunya kak Ferry sudah bersedia kami interview.

Narasumber : Sama-sama.

**LAMPIRAN 2 - Transkrip Wawancara mengenai *User Feedback******Transkrip Wawancara dengan Henky Tornado***

Penulis : Siapa nama anda?

Narasumber : Henky Tornado.

Penulis : Apa pekerjaan anda sekarang?

Narasumber : Senior Penetration Tester Consultant dan Team Lead.

Penulis : Apakah User interface mudah dipahami?

Narasumber : Iya.

Penulis : Apakah seluruh fitur mudah digunakan dan dipahami?

Narasumber : Iya.

Penulis : Apakah seluruh fitur berjalan dengan baik?

Narasumber : Iya sudah berjalan dengan sesuai dengan yang sudah dijelaskan.

Penulis : Apakah fitur project dapat membantu atau mempermudah anda dalam kolaborasi dengan pentester lain dan client?

Narasumber : Iya dapat mempermudah dalam konteks monitoring dan juga management project.

Penulis : Apakah fitur Generate Report dapat mempercepat proses pentest?

Narasumber : Iya dapat mempermudah, tapi harus dibuat logging, sehingga kita dapat tau siapa saja yang sudah pernah mendownload report tersebut, karena jika terjadi missunderstanding akan lebih mudah di validasi siapa saja yang sudah pernah mengakses report tersebut.

Penulis : Apakah fitur Resources dapat membantu atau mempermudah anda saat melakukan pentest?

Narasumber : Dapat mempermudah tapi harus diperbanyak default dari developer dengan memasukan standar yang sudah ada sehingga nantinya user tinggal menambahkan teknik baru dan tak perlu mengulang lagi untuk mengisi secara manual.

Penulis : Apakah Aplikasi Vulnerability Management ini dapat membantu atau mempermudah pekerjaan anda?

Narasumber : Iya dapat membantu karena akan sangat mengikis waktu dan juga meningkatkan efisiensi dalam perihal komunikasi.

Penulis : Apakah ada saran terhadap Aplikasi Vulnerability Management ini?

Narasumber : Tolong jangan menciptakan fitur yang dapat menjadi musibah untuk kita semua seperti change password, karena harusnya password bersifat pribadi dan hanya bisa diganti oleh orang tersebut karena password secara konseptual merupakan salah satu faktor yang digunakan dalam konsep autentikasi dimana itu menjadi sebuah validasi bahwa orang tersebut memang real, dikarenakan tersebut harusnya penggantian password hanya dapat dilakukan oleh user itu sendiri, seandainya tetap diperlukan maka hanya bisa dibentuk menjadi reset password dimana itu tidak dapat mengganti password dari orang lain

***Transkrip Wawancara dengan Rafi Ramzy***

Penulis : Siapa nama anda?

Narasumber : Rafi Ramzy.

Penulis : Apa pekerjaan anda sekarang?

Narasumber : Senior Penetration Tester.

Penulis : Apakah User interface mudah dipahami?

Narasumber : Ya sangat mudah, sistemnya juga ok.

Penulis : Apakah seluruh fitur mudah digunakan dan dipahami?

Narasumber : Mudah dipahami penggunaannya, tapi alangkah baiknya di buat faq or docs.

Penulis : Apakah seluruh fitur berjalan dengan baik?

Narasumber : So far oke berjalan dengan baik.

Penulis : Apakah fitur project dapat membantu atau mempermudah anda dalam kolaborasi dengan pentester lain dan client?

Narasumber : Ya sangat mudah.

Penulis : Apakah fitur Generate Report dapat mempercepat proses pentest?

Narasumber : Ya, mempercepat pentest result delivery.

Penulis : Apakah fitur Resources dapat membantu atau mempermudah anda saat melakukan pentest?

Narasumber : Sepertinya lumayan membantu.

Penulis : Apakah Aplikasi Vulnerability Management ini dapat membantu atau mempermudah pekerjaan anda?

Narasumber : Ya, dapat dengan mudah karena tidak perlu adanya pembuatan report, dan checklist sudah tersedia setiap mulai project baru.

Penulis : Apakah ada saran terhadap Aplikasi Vulnerability Management ini?

Narasumber : Pada aplikasi tersebut terdapat input untuk poc, tetapi gambar terlihat terpisah dengan deskripsi. sebaiknya disatukan karena deskripsi dan evidence merupakan sebuah rangkaian kesatuan.

***Transkrip Wawancara dengan Christoval Leaved***

- Penulis : Siapa nama anda?
- Narasumber : Christoval Leaved.
- Penulis : Apa pekerjaan anda sekarang?
- Narasumber : Security Consultant di Vantage Point Security.
- Penulis : Apakah User interface mudah dipahami?
- Narasumber : Yes, secara keseluruhan User Interface mudah dipahami. Terlihat simple dan lumayan lengkap untuk Vulnerable Management.
- Penulis : Apakah seluruh fitur mudah digunakan dan dipahami?
- Narasumber : Yes, secara keseluruhan fitur sudah mencakup kebutuhan pentester untuk melakukan reporting.
- Penulis : Apakah seluruh fitur berjalan dengan baik?
- Narasumber : Yes, semua fitur berjalan dengan baik sesuai penjelasan sebelumnya.
- Penulis : Apakah fitur project dapat membantu atau mempermudah anda dalam kolaborasi dengan pentester lain dan client?
- Narasumber : Yes, fitur tersebut dapat membantu client untuk melihat progress pentest secara keseluruhan.
- Penulis : Apakah fitur Generate Report dapat mempercepat proses pentest?
- Narasumber : Yes, fitur tersebut membantu pentester untuk mempercepat proses reporting dikarenakan pentester tidak membutuhkan templating dalam reporting.
- Penulis : Apakah fitur Resources dapat membantu atau mempermudah anda saat melakukan pentest?
- Narasumber : Yes, fitur tersebut dapat membantu pentester untuk mencari info mengenai pentest.
- Penulis : Apakah Aplikasi Vulnerability Management ini dapat membantu atau mempermudah pekerjaan anda?
- Narasumber : Yes, secara keseluruhan fitur2 yang ada di aplikasi ini dapat membantu pentester untuk mempercepat proses pentest.
- Penulis : Apakah ada saran terhadap Aplikasi Vulnerability Management ini?
- Narasumber : Yes, mungkin untuk enhancement bisa dibuat database finding yang sudah memuat deskripsi seperti background,impact,cvss,&dll sehingga ketika input finding sudah tinggal dipilih.

***Transkrip Wawancara dengan Ming Hong***

Penulis : Siapa nama anda?

Narasumber : Ming Hong.

Penulis : Apa pekerjaan anda sekarang?

Narasumber : Security Consultant di Vantage Point Security.

Penulis : Apakah User interface mudah dipahami?

Narasumber : Iya, dikarenakan penggunaan desain yang minimalis dan konsisten, sehingga fitur dapat diakses lebih mudah.

Penulis : Apakah seluruh fitur mudah digunakan dan dipahami?

Narasumber : Iya, dikarenakan UI & UX yang diterapkan sangat baik, sehingga fitur juga mudah dipahami.

Penulis : Apakah seluruh fitur berjalan dengan baik?

Narasumber : Iya, setelah dilakukan pengujian, fitur dinilai dapat berjalan dengan baik.

Penulis : Apakah fitur project dapat membantu atau mempermudah anda dalam kolaborasi dengan pentester lain dan client?

Narasumber : Iya, fitur pada website dapat mempermudah pemaparan finding.

Penulis : Apakah fitur Generate Report dapat mempercepat proses pentest?

Narasumber : Iya, dikarenakan pentester tidak perlu membuat kerangka laporan dari awal secara manual.

Penulis : Apakah fitur Resources dapat membantu atau mempermudah anda saat melakukan pentest?

Narasumber : Iya, dikarenakan fitur tersebut dapat membantu pentester menjelaskan finding ke pentester lain, sehingga mempermudah proses kolaborasi.

Penulis : Apakah Aplikasi Vulnerability Management ini dapat membantu atau mempermudah pekerjaan anda?

Narasumber : Iya, dikarenakan fitur-fiturnya yang dapat mempermudah proses listing finding dan generate report.

Penulis : Apakah ada saran terhadap Aplikasi Vulnerability Management ini?

Narasumber : Sebaiknya ditambahkan dokumentasi pada setiap fitur agar pentester yang masih pemula dapat lebih memahami fitur lebih baik.

***Transkrip Wawancara dengan Ferry Alifani***

Penulis : Siapa nama anda?

Narasumber : Ferry Alif.

Penulis : Apa pekerjaan anda sekarang?

Narasumber : Cyber Security Engineer.

Penulis : Apakah User interface mudah dipahami?

Narasumber : Yes.

Penulis : Apakah seluruh fitur mudah digunakan dan dipahami?

Narasumber : Yes.

Penulis : Apakah seluruh fitur berjalan dengan baik?

Narasumber : Yes.

Penulis : Apakah fitur project dapat membantu atau mempermudah anda dalam kolaborasi dalam project pentest?

Narasumber : Yes.

Penulis : Apakah fitur Generate Report dapat mempermudah dalam mendapatkan Laporan Pentest setiap waktu dengan Kondisi Pentest yang terbaru?

Narasumber : Yes.

Penulis : Apakah Aplikasi Vulnerability Management ini dapat membantu atau mempermudah pekerjaan anda?

Narasumber : Yes.

Penulis : Apakah ada saran terhadap Aplikasi Vulnerability Management ini?

Narasumber : Ditambahkan view page yang menampilkan sisa mandays dari masing-masing client, Ditambahkan status risk accepted untuk masing-masing temuan, dan client page diberikan hak akses untuk memasukkan komen untuk temuan yang risk accepted, dan Reportnya pentestnya diberikan password.

## LAMPIRAN 3 - Persetujuan Transkrip Wawancara dengan Narasumber

### *Persetujuan Transkrip Wawancara dengan Henky Tornado*

#### Approval Interview User Requirement dan User Feedback - Henky Tornado

RYO CHRIS SAMUEL  
Kepada: Henky Tornado

Selamat sore,

Berkenaan dengan wawancara yang dilakukan sebelumnya dengan Bapak Henky Tornado sebagai narasumber untuk pengumpulan data Tugas Akhir kami, kami mahasiswa Universitas Bina Nusantara jurusan Cyber Security dari Kelompok Skripsi Vulnerability Management yang beranggotakan:

1. Stanley Halim
2. Alfredo Thomas Nofanski
3. Ryo Chris Samuel

Mohon izin untuk approve transkrip wawancara yang telah dilakukan pada link transkrip sebagai berikut  
[https://docs.google.com/document/d/1tyLG18ddwmK\\_ZkP0JaH1qDPZgQDR0Qtneeo8\\_djbE4/edit?usp=sharing](https://docs.google.com/document/d/1tyLG18ddwmK_ZkP0JaH1qDPZgQDR0Qtneeo8_djbE4/edit?usp=sharing) dan <https://docs.google.com/document/d/1Urdz4fZLrPgByf1hGhIFcxBlVZH-58ZyYe-gt91274/edit?usp=sharing>

**Transkrip Wawancara User Feedback dengan Henky Tornado**

Transkrip Wawancara dengan Henky Tornado Penulis : Siapa nama anda?  
 Narasumber : Henky Tornado. Penulis : Apa pekerjaan anda sekarang?  
 Narasumber : Senior Penetration Tester Consultant dan Team Lead. Penulis : Apakah User interface mudah dipahami? Narasumber : Iya. Penulis : Apakah seluruh fitur...

docs.google.com

#### Re: Approval Interview User Requirement dan User Feedback - Henky Tornado

HT Henky Tornado <[henky@itsecasia.com](mailto:henky@itsecasia.com)>  
Kepada: RYO CHRIS SAMUEL

Mulai balasan dengan: [Thank you!](#) [Thank you for the approval.](#) [Great, thanks!](#)

Dear Ryo and team,  
 Approved.  
 Thank you,  
 Regards,  
 Henky Tornado

## **Persetujuan Transkrip Wawancara dengan Rafi Ramzy**

## Persetujuan Transkrip Wawancara dengan Christoval Leaved

### Approval Interview User Requirement dan User Feedback - Christoval Leaved

 RYO CHRIS SAMUEL  
Kepada: christoval.leaved@vantagepoint.co.id

Selamat sore,

Berkenaan dengan wawancara yang dilakukan sebelumnya dengan Bapak Christoval Leaved sebagai narasumber untuk pengumpulan data Tugas Akhir kami, kami mahasiswa Universitas Bina Nusantara jurusan Cyber Security dari Kelompok Skripsi Vulnerability Management yang beranggotakan:

1. Stanley Halim
2. Alfredo Thomas Nolanski
3. Ryo Chris Samuel

Mohon izin untuk approve transkrip wawancara yang telah dilakukan pada link transkrip sebagai berikut [https://docs.google.com/document/d/1TV64XuJwNj7ixylcAMyyV\\_c8MiywPJMUCwzflkyM5c/edit?usp=sharing](https://docs.google.com/document/d/1TV64XuJwNj7ixylcAMyyV_c8MiywPJMUCwzflkyM5c/edit?usp=sharing) dan <https://docs.google.com/document/d/1GuV-tSGQpVzEEVoRuvSVeq8cm3gshjHqQ4vtPjV-o/edit?usp=sharing>

**Transkrip Wawancara User Feedback dengan Christoval Leaved**

Transkrip Wawancara dengan Christoval Leaved Penulis : Siapa nama anda? Narasumber : Christoval Leaved. Penulis : Apa pekerjaan anda sekarang? Narasumber : Security Consultant di Vantage Point Security. Penulis : Apakah User interface mudah dipahami? Narasumber : Yes, secara keseluruhan User Int...  
[docs.google.com](https://docs.google.com)

### RE: Approval Interview User Requirement dan User Feedback - Christoval Leaved

 Christoval Leaved <Christoval.Leaved@vantagepoint.co.id>  
Kepada: RYO CHRIS SAMUEL

Hi Ryo,

Transkrip tersebut sudah OK, thank you for having me.  
Approved, Good Luck!

Best Regards,  
**Christoval Leaved | Security Consultant**  
Office: +62 2152111235 | Mobile: +62 8179271199 E-Mail: [christoval.leaved@vantagepoint.co.id](mailto:christoval.leaved@vantagepoint.co.id)



**VANTAGEPOINT**  
Security at the Speed of Development



**PT. Vantage Point Security Indonesia**  
World Trade Centre I, Lt. 16, Jl. Jend. Sudirman Kav. 29-31, RT 008 RW 003, Jakarta Selatan, DKI. Jakarta, Indonesia. 12920.

*Security at the Speed of Development*

The contents of this e-mail message and any attachments are confidential and are intended solely for addressee. The information may also be legally privileged. This transmission is sent in trust, for the sole purpose of delivery to the intended recipient. If you have received this transmission in error, any use, reproduction or dissemination of this transmission is strictly prohibited. If you are not the intended recipient, please immediately notify the sender by reply e-mail or phone and delete this message and its attachments, if any.

From: RYO CHRIS SAMUEL <[christoval.leaved@vantagepoint.co.id](mailto:christoval.leaved@vantagepoint.co.id)>

## Persetujuan Transkrip Wawancara dengan Ming Hong

**Approval Interview User Requirement dan User Feedback - Ming Hong**

 RYO CHRIS SAMUEL  
Kepada: ming.hong@vantagepoint.co.id

Selamat sore,

Berkenaan dengan wawancara yang dilakukan sebelumnya dengan Bapak Ming Hong sebagai narasumber untuk pengumpulan data Tugas Akhir kami, kami mahasiswa Univeristas Bina Nusantara jurusan Cyber Security dari Kelompok Skripsi Vulnerability Management yang beranggotakan:

1. Stanley Halim
2. Alfredo Thomas Nofanski
3. Ryo Chris Samuel

Mohon izin untuk approve transkrip wawancara yang telah dilakukan pada link transkrip sebagai berikut <https://docs.google.com/document/d/1URFIAGLOXF69tNsv7ZYiHukdUwUldl8XZTbdf2e/edit?usp=sharing> dan [https://docs.google.com/document/d/1Z3uHhWxz2\\_Mta5423alUleYzXWBrfkQiuI97SSNvWo/edit?usp=sharing](https://docs.google.com/document/d/1Z3uHhWxz2_Mta5423alUleYzXWBrfkQiuI97SSNvWo/edit?usp=sharing)

**Transkrip Wawancara User Feedback dengan Ming Hong**

Transkrip Wawancara dengan Ming Hong Penulis : Siapa nama anda?  
 Narasumber : Ming Hong. Penulis : Apa pekerjaan anda sekarang?  
 Narasumber : Security Consultant di Vantage Point Security. Penulis : Apakah User interface mudah dipahami? Narasumber : Iya, dikarenakan penggunaan desain yang minimali...  
[docs.google.com](https://docs.google.com)

**RE: Approval Interview User Requirement dan User Feedback - Ming Hong**

 Ming Hong <ming.hong@vantagepoint.co.id>  
Kepada: RYO CHRIS SAMUEL

Dear Ryo & Team,

Approved ya.

Best Regards,  
**Ming Hong | Security Consultant**  
 Office: +62 215211235 | Mobile: +62 8119121307 E-Mail: [ming.hong@vantagepoint.co.id](mailto:ming.hong@vantagepoint.co.id)

**PT Vantage Point Security Indonesia**  
 World Trade Centre 1, Lt. 16, Jl. Jend. Sudirman Kav. 29-31, RT 008 RW 003, Jakarta Selatan, DKI. Jakarta, Indonesia, 12920.  
*Security at the Speed of Development*

The contents of this e-mail message and any attachments are confidential and are intended solely for addressee. The information may also be legally privileged. This transmission is sent in trust, for the sole purpose of delivery to the intended recipient. If you have received this transmission in error, any use, reproduction or dissemination of this transmission is strictly prohibited. If you are not the intended recipient, please immediately notify the sender by reply e-mail or phone and delete this message and its attachments, if any.

## Persetujuan Transkrip Wawancara dengan Ferry Alifani

Approval Interview User Requirement dan User Feedback - Ferry Alifani

 RYO CHRIS SAMUEL  
Kepada: ferralifan@gmail.com

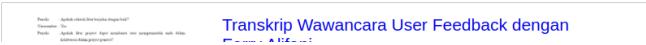
Selamat sore,

Berkenaan dengan wawancara yang dilakukan sebelumnya dengan Bapak Ferry Alifani sebagai narasumber untuk pengumpulan data Tugas Akhir kami, kami mahasiswa Universitas Bina Nusantara jurusan Cyber Security dari Kelompok Skripsi Vulnerability Management yang beranggotakan:

1. Stanley Halim
2. Alfredo Thomas Nofanski
3. Ryo Chris Samuel

Mohon izin untuk approve transkrip wawancara yang telah dilakukan pada link transkrip sebagai berikut <https://docs.google.com/document/d/1EsRbLPPiy3sCWbSD0ATasGcduANICKpQUXMCEf9j28/edit?usp=sharing> dan <https://docs.google.com/document/d/1L6jrxwac7CCvy2ldGeihPznBsuMFZV7vOK0JNHcZvU/edit?usp=sharing>





Re: Approval Interview User Requirement dan User Feedback - Ferry Alifani

① Beberapa konten dalam pesan ini telah diblokir karena pengirim tidak termasuk dalam daftar Pengirim aman Anda. [Saya percaya konten dari ferralifan@gmail.com.](#) | [Tampilkan konten yang diblokir](#)

 Ferralifan <ferralifan@gmail.com>  
Kepada: RYO CHRIS SAMUEL

Halo All, approved yah.

**LAMPIRAN 4 - Surat Non Survei**

**Surat Non Survei**  
No : **006/HoP.CSI/I/2023**

Yang bertandatangan di bawah ini :

Nama : **Yohan Muliono, S.Kom., M.TI., CND, CEHmaster, eMAPT.**  
Jabatan : Head of Cyber Security Program

Menyatakan bahwa mahasiswa berikut ini :

No.	N.I.M	Nama Mahasiswa
1.	2301859146	STANLEY HALIM
2.	2301924536	ALFREDO THOMAS NOFANSKI
3.	2301931503	RYO CHRIS SAMUEL

Sudah melakukan survei di Jurusan Cyber Security Program Universitas Bina Nusantara dengan judul skripsi:

**“APLIKASI MANAJEMEN KERENTANAN BERDASARKAN HASIL UJI KEAMANAN BERBASIS WEBSITE”**

di Lingkungan Universitas Bina Nusantara.

Demikian surat pernyataan ini dibuat untuk digunakan sebagaimana mestinya.

Jakarta, 11 Januari 2023



**Yohan Muliono, S.Kom., M.TI., CND, CEHmaster, eMAPT.**  
Head of Cyber Security Program

**RIWAYAT HIDUP****PERSONAL INFORMATION**

**Binusian ID** : 2301859146  
**Full Name** : Stanley Halim  
**E-Mail** : stanley.halim@binus.ac.id  
**Address** **Current**  
JL Permata Taman Palem Blok D8 No.1  
Kelurahan Pegadungan, Kecamatan Kalideres  
Jakarta Barat, DKI Jakarta  
**Permanent**  
JL Permata Taman Palem Blok D8 No.1  
Kelurahan Pegadungan, Kecamatan Kalideres  
Jakarta Barat, DKI Jakarta  
**Phone Numbers** Home : (021) 54397031  
Mobile : 081908013871  
**Gender** : Male  
**Birth Place / Date** : Jakarta, 25 Oktober 2001  
**Nationality** : Indonesia  
**Marital Status** : Single  
**Religion** : Buddha

**FORMAL EDUCATION**

2019 – 2023	Bina Nusantara University
2016 – 2019	SMA Cinta Kasih Tzu Chi
2013 – 2016	SMP Cendrawasih
2007 – 2013	SD Cendrawasih

**INFORMAL EDUCATION****PERSONAL CERTIFICATION****ORGANIZATION EXPERIENCE**

2020 - 2022	Member of Cyber Security Community (CSC)
2020 - present	Active member Petir Cyber Security Binus University

**WORKING EXPERIENCE**



### PERSONAL INFORMATION

**Binusian ID** : 2301924536  
**Full Name** : Alfredo Thomas Nofanski  
**E-Mail** : alfredo.nofanski@binus.ac.id  
**Address** **Current**  
Gg Buntu No 03, RT 02 / RW 04, Kelurahan  
Keagungan, Kecamatan Taman Sari, Jakarta Barat,  
DKI Jakarta  
**Permanent**  
Gg Buntu No 03, RT 02 / RW 04, Kelurahan  
Keagungan, Kecamatan Taman Sari, Jakarta Barat,  
DKI Jakarta  
**Phone Numbers** Home : -  
Mobile : 081334261321  
**Gender** : Male  
**Birth Place / Date** : Pati, 25 November 2001  
**Nationality** : Indonesia  
**Marital Status** : Single  
**Religion** : Islam

**FORMAL EDUCATION**

2019 – 2023	Bina Nusantara University
2016 – 2019	SMA Negeri 1 Pati
2013 – 2016	SMP Negeri 1 Tayu
2007 – 2013	SD Negeri Grogolan 02

**INFORMAL EDUCATION****PERSONAL CERTIFICATION****ORGANIZATION EXPERIENCE**

2020-2022	Cyber Security Community (CSC)
-----------	--------------------------------

**WORKING EXPERIENCE**

2022-2023	Internship ITSEC Asia
-----------	-----------------------



### PERSONAL INFORMATION

**Binusian ID** : 2301931503  
**Full Name** : Ryo Chris Samuel  
**E-Mail** : ryo.samuel@binus.ac.id  
**Address**  
**Current**  
Jl Mayjen Ishak Djuarsa, Gg Masjid II, RT 2 / RW 11,  
No. 59B  
Kelurahan Loji, Kecamatan Bogor Barat  
Bogor, Jawa Barat  
**Permanent**  
Jl Mayjen Ishak Djuarsa, Gg Masjid II, RT 2 / RW 11,  
No. 59B  
Kelurahan Loji, Kecamatan Bogor Barat  
Bogor, Jawa Barat  
**Phone Numbers** Home : (0251) 8330950  
Mobile : 083112355739  
**Gender** : Male  
**Birth Place / Date** : Bogor, 06 Juni 2001  
**Nationality** : Indonesia  
**Marital Status** : Single  
**Religion** : Katolik

**FORMAL EDUCATION**

2019 – 2023	Bina Nusantara University
2016 – 2019	SMA Budi Mulia Bogor
2013 – 2016	SMP Negeri 7 Bogor
2007 – 2013	SD Mardi Yuana Bogor

**INFORMAL EDUCATION****PERSONAL CERTIFICATION****ORGANIZATION EXPERIENCE**

2020 - 2022	Member of Cyber Security Community (CSC)
-------------	--

**WORKING EXPERIENCE**

Feb 2022 – Feb 2023	Internship ITSEC Asia
---------------------	-----------------------