



KERJA PRAKTIK - EF234603

Rancang Bangun Website Portal Berita Menggunakan Framework Laravel Untuk PT. Winnicode Garuda Teknologi

PT. Winnicode Garuda Teknologi
Jl. Asia Afrika No. 158, Kb. Pisang, Kec. Sumur Bandung, Kota Bandung, Jawa Barat 40261
Periode: Agustus 2024 – November 2024

Oleh:

David Ralphwaldo Martuaraja

05111940000190

Pembimbing Jurusan

Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D.

Pembimbing Lapangan

Muhamad Widyantoro, S. Kom

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2024



KERJA PRAKTIK - EF234603

Rancang Bangun Website Portal Berita Menggunakan Framework Laravel Untuk PT. Winnicode Garuda Teknologi

PT. Winnicode Garuda Teknologi
Jl. Asia Afrika No. 158, Kb. Pisang, Kec. Sumur Bandung, Kota Bandung, Jawa Barat 40261
Periode: Agustus 2024 – November 2020

Oleh:

David Ralphwaldo Martuaraja

0511194000190

Pembimbing Jurusan

Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D.

Pembimbing Lapangan

Muhamad Widyantoro, S. Kom.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2024

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

DAFTAR ISI	iv
DAFTAR GAMBAR	xx
DAFTAR TABEL	xxix
DAFTAR KODE SUMBER	xxxiii
LEMBAR PENGESAHAN	xxxv
KATA PENGANTAR	xxxix
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan	2
1.3. Manfaat	2
1.4. Rumusan Masalah	2
1.5. Lokasi dan Waktu Kerja Praktik	2
1.6. Metodologi Kerja Praktik	3
1.6.1. Perumusan Masalah	3
1.6.2. Studi Literatur	3
1.6.3. Analisis dan Perancangan Sistem	3
1.6.4. Implementasi Sistem	3
1.6.5. Uji Coba dan Evaluasi	3
1.6.6. Kesimpulan dan Saran	4
1.7. Sistematika Laporan	4
1.7.1. Bab I Pendahuluan	4

1.7.2.	Bab II Profil Perusahaan	4
1.7.3.	Bab III Tinjauan Pustaka	4
1.7.4.	Bab IV Analisis dan Perancangan Sistem	4
1.7.5.	Bab V Implementasi Sistem	4
1.7.6.	Bab VI Pengujian dan Evaluasi	4
1.7.7.	Bab VII Kesimpulan dan Saran	5
	BAB II PROFIL PERUSAHAAN	7
2.1.	Profil PT. Winnicode Garuda Teknologi	7
2.2.	Lokasi	7
	BAB III TINJAUAN PUSTAKA	10
3.1.	Pemrograman Web	10
3.2.	HTML	10
3.3.	Javascript	11
3.4.	PHP	11
3.5.	Laravel	11
3.6.	Laragon	12
3.7.	Web Server Apache	12
3.8.	MySQL	12
3.9.	Website Portal Berita	13
	BAB IV ANALISIS DAN PERANCANGAN SISTEM	15
4.1.	Analisis Sistem	15
4.1.1.	Definisi Umum Aplikasi	15
4.1.2.	Analisis Kebutuhan Fungsional	15

4.2.	Diagram Kasus Penggunaan	18
4.3.	Spesifikasi Kasus Penggunaan	21
4.3.1.	Menampilkan <i>Homepage</i>	21
4.3.2.	Melakukan Pencarian	22
4.3.3.	Melakukan <i>Login</i>	23
4.3.4.	Melakukan Registrasi	24
4.3.5.	Melakukan <i>Logout</i>	25
4.3.6.	Melakukan <i>Reset Password</i>	26
4.3.7.	Menampilkan <i>Dashboard</i>	27
4.3.8.	Memperbarui Data Akun	29
4.3.9.	Menghapus Akun	30
4.3.10.	Menampilkan Berita	31
4.3.11.	Memberikan atau Menghapus <i>Vote</i> pada Berita	32
4.3.12.	Menampilkan <i>Bookmarks</i>	33
4.3.13.	Menambahkan atau Menghapus Berita dari <i>Bookmarks</i>	34
4.3.14.	Melihat Balasan Komentar	36
4.3.15.	Menambahkan Komentar pada Berita	37
4.3.16.	Membalas Komentar	38
4.3.17.	Mengedit Komentar	39
4.3.18.	Menghapus Komentar Milik Sendiri	41
4.3.19.	Memberikan atau Menghapus <i>Vote</i> pada Komentar	42

4.3.20.	Melaporkan <i>User</i>	44
4.3.21.	Melaporkan Berita	45
4.3.22.	Melaporkan Komentar	46
4.3.23.	Menampilkan Daftar Laporan <i>User</i>	47
4.3.24.	Melihat Sebuah Laporan <i>User</i>	48
4.3.25.	Melihat Semua Laporan untuk Satu <i>User</i>	49
4.3.26.	Menghapus Sebuah Laporan <i>User</i>	51
4.3.27.	Menampilkan Daftar Laporan Berita	52
4.3.28.	Melihat Sebuah Laporan Berita	53
4.3.29.	Melihat Semua Laporan untuk Satu Berita	54
4.3.30.	Menghapus Sebuah Laporan Berita	55
4.3.31.	Menampilkan Daftar Laporan Komentar	56
4.3.32.	Melihat Sebuah Laporan Komentar	57
4.3.33.	Melihat Semua Laporan untuk Satu Komentar	58
4.3.34.	Menghapus Sebuah Laporan Komentar	60
4.3.35.	Menampilkan Daftar <i>User</i>	61
4.3.36.	Membuat <i>User</i> Baru	62
4.3.37.	Menampilkan Informasi Sebuah <i>User</i>	63
4.3.38.	Memperbarui Data Sebuah <i>User</i>	64
4.3.39.	Menampilkan Daftar Komentar untuk Sebuah Berita	65
4.3.40.	Menampilkan Sebuah Komentar	66

4.3.41.	Menampilkan Daftar Balasan untuk Sebuah Komentar	67
4.3.42.	Menghapus Komentar <i>User Lain</i>	69
4.3.43.	Menampilkan Daftar Berita	70
4.3.44.	Membuat Berita Baru	71
4.3.45.	Mengedit Sebuah Berita	72
4.3.46.	Menghapus Sebuah Berita	73
4.3.47.	Mengupload <i>File</i> untuk Sebuah Berita	75
4.3.48.	Menghapus <i>File</i> yang Diupload untuk Sebuah Berita	76
4.4.	Diagram Aktivitas	78
4.4.1.	Menampilkan <i>Homepage</i>	78
4.4.2.	Melakukan Pencarian	79
4.4.3.	Melakukan <i>Login</i>	79
4.4.4.	Melakukan Registrasi	80
4.4.5.	Melakukan <i>Logout</i>	81
4.4.6.	Melakukan <i>Reset Password</i>	82
4.4.7.	Menampilkan <i>Dashboard</i>	84
4.4.8.	Memperbarui Data Akun	84
4.4.9.	Menghapus Akun	85
4.4.10.	Menampilkan Berita	86
4.4.11.	Memberikan atau Menghapus <i>Vote</i> pada Berita	87
4.4.12.	Menampilkan <i>Bookmarks</i>	88

4.4.13.	Menambahkan atau Menghapus Berita dari <i>Bookmarks</i>	89
4.4.14.	Melihat Balasan Komentar	89
4.4.15.	Menambahkan Komentar pada Berita	90
4.4.16.	Membalas Komentar	91
4.4.17.	Mengedit Komentar	92
4.4.18.	Menghapus Komentar Milik Sendiri	93
4.4.19.	Memberikan atau Menghapus <i>Vote</i> pada Komentar	94
4.4.20.	Melaporkan <i>User</i>	94
4.4.21.	Melaporkan Berita	95
4.4.22.	Melaporkan Komentar	96
4.4.23.	Menampilkan Daftar Laporan <i>User</i>	97
4.4.24.	Melihat Sebuah Laporan <i>User</i>	98
4.4.25.	Melihat Semua Laporan untuk Satu <i>User</i>	99
4.4.26.	Menghapus Sebuah Laporan <i>User</i>	99
4.4.27.	Menampilkan Daftar Laporan Berita	100
4.4.28.	Melihat Sebuah Laporan Berita	101
4.4.29.	Melihat Semua Laporan untuk Satu Berita	102
4.4.30.	Menghapus Sebuah Laporan Berita	103
4.4.31.	Menampilkan Daftar Laporan Komentar	103
4.4.32.	Melihat Sebuah Laporan Komentar	104
4.4.33.	Melihat Semua Laporan untuk Satu Komentar	105

4.4.34.	Menghapus Sebuah Laporan Komentar	106
4.4.35.	Menampilkan Daftar <i>User</i>	107
4.4.36.	Membuat <i>User</i> Baru	108
4.4.37.	Menampilkan Informasi Sebuah <i>User</i>	109
4.4.38.	Memperbarui Data Sebuah <i>User</i>	110
4.4.39.	Menampilkan Daftar Komentar untuk Sebuah Berita	111
4.4.40.	Menampilkan Sebuah Komentar	112
4.4.41.	Menampilkan Daftar Balasan untuk Sebuah Komentar	113
4.4.42.	Menghapus Komentar <i>User</i> Lain	114
4.4.43.	Menampilkan Daftar Berita	115
4.4.44.	Membuat Berita Baru	116
4.4.45.	Mengedit Sebuah Berita	117
4.4.46.	Menghapus Sebuah Berita	118
4.4.47.	Mengupload <i>File</i> untuk Sebuah Berita	119
4.4.48.	Menghapus <i>File</i> yang Diupload untuk Sebuah Berita	120
4.5.	Desain Basis Data	121
4.5.1.	Desain Konseptual Data	121
4.5.2.	Desain Fisik Data	122
4.5.3.	Struktur Tabel <i>Users</i>	123
4.5.4.	Struktur Tabel <i>News</i>	124
4.5.5.	Struktur Tabel <i>Uploaded_Files</i>	125

4.5.6.	Struktur Tabel <i>Comments</i>	126
4.5.7.	Struktur Tabel <i>Votes</i>	127
4.5.8.	Struktur Tabel <i>Bookmarks</i>	128
4.5.9.	Struktur Tabel <i>Reports</i>	128
BAB V IMPLEMENTASI SISTEM		131
5.1.	Implementasi <i>Controller</i>	131
5.1.1	RegisteredUserController	131
5.1.2	AuthenticatedSessionController	132
5.1.3	PasswordResetLinkController	133
5.1.4	NewPasswordController	134
5.1.5	DashboardController	136
5.1.6	NewsController	139
5.1.7	CommentController	144
5.1.8	ReportController	148
5.1.9	CommentManagementController	151
5.1.10	NewsManagementController	153
5.1.11	UserManagementController	158
5.1.12	ReportManagementController	162
5.2.	Implementasi <i>User Interface</i>	166
5.2.1	Halaman <i>Login</i>	166
5.2.2	Halaman Registrasi	167
5.2.3	Halaman <i>Forgot Password</i>	168
5.2.4	Halaman Mengubah <i>Password</i>	168

5.2.5	Halaman <i>Homepage</i>	169
5.2.6	Halaman Pencarian Berita	170
5.2.7	Halaman <i>Bookmarks</i>	170
5.2.8	Halaman Berita	171
5.2.9	Halaman Balasan – Balasan Komentar	172
5.2.10	Halaman <i>Dashboard</i>	173
5.2.11	Halaman Mengedit Komentar	173
5.2.12	Halaman Melaporkan <i>User</i>	174
5.2.13	Halaman Melaporkan Berita	175
5.2.14	Halaman Melaporkan Komentar	175
5.2.15	Halaman Daftar Komentar dari Sebuah Berita	176
5.2.16	Halaman Daftar Balasan dari Sebuah Komentar	177
5.2.17	Halaman Menampilkan Sebuah Komentar	178
5.2.18	Halaman Daftar <i>User</i>	179
5.2.19	Halaman Membuat <i>User</i> Baru	180
5.2.20	Halaman Memperbarui Data <i>User</i>	181
5.2.21	Halaman Daftar Laporan <i>User</i>	182
5.2.22	Halaman Daftar Laporan Berita	183
5.2.23	Halaman Daftar Laporan Komentar	184
5.2.24	Halaman Menampilkan Laporan <i>User</i>	185
5.2.25	Halaman Menampilkan Laporan Berita	186
5.2.26	Halaman Menampilkan Laporan Komentar	186

5.2.27	Halaman Daftar Berita	187
5.2.28	Halaman Membuat Berita Baru	188
5.2.29	Halaman Memperbarui Sebuah Berita	189
BAB VI PENGUJIAN DAN EVALUASI		192
6.1.	Skenario Pengujian	192
6.1.1.	Menampilkan <i>Homepage</i>	192
6.1.2.	Melakukan Pencarian	192
6.1.3.	Melakukan <i>Login</i>	193
6.1.4.	Melakukan <i>Registrasi</i>	193
6.1.5.	Melakukan <i>Logout</i>	194
6.1.6.	Melakukan <i>Reset Password</i>	194
6.1.7.	Menampilkan <i>Dashboard</i>	195
6.1.8.	Memperbarui Data Akun	195
6.1.9.	Menghapus Akun	196
6.1.10.	Menampilkan Berita	196
6.1.11.	Memberikan atau Menghapus <i>Vote</i> pada Berita	197
6.1.12.	Menampilkan <i>Bookmarks</i>	197
6.1.13.	Menambahkan atau Menghapus Berita dari <i>Bookmarks</i>	198
6.1.14.	Melihat Balasan Komentar	198
6.1.15.	Menambahkan Komentar pada Berita	199
6.1.16.	Membalas Komentar	199
6.1.17.	Mengedit Komentar	200

6.1.18.	Menghapus Komentar	200
6.1.19.	Memberikan atau Menghapus <i>Vote</i> pada Komentar	201
6.1.20.	Melaporkan <i>User</i>	201
6.1.21.	Melaporkan Berita	202
6.1.22.	Melaporkan Komentar	202
6.1.23.	Menampilkan Daftar Laporan <i>User</i>	203
6.1.24.	Melihat Sebuah Laporan <i>User</i>	203
6.1.25.	Melihat Semua Laporan untuk Satu <i>User</i>	204
6.1.26.	Menghapus Sebuah Laporan <i>User</i>	204
6.1.27.	Menampilkan Daftar Laporan Berita	205
6.1.28.	Melihat Sebuah Laporan Berita	205
6.1.29.	Melihat Semua Laporan untuk Satu Berita	205
6.1.30.	Menghapus Sebuah Laporan Berita	206
6.1.31.	Menampilkan Daftar Laporan Komentar	206
6.1.32.	Melihat Sebuah Laporan Komentar	207
6.1.33.	Melihat Semua Laporan untuk Satu Komentar	207
6.1.34.	Menghapus Sebuah Laporan Komentar	208
6.1.35.	Menampilkan Daftar <i>User</i>	208
6.1.36.	Membuat <i>User</i> Baru	209
6.1.37.	Menampilkan Informasi Sebuah <i>User</i>	209
6.1.38.	Memperbarui Data Sebuah <i>User</i>	210

6.1.39.	Menampilkan Daftar Komentar untuk Sebuah Berita	210
6.1.40.	Menampilkan Sebuah Komentar	211
6.1.41.	Menampilkan Daftar Balasan untuk Sebuah Komentar	211
6.1.42.	Menghapus Komentar <i>User Lain</i>	212
6.1.43.	Menampilkan Daftar Berita	212
6.1.44.	Membuat Berita Baru	213
6.1.45.	Mengedit Sebuah Berita	213
6.1.46.	Menghapus Sebuah Berita	214
6.1.47.	Mengupload <i>File</i> untuk Sebuah Berita	214
6.1.48.	Menghapus <i>File</i> yang Diupload untuk Sebuah Berita	215
6.2.	Evaluasi Pengujian	215
6.2.1.	Menampilkan <i>Homepage</i>	227
6.2.2.	Melakukan Pencarian	228
6.2.3.	Melakukan <i>Login</i>	229
6.2.4.	Melakukan Registrasi	231
6.2.5.	Melakukan <i>Logout</i>	232
6.2.6.	Melakukan <i>Reset Password</i>	232
6.2.7.	Menampilkan <i>Dashboard</i>	235
6.2.8.	Memperbarui Data Akun	235
6.2.9.	Menghapus Akun	236
6.2.10.	Menampilkan Berita	237

6.2.11.	Memberikan atau Menghapus <i>Vote</i> pada Berita	238
6.2.12.	Menampilkan <i>Bookmarks</i>	239
6.2.13.	Menambahkan atau Menghapus Berita dari <i>Bookmarks</i>	239
6.2.14.	Melihat Balasan Komentar	241
6.2.15.	Menambahkan Komentar pada Berita	241
6.2.16.	Membalas Komentar	243
6.2.17.	Mengedit Komentar	244
6.2.18.	Menghapus Komentar	245
6.2.19.	Memberikan atau Menghapus <i>Vote</i> pada Komentar	245
6.2.20.	Melaporkan <i>User</i>	247
6.2.21.	Melaporkan Berita	248
6.2.22.	Melaporkan Komentar	249
6.2.23.	Menampilkan Daftar Laporan <i>User</i>	250
6.2.24.	Melihat Sebuah Laporan <i>User</i>	250
6.2.25.	Melihat Semua Laporan untuk Satu <i>User</i>	251
6.2.26.	Menghapus Sebuah Laporan <i>User</i>	252
6.2.27.	Menampilkan Daftar Laporan Berita	252
6.2.28.	Melihat Sebuah Laporan Berita	253
6.2.29.	Melihat Semua Laporan untuk Satu Berita	254
6.2.30.	Menghapus Sebuah Laporan Berita	254
6.2.31.	Menampilkan Daftar Laporan Komentar	255

6.2.32.	Melihat Sebuah Laporan Komentar	256
6.2.33.	Melihat Semua Laporan untuk Satu Komentar	256
6.2.34.	Menghapus Sebuah Laporan Komentar	257
6.2.35.	Menampilkan Daftar <i>User</i>	258
6.2.36.	Membuat <i>User</i> Baru	258
6.2.37.	Menampilkan Informasi Sebuah <i>User</i>	260
6.2.38.	Memperbarui Data Sebuah <i>User</i>	261
6.2.39.	Menampilkan Daftar Komentar Sebuah Berita	261
6.2.40.	Menampilkan Sebuah Komentar	262
6.2.41.	Menampilkan Daftar Balasan untuk Sebuah Komentar	263
6.2.42.	Menghapus Komentar <i>User</i> Lain	263
6.2.43.	Menampilkan Daftar Berita	264
6.2.44.	Membuat Berita Baru	265
6.2.45.	Mengedit Sebuah Berita	265
6.2.46.	Menghapus Sebuah Berita	266
6.2.47.	Mengupload <i>File</i> untuk Sebuah Berita	267
6.2.48.	Menghapus <i>File</i> yang Diupload untuk Sebuah Berita	268
BAB VII KESIMPULAN DAN SARAN		270
7.1.	Kesimpulan	270
7.2.	Saran	270

DAFTAR PUSTAKA	272
BIODATA PENULIS	275

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 4.1 Diagram Kasus Penggunaan	20
Gambar 4.2 Diagram Aktivitas Menampilkan Homepage	78
Gambar 4.3 Diagram Aktivitas Melakukan Pencarian	79
Gambar 4.4 Diagram Aktivitas Melakukan Login	80
Gambar 4.5 Diagram Aktivitas Melakukan Registrasi.....	81
Gambar 4.6 Diagram Aktivitas Melakukan Logout	82
Gambar 4.7 Diagram Aktivitas Melakukan Reset Password	83
Gambar 4.8 Diagram Aktivitas Menampilkan Dashboard	84
Gambar 4.9 Diagram Aktivitas Memperbarui Data Akun	85
Gambar 4.10 Diagram Aktivitas Menghapus Akun	86
Gambar 4.11 Diagram Aktivitas Menampilkan Berita.....	87
Gambar 4.12 Diagram Aktivitas Memberikan atau Menghapus Vote pada Berita.....	88
Gambar 4.13 Diagram Aktivitas Menampilkan Bookmarks	88
Gambar 4.14 Diagram Aktivitas Menambahkan atau Menghapus Berita dari Bookmarks.....	89
Gambar 4.15 Diagram Aktivitas Melihat Balasan Komentar.....	90
Gambar 4.16 Diagram Aktivitas Menambahkan Komentar pada Berita	91
Gambar 4.17 Diagram Aktivitas Membalas Komentar	92
Gambar 4.18 Diagram Aktivitas Mengedit Komentar	93
Gambar 4.19 Diagram Aktivitas Menghapus Komentar Milik Sendiri	93
Gambar 4.20 Diagram Aktivitas Memberikan atau Menghapus Vote pada Komentar.....	94
Gambar 4.21 Diagram Aktivitas Melaporkan User.....	95
Gambar 4.22 Diagram Aktivitas Melaporkan Berita.....	96
Gambar 4.23 Diagram Aktivitas Melaporkan Komentar	97

Gambar 4.24 Diagram Aktivitas Melihat Daftar Laporan User ..	98
Gambar 4.25 Diagram Aktivitas Melihat Sebuah Laporan User.	98
Gambar 4.26 Diagram Aktivitas Melihat Semua Laporan untuk Satu User	99
Gambar 4.27 Diagram Aktivitas Menghapus Sebuah Laporan User	100
Gambar 4.28 Diagram Aktivitas Menampilkan Daftar Laporan Berita	101
Gambar 4.29 Diagram Aktivitas Melihat Sebuah Laporan Berita	101
Gambar 4.30 Diagram Aktivitas Melihat Semua Laporan untuk Satu Berita	102
Gambar 4.31 Diagram Aktivitas Menghapus Sebuah Laporan Berita	103
Gambar 4.32 Diagram Aktivitas Menampilkan Daftar Laporan Komentar	104
Gambar 4.33 Diagram Aktivitas Melihat Sebuah Laporan Komentar	105
Gambar 4.34 Diagram Aktivitas Melihat Semua Laporan untuk Satu Komentar	106
Gambar 4.35 Diagram Aktivitas Menghapus Sebuah Laporan Komentar	107
Gambar 4.36 Diagram Aktivitas Menampilkan Daftar User.....	108
Gambar 4.37 Diagram Aktivitas Membuat User Baru	109
Gambar 4.38 Diagram Aktivitas Menampilkan Informasi Sebuah User	110
Gambar 4.39 Diagram Aktivitas Memperbarui Data Sebuah User	111
Gambar 4.40 Diagram Aktivitas Menampilkan Daftar Komentar untuk Sebuah Berita	112

Gambar 4.41 Diagram Aktivitas Menampilkan Sebuah Komentar	113
Gambar 4.42 Diagram Aktivitas Menampilkan Daftar Balasan untuk Sebuah Komentar	114
Gambar 4.43 Diagram Aktivitas Menghapus Komentar User Lain	115
Gambar 4.44 Diagram Aktivitas Menampilkan Daftar Berita ..	116
Gambar 4.45 Diagram Aktivitas Membuat Berita Baru.....	117
Gambar 4.46 Diagram Aktivitas Mengedit Sebuah Berita.....	118
Gambar 4.47 Diagram Aktivitas Menghapus Sebuah Berita	119
Gambar 4.48 Diagram Aktivitas Mengupload File untuk Sebuah Berita	120
Gambar 4.49 Diagram Aktivitas Menghapus File yang Diupload untuk Sebuah Berita	121
Gambar 4.50 Desain Konseptual Data Website Portal Berita ...	122
Gambar 4.51 Desain Fisik Data Website Portal Berita	123
Gambar 5.1 Implementasi Antarmuka Halaman Login.....	167
Gambar 5.2 Implementasi Antarmuka Halaman Registrasi	167
Gambar 5.3 Implementasi Antarmuka Halaman Forgot Password	168
Gambar 5.4 Implementasi Antarmuka Halaman Mengubah Password.....	169
Gambar 5.5 Implementasi Antarmuka Halaman Homepage	169
Gambar 5.6 Implementasi Antarmuka Halaman Pencarian Berita	170
Gambar 5.7 Implementasi Antarmuka Halaman Bookmarks....	171
Gambar 5.8 Implementasi Antarmuka Halaman Berita	172
Gambar 5.9 Implementasi Antarmuka Halaman Balasan - Balasan Komentar	172
Gambar 5.10 Implementasi Antarmuka Halaman Dashboard...	173

Gambar 5.11 Implementasi Antarmuka Halaman Mengedit Komentar	174
Gambar 5.12 Implementasi Antarmuka Halaman Melaporkan User	174
Gambar 5.13 Implementasi Antarmuka Halaman Melaporkan Berita	175
Gambar 5.14 Implementasi Antarmuka Halaman Melaporkan Komentar	176
Gambar 5.15 Implementasi Antarmuka Halaman Daftar Komentar dari Sebuah Berita	177
Gambar 5.16 Implementasi Antarmuka Halaman Daftar Balasan dari Sebuah Komentar	178
Gambar 5.17 Implementasi Antarmuka Halaman Menampilkan Sebuah Komentar	179
Gambar 5.18 Implementasi Antarmuka Halaman Daftar User .	180
Gambar 5.19 Implementasi Antarmuka Halaman Membuat User Baru	181
Gambar 5.20 Implementasi Antarmuka Halaman Memperbarui Data User	182
Gambar 5.21 Implementasi Antarmuka Halaman Daftar Laporan User	183
Gambar 5.22 Implementasi Antarmuka Halaman Daftar Laporan Berita	184
Gambar 5.23 Implementasi Antarmuka Halaman Daftar Laporan Komentar	185
Gambar 5.24 Implementasi Antarmuka Halaman Menampilkan Laporan User	185
Gambar 5.25 Implementasi Antarmuka Halaman Menampilkan Laporan Berita.....	186
Gambar 5.26 Implementasi Antarmuka Halaman Menampilkan Laporan Komentar.....	187

Gambar 5.27 Implementasi Antarmuka Halaman Daftar Berita	188
Gambar 5.28 Implementasi Antarmuka Halaman Membuat Berita Baru	189
Gambar 5.29 Implementasi Antarmuka Halaman Memperbarui Sebuah Berita	190
Gambar 6.1 Hasil Pengujian Menampilkan Homepage	228
Gambar 6.2 Hasil Pengujian Melakukan Pencarian	228
Gambar 6.3 Hasil Pengujian Melakukan Login Gagal karena Email dan Password Tidak Sesuai.....	229
Gambar 6.4 Hasil Pengujian Melakukan Login Gagal karena Akun Diblokir.....	230
Gambar 6.5 Hasil Pengujian Melakukan Login yang Berhasil .	230
Gambar 6.6 Hasil Pengujian Melakukan Registrasi yang Gagal	231
Gambar 6.7 Hasil Pengujian Melakukan Registrasi yang Berhasil	231
Gambar 6.8 Hasil Pengujian Melakukan Logout	232
Gambar 6.9 Hasil Pengujian Melakukan Reset Password Gagal karena Email Tidak Ditemukan.....	233
Gambar 6.10 Hasil Pengujian Melakukan Reset Password Email Ditemukan	233
Gambar 6.11 Hasil Pengujian Melakukan Reset Password Gagal karena Password Tidak Sama.....	234
Gambar 6.12 Hasil Pengujian Melakukan Reset Password Berhasil.....	234
Gambar 6.13 Hasil Pengujian Menampilkan Dashboard	235
Gambar 6.14 Hasil Pengujian Memperbarui Data Akun Gagal	236
Gambar 6.15 Hasil Pengujian Memperbarui Data Akun Berhasil	236
Gambar 6.16 Hasil Pengujian Menghapus Akun	237
Gambar 6.17 Hasil Pengujian Menampilkan Berita.....	237

Gambar 6.18 Hasil Pengujian Memberikan atau Menghapus Vote pada Berita 1	238
Gambar 6.19 Hasil Pengujian Memberikan atau Menghapus Vote pada Berita 2.....	238
Gambar 6.20 Hasil Pengujian Menampilkan Bookmarks	239
Gambar 6.21 Hasil Pengujian Menambahkan atau Menghapus Berita dari Bookmarks 1.....	240
Gambar 6.22 Hasil Pengujian Menambahkan atau Menghapus Berita dari Bookmarks 2.....	240
Gambar 6.23 Hasil Pengujian Melihat Balasan Komentar	241
Gambar 6.24 Hasil Pengujian Menambahkan Komentar pada Berita Gagal.....	242
Gambar 6.25 Hasil Pengujian Menambahkan Komentar pada Berita Berhasil	242
Gambar 6.26 Hasil Pengujian Membalas Komentar Gagal.....	243
Gambar 6.27 Hasil Pengujian Membalas Komentar Berhasil	243
Gambar 6.28 Hasil Pengujian Mengedit Komentar Gagal	244
Gambar 6.29 Hasil Pengujian Mengedit Komentar Berhasil	244
Gambar 6.30 Hasil Pengujian Menghapus Komentar	245
Gambar 6.31 Hasil Pengujian Memberikan atau Menghapus Vote pada Komentar 1	246
Gambar 6.32 Hasil Pengujian Memberikan atau Menghapus Vote pada Komentar 2	246
Gambar 6.33 Hasil Pengujian Melaporkan User Gagal	247
Gambar 6.34 Hasil Pengujian Melaporkan User Berhasil.....	247
Gambar 6.35 Hasil Pengujian Melaporkan Berita Gagal	248
Gambar 6.36 Hasil Pengujian Melaporkan Berita Berhasil	248
Gambar 6.37 Hasil Pengujian Melaporkan Komentar Gagal	249
Gambar 6.38 Hasil Pengujian Melaporkan Komentar Berhasil	249
Gambar 6.39 Hasil Pengujian Menampilkan Daftar Laporan User	250

Gambar 6.40 Hasil Pengujian Melihat Sebuah Laporan User...	251
Gambar 6.41 Hasil Pengujian Melihat Semua Laporan untuk Satu User	251
Gambar 6.42 Hasil Pengujian Menghapus Sebuah Laporan User	252
Gambar 6.43 Hasil Pengujian Menampilkan Daftar Laporan Berita	253
Gambar 6.44 Hasil Pengujian Melihat Sebuah Laporan Berita.	253
Gambar 6.45 Hasil Pengujian Melihat Semua Laporan untuk Satu Berita	254
Gambar 6.46 Hasil Pengujian Menghapus Sebuah Laporan Berita	255
Gambar 6.47 Hasil Pengujian Menampilkan Daftar Laporan Komentar	255
Gambar 6.48 Hasil Pengujian Melihat Sebuah Laporan Komentar	256
Gambar 6.49 Hasil Pengujian Melihat Semua Laporan untuk Satu Komentar	257
Gambar 6.50 Hasil Pengujian Menghapus Sebuah Laporan Komentar	257
Gambar 6.51 Hasil Pengujian Menampilkan Daftar User	258
Gambar 6.52 Hasil Pengujian Membuat User Baru Gagal karena Email Sudah Digunakan	259
Gambar 6.53 Hasil Pengujian Membuat User Baru Gagal karena Data Tidak Valid	259
Gambar 6.54 Hasil Pengujian Membuat User Baru Berhasil....	260
Gambar 6.55 Hasil Pengujian Menampilkan Informasi Sebuah User	260
Gambar 6.56 Hasil Pengujian Memperbarui Data Sebuah User	261
Gambar 6.57 Hasil Pengujian Menampilkan Daftar Komentar Sebuah Berita	262

Gambar 6.58 Hasil Pengujian Menampilkan Sebuah Komentar	262
Gambar 6.59 Hasil Pengujian Melihat Daftar Balasan untuk Sebuah Komentar	263
Gambar 6.60 Hasil Pengujian Menghapus Komentar User Lain	264
Gambar 6.61 Hasil Pengujian Menampilkan Daftar Berita.....	264
Gambar 6.62 Hasil Pengujian Membuat Berita Baru.....	265
Gambar 6.63 Hasil Pengujian Mengedit Sebuah Berita Gagal .	266
Gambar 6.64 Hasil Pengujian Mengedit Sebuah Berita Berhasil	266
Gambar 6.65 Hasil Pengujian Menghapus Sebuah Berita.....	267
Gambar 6.66 Hasil Pengujian Mengupload File untuk Sebuah Berita	267
Gambar 6.67 Hasil Pengujian Menghapus File yang Diupload untuk Sebuah Berita	268

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 4.1 Daftar Kebutuhan Fungsional	16
Tabel 4.2 Spesifikasi Kasus Penggunaan Menampilkan Homepage.....	21
Tabel 4.3 Spesifikasi Kasus Penggunaan Melakukan Pencarian	22
Tabel 4.4 Spesifikasi Kasus Penggunaan Melakukan Login.....	23
Tabel 4.5 Spesifikasi Kasus Penggunaan Melakukan Registrasi	24
Tabel 4.6 Spesifikasi Kasus Penggunaan Melakukan Logout.....	25
Tabel 4.7 Spesifikasi Kasus Penggunaan Melakukan Reset Password.....	26
Tabel 4.8 Spesifikasi Kasus Penggunaan Menampilkan Dashboard.....	28
Tabel 4.9 Spesifikasi Kasus Penggunaan Memperbarui Data Akun	29
Tabel 4.10 Spesifikasi Kasus Penggunaan Menghapus Akun.....	30
Tabel 4.11 Spesifikasi Kasus Penggunaan Menampilkan Berita	31
Tabel 4.12 Spesifikasi Kasus Penggunaan Memberikan atau Menghapus Vote pada Berita	32
Tabel 4.13 Spesifikasi Kasus Penggunaan Menampilkan Bookmarks	33
Tabel 4.14 Spesifikasi Kasus Penggunaan Menambahkan atau Menghapus Berita dari Bookmarks	35
Tabel 4.15 Spesifikasi Kasus Penggunaan Melihat Balasan Komentar	36
Tabel 4.16 Spesifikasi Kasus Penggunaan Menambahkan Komentar pada Berita.....	37
Tabel 4.17 Spesifikasi Kasus Penggunaan Membalas Komentar	38
Tabel 4.18 Spesifikasi Kasus Penggunaan Mengedit Komentar	40
Tabel 4.19 Spesifikasi Kasus Penggunaan Menghapus Komentar Milik Sendiri.....	41

Tabel 4.20 Spesifikasi Kasus Penggunaan Memberikan atau Menghapus Vote pada Komentar	42
Tabel 4.21 Spesifikasi Kasus Penggunaan Melaporkan User	44
Tabel 4.22 Spesifikasi Kasus Penggunaan Melaporkan Berita	45
Tabel 4.23 Spesifikasi Kasus Penggunaan Melaporkan Komentar	46
Tabel 4.24 Spesifikasi Kasus Penggunaan Menampilkan Daftar Laporan User	48
Tabel 4.25 Spesifikasi Kasus Penggunaan Melihat Sebuah Laporan User	49
Tabel 4.26 Spesifikasi Kasus Penggunaan Melihat Semua Laporan untuk Satu User	50
Tabel 4.27 Spesifikasi Kasus Penggunaan Menghapus Sebuah Laporan User	51
Tabel 4.28 Spesifikasi Kasus Penggunaan Menampilkan Daftar Laporan Berita	52
Tabel 4.29 Spesifikasi Kasus Penggunaan Melihat Sebuah Laporan Berita	53
Tabel 4.30 Spesifikasi Kasus Penggunaan Melihat Semua Laporan untuk Satu Berita	54
Tabel 4.31 Spesifikasi Kasus Penggunaan Menghapus Sebuah Laporan Berita	55
Tabel 4.32 Spesifikasi Kasus Penggunaan Menampilkan Daftar Laporan Komentar	56
Tabel 4.33 Spesifikasi Kasus Penggunaan Melihat Sebuah Laporan Komentar	58
Tabel 4.34 Spesifikasi Kasus Penggunaan Melihat Semua Laporan untuk Satu Komentar	59
Tabel 4.35 Spesifikasi Kasus Penggunaan Menghapus Sebuah Laporan Komentar	60

Tabel 4.36 Spesifikasi Kasus Penggunaan Menampilkan Daftar User	61
Tabel 4.37 Spesifikasi Kasus Penggunaan Membuat User Baru.	62
Tabel 4.38 Spesifikasi Kasus Penggunaan Menampilkan Informasi Sebuah User	63
Tabel 4.39 Spesifikasi Kasus Penggunaan Memperbarui Data Sebuah User.....	64
Tabel 4.40 Spesifikasi Kasus Penggunaan Menampilkan Daftar Komentar untuk Sebuah Berita	65
Tabel 4.41 Spesifikasi Kasus Penggunaan Menampilkan Sebuah Komentar	67
Tabel 4.42 Spesifikasi Kasus Penggunaan Menampilkan Daftar Balasan untuk Sebuah Komentar.....	68
Tabel 4.43 Spesifikasi Kasus Penggunaan Menghapus Komentar User Lain	69
Tabel 4.44 Spesifikasi Kasus Penggunaan Menampilkan Daftar Berita	70
Tabel 4.45 Spesifikasi Kasus Penggunaan Membuat Berita Baru	71
Tabel 4.46 Spesifikasi Kasus Penggunaan Mengedit Sebuah Berita	72
Tabel 4.47 Spesifikasi Kasus Penggunaan Menghapus Sebuah Berita	74
Tabel 4.48 Spesifikasi Kasus Penggunaan Mengupload File untuk Sebuah Berita	75
Tabel 4.49 Spesifikasi Kasus Penggunaan Menghapus File yang Diupload untuk Sebuah Berita	77
Tabel 4.50 Struktur Tabel Users.....	123
Tabel 4.51 Struktur Tabel News.....	124
Tabel 4.52 Struktur Tabel Uploaded Files	125
Tabel 4.53 Struktur Tabel Comments	126

Tabel 4.54 Struktur Tabel Votes	127
Tabel 4.55 Struktur Tabel Bookmarks	128
Tabel 4.56 Struktur Tabel Reports	129
Tabel 6.1 Hasil Evaluasi Pengujian Fungsionalitas	216

DAFTAR KODE SUMBER

Kode Sumber 5.1 Implementasi RegisteredUserController	132
Kode Sumber 5.2 Implementasi AuthenticatedSessionController	133
Kode Sumber 5.3 Implementasi PasswordResetLinkController	134
Kode Sumber 5.4 Implementasi NewPasswordController	136
Kode Sumber 5.5 Implementasi DashboardController.....	138
Kode Sumber 5.6 Implementasi NewsController.....	143
Kode Sumber 5.7 Implementasi CommentController	148
Kode Sumber 5.8 Implementasi ReportController	151
Kode Sumber 5.9 Implementasi CommentManagementController	153
Kode Sumber 5.10 Implementasi NewsManagementController	158
Kode Sumber 5.11 Implementasi UserManagementController	162
Kode Sumber 5.12 Implementasi ReportManagementController	166

[Halaman ini sengaja dikosongkan]

**LEMBAR PENGESAHAN
KERJA PRAKTIK**

**Rancang Bangun Website Portal Berita Menggunakan
Framework Laravel Untuk PT. Winnicode Garuda
Teknologi**

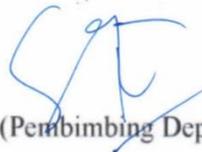
Oleh:

David Ralphwaldo Martuaraja

05111940000190

Disetujui oleh Pembimbing Kerja Praktik:

1. Shintami Chusnul Hidayati,
S. Kom., M.Sc., Ph. D.
NIP. 198106202005011003



(Pembimbing Departemen)

2. Muhamad Widyantoro,
S. Kom.



(Pembimbing Lapangan)

[Halaman ini sengaja dikosongkan]

Rancang Bangun Website Portal Berita Menggunakan Framework Laravel Untuk PT. Winnicode Garuda Teknologi

Nama Mahasiswa : David Ralphwaldo Martuaraja
NRP : 05111940000190

Departemen : Teknik Informatika FTEIC-ITS
Pembimbing Departemen : Shintami Chusnul Hidayati, S. Kom.,
M. Sc., Ph. D.
Pembimbing Lapangan : Muhamad Widyantoro, S. Kom.

ABSTRAK

PT. Winnicode Garuda Teknologi adalah perusahaan yang bergerak di bidang media publikasi dan layanan digital. Produk yang saya kerjakan selama kegiatan kerja praktik di PT. Winnicode Garuda Teknologi adalah website portal berita. Website portal berita ini dibuat dengan tujuan untuk menyediakan platform yang menyediakan artikel-artikel berita yang dapat diakses dengan mudah oleh penggunanya, dan juga dapat digunakan untuk mengelola artikel-artikel berita yang ada.. Website portal berita ini dibuat menggunakan framework Laravel. Beberapa fitur yang diimplementasikan pada website portal berita ini adalah fitur pencarian berita, pengelolaan artikel-artikel berita, dan pengelolaan user. Pembuatan website dilakukan menggunakan development environment Laragon, yang dilengkapi dengan web server apache, framework Laravel, dan database MySQL.

Kata Kunci : Website, Portal Berita, Laravel

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas penyertaan dan karuniaNya sehingga penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Rancang Bangun Website Portal Berita Menggunakan Framework Laravel Untuk PT. Winnicode Garuda Teknologi.

Penulis menyadari bahwa masih banyak kekurangan baik dalam melaksanakan kerja praktik maupun penyusunan buku laporan kerja praktik ini. Namun penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang tua penulis.
2. Ibu Shintami Chusnul Hidayati S.Kom., M.Sc., Ph.D. selaku dosen pembimbing kerja praktik.
3. Bapak Muhamad Widyantoro, S.Kom., selaku pembimbing lapangan selama kerja praktik berlangsung.
4. Teman-teman penulis yang senantiasa memberikan semangat ketika penulis melaksanakan KP.

Surabaya, 15 November 2024
David Ralphwaldo Martuaraja

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi memiliki dampak yang sangat besar pada penyebaran informasi. Penyebaran informasi kini dapat dilakukan secara *online*, yang memungkinkan informasi tersebut untuk diakses secara mudah oleh siapapun. Salah satu cara untuk menyediakan layanan dan informasi secara *online* adalah dengan menggunakan sebuah *website*. *Website* adalah sebuah aplikasi yang berisi layanan dan informasi yang disediakan secara *online* yang dapat diakses menggunakan sebuah *web browser*. *Website* sendiri banyak jenisnya, seperti *website* mesin pencarian, pembelajaran, bank, dan lain-lain. Salah satu jenis *website* yang cukup populer adalah *website* portal berita. *Website* portal berita adalah sebuah *website* yang dapat digunakan untuk menyediakan dan mengelola artikel-artikel berita yang dapat diakses oleh penggunanya.

PT. Winnicode Garuda Teknologi adalah sebuah perusahaan yang bergerak di bidang media publikasi dan layanan digital. Untuk saat ini, PT. Winnicode Garuda Teknologi sedang membutuhkan sebuah *website* portal berita. *Website* portal berita ini dibuat dengan tujuan untuk menyediakan *platform* yang dapat digunakan oleh penggunanya untuk mendapatkan informasi dan berita terkini dan untuk melakukan pengelolaan artikel-artikel berita yang ada dengan mudah. Pada saat kerja praktik, saya mendapat kesempatan untuk merancang dan membangun aplikasi *website* portal berita tersebut.

1.2. Tujuan

Kegiatan kerja praktik ini dilakukan dengan tujuan menyelesaikan kewajiban nilai kerja praktik sebesar 4 sks dan untuk membantu PT. Winnicode Garuda Teknologi dalam pembuatan aplikasi *website* portal berita menggunakan *framework* Laravel.

1.3. Manfaat

Manfaat yang diperoleh dari adanya *website* portal berita adalah adanya *website* yang dapat digunakan oleh masyarakat sebagai sumber berita dan informasi yang terkini. Selain itu, *website* portal berita ini juga dapat digunakan untuk mengelola artikel-artikel berita dengan mudah.

1.4. Rumusan Masalah

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana proses rancang bangun aplikasi *website* portal berita menggunakan framework Laravel yang memenuhi kebutuhan pengguna?
2. Bagaimana proses rancang bangun fitur-fitur dari aplikasi *website* portal berita agar dapat mempermudah proses pengelolaan artikel-artikel berita?

1.5. Lokasi dan Waktu Kerja Praktik

Kegiatan kerja praktik dilakukan secara *remote*, dan pelaksanaan kerja praktik berlangsung dari bulan Agustus 2024 sampai dengan bulan November 2024.

1.6. Metodologi Kerja Praktik

Metodologi dalam pembuatan buku kerja praktik meliputi :

1.6.1. Perumusan Masalah

Untuk mengetahui kebutuhan dari *website* yang dibuat, kami bertemu dengan mentor dan client kami untuk mendiskusikan fitur-fitur dan kebutuhan yang diperlukan oleh *website* portal berita. Kami diberikan penjelasan dari tiap fitur dan kebutuhan yang diperlukan oleh *website*.

1.6.2. Studi Literatur

Setelah diberikan kebutuhan dan gambaran dari *website* portal berita yang akan dibuat, kami diberitahu tinjauan apa saja yang akan diimplementasikan dalam pengembangan *website*. Tinjauan yang dipakai meliputi framework Laravel, web server apache, database MySQL, dan lain-lain.

1.6.3. Analisis dan Perancangan Sistem

Pada tahap ini dilakukan perancangan sistem sesuai dengan fitur-fitur dan kebutuhan yang telah dijelaskan dan tinjauan yang diberikan.

1.6.4. Implementasi Sistem

Implementasi merupakan realisasi dari tahap analisis dan perancangan sistem. Pada tahap ini dilakukan pembangunan *website* yang sudah dirancang.

1.6.5. Uji Coba dan Evaluasi

Uji coba dan evaluasi dilakukan untuk memastikan bahwa *website* yang telah dibangun memenuhi kebutuhan yang sudah direncanakan.

1.6.6. Kesimpulan dan Saran

Pengujian yang dilakukan ini telah memenuhi syarat yang diinginkan, dan berjalan dengan baik dan lancar.

1.7. Sistematika Laporan

Laporan kerja praktik ini terdiri dari tujuh bab dengan rincian sebagai berikut:

1.7.1. Bab I Pendahuluan

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

1.7.2. Bab II Profil Perusahaan

Bab ini berisi profil perusahaan dan lokasi dari PT. Winnicode Garuda Teknologi.

1.7.3. Bab III Tinjauan Pustaka

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

1.7.4. Bab IV Analisis dan Perancangan Sistem

Bab ini berisi mengenai tahap analisis sistem aplikasi dalam menyelesaikan proyek kerja praktik.

1.7.5. Bab V Implementasi Sistem

Bab ini berisi uraian tahap-tahap yang dilakukan untuk proses implementasi aplikasi.

1.7.6. Bab VI Pengujian dan Evaluasi

Bab ini berisi hasil uji coba dan evaluasi dari aplikasi yang telah dikembangkan selama pelaksanaan kerja praktik.

1.7.7. Bab VII Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

[Halaman ini sengaja dikosongkan]

BAB II

PROFIL PERUSAHAAN

2.1. Profil PT. Winnicode Garuda Teknologi

PT. Winnicode Garuda Teknologi adalah perusahaan yang bergerak di bidang media publikasi dan layanan digital, yang didirikan pada tahun 2020 di Bandung. PT. Winnicode Garuda Teknologi memiliki fokus utama pada publikasi berita, artikel, jurnalistik, dan menyediakan layanan digital. PT. Winnicode Garuda Teknologi mengkhususkan diri dalam pembuatan dan distribusi konten-konten berkualitas seperti berita terkini, artikel mendalam, dan tulisan jurnalistik yang dirancang untuk memberikan informasi yang akurat, terpercaya, dan relevan kepada masyarakat.

2.2. Visi dan Misi Perusahaan

PT. Winnicode Garuda Teknologi memiliki visi untuk menjadi pemimpin dalam industri media publikasi dan layanan digital di Indonesia, dengan fokus pada inovasi, kualitas, dan integritas dalam setiap proyek yang dikerjakannya. PT. Winnicode Garuda Teknologi memiliki misi sebagai berikut:

1. Menyediakan layanan media publikasi yang kredibel dan berdampak, termasuk berita, artikel, dan konten jurnalistik yang berkualitas.
2. Mengembangkan solusi digital yang efisien dan efektif untuk mendukung transformasi bisnis ke arah digitalisasi.

3. Membangun kemitraan yang berkelanjutan dengan klien melalui layanan yang unggul dan kepuasan pelanggan.
4. Menginspirasi dan mendorong perkembangan teknologi di Indonesia melalui inovasi dan keahlian.

2.3. Lokasi

Jl. Asia Afrika No. 158, Kb. Pisang, Kec. Sumur Bandung, Bandung, Jawa Barat 40261

[Halaman ini sengaja dikosongkan]

BAB III

TINJAUAN PUSTAKA

Bab ini membahas tentang dasar teori yang digunakan selama proses pengerjaan dan implementasi dari aplikasi *website* portal berita.

3.1. Pemrograman Web

Pemrograman *web* adalah proses pengembangan aplikasi yang dijalankan pada sebuah *server* di *internet* yang dapat dioperasikan melalui sebuah *web browser*. Pemrograman *web* biasanya dibagi menjadi dua, yaitu pemrograman *front-end* dan pemrograman *back-end*. Pemrograman *front-end* bertujuan untuk membangun tampilan dan halaman-halaman *website* yang interaktif dan biasanya dilakukan menggunakan HTML, CSS, dan bahasa pemrograman JavaScript. Pemrograman *back-end* bertujuan untuk membangun fitur-fitur dari *website*, mengelola interaksi dengan *database*, dan mengelola data yang biasanya dilakukan menggunakan bahasa pemrograman seperti PHP, Javascript, Java, dan Python.

3.2. HTML

Hypertext Markup Language atau HTML adalah sebuah bahasa *markup* yang digunakan dalam sebuah halaman *website*. HTML digunakan untuk menyusun struktur dan isi dari sebuah halaman *website*. HTML biasanya digabungkan dengan CSS untuk membuat tampilannya lebih menarik, dan dengan Javascript untuk membuatnya interaktif.

3.3. Javascript

Javascript adalah sebuah bahasa pemrograman tingkat tinggi yang dinamis. Javascript merupakan salah satu teknologi inti yang digunakan dalam pemrograman *web*, selain HTML dan CSS. Javascript biasanya digunakan bersama dengan HTML dan CSS untuk menghasilkan halaman *website* yang interaktif.

3.4. PHP

PHP: Hypertext Processor atau PHP adalah sebuah bahasa pemrograman yang umumnya digunakan dalam pengembangan aplikasi *web*. Kode PHP dijalankan oleh *PHP Interpreter* yang berjalan pada sebuah *web server*. Hasil dari eksekusi kode PHP dapat berupa tipe data apapun, seperti kode HTML atau bahkan data gambar, yang nantinya akan dikembalikan sebagai bagian dari *HTTP response*. PHP memiliki banyak *web framework* yang dapat memudahkan proses pengembangan aplikasi *web* seperti Laravel, CodeIgniter, Symfony, dan Phalcon.

3.5. Laravel

Laravel adalah sebuah *web framework* berbasis bahasa pemrograman PHP yang bersifat *open-source*. Laravel menggunakan arsitektur Model-View-Controller atau MVC untuk pengembangan aplikasinya. Laravel menyediakan fitur-fitur yang dapat membuat proses pengembangan *website* lebih mudah dan efisien seperti fitur *authentication*, *authorization*, *database migrations*, *Eloquent ORM*, dan lain-lain.

3.6. Laragon

Laragon merupakan sebuah *universal development environment* yang dapat digunakan dengan bahasa pemrograman seperti PHP, Node.js, Python, Java, Go, dan Ruby. Laragon bersifat *portable*, ringan, dan mudah digunakan. Instalasi dari Laragon sudah termasuk bahasa pemrograman PHP, *web server* apache, dan *database* MySQL.

3.7. Web Server Apache

Web server digunakan untuk menerima *request* yang dikirim dari browser pengguna *website*, memproses *request* tersebut, dan memberikan *response* kepada pengguna. Salah satu *web server* yang cukup populer adalah *Apache web server*. *Apache web server* adalah perangkat lunak *web server* yang bersifat *open-source*. *Apache* dapat digunakan dengan bahasa-bahasa pemrograman seperti PHP, Python, Tcl, dan Perl. *Apache* dilengkapi dengan fitur-fitur seperti pesan-pesan kesalahan yang dapat dikonfigurasi, *database authentication*, *compression*, *virtual hosting*, dan lain-lain.

3.8. MySQL

MySQL merupakan sebuah sistem manajemen *database* relasional yang bersifat *open-source*. MySQL mengorganisir data menjadi satu atau beberapa tabel data yang dapat berelasi dengan tabel-tabel lainnya. Operasi-operasi untuk membuat, memodifikasi, dan mengambil data dari tabel-tabel yang ada dilakukan menggunakan *query*. MySQL akan menjalankan *query* yang diterimanya

dan melakukan operasi pada *database* sesuai dengan perintah yang ada di *query* tersebut.

3.9. Website Portal Berita

Website portal berita adalah aplikasi *web* yang berfungsi sebagai *platform* untuk mengelola dan menyediakan berita dan informasi terkini yang relevan kepada penggunanya. *Website* portal biasanya mengelompokkan artikel-artikel ke dalam berbagai jenis kategori agar lebih mudah digunakan oleh penggunanya. Selain itu, *website* portal berita biasanya juga dilengkapi fitur-fitur yang dapat memudahkan penggunanya. Beberapa contoh *website* portal berita yang cukup populer adalah Detik.com, Kompas.com, dan Tribunnews.com.

[Halaman ini sengaja dikosongkan]

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahapan analisis dan perancangan sistem dari aplikasi *website* portal berita yang meliputi analisis sistem, diagram kasus penggunaan, spesifikasi kasus penggunaan, diagram aktivitas, dan desain basis data.

4.1. Analisis Sistem

Subbab ini membahas tahapan analisis sistem dari aplikasi *website* portal berita yang dibagi menjadi dua, yaitu definisi umum aplikasi dan analisis kebutuhan fungsional.

4.1.1. Definisi Umum Aplikasi

Secara umum, aplikasi *website* portal berita yang dibuat merupakan sistem yang dapat digunakan oleh penggunanya untuk membaca artikel-artikel berita dan untuk melakukan pengelolaan artikel berita. *Website* portal berita yang dibuat juga akan memiliki fitur-fitur lain, seperti untuk melakukan pengelolaan *user*, pengelolaan komentar, pengelolaan laporan, dan lain-lain. Pada sistem ini terdapat tiga jenis *user*, yaitu *user* biasa, *editor*, dan *administrator*, yang masing-masing memiliki hak akses yang berbeda.

4.1.2. Analisis Kebutuhan Fungsional

Dari diskusi dengan pembimbing lapangan dan *client*, kami mendapatkan daftar kebutuhan fungsional yang diperlukan oleh *website* portal berita. Kebutuhan fungsional dari *website* portal berita dapat dilihat pada Tabel 4.1.

Tabel 4.1 Daftar Kebutuhan Fungsional

Kode Kebutuhan	Deskripsi Kebutuhan
KF-001	Menampilkan <i>homepage</i>
KF-002	Melakukan pencarian
KF-003	Melakukan <i>login</i>
KF-004	Melakukan registrasi
KF-005	Melakukan <i>logout</i>
KF-006	Melakukan <i>reset password</i>
KF-007	Menampilkan dashboard
KF-008	Memperbarui data akun
KF-009	Menghapus akun
KF-010	Menampilkan berita
KF-011	Memberikan atau menghapus <i>vote</i> pada berita
KF-012	Menampilkan <i>bookmarks</i>
KF-013	Menambahkan atau menghapus berita dari <i>bookmarks</i>
KF-014	Melihat balasan komentar
KF-015	Menambahkan komentar pada berita
KF-016	Membalas komentar
KF-017	Mengedit komentar
KF-018	Menghapus komentar
KF-019	Memberikan atau menghapus <i>vote</i> pada komentar
KF-020	Melaporkan <i>user</i>
KF-021	Melaporkan berita
KF-022	Melaporkan komentar
KF-023	Menampilkan daftar laporan <i>user</i>
KF-024	Melihat sebuah laporan <i>user</i>
KF-025	Melihat semua laporan untuk satu <i>user</i>

KF-026	Menghapus sebuah laporan <i>user</i>
KF-027	Menampilkan daftar laporan berita
KF-028	Melihat sebuah laporan berita
KF-029	Melihat semua laporan untuk satu berita
KF-030	Menghapus sebuah laporan berita
KF-031	Menampilkan daftar laporan komentar
KF-032	Melihat sebuah laporan komentar
KF-033	Melihat semua laporan untuk satu komentar
KF-034	Menghapus sebuah laporan komentar
KF-035	Menampilkan daftar <i>user</i>
KF-036	Membuat <i>user</i> baru
KF-037	Menampilkan informasi sebuah <i>user</i>
KF-038	Memperbarui data sebuah <i>user</i>
KF-039	Menampilkan daftar komentar sebuah berita
KF-040	Menampilkan sebuah komentar
KF-041	Menampilkan daftar balasan untuk sebuah komentar
KF-042	Menghapus sebuah komentar
KF-043	Menampilkan daftar berita
KF-044	Membuat berita baru
KF-045	Mengedit sebuah berita
KF-046	Menghapus sebuah berita
KF-047	Mengupload <i>file</i> untuk sebuah berita

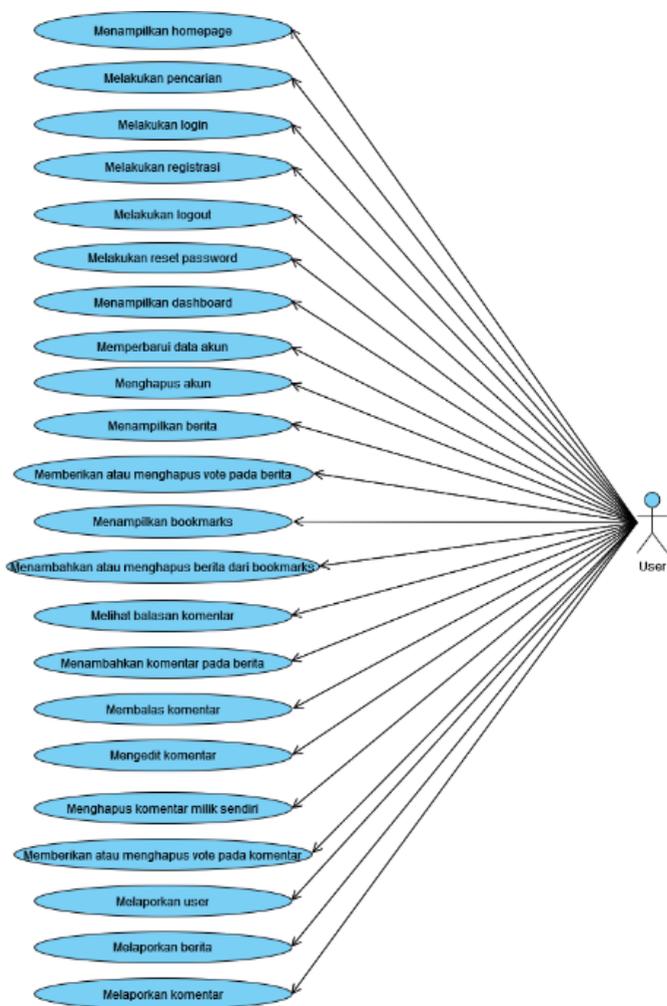
KF-048	Menghapus <i>file</i> yang diupload untuk sebuah berita
--------	---

4.2. Diagram Kasus Penggunaan

Daftar kebutuhan fungsional dari *website* portal berita dapat dibuatkan sebuah diagram kasus penggunaan atau *use case diagram* agar lebih mudah dipahami. *Use case diagram* dari *website* portal berita dapat dilihat pada



Gambar 4.1.





Gambar 4.1 Diagram Kasus Penggunaan

4.3. Spesifikasi Kasus Penggunaan

Subbab ini menjelaskan spesifikasi kasus penggunaan dari setiap kebutuhan fungsional aplikasi *website* portal berita.

4.3.1. Menampilkan *Homepage*

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menampilkan *homepage*. Spesifikasi kasus penggunaan untuk kebutuhan menampilkan *homepage* dapat dilihat pada Tabel 4.2.

Tabel 4.2 Spesifikasi Kasus Penggunaan Menampilkan Homepage

Nama	Menampilkan <i>homepage</i>
Kode	UC001
Deskripsi	Ketika seorang pengguna mengunjungi <i>homepage website</i> , pengguna dapat melihat artikel-artikel berita yang terkini.
Tipe	Fungsional
Pemicu	Pengguna mengunjungi halaman <i>homepage website</i>
Aktor	Pengguna
Kondisi Awal	Pengguna belum berada di halaman <i>homepage</i> .
Kondisi Akhir	Pengguna dapat membuka halaman <i>homepage</i> dari <i>website</i> .
Alur Kejadian Normal	1. Aktor mengetikkan URL dari <i>website</i> pada <i>address bar</i> dari <i>web browser</i> yang digunakan dan mengunjungi URL tersebut. 2. Halaman <i>homepage</i> ditampilkan kepada aktor.

Alur Kejadian Alternatif	1. Aktor mengklik <i>link</i> menuju <i>homepage</i> yang ada di <i>navigation bar</i> . 2. Halaman <i>homepage</i> ditampilkan kepada aktor.
Pengecualian	-

4.3.2. Melakukan Pencarian

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah melakukan pencarian. Spesifikasi kasus penggunaan untuk kebutuhan melakukan pencarian dapat dilihat pada Tabel 4.3.

Tabel 4.3 Spesifikasi Kasus Penggunaan Melakukan Pencarian

Nama	Melakukan pencarian
Kode	UC002
Deskripsi	Pengguna dapat menggunakan fitur pencarian untuk melakukan pencarian untuk berita dengan judul yang memiliki kata kunci tertentu
Tipe	Fungsional
Pemicu	Pengguna menggunakan fitur pencarian
Aktor	Pengguna
Kondisi Awal	Pengguna belum berada di halaman hasil pencarian
Kondisi Akhir	Pengguna berada di halaman hasil pencarian yang berisi artikel-artikel berita dengan judul yang memiliki kata kunci yang digunakan.

Alur Kejadian Normal	1. Aktor mengetikkan kata kunci untuk berita yang ingin dicari pada <i>input</i> pencarian yang ada di <i>navigation bar</i> dan mengklik tombol untuk melakukan pencarian. 2. Halaman hasil pencarian yang berisi artikel-artikel berita dengan judul yang memiliki kata kunci yang digunakan ditampilkan kepada aktor.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.3. Melakukan *Login*

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah melakukan *login*. Spesifikasi kasus penggunaan untuk kebutuhan melakukan *login* dapat dilihat pada Tabel 4.4.

Tabel 4.4 Spesifikasi Kasus Penggunaan Melakukan Login

Nama	Melakukan <i>login</i>
Kode	UC003
Deskripsi	Pengguna melakukan <i>login</i> agar mendapatkan akses ke fitur-fitur yang tersedia hanya untuk pengguna yang sudah melakukan <i>login</i> .
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>login</i>
Aktor	Pengguna
Kondisi Awal	Pengguna belum melakukan <i>login</i>

Kondisi Akhir	Pengguna berhasil melakukan <i>login</i> dan masuk ke akunnya.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor mengklik tombol <i>login</i> yang berada di <i>navigation bar</i>. 2. Aktor memasukkan <i>username</i> dan <i>password</i> untuk akun miliknya ke dalam <i>form login</i> dan mengklik tombol <i>login</i>. 3. Aktor berhasil melakukan <i>login</i> ke dalam sistem dan diberikan sesi dengan hak akses sesuai dengan tipe akunnya.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.4. Melakukan Registrasi

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah melakukan registrasi. Spesifikasi kasus penggunaan untuk kebutuhan melakukan registrasi dapat dilihat pada Tabel 4.5.

Tabel 4.5 Spesifikasi Kasus Penggunaan Melakukan Registrasi

Nama	Melakukan registrasi
Kode	UC004
Deskripsi	Pengguna yang tidak memiliki akun dapat melakukan registrasi untuk membuat akun baru.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>register</i>
Aktor	Pengguna

Kondisi Awal	Pengguna belum mempunyai akun.
Kondisi Akhir	Pengguna mempunyai akun untuk masuk ke dalam sistem.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor mengklik tombol <i>login</i> yang ada di <i>navigation bar</i>. 2. Aktor mengklik <i>link</i> menuju halaman registrasi. 3. Aktor memasukkan informasi untuk akun yang akan dibuat dan mengklik tombol <i>register</i>. 4. Aktor dibuatkan akun baru oleh sistem yang dapat digunakan untuk <i>login</i>.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.5. Melakukan *Logout*

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah melakukan *logout*. Spesifikasi kasus penggunaan untuk kebutuhan melakukan *logout* dapat dilihat pada Tabel 4.6.

Tabel 4.6 Spesifikasi Kasus Penggunaan Melakukan Logout

Nama	Melakukan <i>logout</i>
Kode	UC005
Deskripsi	Pengguna dapat melakukan <i>logout</i> untuk keluar dari akunnya dan mengakhiri sesi.
Tipe	Fungsional

Pemicu	Pengguna mengklik tombol <i>logout</i>
Aktor	Pengguna
Kondisi Awal	Pengguna masih masuk di dalam sistem menggunakan akunnya dan memiliki sesi.
Kondisi Akhir	Pengguna keluar dari akunnya dan mengakhiri sesi.
Alur Kejadian Normal	1. Aktor mengklik tombol <i>logout</i> yang ada di <i>navigation bar</i> . 2. Aktor berhasil keluar dari akunnya dan mengakhiri sesi.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.6. Melakukan *Reset Password*

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah melakukan *reset password*. Spesifikasi kasus penggunaan untuk kebutuhan melakukan *reset password* dapat dilihat pada Tabel 4.7.

Tabel 4.7 Spesifikasi Kasus Penggunaan Melakukan Reset Password

Nama	Melakukan <i>reset password</i>
Kode	UC006
Deskripsi	Pengguna dapat melakukan <i>reset password</i> jika lupa <i>password</i> akun miliknya.
Tipe	Fungsional

Pemicu	Pengguna mengklik tombol <i>forgot password</i> pada halaman <i>login</i>
Aktor	Pengguna
Kondisi Awal	Pengguna lupa <i>password</i> yang digunakan untuk akunnya dan tidak bisa melakukan <i>login</i> .
Kondisi Akhir	Pengguna berhasil melakukan <i>reset password</i> untuk akunnya dan bisa melakukan <i>login</i> kembali.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor mengklik tombol <i>login</i> di <i>navigation bar</i>. 2. Aktor mengklik tombol <i>forgot password</i>. 3. Aktor memasukkan <i>email</i> yang digunakan untuk akunnya dan mengklik tombol <i>send password reset link</i>. 4. Aktor menerima <i>email</i> yang berisi <i>link</i> untuk melakukan <i>reset password</i>. 5. Aktor menggunakan <i>link</i> yang dikirimkan oleh sistem untuk membuat <i>password</i> baru untuk akunnya. 6. Aktor berhasil mengganti <i>password</i> akunnya dan bisa melakukan <i>login</i>.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.7. Menampilkan *Dashboard*

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menampilkan *dashboard*. Spesifikasi kasus penggunaan untuk kebutuhan menampilkan *dashboard* dapat dilihat pada Tabel 4.8.

Tabel 4.8 Spesifikasi Kasus Penggunaan Menampilkan Dashboard

Nama	Menampilkan <i>dashboard</i>
Kode	UC007
Deskripsi	Pengguna yang telah melakukan <i>login</i> dapat menampilkan <i>dashboard</i> .
Tipe	Fungsional
Pemicu	Pengguna melakukan <i>login</i> atau mengklik <i>link</i> menuju <i>dashboard</i> yang ada di <i>navigation bar</i> .
Aktor	Pengguna
Kondisi Awal	Pengguna tidak berada di halaman <i>dashboard</i> .
Kondisi Akhir	Pengguna berada di halaman <i>dashboard</i> .
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna mengklik tombol <i>login</i> yang ada di <i>navigation bar</i>. 2. Pengguna mengisi <i>username</i> dan <i>password</i> dari akun miliknya dan melakukan <i>login</i>. 3. Halaman <i>dashboard</i> ditampilkan kepada pengguna.
Alur Kejadian Alternatif	<ol style="list-style-type: none"> 1. Aktor yang sebelumnya telah melakukan <i>login</i> mengklik <i>link</i> menuju <i>dashboard</i> yang ada di <i>navigation bar</i>. 2. Halaman <i>dashboard</i> ditampilkan kepada aktor.

Pengecualian	-
--------------	---

4.3.8. Memperbarui Data Akun

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah memperbarui data akun. Spesifikasi kasus penggunaan untuk kebutuhan memperbarui data akun dapat dilihat pada Tabel 4.9.

Tabel 4.9 Spesifikasi Kasus Penggunaan Memperbarui Data Akun

Nama	Memperbarui data akun
Kode	UC008
Deskripsi	Pengguna dapat memperbarui data akun untuk mengubah nama atau <i>password</i> miliknya.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>update account</i> .
Aktor	Pengguna
Kondisi Awal	Data akun milik pengguna belum berubah.
Kondisi Akhir	Data akun milik pengguna berhasil diubah menjadi data baru yang dimasukkan oleh pengguna.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor melakukan <i>login</i>. 2. Aktor membuka halaman <i>dashboard</i>. 3. Aktor memasukkan data baru untuk akunnya dan mengklik tombol <i>update account</i> untuk memperbarui data akunnya.
Alur Kejadian Alternatif	-

Pengecualian	-
--------------	---

4.3.9. Menghapus Akun

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menghapus akun. Spesifikasi kasus penggunaan untuk kebutuhan menghapus akun dapat dilihat pada Tabel 4.10.

Tabel 4.10 Spesifikasi Kasus Penggunaan Menghapus Akun

Nama	Menghapus akun
Kode	UC009
Deskripsi	Pengguna dapat menghapus akun miliknya.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>delete account</i> .
Aktor	Pengguna
Kondisi Awal	Akun pengguna masih tersimpan di dalam sistem.
Kondisi Akhir	Akun pengguna dihapuskan dari sistem.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor melakukan <i>login</i>. 2. Aktor membuka halaman <i>dashboard</i>. 3. Aktor mengetikkan <i>password</i> akunya pada <i>form delete account</i> dan mengklik tombol <i>delete account</i>. 4. Akun milik aktor dihapuskan dari sistem.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.10. Menampilkan Berita

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menampilkan berita. Spesifikasi kasus penggunaan untuk kebutuhan menampilkan berita dapat dilihat pada Tabel 4.11.

Tabel 4.11 Spesifikasi Kasus Penggunaan Menampilkan Berita

Nama	Menampilkan berita
Kode	UC010
Deskripsi	Pengguna dapat menampilkan berita.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>view</i> pada sebuah <i>item</i> berita.
Aktor	Pengguna
Kondisi Awal	Pengguna tidak berada di halaman berita.
Kondisi Akhir	Pengguna berada di halaman berita dan dapat membaca artikel berita dan melihat komentar-komentar yang ada.
Alur Kejadian Normal	1. Aktor mengklik tombol <i>view news</i> untuk sebuah <i>item</i> berita yang dapat dilakukan di halaman <i>homepage</i> atau halaman hasil pencarian berita. 2. Halaman yang berisi artikel berita dan komentar-komentar ditampilkan kepada actor.
Alur Kejadian Alternatif	-

Pengecualian	-
--------------	---

4.3.11. Memberikan atau Menghapus *Vote* pada Berita

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah memberikan atau menghapus *vote* pada berita. Spesifikasi kasus penggunaan untuk kebutuhan memberikan atau menghapus *vote* pada berita dapat dilihat pada Tabel 4.12.

Tabel 4.12 Spesifikasi Kasus Penggunaan Memberikan atau Menghapus *Vote* pada Berita

Nama	Memberikan atau menghapus <i>vote</i> pada berita
Kode	UC011
Deskripsi	Pengguna dapat memberikan atau menghapus <i>vote</i> pada berita.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>thumbs up</i> atau <i>thumbs down</i> pada sebuah artikel berita.
Aktor	Pengguna
Kondisi Awal	Pengguna belum memberikan <i>vote</i> atau sudah memberikan <i>vote</i> pada sebuah berita.
Kondisi Akhir	Jika pengguna belum memberikan <i>vote</i> , maka <i>vote</i> akan dibuatkan untuk berita tersebut. Jika pengguna sudah memiliki <i>vote</i> dan memberikan <i>vote</i> dengan jenis yang berbeda, maka jenis <i>vote</i> dari pengguna yang sekarang akan diperbarui. Jika pengguna

	sudah memiliki <i>vote</i> dan memberikan yang sama, maka <i>vote</i> milik pengguna akan dihapuskan.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor membuka halaman berita dari sebuah artikel berita. 2. Aktor mengklik tombol <i>thumbs up</i> atau <i>thumbs down</i>. 3. Jika aktor belum memiliki <i>vote</i> untuk berita tersebut, maka <i>vote</i> akan dibuatkan. Jika aktor memberikan <i>vote</i> yang sama dengan <i>vote</i> yang sebelumnya telah diberikan, maka <i>vote</i> dihapuskan. Jika aktor memberikan <i>vote</i> yang berbeda dengan <i>vote</i> yang sebelumnya telah diberikan, maka jenis <i>vote</i> akan diperbarui.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.12. Menampilkan *Bookmarks*

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menampilkan *bookmarks*. Spesifikasi kasus penggunaan untuk kebutuhan menampilkan *bookmarks* dapat dilihat pada Tabel 4.13.

Tabel 4.13 Spesifikasi Kasus Penggunaan Menampilkan *Bookmarks*

Nama	Menampilkan <i>bookmarks</i>
Kode	UC012

Deskripsi	Pengguna dapat menampilkan artikel-artikel berita yang sudah ditambahkan ke <i>bookmarks</i> .
Tipe	Fungsional
Pemicu	Pengguna mengklik <i>link</i> ke halaman <i>bookmarks</i> yang ada di <i>navigation bar</i> .
Aktor	Pengguna
Kondisi Awal	Pengguna belum berada di halaman <i>bookmarks</i> .
Kondisi Akhir	Pengguna berada di halaman <i>bookmarks</i> dan dapat melihat artikel-artikel berita yang sudah ditambahkan ke <i>bookmarks</i> .
Alur Kejadian Normal	1. Aktor mengklik <i>link</i> menuju halaman <i>bookmarks</i> yang ada di <i>navigation bar</i> . 2. Halaman <i>bookmarks</i> ditampilkan kepada aktor.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.13. Menambahkan atau Menghapus Berita dari *Bookmarks*

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menambahkan berita ke *bookmarks*. Spesifikasi kasus penggunaan untuk kebutuhan menambahkan berita ke *bookmarks* dapat dilihat pada Tabel 4.14.

Tabel 4.14 Spesifikasi Kasus Penggunaan Menambahkan atau Menghapus Berita dari Bookmarks

Nama	Menambahkan atau menghapus berita dari <i>bookmarks</i>
Kode	UC013
Deskripsi	Pengguna dapat menambahkan atau menghapus berita dari <i>bookmarks</i> .
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>add to bookmarks</i> atau <i>remove from bookmarks</i> pada sebuah artikel berita.
Aktor	Pengguna
Kondisi Awal	Berita yang belum ditambahkan ke <i>bookmarks</i> tidak disimpan di <i>bookmarks</i> dan berita yang sudah ditambahkan ke <i>bookmarks</i> disimpan di <i>bookmarks</i> .
Kondisi Akhir	Berita yang ditambahkan ke <i>bookmarks</i> disimpan di <i>bookmarks</i> dan berita yang dihapus dari <i>bookmarks</i> tidak lagi disimpan di <i>bookmarks</i> .
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor sudah berada di <i>homepage</i> atau halaman hasil pencarian artikel berita. 2. Aktor mengklik tombol <i>add to bookmarks</i> pada berita yang ingin ditambahkan ke <i>bookmarks</i>, atau tombol <i>remove from bookmarks</i> pada berita yang ingin dihapus dari <i>bookmarks</i>.

	3. Berita ditambahkan ke <i>bookmarks</i> atau dihapus dari <i>bookmarks</i> .
Alur Kejadian Alternatif	1. Aktor membuka halaman berita dari sebuah artikel berita. 2. Aktor mengklik tombol <i>add to bookmarks</i> pada berita yang ingin ditambahkan ke <i>bookmarks</i> atau tombol <i>remove from bookmarks</i> pada berita yang ingin dihapus dari <i>bookmarks</i> . 3. Berita ditambahkan ke <i>bookmarks</i> atau dihapus dari <i>bookmarks</i> .
Pengecualian	-

4.3.14. Melihat Balasan Komentar

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah melihat balasan komentar. Spesifikasi kasus penggunaan untuk kebutuhan melihat balasan komentar dapat dilihat pada Tabel 4.15.

Tabel 4.15 Spesifikasi Kasus Penggunaan Melihat Balasan Komentar

Nama	Melihat balasan komentar
Kode	UC014
Deskripsi	Pengguna dapat melihat balasan komentar.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>view replies</i> .
Aktor	Pengguna

Kondisi Awal	Pengguna tidak berada di halaman yang berisi balasan-balasan untuk sebuah komentar pada sebuah berita.
Kondisi Akhir	Pengguna berada di halaman yang berisi balasan-balasan untuk sebuah komentar pada sebuah berita.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor membuka halaman berita dari sebuah artikel berita. 2. Aktor mengklik tombol <i>view replies</i> pada komentar yang ingin ditampilkan balasan-balasannya. 3. Aktor dibawa ke halaman yang berisi balasan-balasan untuk komentar yang dipilih.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.15. Menambahkan Komentar pada Berita

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menambahkan komentar pada berita. Spesifikasi kasus penggunaan untuk kebutuhan menambahkan komentar pada berita dapat dilihat pada Tabel 4.16.

Tabel 4.16 Spesifikasi Kasus Penggunaan Menambahkan Komentar pada Berita

Nama	Menambahkan komentar pada berita
Kode	UC015

Deskripsi	Pengguna dapat menambahkan komentar pada berita.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>post comment</i> .
Aktor	Pengguna
Kondisi Awal	Komentar yang ingin ditambahkan pengguna tidak disimpan di dalam sistem.
Kondisi Akhir	Komentar yang dibuat oleh pengguna disimpan di dalam sistem.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor membuka halaman berita dari sebuah artikel berita. 2. Aktor menuliskan isi komentar pada <i>field</i> isi komentar, kemudian mengklik tombol <i>post comment</i>. 3. Komentar untuk berita dibuatkan dan disimpan di dalam sistem.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.16. Membalas Komentar

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah membalas komentar. Spesifikasi kasus penggunaan untuk kebutuhan membalas komentar dapat dilihat pada Tabel 4.17.

Tabel 4.17 Spesifikasi Kasus Penggunaan Membalas Komentar

Nama	Membalas komentar
------	-------------------

Kode	UC016
Deskripsi	Pengguna dapat membalas komentar.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>reply</i> .
Aktor	Pengguna
Kondisi Awal	Balasan untuk komentar yang ingin ditambahkan pengguna tidak disimpan di dalam sistem.
Kondisi Akhir	Balasan untuk komentar yang dibuat oleh pengguna disimpan di dalam sistem.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor membuka halaman berita dari sebuah artikel berita. 2. Aktor mengklik tombol <i>reply</i> pada komentar yang ingin dibalas. 3. Aktor menuliskan isi balasan pada <i>field</i> isi balasan, kemudian mengklik tombol <i>post reply</i>. 4. Balasan untuk komentar dibuatkan dan disimpan di dalam sistem.
Alur Kejadian Alternatif	<ol style="list-style-type: none"> 1. Aktor membuka halaman balasan dari sebuah komentar. 2. Aktor menuliskan isi balasan pada <i>field</i> isi balasan, kemudian mengklik tombol <i>post reply</i>. 3. Balasan untuk komentar dibuatkan dan disimpan di dalam sistem.
Pengecualian	-

4.3.17. Mengedit Komentar

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah mengedit komentar. Spesifikasi kasus penggunaan untuk kebutuhan mengedit komentar dapat dilihat pada Tabel 4.18.

Tabel 4.18 Spesifikasi Kasus Penggunaan Mengedit Komentar

Nama	Mengedit komentar
Kode	UC017
Deskripsi	Pengguna dapat mengedit komentar yang sebelumnya telah dibuat olehnya.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>edit</i> pada komentar yang sebelumnya telah dibuat olehnya.
Aktor	Pengguna
Kondisi Awal	Komentar yang sebelumnya dibuat oleh pengguna isinya masih sama.
Kondisi Akhir	Komentar yang sebelumnya dibuat oleh pengguna isinya diperbarui dan perubahannya disimpan di dalam sistem.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor membuka halaman berita dari sebuah artikel berita atau halaman balasan dari sebuah komentar. 2. Aktor mengklik tombol <i>edit</i> pada komentar atau balasan yang sebelumnya telah dibuat olehnya. 3. Aktor dibawa ke halaman untuk memperbarui isi komentar atau balasan miliknya.

	4. Aktor memperbarui isi dari komentar atau balasan miliknya, kemudian mengklik tombol <i>save</i> . 5. Perubahan untuk komentar atau balasan disimpan di dalam sistem.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.18. Menghapus Komentar Milik Sendiri

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menghapus komentar milik sendiri. Spesifikasi kasus penggunaan untuk kebutuhan menghapus komentar milik sendiri dapat dilihat pada Tabel 4.19.

Tabel 4.19 Spesifikasi Kasus Penggunaan Menghapus Komentar Milik Sendiri

Nama	Menghapus komentar milik sendiri
Kode	UC018
Deskripsi	Pengguna dapat menghapus komentar miliknya.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>delete</i> pada komentar yang sebelumnya dibuat olehnya.
Aktor	Pengguna
Kondisi Awal	Komentar milik pengguna masih disimpan di dalam sistem.

Kondisi Akhir	Komentar milik pengguna dihapuskan dari sistem.
Alur Kejadian Normal	1. Aktor membuka halaman berita dari sebuah artikel berita atau halaman balasan dari sebuah komentar. 2. Aktor mengklik tombol <i>delete</i> pada komentar miliknya. 3. Komentar milik aktor dihapuskan dari sistem.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.19. Memberikan atau Menghapus *Vote* pada Komentar

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah memberikan atau menghapus *vote* pada komentar. Spesifikasi kasus penggunaan untuk kebutuhan memberikan atau menghapus *vote* pada komentar dapat dilihat pada Tabel 4.20.

Tabel 4.20 Spesifikasi Kasus Penggunaan Memberikan atau Menghapus *Vote* pada Komentar

Nama	Memberikan atau menghapus <i>vote</i> pada komentar
Kode	UC019
Deskripsi	Pengguna dapat memberikan atau menghapus <i>vote</i> pada komentar.
Tipe	Fungsional

Pemicu	Pengguna mengklik tombol <i>thumbs up</i> atau <i>thumbs down</i> pada sebuah komentar.
Aktor	Pengguna
Kondisi Awal	Pengguna belum memberikan <i>vote</i> atau sudah memberikan <i>vote</i> pada sebuah komentar.
Kondisi Akhir	Jika pengguna belum memberikan <i>vote</i> , maka <i>vote</i> akan dibuatkan untuk komentar tersebut. Jika pengguna sudah memiliki <i>vote</i> dan memberikan <i>vote</i> dengan jenis yang berbeda, maka jenis <i>vote</i> dari pengguna yang sekarang akan diperbarui. Jika pengguna sudah memiliki <i>vote</i> dan memberikan yang sama, maka <i>vote</i> milik pengguna akan dihapuskan.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor membuka halaman berita dari sebuah artikel berita atau halaman balasan untuk sebuah komentar. 2. Aktor mengklik tombol <i>thumbs up</i> atau <i>thumbs down</i> pada komentar yang ingin diberikan <i>vote</i>. 3. Jika aktor belum memiliki <i>vote</i> untuk komentar tersebut, maka <i>vote</i> akan dibuatkan. Jika aktor memberikan <i>vote</i> yang sama dengan <i>vote</i> yang sebelumnya telah diberikan, maka <i>vote</i> dihapuskan. Jika aktor memberikan <i>vote</i> yang berbeda

	dengan <i>vote</i> yang sebelumnya telah diberikan, maka jenis <i>vote</i> akan diperbarui.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.20. Melaporkan *User*

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah melaporkan *user*. Spesifikasi kasus penggunaan untuk kebutuhan melaporkan *user* dapat dilihat pada Tabel 4.21.

Tabel 4.21 Spesifikasi Kasus Penggunaan Melaporkan User

Nama	Melaporkan <i>user</i>
Kode	UC020
Deskripsi	Pengguna dapat melaporkan <i>user</i> .
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>report user</i> pada sebuah komentar.
Aktor	Pengguna
Kondisi Awal	Laporan milik pengguna untuk <i>user</i> yang ingin dilaporkan belum dibuatkan.
Kondisi Akhir	Laporan milik pengguna untuk <i>user</i> yang ingin dilaporkan dibuatkan dan disimpan di dalam sistem.
Alur Kejadian Normal	1. Aktor membuka halaman berita dari sebuah artikel berita

	<p>atau halaman balasan untuk sebuah komentar.</p> <p>2. Aktor mengklik tombol <i>report user</i> pada komentar yang dibuat oleh <i>user</i> yang ingin dilaporkan.</p> <p>3. Aktor mengisi alasan membuat laporan, kemudian mengklik tombol <i>report</i>.</p> <p>4. Laporan dibuatkan dan disimpan di dalam sistem.</p>
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.21. Melaporkan Berita

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah melaporkan berita. Spesifikasi kasus penggunaan untuk kebutuhan melaporkan berita dapat dilihat pada Tabel 4.22.

Tabel 4.22 Spesifikasi Kasus Penggunaan Melaporkan Berita

Nama	Melaporkan berita
Kode	UC021
Deskripsi	Pengguna dapat melaporkan berita.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>report news</i> .
Aktor	Pengguna
Kondisi Awal	Laporan milik pengguna untuk berita yang ingin dilaporkan belum dibuatkan.

Kondisi Akhir	Laporan milik pengguna untuk berita yang ingin dilaporkan dibuatkan dan disimpan di dalam sistem.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor membuka halaman berita dari sebuah artikel berita. 2. Aktor mengklik tombol <i>report news</i>. 3. Aktor mengisi alasan membuat laporan, kemudian mengklik tombol <i>report</i>. 4. Laporan dibuatkan dan disimpan di dalam sistem.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.22. Melaporkan Komentar

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah melaporkan komentar. Spesifikasi kasus penggunaan untuk kebutuhan melaporkan komentar dapat dilihat pada Tabel 4.23.

Tabel 4.23 Spesifikasi Kasus Penggunaan Melaporkan Komentar

Nama	Melaporkan komentar
Kode	UC022
Deskripsi	Pengguna dapat melaporkan komentar.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>report comment</i> .
Aktor	Pengguna

Kondisi Awal	Laporan milik pengguna untuk komentar yang ingin dilaporkan belum dibuatkan.
Kondisi Akhir	Laporan milik pengguna untuk komentar yang ingin dilaporkan dibuatkan dan disimpan di dalam sistem.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor membuka halaman berita dari sebuah artikel berita atau halaman balasan dari sebuah komentar. 2. Aktor mengklik tombol <i>report comment</i> pada komentar yang ingin dilaporkan. 3. Aktor mengisi alasan membuat laporan, kemudian mengklik tombol <i>report</i>. 4. Laporan dibuatkan dan disimpan di dalam sistem.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.23. Menampilkan Daftar Laporan *User*

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menampilkan daftar laporan *user*. Spesifikasi kasus penggunaan untuk kebutuhan menampilkan daftar laporan *user* dapat dilihat pada Tabel 4.24.

Tabel 4.24 Spesifikasi Kasus Penggunaan Menampilkan Daftar Laporan User

Nama	Menampilkan daftar laporan <i>user</i>
Kode	UC023
Deskripsi	Pengguna dapat menampilkan daftar laporan <i>user</i> .
Tipe	Fungsional
Pemicu	Pengguna mengklik <i>link</i> menuju halaman <i>user report management</i> yang ada di <i>navigation bar</i> .
Aktor	Pengguna bertipe <i>administrator</i>
Kondisi Awal	Pengguna tidak berada di halaman <i>user report management</i> yang berisi daftar laporan <i>user</i> .
Kondisi Akhir	Pengguna berada di halaman <i>user report management</i> yang berisi daftar laporan <i>user</i> .
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor mengklik <i>link</i> menuju halaman <i>user report management</i> yang ada di menu <i>administrator</i> di <i>navigation bar</i>. 2. Aktor dibawa ke halaman <i>user report management</i> yang berisi daftar laporan komentar.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.24. Melihat Sebuah Laporan *User*

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah melihat sebuah laporan *user*. Spesifikasi kasus penggunaan untuk kebutuhan melihat sebuah laporan *user* dapat dilihat pada Tabel 4.25.

Tabel 4.25 Spesifikasi Kasus Penggunaan Melihat Sebuah Laporan User

Nama	Melihat sebuah laporan <i>user</i>
Kode	UC032
Deskripsi	Pengguna dapat melihat sebuah laporan <i>user</i> .
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>view</i> pada sebuah laporan <i>user</i> .
Aktor	Pengguna bertipe <i>administrator</i> .
Kondisi Awal	Pengguna tidak berada di halaman yang berisi informasi sebuah laporan <i>user</i> .
Kondisi Akhir	Pengguna berada di halaman yang berisi informasi sebuah laporan <i>user</i> .
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor membuka halaman <i>user report management</i>. 2. Aktor mengklik tombol <i>view</i> pada laporan yang ingin dilihat. 3. Aktor dibawa ke halaman yang berisi informasi dari laporan tersebut.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.25. Melihat Semua Laporan untuk Satu *User*

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah melihat semua laporan untuk satu *user*. Spesifikasi kasus penggunaan untuk

kebutuhan melihat semua laporan untuk satu *user* dapat dilihat pada Tabel 4.26.

Tabel 4.26 Spesifikasi Kasus Penggunaan Melihat Semua Laporan untuk Satu User

Nama	Melihat semua laporan untuk satu <i>user</i>
Kode	UC025
Deskripsi	Pengguna dapat melihat semua laporan untuk satu <i>user</i> .
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>view reports</i> .
Aktor	Pengguna bertipe <i>administrator</i>
Kondisi Awal	Pengguna tidak berada di halaman yang berisi laporan-laporan untuk satu <i>user</i> .
Kondisi Akhir	Pengguna berada di halaman yang berisi laporan-laporan untuk satu <i>user</i> .
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor membuka halaman <i>user management</i>. 2. Aktor mengklik tombol <i>view</i> pada <i>user</i> yang laporannya ingin dilihat. 3. Aktor mengklik tombol <i>view reports</i>. 4. Aktor dibawa ke halaman yang berisi daftar laporan-laporan untuk <i>user</i> tersebut.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.26. Menghapus Sebuah Laporan *User*

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menghapus sebuah laporan *user*. Spesifikasi kasus penggunaan untuk kebutuhan menghapus sebuah laporan *user* dapat dilihat pada Tabel 4.27.

Tabel 4.27 Spesifikasi Kasus Penggunaan Menghapus Sebuah Laporan User

Nama	Menghapus sebuah laporan <i>user</i>
Kode	UC026
Deskripsi	Pengguna dapat menghapus sebuah laporan <i>user</i> .
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>delete</i> .
Aktor	Pengguna bertipe <i>administrator</i>
Kondisi Awal	Laporan <i>user</i> masih disimpan di dalam sistem.
Kondisi Akhir	Laporan <i>user</i> dihapuskan dari sistem.
Alur Kejadian Normal	1. Aktor membuka halaman <i>user report management</i> . 2. Aktor mengklik tombol <i>view</i> pada laporan yang ingin dihapus. 3. Aktor mengklik tombol <i>delete</i> . 4. Laporan dihapuskan dari sistem.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.27. Menampilkan Daftar Laporan Berita

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menampilkan daftar laporan berita. Spesifikasi kasus penggunaan untuk kebutuhan menampilkan daftar laporan berita dapat dilihat pada Tabel 4.28.

Tabel 4.28 Spesifikasi Kasus Penggunaan Menampilkan Daftar Laporan Berita

Nama	Menampilkan daftar laporan komentar
Kode	UC027
Deskripsi	Pengguna dapat menampilkan daftar laporan berita.
Tipe	Fungsional
Pemicu	Pengguna mengklik <i>link</i> menuju halaman <i>news report management</i> yang ada di <i>navigation bar</i> .
Aktor	Pengguna bertipe <i>administrator</i>
Kondisi Awal	Pengguna tidak berada di halaman <i>news report management</i> yang berisi daftar laporan <i>user</i> .
Kondisi Akhir	Pengguna tidak berada di halaman <i>news report management</i> yang berisi daftar laporan <i>user</i> .
Alur Kejadian Normal	1. Aktor mengklik <i>link</i> menuju halaman <i>news report management</i> yang ada di menu <i>administrator</i> di <i>navigation bar</i> .

	2. Aktor dibawa ke halaman <i>news report management</i> yang berisi daftar laporan komentar.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.28. Melihat Sebuah Laporan Berita

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah melihat sebuah laporan berita. Spesifikasi kasus penggunaan untuk kebutuhan melihat sebuah laporan berita dapat dilihat pada Tabel 4.29.

Tabel 4.29 Spesifikasi Kasus Penggunaan Melihat Sebuah Laporan Berita

Nama	Melihat sebuah laporan berita
Kode	UC028
Deskripsi	Pengguna dapat melihat sebuah laporan berita.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>view</i> pada sebuah laporan berita.
Aktor	Pengguna bertipe <i>administrator</i> .
Kondisi Awal	Pengguna tidak berada di halaman yang berisi informasi sebuah laporan berita.
Kondisi Akhir	Pengguna berada di halaman yang berisi informasi sebuah laporan berita.
Alur Kejadian Normal	1. Aktor membuka halaman <i>news report management</i> .

	2. Aktor mengklik tombol <i>view</i> pada laporan yang ingin dilihat. 3. Aktor dibawa ke halaman yang berisi informasi dari laporan tersebut.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.29. Melihat Semua Laporan untuk Satu Berita

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah melihat semua laporan untuk satu berita. Spesifikasi kasus penggunaan untuk kebutuhan melihat semua laporan untuk satu berita dapat dilihat pada Tabel 4.30.

Tabel 4.30 Spesifikasi Kasus Penggunaan Melihat Semua Laporan untuk Satu Berita

Nama	Melihat semua laporan untuk satu berita
Kode	UC029
Deskripsi	Pengguna dapat melihat semua laporan untuk satu berita.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>view reports</i> .
Aktor	Pengguna bertipe <i>administrator</i>
Kondisi Awal	Pengguna tidak berada di halaman yang berisi laporan-laporan untuk satu berita.

Kondisi Akhir	Pengguna berada di halaman yang berisi laporan-laporan untuk satu berita.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor membuka halaman <i>news management</i>. 2. Aktor mengklik tombol <i>view</i> pada berita yang laporan-laporannya ingin dilihat. 3. Aktor mengklik tombol <i>view reports</i>. 4. Aktor dibawa ke halaman yang berisi daftar laporan-laporan untuk berita tersebut.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.30. Menghapus Sebuah Laporan Berita

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menghapus sebuah laporan berita. Spesifikasi kasus penggunaan untuk kebutuhan menghapus sebuah laporan berita dapat dilihat pada Tabel 4.31.

Tabel 4.31 Spesifikasi Kasus Penggunaan Menghapus Sebuah Laporan Berita

Nama	Menghapus sebuah laporan berita
Kode	UC030
Deskripsi	Pengguna dapat menghapus sebuah laporan berita.
Tipe	Fungsional

Pemicu	Pengguna mengklik tombol <i>delete</i> .
Aktor	Pengguna bertipe <i>administrator</i>
Kondisi Awal	Laporan berita masih disimpan di dalam sistem.
Kondisi Akhir	Laporan berita dihapuskan dari sistem.
Alur Kejadian Normal	1. Aktor membuka halaman <i>news report management</i> . 2. Aktor mengklik tombol <i>view</i> pada laporan yang ingin dihapus. 3. Aktor mengklik tombol <i>delete</i> . 4. Laporan dihapuskan dari sistem.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.31. Menampilkan Daftar Laporan Komentar

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menampilkan daftar laporan komentar. Spesifikasi kasus penggunaan untuk kebutuhan menampilkan daftar laporan komentar dapat dilihat pada Tabel 4.32.

Tabel 4.32 Spesifikasi Kasus Penggunaan Menampilkan Daftar Laporan Komentar

Nama	Menampilkan daftar laporan komentar
Kode	UC031
Deskripsi	Pengguna dapat menampilkan daftar laporan komentar.

Tipe	Fungsional
Pemicu	Pengguna mengklik <i>link</i> menuju halaman <i>comment report management</i> yang ada di <i>navigation bar</i> .
Aktor	Pengguna bertipe <i>administrator</i>
Kondisi Awal	Pengguna tidak berada di halaman <i>comment report management</i> yang berisi daftar laporan <i>user</i> .
Kondisi Akhir	Pengguna tidak berada di halaman <i>comment report management</i> yang berisi daftar laporan <i>user</i> .
Alur Kejadian Normal	1. Aktor mengklik <i>link</i> menuju halaman <i>comment report management</i> yang ada di menu <i>administrator</i> di <i>navigation bar</i> . 2. Aktor dibawa ke halaman <i>comment report management</i> yang berisi daftar laporan komentar.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.32. Melihat Sebuah Laporan Komentar

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah melihat sebuah laporan komentar. Spesifikasi kasus penggunaan untuk kebutuhan melihat sebuah laporan komentar dapat dilihat pada Tabel 4.33.

Tabel 4.33 Spesifikasi Kasus Penggunaan Melihat Sebuah Laporan Komentar

Nama	Melihat sebuah laporan komentar
Kode	UC032
Deskripsi	Pengguna dapat melihat sebuah laporan komentar.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>view</i> pada sebuah laporan komentar.
Aktor	Pengguna bertipe <i>administrator</i> .
Kondisi Awal	Pengguna tidak berada di halaman yang berisi informasi sebuah laporan komentar.
Kondisi Akhir	Pengguna berada di halaman yang berisi informasi sebuah laporan komentar.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor membuka halaman <i>comment report management</i>. 2. Aktor mengklik tombol <i>view</i> pada laporan yang ingin dilihat. 3. Aktor dibawa ke halaman yang berisi informasi dari laporan tersebut.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.33. Melihat Semua Laporan untuk Satu Komentar

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah melihat semua laporan untuk satu komentar. Spesifikasi kasus penggunaan untuk

kebutuhan melihat semua laporan untuk satu komentar dapat dilihat pada Tabel 4.34.

Tabel 4.34 Spesifikasi Kasus Penggunaan Melihat Semua Laporan untuk Satu Komentar

Nama	Melihat semua laporan untuk satu komentar
Kode	UC033
Deskripsi	Pengguna dapat melihat semua laporan untuk satu komentar.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>view reports</i> .
Aktor	Pengguna bertipe <i>administrator</i>
Kondisi Awal	Pengguna tidak berada di halaman yang berisi laporan-laporan untuk satu komentar.
Kondisi Akhir	Pengguna berada di halaman yang berisi laporan-laporan untuk satu komentar.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor membuka halaman <i>comment management</i> untuk sebuah artikel berita. 2. Aktor mengklik tombol <i>view</i> pada komentar yang laporannya ingin dilihat. 3. Aktor mengklik tombol <i>view reports</i>. 4. Aktor dibawa ke halaman yang berisi daftar laporan-laporan untuk komentar tersebut.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.34. Menghapus Sebuah Laporan Komentar

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menghapus sebuah laporan komentar. Spesifikasi kasus penggunaan untuk kebutuhan menghapus sebuah laporan komentar dapat dilihat pada Tabel 4.35.

Tabel 4.35 Spesifikasi Kasus Penggunaan Menghapus Sebuah Laporan Komentar

Nama	Menghapus sebuah laporan komentar
Kode	UC034
Deskripsi	Pengguna dapat menghapus sebuah laporan komentar.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>delete</i> .
Aktor	Pengguna bertipe <i>administrator</i>
Kondisi Awal	Laporan komentar masih disimpan di dalam sistem.
Kondisi Akhir	Laporan komentar dihapuskan dari sistem.
Alur Kejadian Normal	1. Aktor membuka halaman <i>comment report management</i> . 2. Aktor mengklik tombol <i>view</i> pada laporan yang ingin dihapus. 3. Aktor mengklik tombol <i>delete</i> . 4. Laporan dihapuskan dari sistem.
Alur Kejadian Alternatif	-

Pengecualian	-
--------------	---

4.3.35. Menampilkan Daftar *User*

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menampilkan daftar *user*. Spesifikasi kasus penggunaan untuk kebutuhan menampilkan daftar *user* dapat dilihat pada Tabel 4.36.

Tabel 4.36 Spesifikasi Kasus Penggunaan Menampilkan Daftar User

Nama	Menampilkan daftar <i>user</i>
Kode	UC035
Deskripsi	Pengguna dapat menampilkan daftar <i>user</i> .
Tipe	Fungsional
Pemicu	Pengguna mengklik <i>link</i> menuju halaman <i>user management</i> yang ada di <i>navigation bar</i> .
Aktor	Pengguna bertipe <i>administrator</i>
Kondisi Awal	Pengguna tidak berada di halaman <i>user management</i> yang berisi daftar <i>user</i> .
Kondisi Akhir	Pengguna berada di halaman <i>user management</i> yang berisi daftar <i>user</i> .
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor mengklik <i>link</i> menuju halaman <i>user management</i> yang ada di menu <i>administrator</i> di <i>navigation bar</i>. 2. Aktor dibawa ke halaman <i>user management</i> yang berisi daftar semua <i>user</i> yang terdaftar.

Alur Kejadian Alternatif	-
Pengecualian	-

4.3.36. Membuat *User* Baru

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah membuat *user* baru. Spesifikasi kasus penggunaan untuk kebutuhan membuat *user* baru dapat dilihat pada Tabel 4.37.

Tabel 4.37 Spesifikasi Kasus Penggunaan Membuat *User* Baru

Nama	Membuat <i>user</i> baru
Kode	UC036
Deskripsi	Pengguna dapat membuat <i>user</i> baru.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>create user</i> .
Aktor	Pengguna bertipe <i>administrator</i>
Kondisi Awal	Tidak ada <i>user</i> baru yang ditambahkan ke dalam sistem.
Kondisi Akhir	<i>User</i> yang dibuat ditambahkan ke dalam sistem.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor membuka halaman <i>user management</i>. 2. Aktor mengklik tombol '+' 3. Aktor mengisikan data dari <i>user</i> yang ingin dibuat, kemudian mengklik tombol <i>create user</i>. 4. <i>User</i> dibuat dan ditambahkan ke dalam sistem.

Alur Kejadian Alternatif	-
Pengecualian	-

4.3.37. Menampilkan Informasi Sebuah *User*

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menampilkan informasi sebuah *user*. Spesifikasi kasus penggunaan untuk kebutuhan menampilkan informasi sebuah *user* dapat dilihat pada Tabel 4.38.

Tabel 4.38 Spesifikasi Kasus Penggunaan Menampilkan Informasi Sebuah *User*

Nama	Menampilkan informasi sebuah <i>user</i>
Kode	UC037
Deskripsi	Pengguna dapat menampilkan informasi sebuah <i>user</i> .
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>view</i> .
Aktor	Pengguna bertipe <i>administrator</i>
Kondisi Awal	Pengguna tidak berada di halaman yang berisi informasi sebuah <i>user</i> .
Kondisi Akhir	Pengguna berada di halaman yang berisi informasi <i>user</i> yang dipilih.
Alur Kejadian Normal	1. Aktor membuka halaman <i>user management</i> . 2. Aktor mengklik tombol <i>view</i> pada <i>user</i> yang ingin dilihat.

	3. Aktor dibawa ke halaman yang berisi informasi dari <i>user</i> yang dipilih.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.38. Memperbarui Data Sebuah *User*

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah memperbarui data sebuah *user*. Spesifikasi kasus penggunaan untuk kebutuhan memperbarui data sebuah *user* dapat dilihat pada Tabel 4.39.

Tabel 4.39 Spesifikasi Kasus Penggunaan Memperbarui Data Sebuah *User*

Nama	Memperbarui data sebuah <i>user</i>
Kode	UC038
Deskripsi	Pengguna dapat memperbarui data sebuah <i>user</i> .
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>save</i> .
Aktor	Pengguna bertipe <i>administrator</i> .
Kondisi Awal	Data dari <i>user</i> yang dipilih tidak berubah.
Kondisi Akhir	Data dari <i>user</i> yang dipilih berhasil diubah dan perubahannya disimpan di dalam sistem.
Alur Kejadian Normal	1. Aktor membuka halaman <i>user management</i> .

	2. Aktor mengklik tombol <i>view</i> pada <i>user</i> yang datanya ingin diubah. 3. Aktor memperbarui data <i>user</i> dan mengklik tombol <i>save</i> . 4. Data dari <i>user</i> tersebut diperbarui sesuai perubahan yang telah dilakukan dan disimpan di dalam sistem.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.39. Menampilkan Daftar Komentar untuk Sebuah Berita

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menampilkan daftar komentar untuk sebuah berita. Spesifikasi kasus penggunaan untuk kebutuhan menampilkan daftar komentar untuk sebuah berita dapat dilihat pada Tabel 4.40.

Tabel 4.40 Spesifikasi Kasus Penggunaan Menampilkan Daftar Komentar untuk Sebuah Berita

Nama	Menampilkan daftar komentar untuk sebuah berita
Kode	UC039
Deskripsi	Pengguna dapat menampilkan daftar komentar untuk sebuah berita.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>comment management</i> .

Aktor	Pengguna bertipe <i>administrator</i> .
Kondisi Awal	Pengguna tidak berada di halaman daftar komentar untuk sebuah berita.
Kondisi Akhir	Pengguna berada di halaman daftar komentar untuk sebuah berita.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor membuka halaman berita dari sebuah artikel berita. 2. Aktor mengklik tombol <i>comment management</i>. 3. Aktor dibawa ke halaman daftar komentar untuk berita tersebut.
Alur Kejadian Alternatif	<ol style="list-style-type: none"> 1. Aktor membuka halaman <i>news management</i>. 2. Aktor mengklik tombol <i>view</i> pada berita yang daftar komentarnya ingin ditampilkan. 3. Aktor mengklik tombol <i>comment management</i>. 4. Aktor dibawa ke halaman daftar komentar untuk berita tersebut.
Pengecualian	-

4.3.40. Menampilkan Sebuah Komentar

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menampilkan sebuah komentar. Spesifikasi kasus penggunaan untuk kebutuhan menampilkan sebuah komentar dapat dilihat pada Tabel 4.41.

Tabel 4.41 Spesifikasi Kasus Penggunaan Menampilkan Sebuah Komentar

Nama	Menampilkan sebuah komentar
Kode	UC040
Deskripsi	Pengguna dapat menampilkan sebuah komentar.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>view</i> .
Aktor	Pengguna bertipe <i>administrator</i> .
Kondisi Awal	Pengguna tidak berada di halaman yang berisi data komentar.
Kondisi Akhir	Pengguna berada di halaman yang berisi data komentar yang dipilih.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor membuka halaman <i>comment management</i> untuk sebuah artikel berita. 2. Aktor mengklik tombol <i>view</i> pada komentar yang ingin dilihat. 3. Halaman berisi data komentar yang dipilih ditampilkan kepada aktor.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.41. Menampilkan Daftar Balasan untuk Sebuah Komentar

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menampilkan daftar balasan untuk sebuah komentar. Spesifikasi kasus penggunaan

untuk kebutuhan menampilkan daftar balasan untuk sebuah komentar dapat dilihat pada Tabel 4.42.

Tabel 4.42 Spesifikasi Kasus Penggunaan Menampilkan Daftar Balasan untuk Sebuah Komentar

Nama	Menampilkan daftar balasan untuk sebuah komentar
Kode	UC041
Deskripsi	Pengguna dapat melihat daftar balasan untuk sebuah komentar.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>view replies</i> .
Aktor	Pengguna bertipe <i>administrator</i> .
Kondisi Awal	Pengguna tidak berada di halaman daftar balasan untuk sebuah komentar.
Kondisi Akhir	Pengguna berada di halaman daftar balasan untuk sebuah komentar.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor membuka halaman <i>comment management</i> untuk sebuah artikel berita. 2. Aktor mengklik tombol <i>view</i> pada komentar yang ingin ditampilkan daftar balasannya. 3. Aktor mengklik tombol <i>view replies</i>. 4. Halaman daftar balasan untuk komentar yang dipilih ditampilkan kepada aktor.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.42. Menghapus Komentar *User* Lain

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menghapus komentar *user* lain. Spesifikasi kasus penggunaan untuk kebutuhan menghapus komentar *user* lain dapat dilihat pada Tabel 4.43.

Tabel 4.43 Spesifikasi Kasus Penggunaan Menghapus Komentar *User* Lain

Nama	Menghapus komentar <i>user</i> lain
Kode	UC042
Deskripsi	Pengguna dapat menghapus komentar milik <i>user</i> lain.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>delete</i> .
Aktor	Pengguna bertipe <i>administrator</i> .
Kondisi Awal	Komentar yang ingin dihapus masih disimpan di dalam sistem.
Kondisi Akhir	Komentar berhasil dihapus dari sistem.
Alur Kejadian Normal	<ol style="list-style-type: none">1. Aktor membuka halaman <i>comment management</i> untuk sebuah berita atau halaman daftar balasan untuk sebuah komentar.2. Aktor mengklik tombol <i>view</i> pada komentar yang ingin dihapus.3. Aktor mengklik tombol <i>delete</i>.4. Komentar dihapuskan dari sistem.

Alur Kejadian Alternatif	-
Pengecualian	-

4.3.43. Menampilkan Daftar Berita

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menampilkan daftar berita. Spesifikasi kasus penggunaan untuk kebutuhan menampilkan daftar berita dapat dilihat pada Tabel 4.44.

Tabel 4.44 Spesifikasi Kasus Penggunaan Menampilkan Daftar Berita

Nama	Menampilkan daftar berita
Kode	UC043
Deskripsi	Pengguna dapat menampilkan daftar artikel-artikel berita yang ada.
Tipe	Fungsional
Pemicu	Pengguna mengklik <i>link</i> menuju halaman <i>news management</i> yang ada di <i>navigation bar</i> .
Aktor	Pengguna bertipe <i>administrator</i> atau <i>editor</i> .
Kondisi Awal	Pengguna tidak berada di halaman <i>news management</i> yang berisi daftar berita.
Kondisi Akhir	Pengguna berada di halaman <i>news management</i> yang berisi daftar berita.
Alur Kejadian Normal	1. Aktor bertipe <i>administrator</i> mengklik <i>link</i> menuju halaman

	<p><i>news management</i> yang ada pada menu <i>administrator</i> di <i>navigation bar</i>.</p> <p>2. Halaman <i>news management</i> yang berisi daftar berita ditampilkan kepada aktor.</p>
Alur Kejadian Alternatif	<p>1. Aktor bertipe <i>editor</i> mengklik <i>link</i> menuju halaman <i>news management</i> yang ada di <i>navigation bar</i>.</p> <p>2. Halaman <i>news management</i> yang berisi daftar berita ditampilkan kepada aktor.</p>
Pengecualian	-

4.3.44. Membuat Berita Baru

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah membuat berita baru. Spesifikasi kasus penggunaan untuk kebutuhan membuat berita baru dapat dilihat pada Tabel 4.45.

Tabel 4.45 Spesifikasi Kasus Penggunaan Membuat Berita Baru

Nama	Membuat berita baru
Kode	UC044
Deskripsi	Pengguna dapat membuat berita baru.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>create news</i> .
Aktor	Pengguna bertipe <i>administrator</i> atau <i>editor</i> .

Kondisi Awal	Tidak ada berita baru yang ditambahkan ke dalam sistem.
Kondisi Akhir	Berita baru berhasil dibuat dan ditambahkan ke dalam sistem.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor membuka halaman <i>news management</i> dan mengklik tombol '+’. 2. Aktor mengisi data untuk berita dan mengklik tombol <i>create news</i>. 3. Berita yang baru akan dibuat dan disimpan di dalam sistem.
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.45. Mengedit Sebuah Berita

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah mengedit sebuah berita. Spesifikasi kasus penggunaan untuk kebutuhan mengedit sebuah berita dapat dilihat pada Tabel 4.46.

Tabel 4.46 Spesifikasi Kasus Penggunaan Mengedit Sebuah Berita

Nama	Mengedit sebuah berita
Kode	UC045
Deskripsi	Pengguna dapat mengedit sebuah berita yang ada.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>save</i> .
Aktor	Pengguna bertipe <i>administrator</i> atau <i>editor</i> .
Kondisi Awal	Isi dari berita tidak berubah.

Kondisi Akhir	Isi dari berita diperbarui sesuai dengan perubahan yang dilakukan.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor bertipe <i>administrator</i> membuka halaman <i>news management</i> dan mengklik tombol <i>view news</i> pada berita yang isinya ingin diubah. 2. Aktor bertipe <i>administrator</i> melakukan perubahan terhadap isi berita dan mengklik tombol <i>update</i>. 3. Isi berita diperbarui dan disimpan di dalam sistem.
Alur Kejadian Alternatif	<ol style="list-style-type: none"> 1. Aktor bertipe <i>editor</i> membuka halaman <i>news management</i> dan mengklik tombol <i>view news</i> pada berita miliknya yang isinya ingin diubah. 2. Aktor bertipe <i>editor</i> melakukan perubahan terhadap isi berita dan mengklik tombol <i>update</i>. 3. Isi berita diperbarui dan disimpan di dalam sistem.
Pengecualian	-

4.3.46. Menghapus Sebuah Berita

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menghapus sebuah berita. Spesifikasi kasus penggunaan untuk kebutuhan menghapus sebuah berita dapat dilihat pada Tabel 4.47.

Tabel 4.47 Spesifikasi Kasus Penggunaan Menghapus Sebuah Berita

Nama	Menghapus sebuah berita
Kode	UC046
Deskripsi	Pengguna dapat menghapus sebuah berita.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>delete</i> .
Aktor	<i>Administrator</i> atau <i>editor</i>
Kondisi Awal	Halaman dan data untuk sebuah berita masih ada di dalam sistem.
Kondisi Akhir	Halaman dan data untuk sebuah berita dihapuskan dari sistem.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor bertipe <i>administrator</i> membuka halaman <i>news management</i>. 2. Aktor bertipe <i>administrator</i> mengklik tombol <i>view</i> pada berita yang ingin dihapus. 3. Aktor bertipe <i>administrator</i> mengklik tombol <i>delete</i> untuk menghapus berita. 4. Berita dihapuskan dari sistem.
Alur Kejadian Alternatif	<ol style="list-style-type: none"> 1. Aktor bertipe <i>editor</i> membuka halaman <i>news management</i>. 2. Aktor bertipe <i>editor</i> mengklik tombol <i>view</i> pada berita miliknya yang ingin dihapus. 3. Aktor bertipe <i>editor</i> mengklik tombol <i>delete</i> untuk menghapus berita. 4. Berita dihapuskan dari sistem.
Pengecualian	-

4.3.47. Mengupload *File* untuk Sebuah Berita

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah mengupload *file* untuk sebuah berita. Spesifikasi kasus penggunaan untuk kebutuhan mengupload *file* untuk sebuah berita dapat dilihat pada Tabel 4.48.

Tabel 4.48 Spesifikasi Kasus Penggunaan Mengupload File untuk Sebuah Berita

Nama	Mengupload <i>file</i> untuk sebuah berita
Kode	UC-047
Deskripsi	Pengguna dapat mengupload sebuah <i>file</i> seperti gambar yang dapat digunakan di dalam artikel berita.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>upload file</i> .
Aktor	Pengguna bertipe <i>administrator</i> atau <i>editor</i> .
Kondisi Awal	<i>File</i> yang dipilih pengguna belum diupload dan disimpan di dalam sistem.
Kondisi Akhir	<i>File</i> yang dipilih pengguna diupload dan disimpan di dalam sistem sehingga dapat digunakan dalam artikel berita.
Alur Kejadian Normal	1. Aktor bertipe <i>administrator</i> membuka halaman <i>news management</i> .

	<p>2. Aktor bertipe <i>administrator</i> mengklik tombol <i>view</i> pada berita yang ingin diupload <i>filenya</i>.</p> <p>3. Aktor bertipe <i>administrator</i> mengklik tombol <i>choose file</i> untuk memilih <i>file</i> yang akan diupload.</p> <p>4. Aktor bertipe <i>administrator</i> mengklik tombol <i>upload file</i> untuk mengupload <i>file</i> yang dipilih.</p> <p>5. <i>File</i> diupload dan disimpan di dalam sistem dan dapat digunakan di dalam artikel berita.</p>
Alur Kejadian Alternatif	<p>1. Aktor bertipe <i>editor</i> membuka halaman <i>news management</i>.</p> <p>2. Aktor bertipe <i>editor</i> mengklik tombol <i>view</i> pada berita miliknya yang ingin diupload <i>filenya</i>.</p> <p>3. Aktor bertipe <i>editor</i> mengklik tombol <i>choose file</i> untuk memilih <i>file</i> yang akan diupload.</p> <p>4. Aktor bertipe <i>editor</i> mengklik tombol <i>upload file</i> untuk mengupload <i>file</i> yang dipilih.</p> <p>5. <i>File</i> diupload dan disimpan di dalam sistem dan dapat digunakan di dalam artikel berita.</p>
Pengecualian	-

4.3.48. Menghapus *File* yang Diupload untuk Sebuah Berita

Salah satu kebutuhan yang ada pada *website* portal berita yang dibuat adalah menghapus *file* yang diupload untuk sebuah berita. Spesifikasi kasus penggunaan

untuk kebutuhan menghapus *file* yang diupload untuk sebuah berita dapat dilihat pada Tabel 4.49.

Tabel 4.49 Spesifikasi Kasus Penggunaan Menghapus File yang Diupload untuk Sebuah Berita

Nama	Menghapus <i>file</i> yang diupload untuk sebuah berita
Kode	UC-048
Deskripsi	Pengguna dapat menghapus <i>file</i> yang sebelumnya telah diupload untuk sebuah berita.
Tipe	Fungsional
Pemicu	Pengguna mengklik tombol <i>delete file</i> .
Aktor	Pengguna bertipe <i>administrator</i> atau <i>editor</i> .
Kondisi Awal	<i>File</i> belum dihapus dan masih disimpan di dalam sistem.
Kondisi Akhir	<i>File</i> dihapus dari sistem.
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aktor bertipe <i>administrator</i> membuka halaman <i>news management</i>. 2. Aktor bertipe <i>administrator</i> mengklik tombol <i>view</i> pada berita yang ingin dihapus <i>filenya</i>. 3. Aktor bertipe <i>administrator</i> mengklik tombol <i>delete file</i> pada <i>file</i> yang ingin dihapus. 4. <i>File</i> yang dipilih dihapuskan dari sistem.
Alur Kejadian Alternatif	<ol style="list-style-type: none"> 1. Aktor bertipe <i>editor</i> membuka halaman <i>news management</i>.

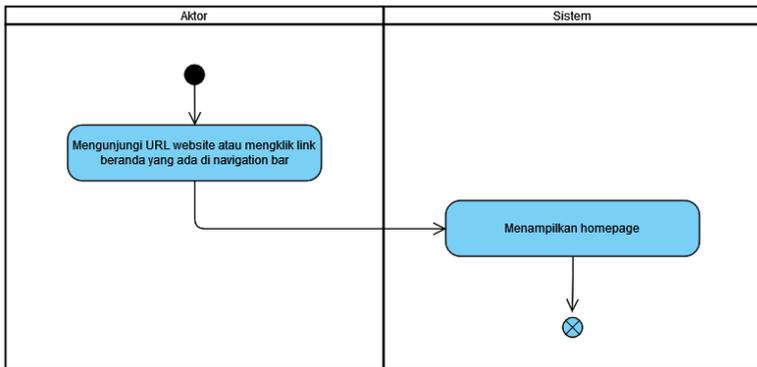
	<p>2. Aktor bertipe <i>editor</i> mengklik tombol <i>view</i> pada berita miliknya yang ingin dihapus <i>filenya</i>.</p> <p>3. Aktor bertipe <i>editor</i> mengklik tombol <i>delete file</i> pada <i>file</i> yang ingin dihapus.</p> <p>4. <i>File</i> yang dipilih dihapuskan dari sistem.</p>
Pengecualian	-

4.4. Diagram Aktivitas

Subbab ini berisikan diagram-diagram aktivitas untuk setiap kebutuhan fungsional yang ada pada aplikasi *website* portal berita.

4.4.1. Menampilkan *Homepage*

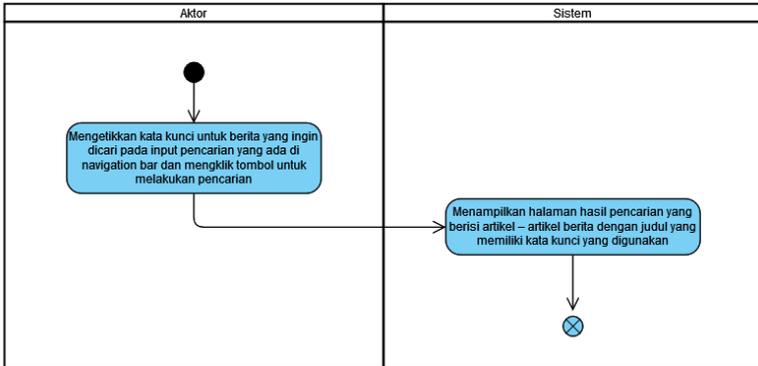
Diagram aktivitas dari kebutuhan menampilkan *homepage* dapat dilihat pada Gambar 4.2.



Gambar 4.2 Diagram Aktivitas Menampilkan Homepage

4.4.2. Melakukan Pencarian

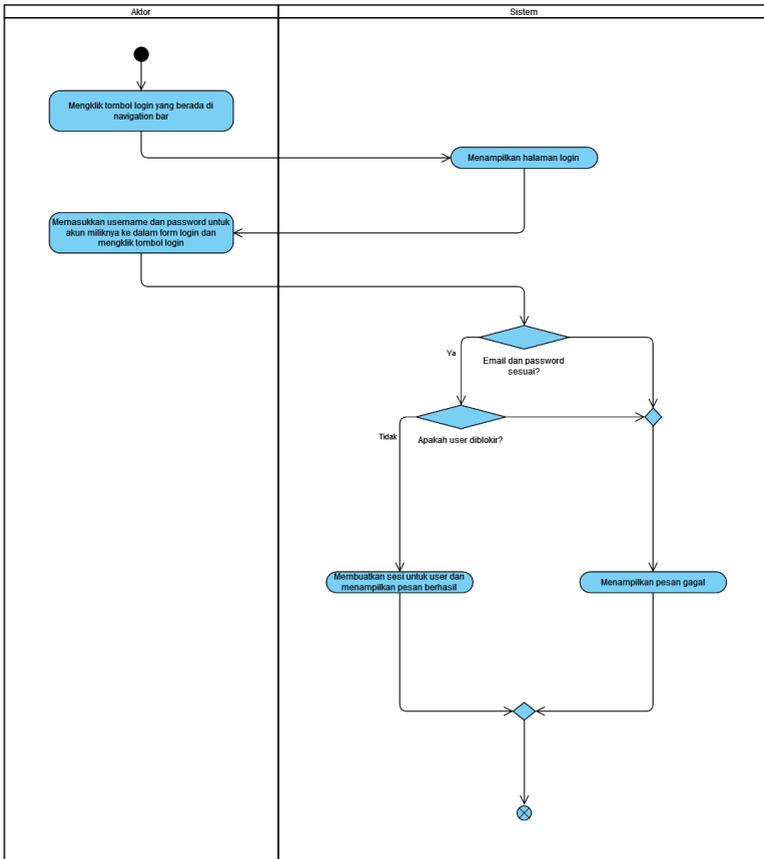
Diagram aktivitas dari kebutuhan melakukan pencarian dapat dilihat pada Gambar 4.3.



Gambar 4.3 Diagram Aktivitas Melakukan Pencarian

4.4.3. Melakukan *Login*

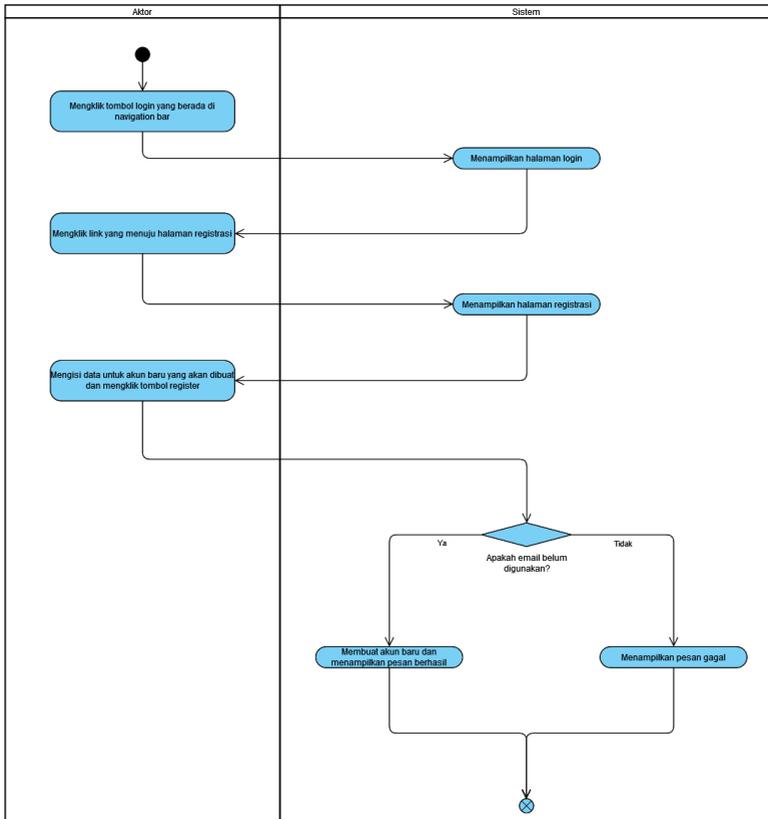
Diagram aktivitas dari kebutuhan melakukan *login* dapat dilihat pada Gambar 4.4.



Gambar 4.4 Diagram Aktivitas Melakukan Login

4.4.4. Melakukan Registrasi

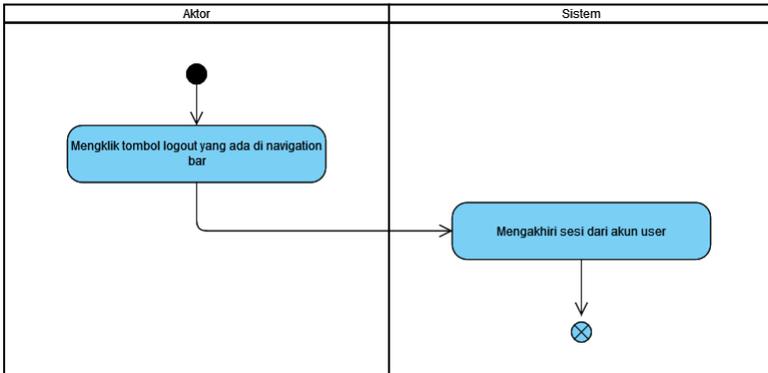
Diagram aktivitas dari kebutuhan melakukan registrasi dapat dilihat pada Gambar 4.5.



Gambar 4.5 Diagram Aktivitas Melakukan Registrasi

4.4.5. Melakukan *Logout*

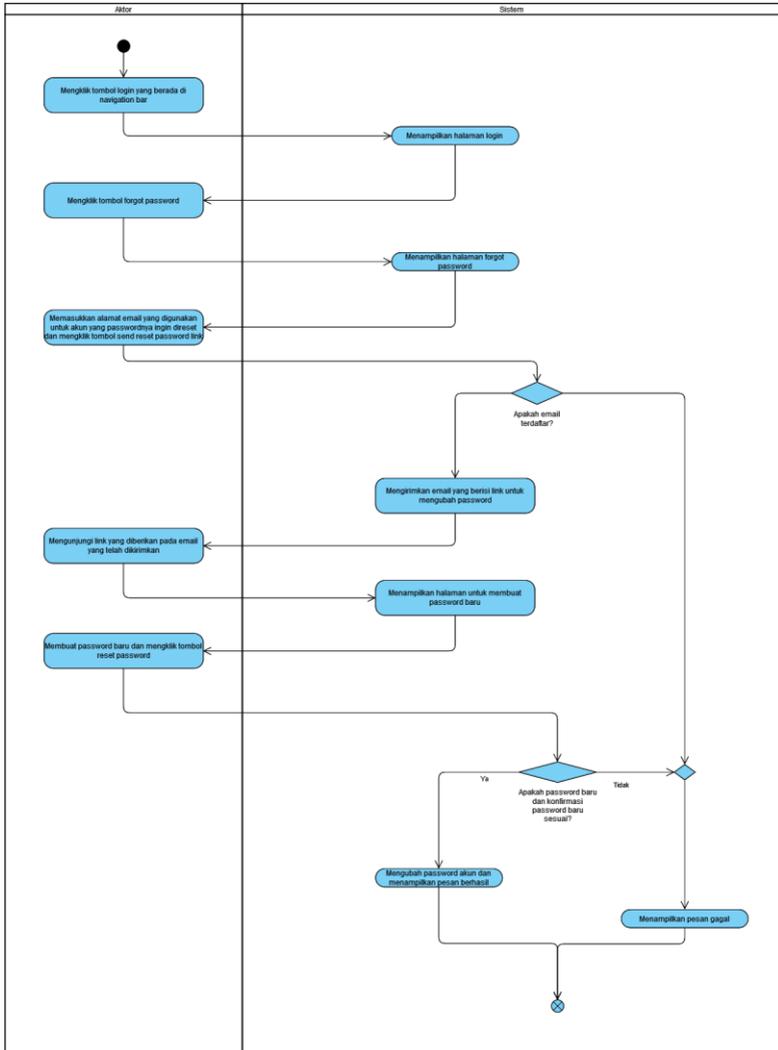
Diagram aktivitas dari kebutuhan melakukan *logout* dapat dilihat pada Gambar 4.6.



Gambar 4.6 Diagram Aktivitas Melakukan Logout

4.4.6. Melakukan *Reset Password*

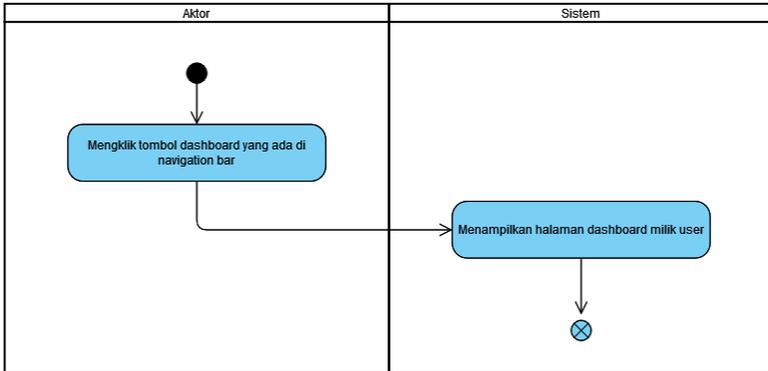
Diagram aktivitas dari kebutuhan melakukan *reset password* dapat dilihat pada Gambar 4.7.



Gambar 4.7 Diagram Aktivitas Melakukan Reset Password

4.4.7. Menampilkan *Dashboard*

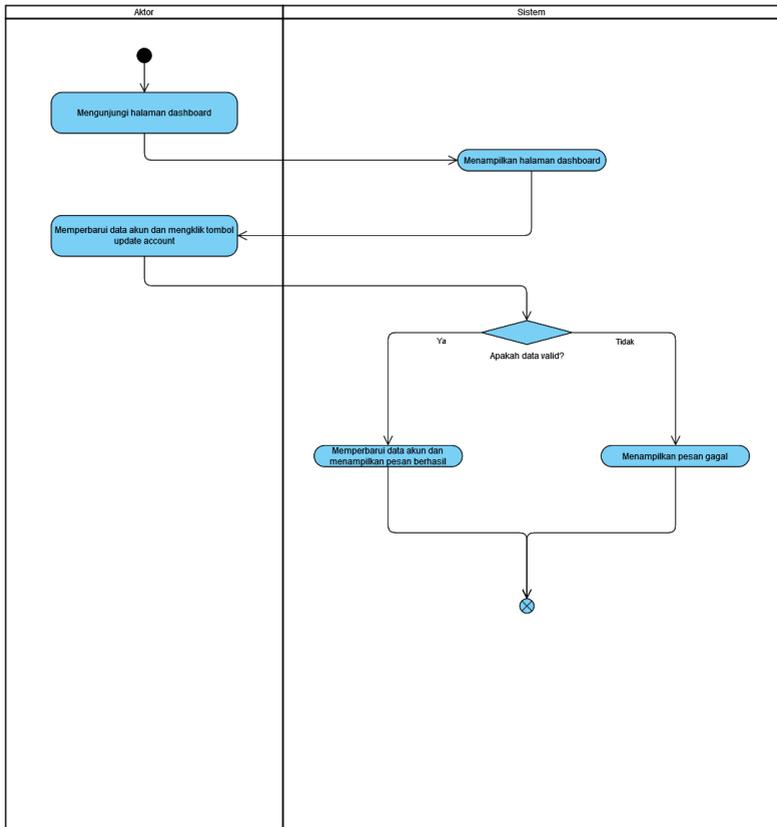
Diagram aktivitas dari kebutuhan menampilkan *dashboard* dapat dilihat pada Gambar 4.8.



Gambar 4.8 Diagram Aktivitas Menampilkan Dashboard

4.4.8. Memperbarui Data Akun

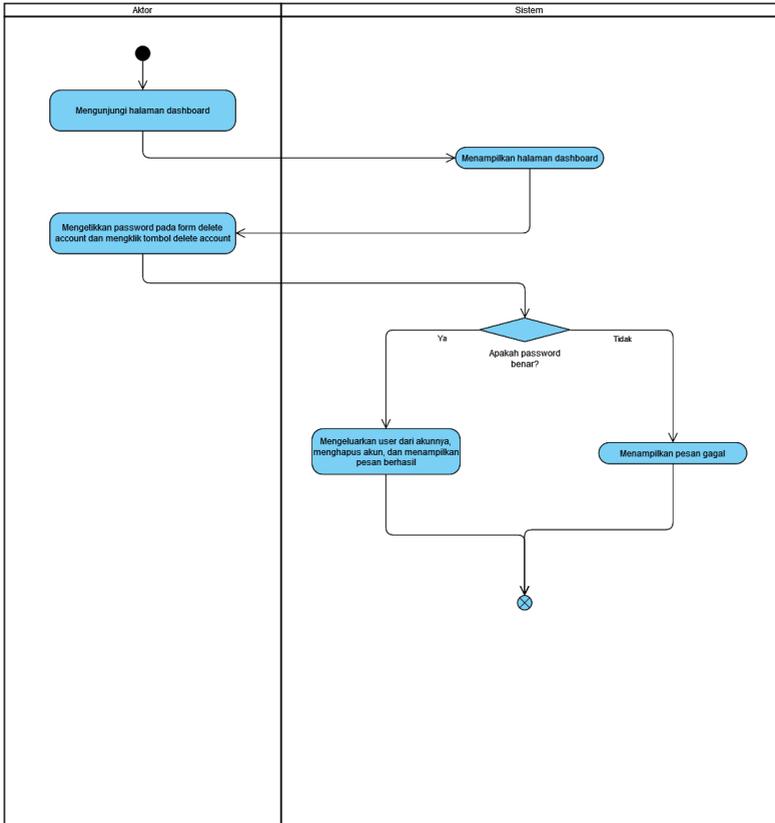
Diagram aktivitas dari kebutuhan memperbarui data akun dapat dilihat pada Gambar 4.9.



Gambar 4.9 Diagram Aktivitas Memperbarui Data Akun

4.4.9. Menghapus Akun

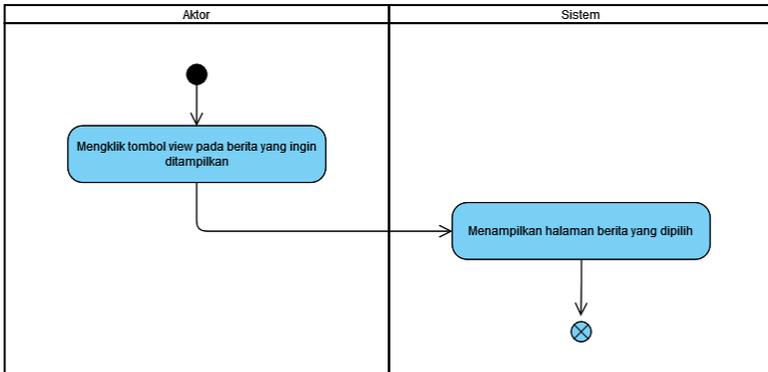
Diagram aktivitas dari kebutuhan menghapus akun dapat dilihat pada Gambar 4.10.



Gambar 4.10 Diagram Aktivitas Menghapus Akun

4.4.10. Menampilkan Berita

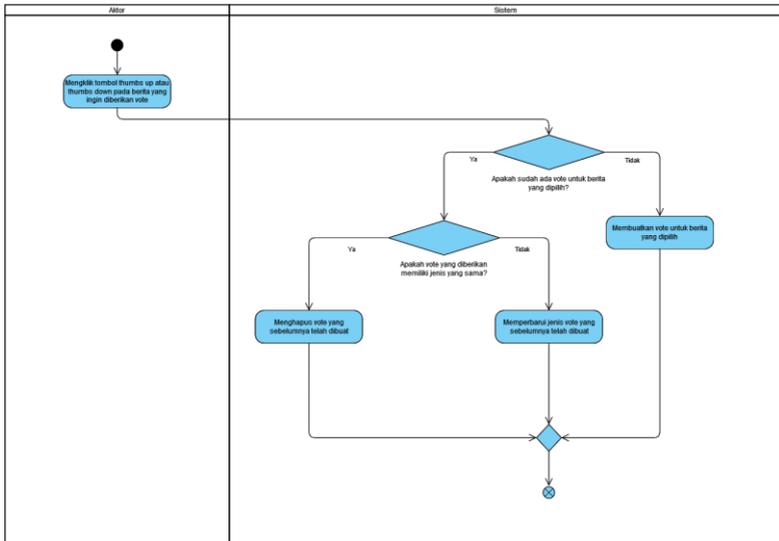
Diagram aktivitas dari kebutuhan menampilkan berita dapat dilihat pada Gambar 4.11.



Gambar 4.11 Diagram Aktivitas Menampilkan Berita

4.4.11. Memberikan atau Menghapus *Vote* pada Berita

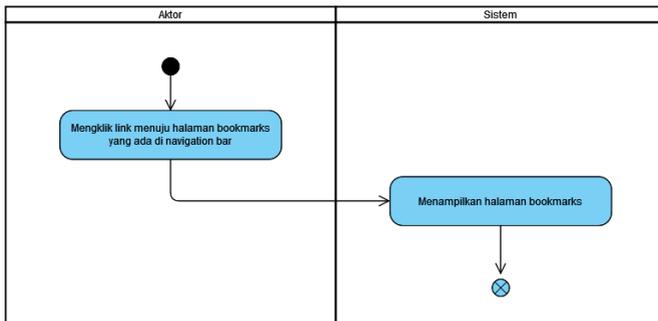
Diagram aktivitas dari kebutuhan memberikan atau menghapus *vote* pada berita dapat dilihat pada Gambar 4.12.



Gambar 4.12 Diagram Aktivitas Memberikan atau Menghapus Vote pada Berita

4.4.12. Menampilkan *Bookmarks*

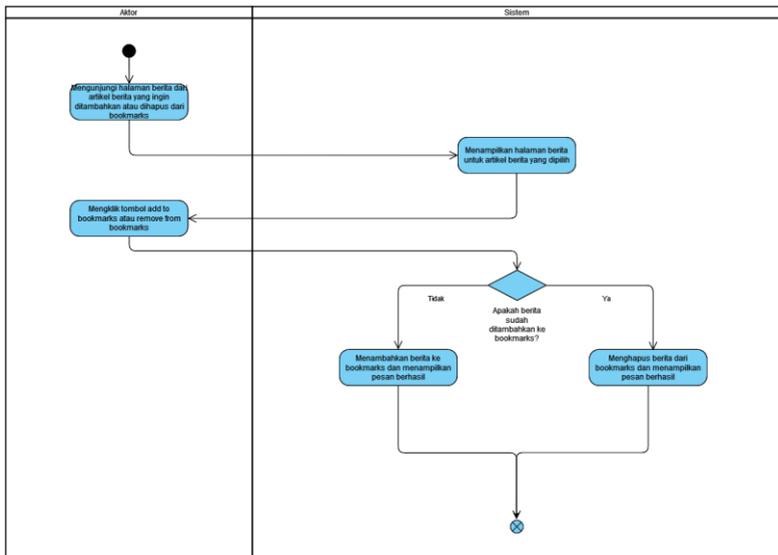
Diagram aktivitas dari kebutuhan menampilkan *bookmarks* dapat dilihat pada Gambar 4.13.



Gambar 4.13 Diagram Aktivitas Menampilkan Bookmarks

4.4.13. Menambahkan atau Menghapus Berita dari *Bookmarks*

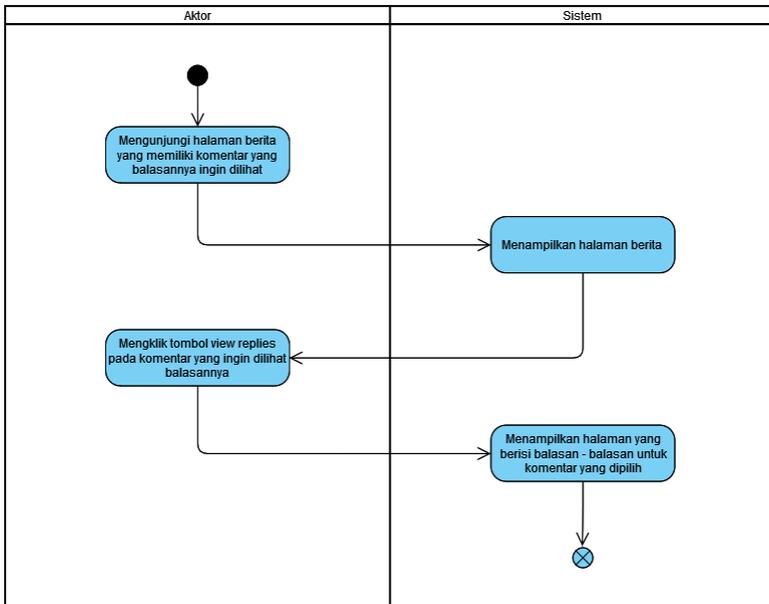
Diagram aktivitas dari kebutuhan menambahkan atau menghapus berita dari *bookmarks* dapat dilihat pada Gambar 4.14.



Gambar 4.14 Diagram Aktivitas Menambahkan atau Menghapus Berita dari Bookmarks

4.4.14. Melihat Balasan Komentar

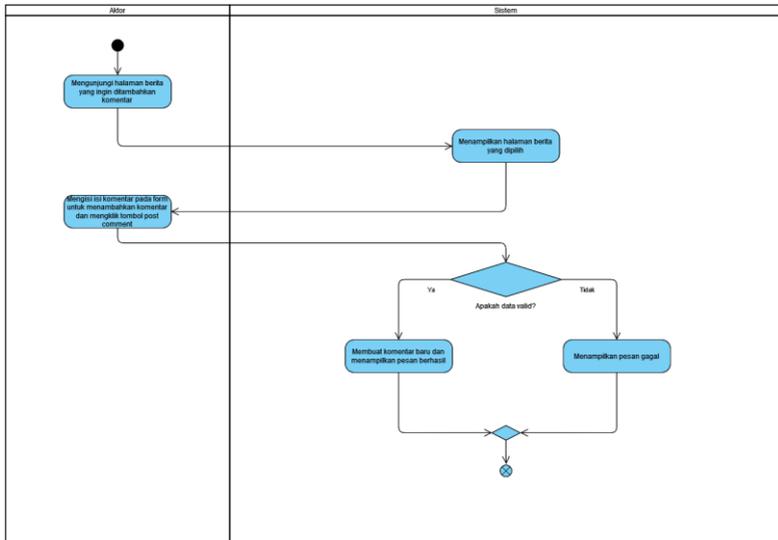
Diagram aktivitas dari kebutuhan melihat balasan komentar dapat dilihat pada Gambar 4.15.



Gambar 4.15 Diagram Aktivitas Melihat Balasan Komentar

4.4.15. Menambahkan Komentar pada Berita

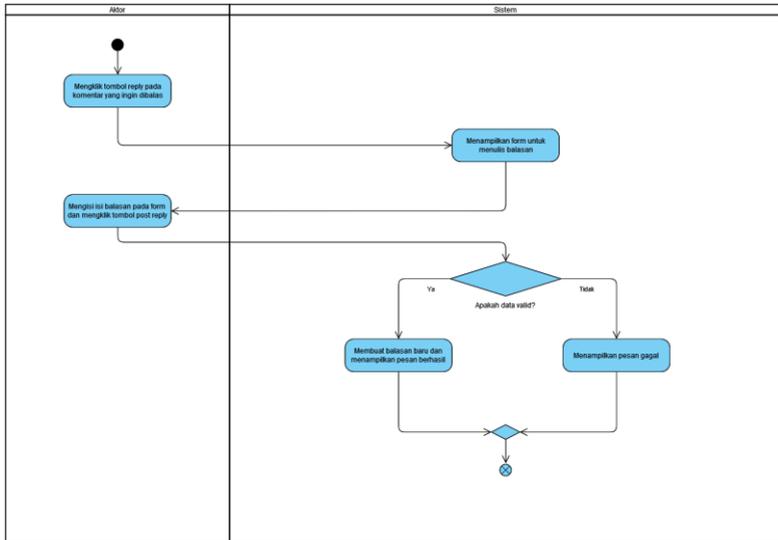
Diagram aktivitas dari kebutuhan menambahkan komentar pada berita dapat dilihat pada Gambar 4.16.



Gambar 4.16 Diagram Aktivitas Menambahkan Komentar pada Berita

4.4.16. Membalas Komentar

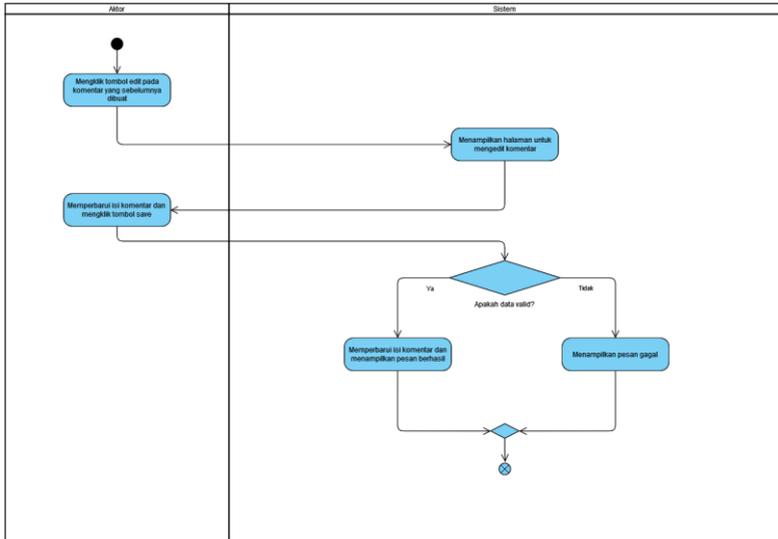
Diagram aktivitas dari kebutuhan membalas komentar dapat dilihat pada Gambar 4.17.



Gambar 4.17 Diagram Aktivitas Membalas Komentar

4.4.17. Mengedit Komentar

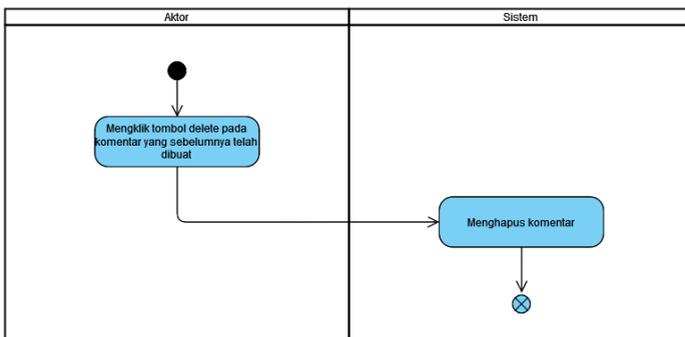
Diagram aktivitas dari kebutuhan mengedit komentar dapat dilihat pada Gambar 4.18.



Gambar 4.18 Diagram Aktivitas Mengedit Komentar

4.4.18. Menghapus Komentar Milik Sendiri

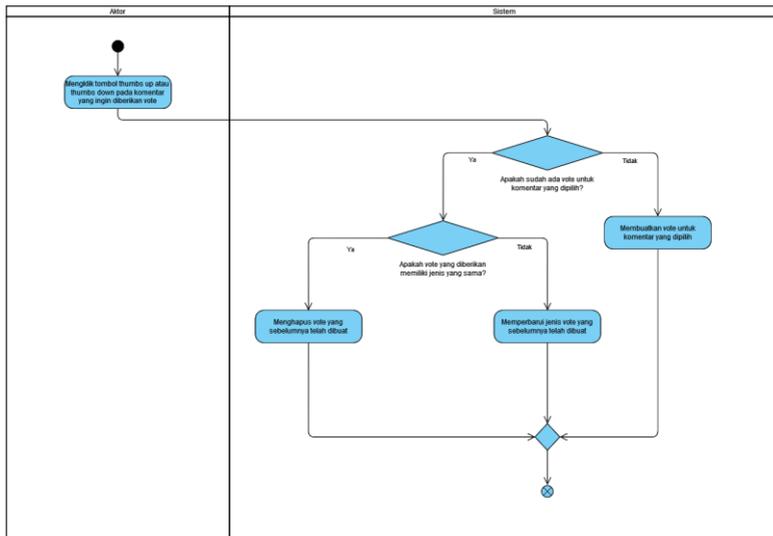
Diagram aktivitas dari kebutuhan menghapus komentar milik sendiri dapat dilihat pada Gambar 4.19.



Gambar 4.19 Diagram Aktivitas Menghapus Komentar Milik Sendiri

4.4.19. Memberikan atau Menghapus *Vote* pada Komentar

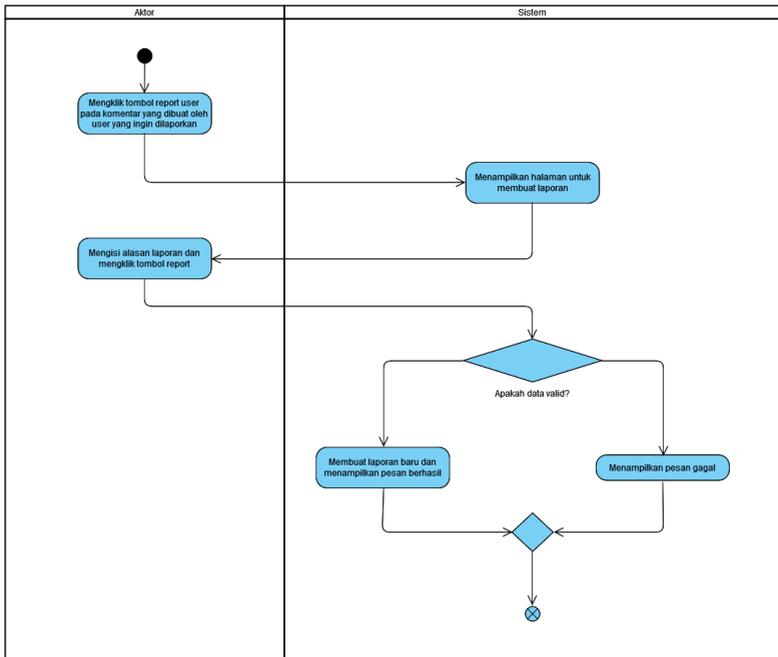
Diagram aktivitas dari kebutuhan memberikan atau menghapus *vote* pada komentar dapat dilihat pada Gambar 4.20.



Gambar 4.20 Diagram Aktivitas Memberikan atau Menghapus *Vote* pada Komentar

4.4.20. Melaporkan *User*

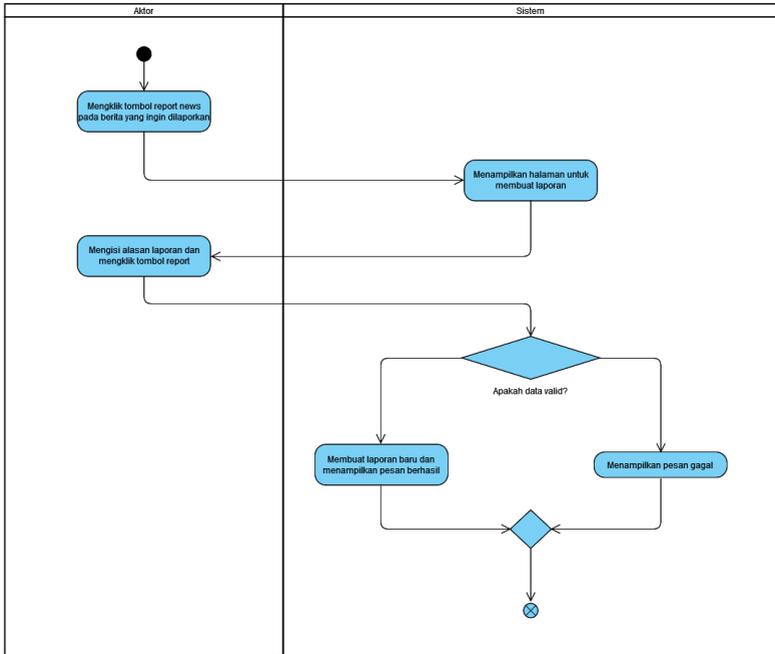
Diagram aktivitas dari kebutuhan melaporkan *user* dapat dilihat pada Gambar 4.21.



Gambar 4.21 Diagram Aktivitas Melaporkan User

4.4.21. Melaporkan Berita

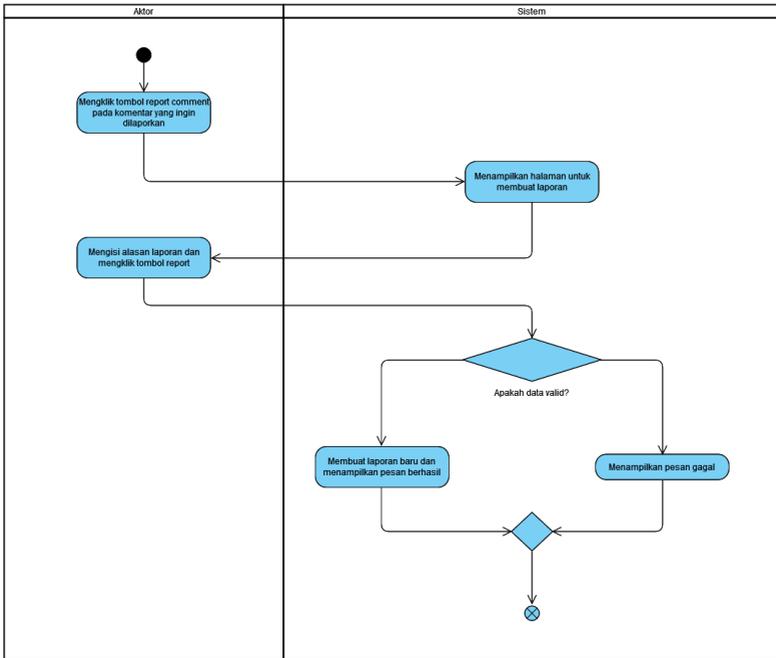
Diagram aktivitas dari kebutuhan melaporkan berita dapat dilihat pada Gambar 4.22.



Gambar 4.22 Diagram Aktivitas Melaporkan Berita

4.4.22. Melaporkan Komentar

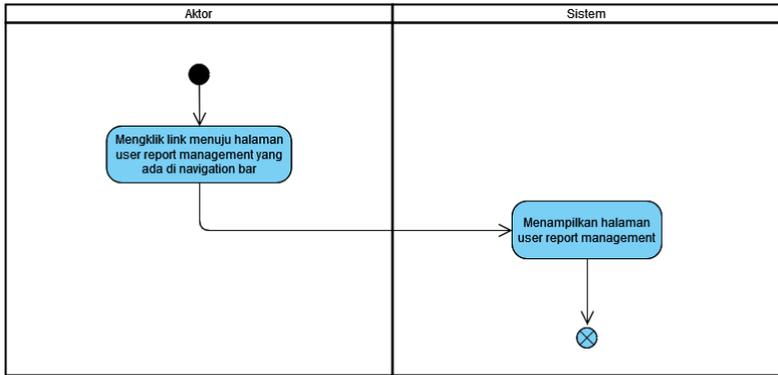
Diagram aktivitas dari kebutuhan melaporkan komentar dapat dilihat pada Gambar 4.23.



Gambar 4.23 Diagram Aktivitas Melaporkan Komentar

4.4.23. Menampilkan Daftar Laporan *User*

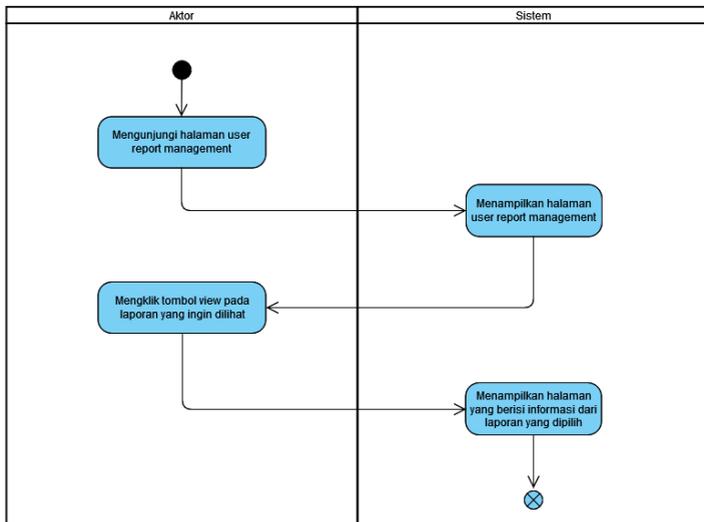
Diagram aktivitas dari kebutuhan menampilkan daftar laporan *user* dapat dilihat pada Gambar 4.24.



Gambar 4.24 Diagram Aktivitas Melihat Daftar Laporan User

4.4.24. **Melihat Sebuah Laporan *User***

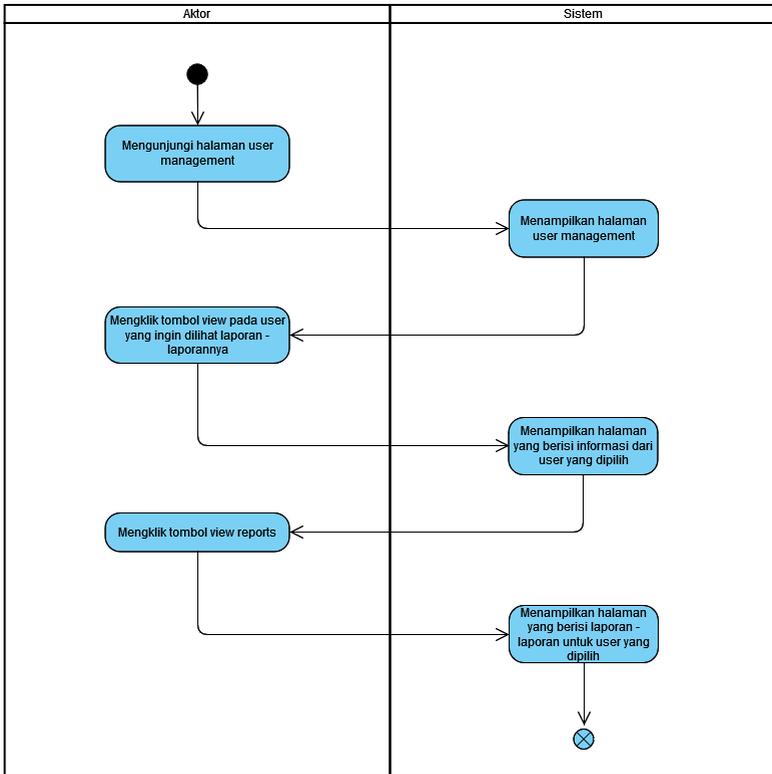
Diagram aktivitas dari kebutuhan melihat sebuah laporan *user* dapat dilihat pada Gambar 4.25.



Gambar 4.25 Diagram Aktivitas Melihat Sebuah Laporan User

4.4.25. Melihat Semua Laporan untuk Satu *User*

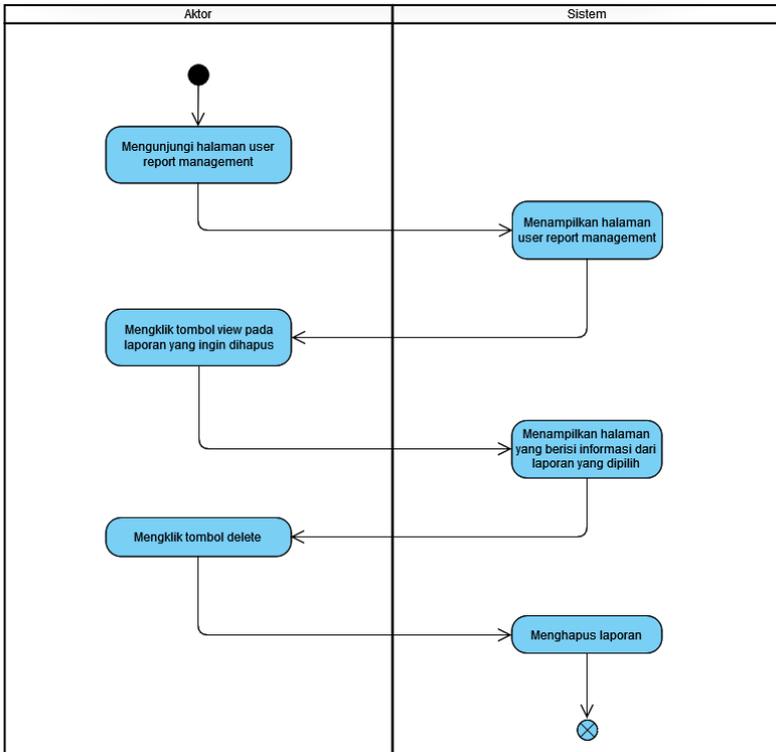
Diagram aktivitas dari kebutuhan melihat semua laporan untuk satu *user* dapat dilihat pada Gambar 4.26.



Gambar 4.26 Diagram Aktivitas Melihat Semua Laporan untuk Satu User

4.4.26. Menghapus Sebuah Laporan *User*

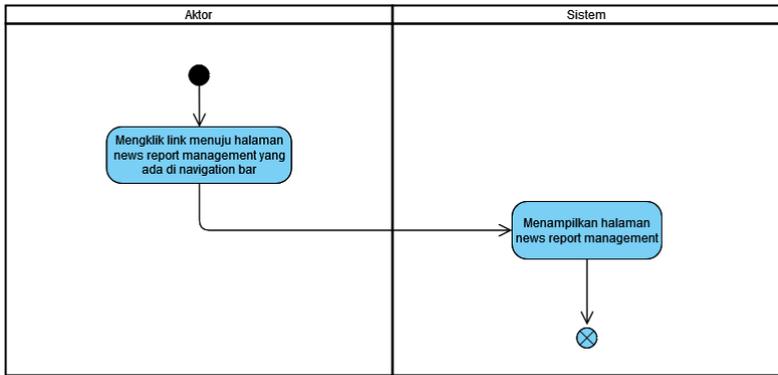
Diagram aktivitas dari kebutuhan menghapus sebuah laporan *user* dapat dilihat pada Gambar 4.27.



Gambar 4.27 Diagram Aktivitas Menghapus Sebuah Laporan User

4.4.27. Menampilkan Daftar Laporan Berita

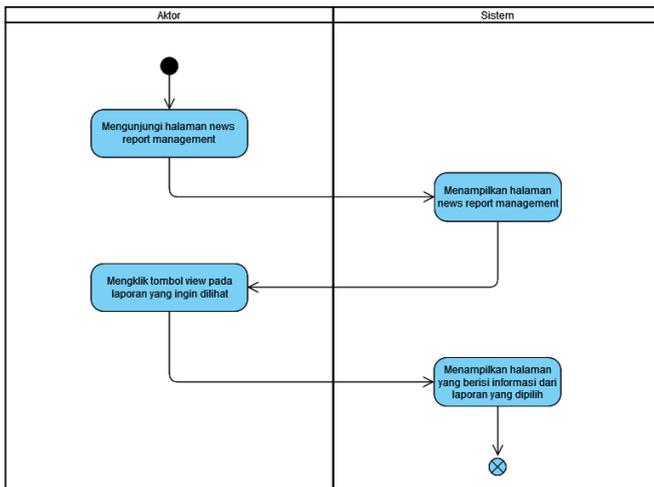
Diagram aktivitas dari kebutuhan menampilkan daftar laporan berita dapat dilihat pada Gambar 4.28.



Gambar 4.28 Diagram Aktivitas Menampilkan Daftar Laporan Berita

4.4.28. Melihat Sebuah Laporan Berita

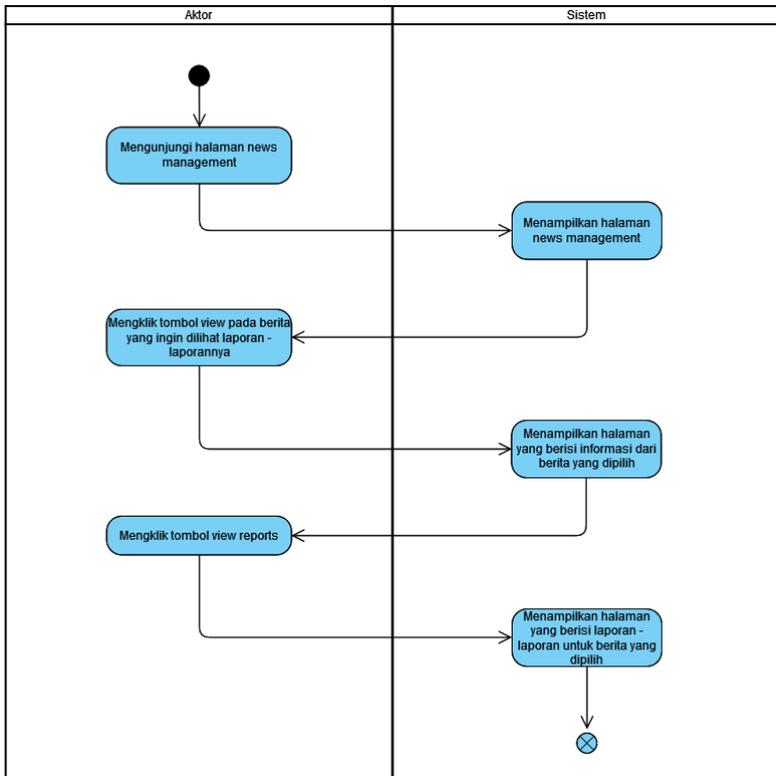
Diagram aktivitas dari kebutuhan melihat sebuah laporan berita dapat dilihat pada Gambar 4.29.



Gambar 4.29 Diagram Aktivitas Melihat Sebuah Laporan Berita

4.4.29. Melihat Semua Laporan untuk Satu Berita

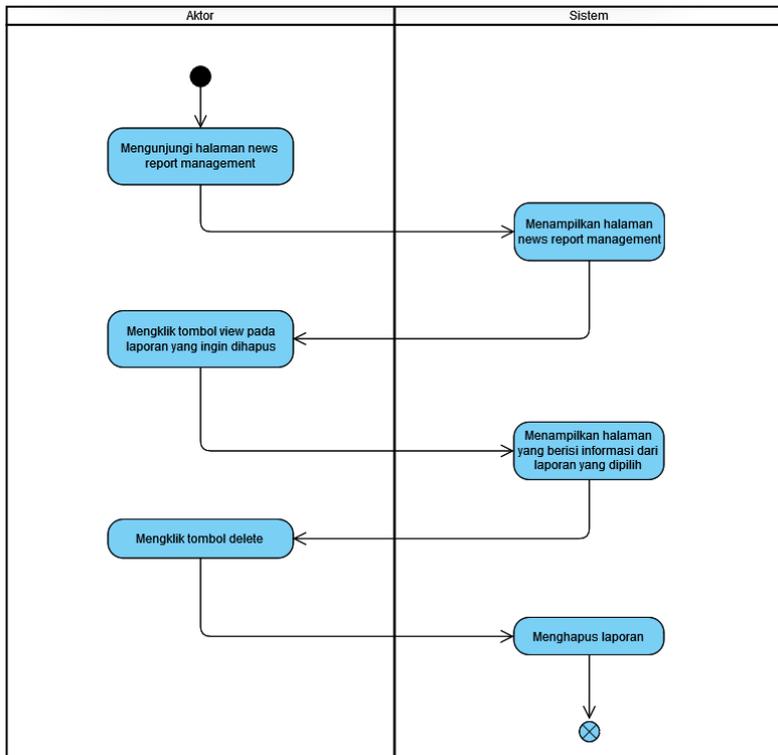
Diagram aktivitas dari kebutuhan melihat semua laporan untuk satu berita dapat dilihat pada Gambar 4.30.



Gambar 4.30 Diagram Aktivitas Melihat Semua Laporan untuk Satu Berita

4.4.30. Menghapus Sebuah Laporan Berita

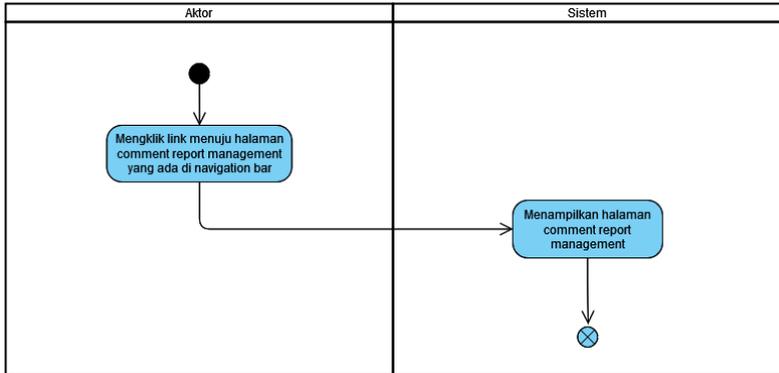
Diagram aktivitas dari kebutuhan menghapus sebuah laporan berita dapat dilihat pada Gambar 4.31.



Gambar 4.31 Diagram Aktivitas Menghapus Sebuah Laporan Berita

4.4.31. Menampilkan Daftar Laporan Komentar

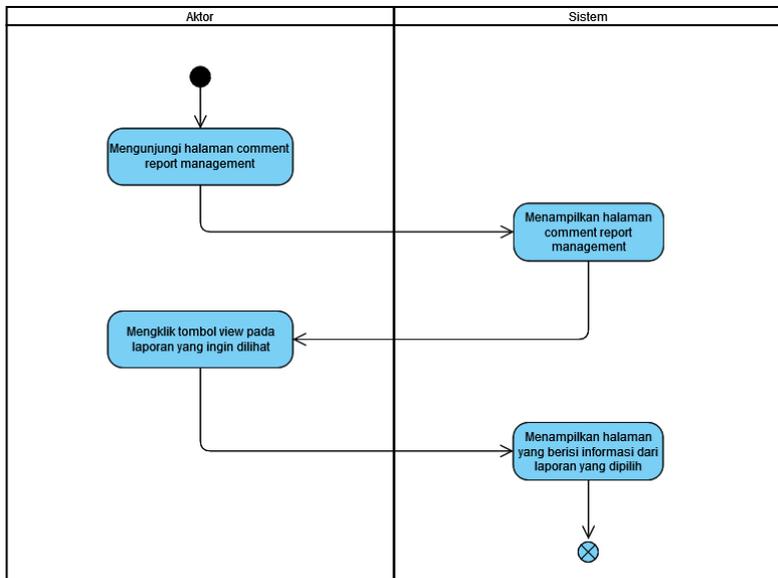
Diagram aktivitas dari kebutuhan menampilkan daftar laporan komentar dapat dilihat pada Gambar 4.32.



Gambar 4.32 Diagram Aktivitas Menampilkan Daftar Laporan Komentar

4.4.32. Melihat Sebuah Laporan Komentar

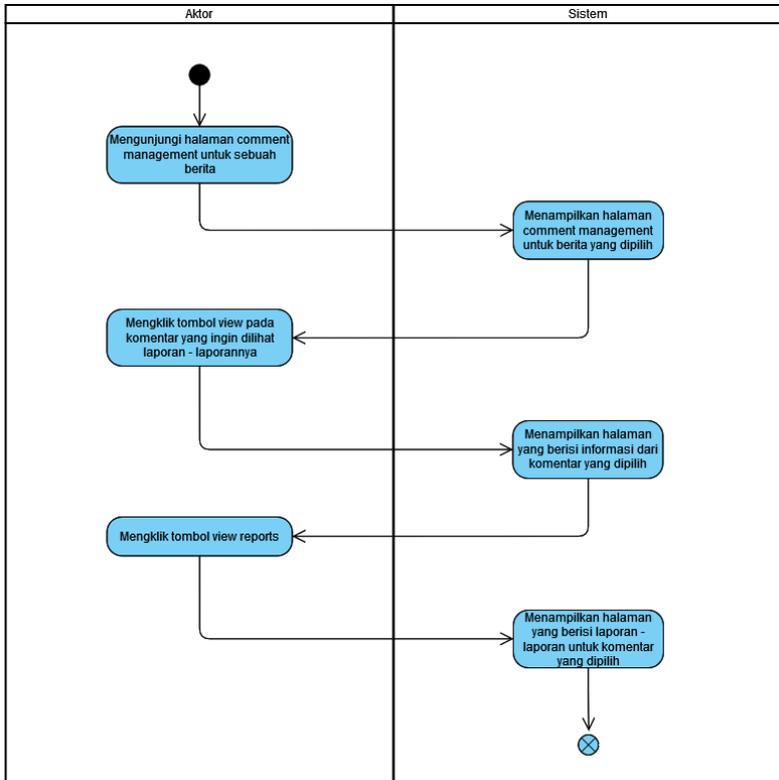
Diagram aktivitas dari kebutuhan melihat sebuah laporan komentar dapat dilihat pada Gambar 4.33.



Gambar 4.33 Diagram Aktivitas Melihat Sebuah Laporan Komentar

4.4.33. Melihat Semua Laporan untuk Satu Komentar

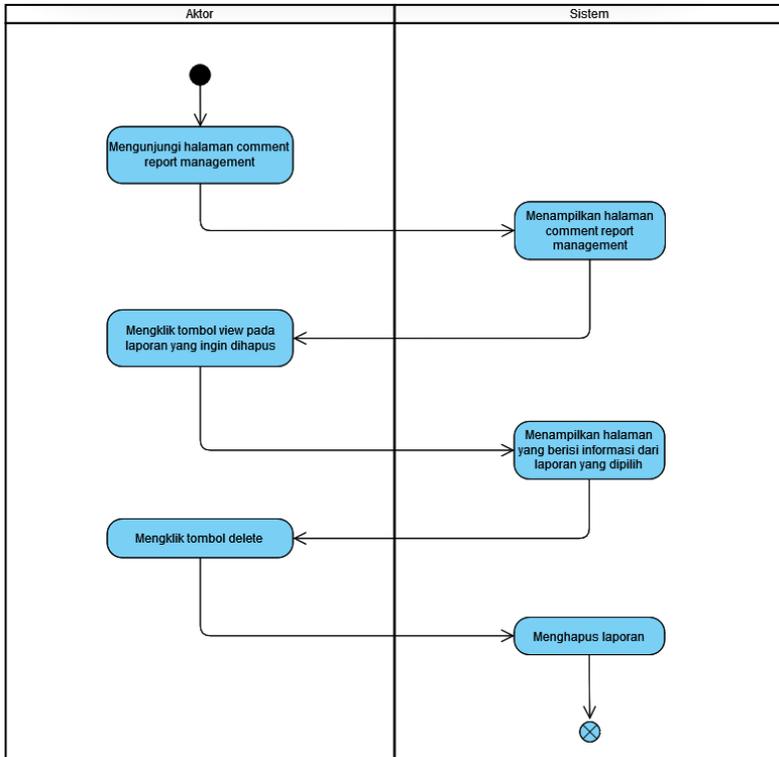
Diagram aktivitas dari kebutuhan melihat semua laporan untuk satu komentar dapat dilihat pada Gambar 4.34.



Gambar 4.34 Diagram Aktivitas Melihat Semua Laporan untuk Satu Komentar

4.4.34. Menghapus Sebuah Laporan Komentar

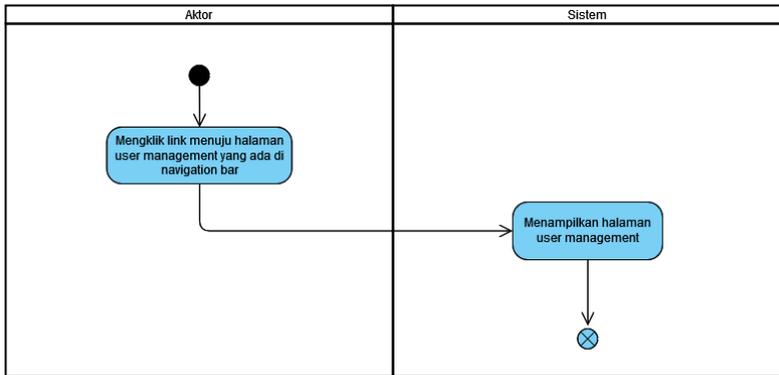
Diagram aktivitas dari kebutuhan menghapus sebuah laporan komentar dapat dilihat pada Gambar 4.35.



Gambar 4.35 Diagram Aktivitas Menghapus Sebuah Laporan Komentar

4.4.35. Menampilkan Daftar *User*

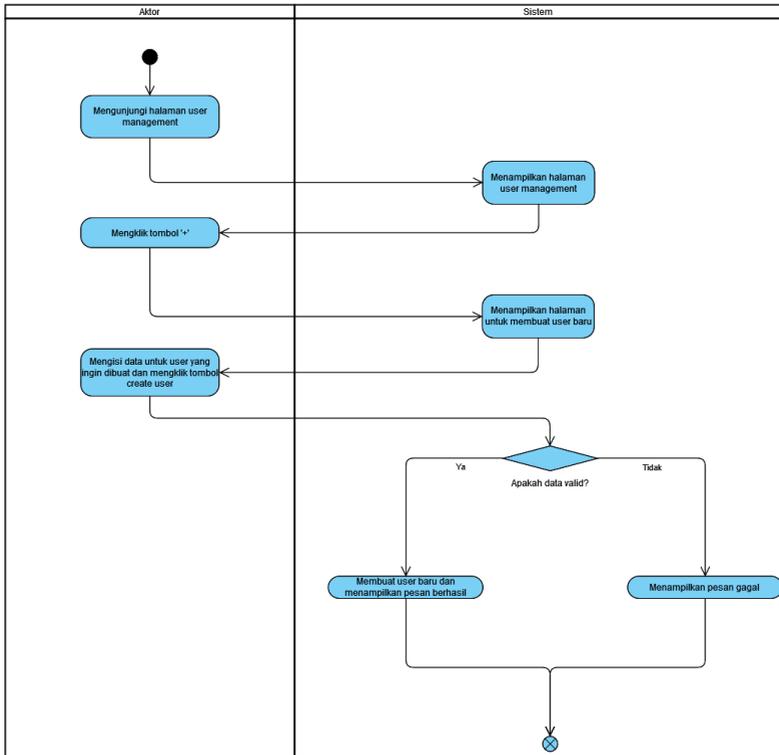
Diagram aktivitas dari kebutuhan menampilkan daftar *user* dapat dilihat pada Gambar 4.36.



Gambar 4.36 Diagram Aktivitas Menampilkan Daftar User

4.4.36. Membuat *User* Baru

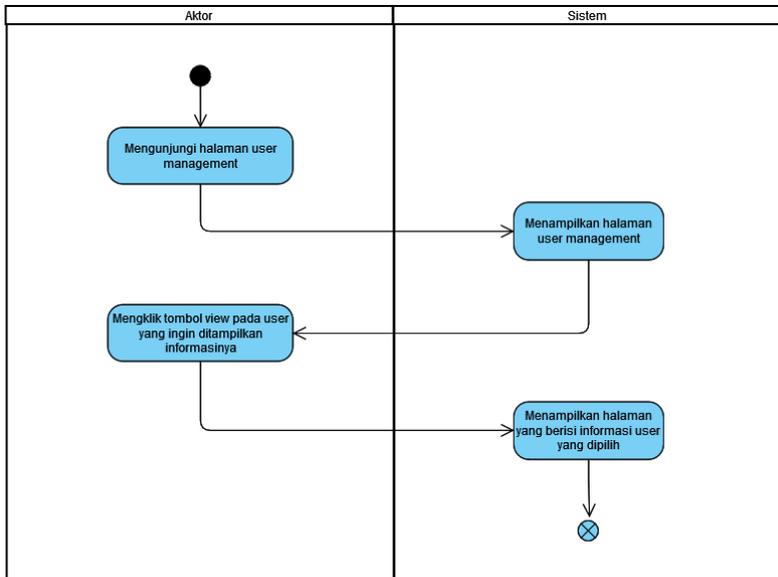
Diagram aktivitas dari kebutuhan membuat *user* baru dapat dilihat pada Gambar 4.37.



Gambar 4.37 Diagram Aktivitas Membuat User Baru

4.4.37. Menampilkan Informasi Sebuah *User*

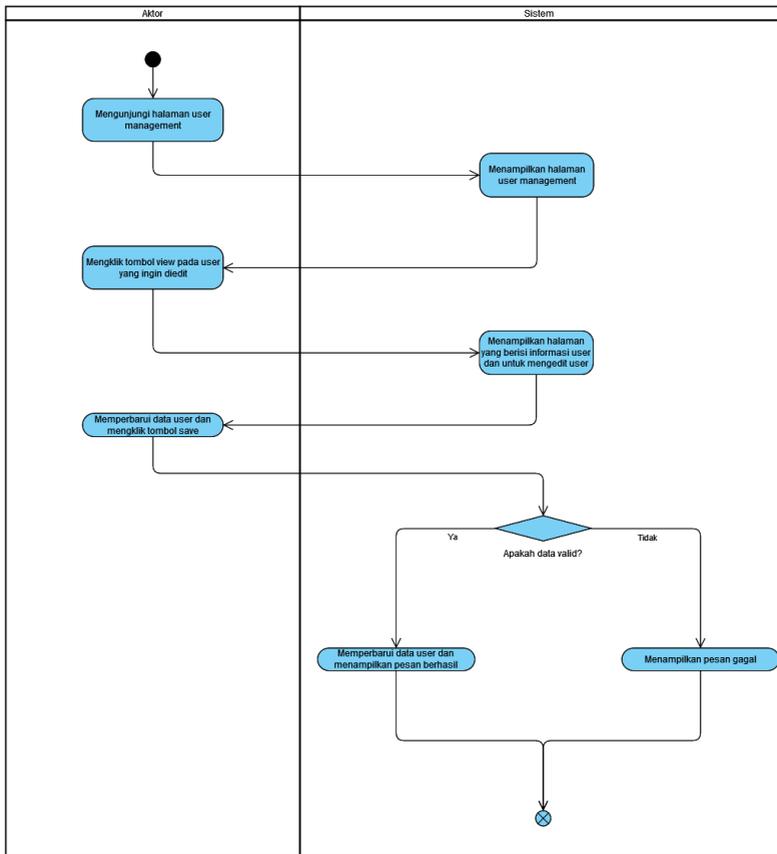
Diagram aktivitas dari kebutuhan menampilkan informasi sebuah *user* dapat dilihat pada Gambar 4.38.



Gambar 4.38 Diagram Aktivitas Menampilkan Informasi Sebuah User

4.4.38. **Memperbarui Data Sebuah User**

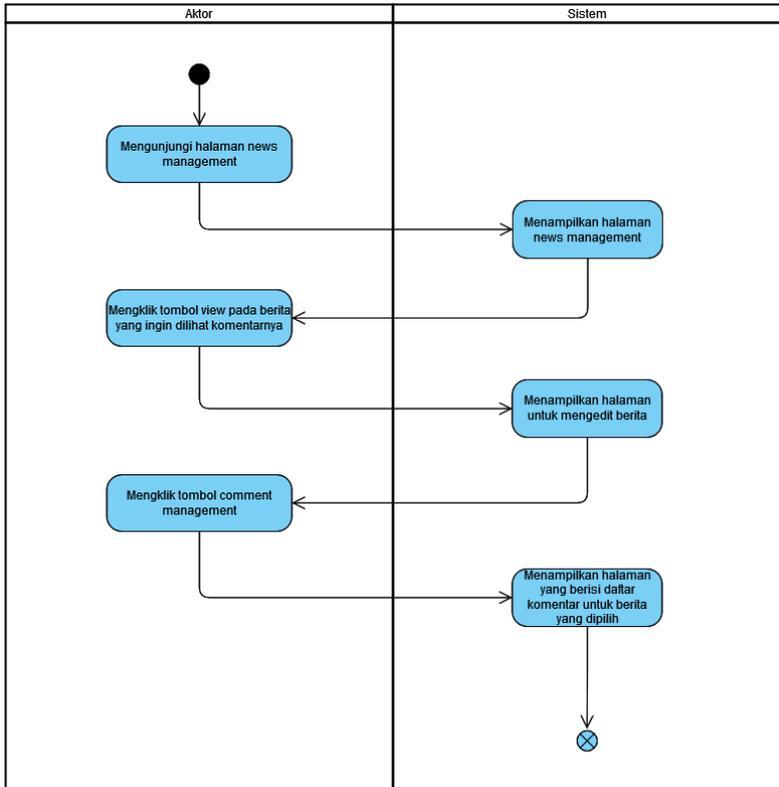
Diagram aktivitas dari kebutuhan memperbarui data sebuah *user* dapat dilihat pada Gambar 4.39.



Gambar 4.39 Diagram Aktivitas Memperbarui Data Sebuah User

4.4.39. Menampilkan Daftar Komentar untuk Sebuah Berita

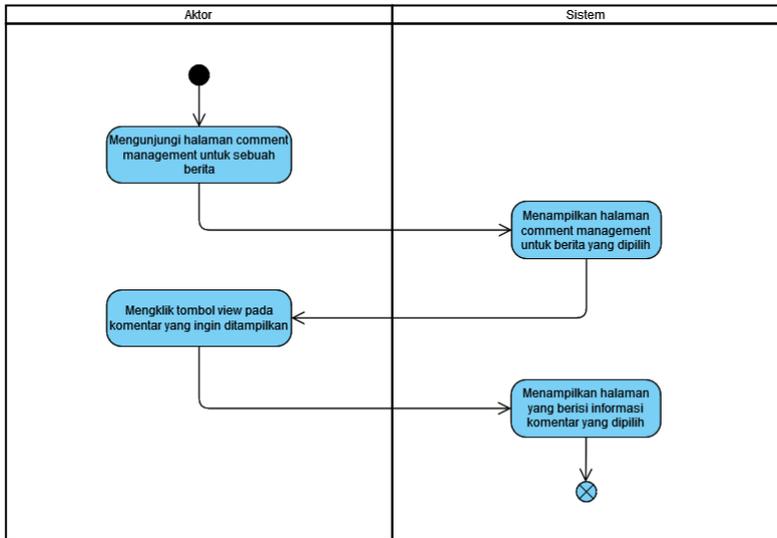
Diagram aktivitas dari kebutuhan menampilkan daftar komentar untuk sebuah berita dapat dilihat pada Gambar 4.40.



Gambar 4.40 Diagram Aktivitas Menampilkan Daftar Komentar untuk Sebuah Berita

4.4.40. Menampilkan Sebuah Komentar

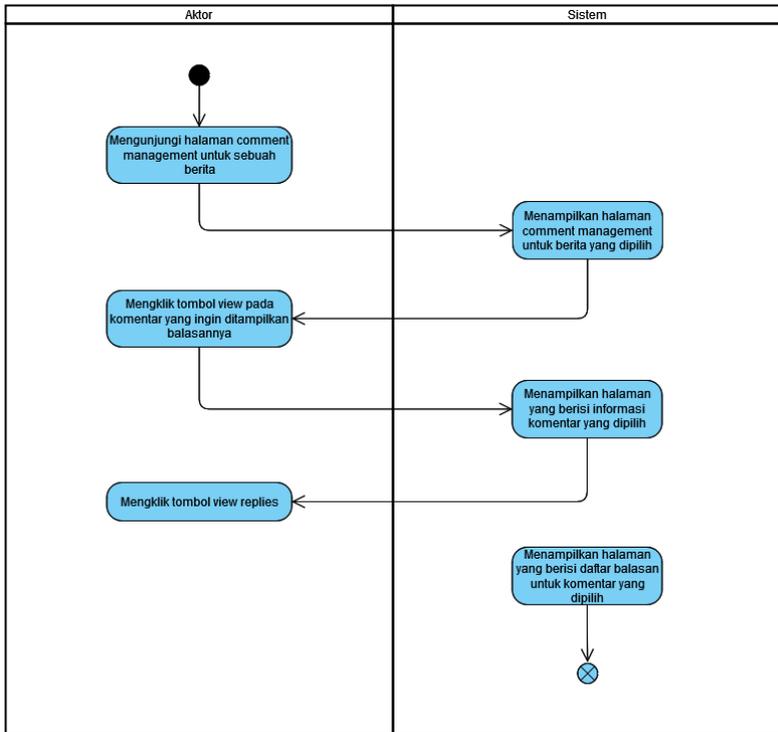
Diagram aktivitas dari kebutuhan menampilkan sebuah komentar dapat dilihat pada Gambar 4.41.



Gambar 4.41 Diagram Aktivitas Menampilkan Sebuah Komentar

4.4.41. Menampilkan Daftar Balasan untuk Sebuah Komentar

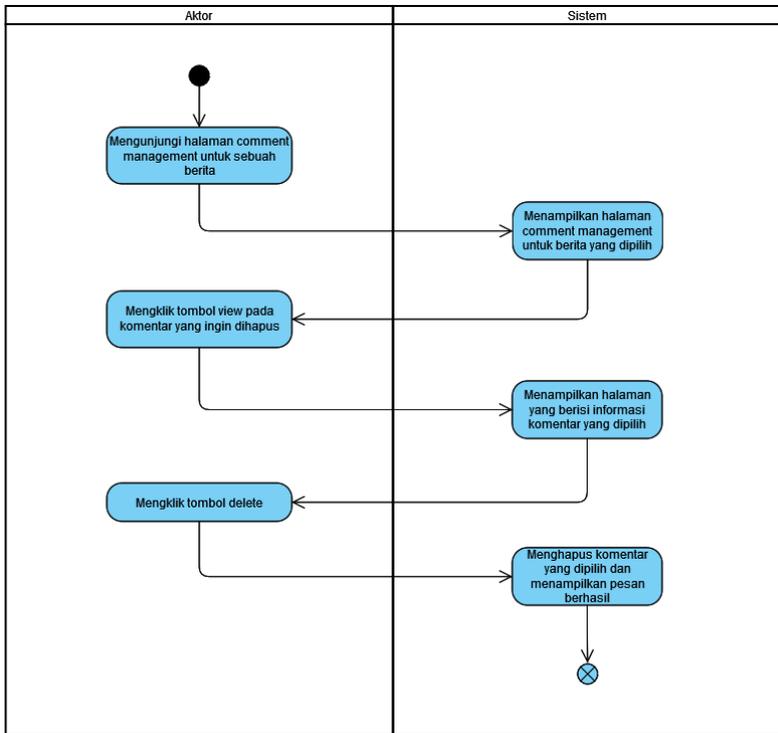
Diagram aktivitas dari kebutuhan menampilkan daftar balasan untuk sebuah komentar dapat dilihat pada Gambar 4.42.



Gambar 4.42 Diagram Aktivitas Menampilkan Daftar Balasan untuk Sebuah Komentar

4.4.42. Menghapus Komentar *User* Lain

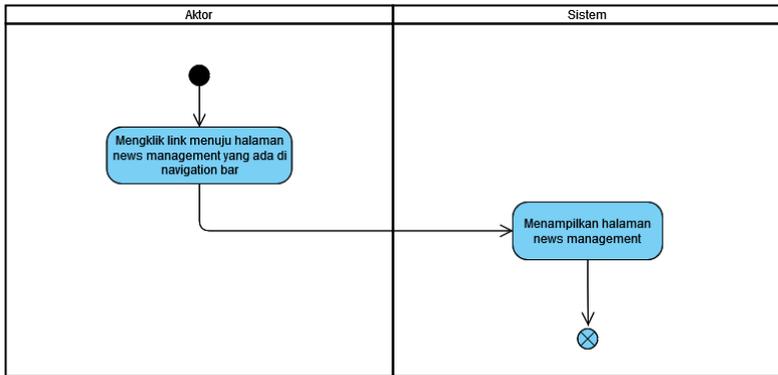
Diagram aktivitas dari kebutuhan menghapus komentar *user* lain dapat dilihat pada Gambar 4.43.



Gambar 4.43 Diagram Aktivitas Menghapus Komentar User Lain

4.4.43. Menampilkan Daftar Berita

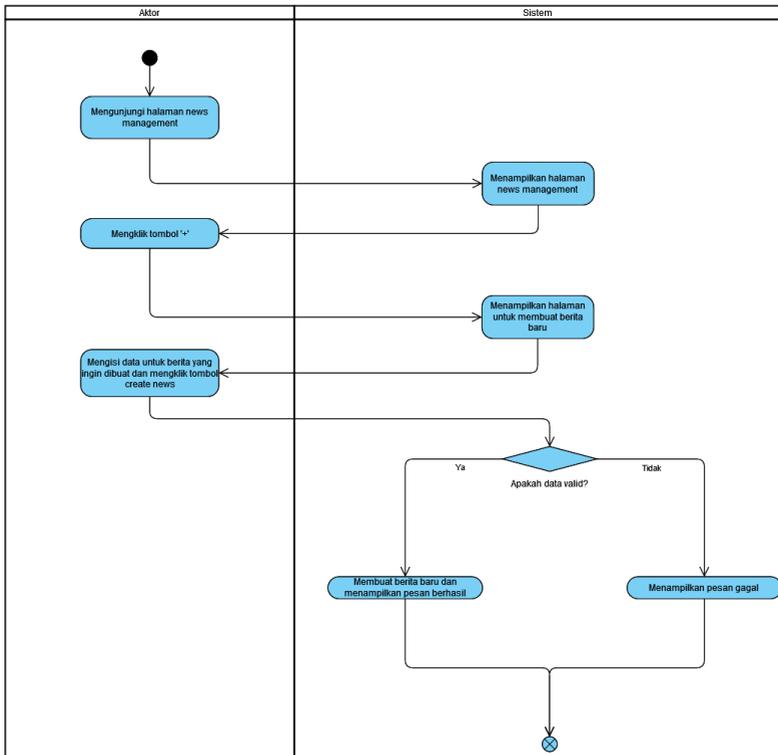
Diagram aktivitas dari kebutuhan menampilkan daftar berita dapat dilihat pada Gambar 4.44.



Gambar 4.44 Diagram Aktivitas Menampilkan Daftar Berita

4.4.44. Membuat Berita Baru

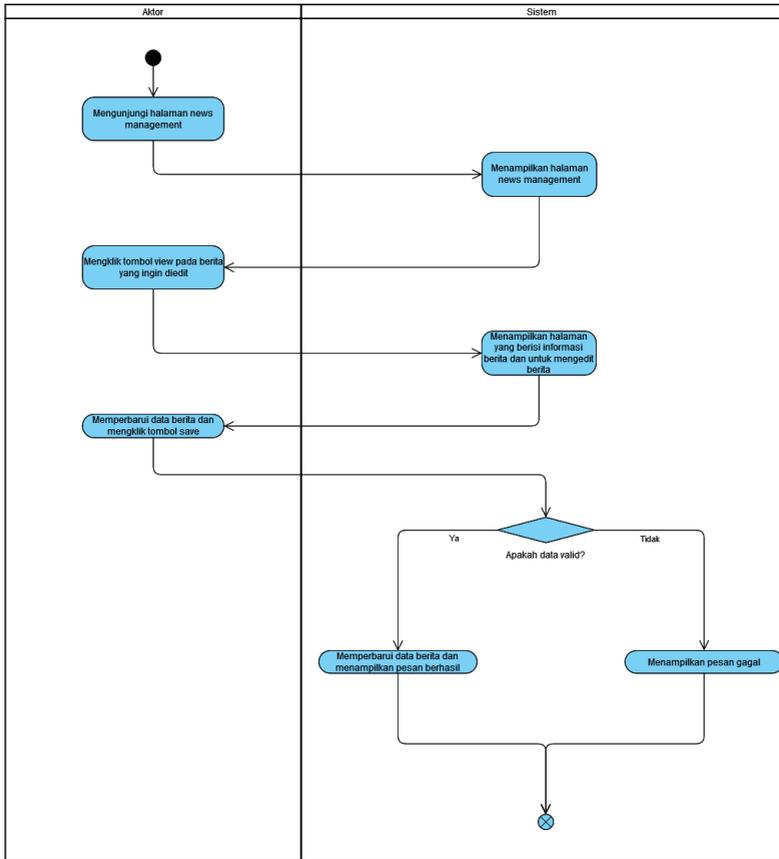
Diagram aktivitas dari kebutuhan membuat berita baru dapat dilihat pada Gambar 4.45.



Gambar 4.45 Diagram Aktivitas Membuat Berita Baru

4.4.45. Mengedit Sebuah Berita

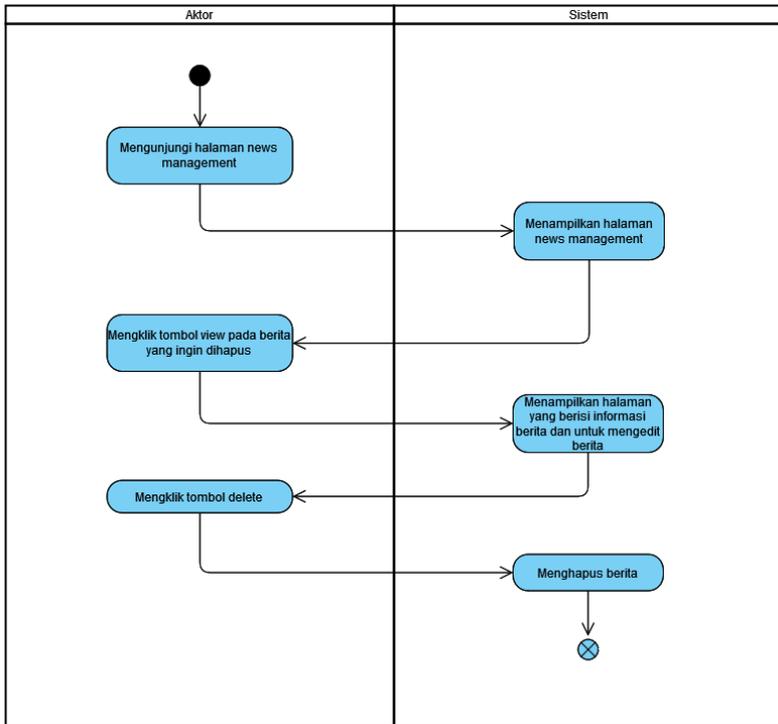
Diagram aktivitas dari kebutuhan mengedit sebuah berita dapat dilihat pada Gambar 4.46.



Gambar 4.46 Diagram Aktivitas Mengedit Sebuah Berita

4.4.46. Menghapus Sebuah Berita

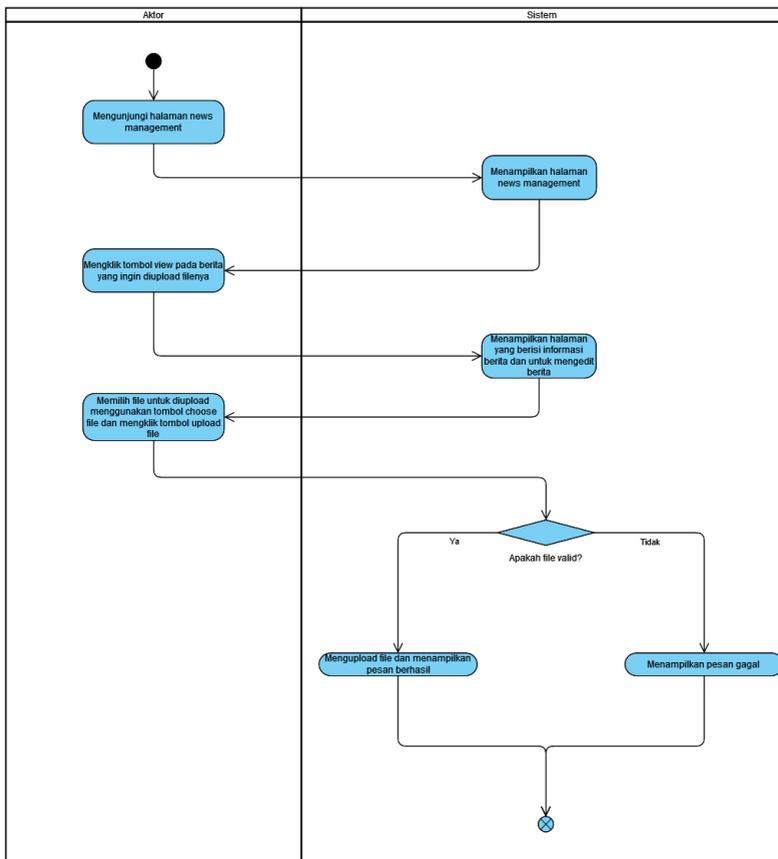
Diagram aktivitas dari kebutuhan menghapus sebuah berita dapat dilihat pada Gambar 4.47.



Gambar 4.47 Diagram Aktivitas Menghapus Sebuah Berita

4.4.47. Mengupload *File* untuk Sebuah Berita

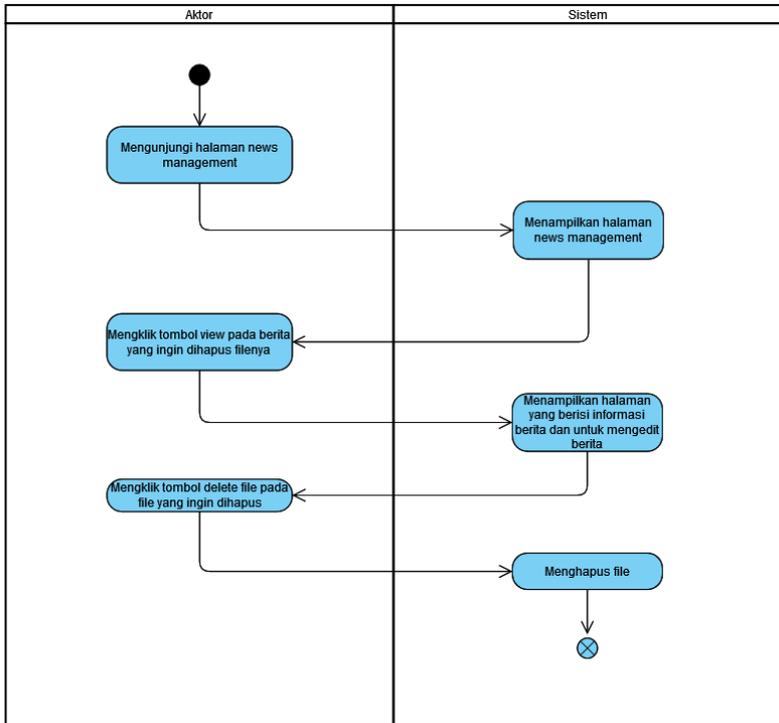
Diagram aktivitas dari kebutuhan menghapus sebuah berita dapat dilihat pada Gambar 4.48.



Gambar 4.48 Diagram Aktivitas Mengupload File untuk Sebuah Berita

4.4.48. Menghapus *File* yang Diupload untuk Sebuah Berita

Diagram aktivitas dari kebutuhan menghapus sebuah berita dapat dilihat pada Gambar 4.49.



Gambar 4.49 Diagram Aktivitas Menghapus File yang Diupload untuk Sebuah Berita

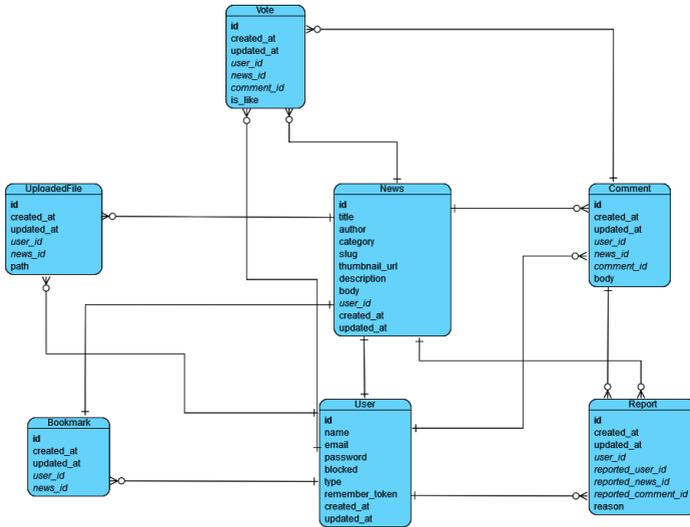
4.5. Desain Basis Data

Subbab ini menjelaskan tentang desain basis data dari aplikasi *website* portal berita yang terdiri dari desain konseptual data, desain fisik data, dan struktur dari setiap tabel *database*.

4.5.1. Desain Konseptual Data

Pembangunan *website* portal berita memerlukan *database* yang digunakan untuk menyimpan semua

data yang diperlukan. Desain konseptual dari *database* dari *website* portal berita yang dirancang dapat dilihat pada Gambar 4.50.



Gambar 4.50 Desain Konseptual Data Website Portal Berita

4.5.2. Desain Fisik Data

Berdasarkan desain konseptual data dapat dibuatkan desain fisik data yang nantinya akan diimplementasikan menggunakan MySQL. Desain fisik database dari *website* portal berita dapat dilihat pada Gambar 4.51.

3	email	varchar	255	<i>Email dari user.</i>
4	password	varchar	255	<i>Password dari user.</i>
5	blocked	tinyint	-	Status pemblokiran <i>user.</i>
6	type	varchar	255	Tipe dari <i>user.</i>
7	remember_token	varchar	100	<i>Remember token</i> yang digunakan oleh Laravel.
8	created_at	timestamp	-	Tanggal <i>user</i> dibuat.
9	updated_at	timestamp	-	Tanggal akun diperbarui.

4.5.4. Struktur Tabel *News*

Tabel *news* digunakan untuk menyimpan informasi dari sebuah artikel berita. Struktur tabel *news* dapat dilihat pada Tabel 4.51.

Tabel 4.51 Struktur Tabel *News*

No.	Nama Field	Tipe Data	Panjang	Keterangan
1	id	bigint	-	<i>Primary key</i> yang bersifat <i>auto increment</i> dari artikel berita.
2	created_at	timestamp	-	Tanggal berita dibuat.
3	updated_at	timestamp	-	Tanggal berita diperbarui.

4	user_id	bigint	-	<i>Foreign key</i> untuk <i>user</i> pemilik artikel berita.
5	title	varchar	256	Judul dari artikel berita.
6	author	varchar	256	Nama penulis artikel berita.
7	category	varchar	256	Kategori dari artikel berita.
8	slug	varchar	256	<i>News slug</i> dari artikel berita.
9	thumbnail_url	varchar	64	URL untuk <i>thumbnail</i> dari artikel berita.
10	description	text	-	Deskripsi singkat dari artikel berita.
11	body	text	-	Isi dari artikel berita.

4.5.5. Struktur Tabel *Uploaded_Files*

Tabel *uploaded files* digunakan untuk menyimpan informasi dari *file* yang diupload untuk sebuah artikel berita. Struktur tabel *uploaded files* dapat dilihat pada Tabel 4.52.

Tabel 4.52 Struktur Tabel *Uploaded Files*

No.	Nama Field	Tipe Data	Panjang	Keterangan
1	id	bigint	-	<i>Primary key</i> yang bersifat <i>auto increment</i> untuk <i>file</i> .

2	created_at	timestamp	-	Tanggal <i>file</i> dibuat.
3	updated_at	timestamp	-	Dibuat secara otomatis oleh Laravel tetapi tidak digunakan.
4	user_id	bigint	-	<i>Foreign key</i> untuk <i>user</i> pemilik <i>file</i> .
5	news_id	bigint	-	<i>Foreign key</i> untuk berita yang mempunyai <i>file</i> .
6	path	varchar	64	URL untuk <i>file</i> yang diupload.

4.5.6. Struktur Tabel *Comments*

Tabel *comments* digunakan untuk menyimpan informasi dari setiap komentar. Struktur tabel *comments* dapat dilihat pada Tabel 4.53.

Tabel 4.53 Struktur Tabel *Comments*

No.	Nama Field	Tipe Data	Panjang	Keterangan
1	id	bigint	-	<i>Primary key</i> yang bersifat <i>auto increment</i> untuk komentar.
2	created_at	timestamp	-	Tanggal komentar dibuat.
3	updated_at	timestamp	-	Tanggal komentar diperbarui.
4	user_id	bigint	-	<i>Foreign key</i> dari <i>user</i> pemilik komentar.

5	news_id	bigint	-	<i>Foreign key</i> untuk berita di mana komentar dibuat.
6	comment_id	bigint	-	<i>Foreign key</i> dari komentar yang dibalas.
7	body	text	-	Isi dari komentar.

4.5.7. Struktur Tabel *Votes*

Tabel *votes* digunakan untuk menyimpan informasi dari setiap *vote*. Struktur tabel *votes* dapat dilihat pada Tabel 4.54.

Tabel 4.54 Struktur Tabel *Votes*

No.	Nama Field	Tipe Data	Panjang	Keterangan
1	id	bigint	-	<i>Primary key</i> yang bersifat <i>auto increment</i> dari <i>vote</i> .
2	created_at	timestamp	-	Tanggal <i>vote</i> dibuat.
3	updated_at	timestamp	-	Dibuat secara otomatis oleh Laravel tetapi tidak digunakan.
4	user_id	bigint	-	<i>Foreign key</i> dari <i>user</i> pemilik <i>vote</i> .
5	news_id	bigint	-	<i>Foreign key</i> dari berita yang diberikan <i>vote</i> .
6	comment_id	bigint	-	<i>Foreign key</i> dari komentar yang diberikan <i>vote</i> .

7	is_like	tinyint	-	Tipe dari <i>vote</i> .
---	---------	---------	---	-------------------------

4.5.8. Struktur Tabel *Bookmarks*

Tabel *bookmarks* digunakan untuk menyimpan informasi dari setiap *bookmark* yang dimiliki oleh *user*. Struktur tabel *bookmarks* dapat dilihat pada Tabel 4.55.

Tabel 4.55 Struktur Tabel *Bookmarks*

No.	Nama Field	Tipe Data	Panjang	Keterangan
1	id	bigint	-	<i>Primary key</i> yang bersifat <i>auto increment</i> dari <i>bookmark</i> .
2	created_at	timestamp	-	Tanggal <i>bookmark</i> dibuat.
3	updated_at	timestamp	-	Dibuat secara otomatis oleh Laravel tetapi tidak digunakan.
4	user_id	bigint	-	<i>Foreign key</i> dari <i>user</i> pemilik <i>bookmark</i> .
5	news_id	bigint	-	<i>Foreign key</i> dari berita yang ditambahkan ke <i>bookmarks</i> .

4.5.9. Struktur Tabel *Reports*

Tabel *reports* digunakan untuk menyimpan informasi dari setiap laporan yang dibuat *user*. Struktur tabel *reports* dapat dilihat pada Tabel 4.56.

Tabel 4.56 Struktur Tabel Reports

No.	Nama Field	Tipe Data	Panjang	Keterangan
1	id	bigint	-	<i>Primary key</i> yang bersifat <i>auto increment</i> dari laporan.
2	created_at	timestamp	-	Tanggal laporan dibuat.
3	updated_at	timestamp	-	Dibuat secara otomatis oleh Laravel tetapi tidak digunakan.
4	user_id	bigint	-	<i>Foreign key</i> dari <i>user</i> pembuat laporan.
5	reported_user_id	bigint	-	<i>Foreign key</i> dari <i>user</i> yang dilaporkan.
6	reported_news_id	bigint	-	<i>Foreign key</i> dari berita yang dilaporkan.
7	reported_comment_id	bigint	-	<i>Foreign key</i> dari komentar yang dilaporkan.

[Halaman ini sengaja dikosongkan]

BAB V IMPLEMENTASI SISTEM

Bab ini membahas tentang implementasi dari aplikasi *website* portal berita yang dibuat. Implementasi dibagi menjadi dua bagian, yaitu implementasi *controller* dan implementasi *user interface*.

5.1. Implementasi *Controller*

Implementasi *controller* berisi logika yang digunakan oleh fitur-fitur yang terdapat pada *website* portal berita seperti untuk menampilkan *homepage*, melakukan pencarian, melakukan *login*, dan lain-lain.

5.1.1 RegisteredUserController

RegisteredUserController adalah *controller* yang berfungsi untuk menangani registrasi *user* baru. Kode sumber untuk RegisteredUserController dapat dilihat pada Kode Sumber 5.1.

```
1. <?php
2.
3. namespace App\Http\Controllers\Auth;
4.
5. use App\Http\Controllers\Controller;
6. use App\Models\User;
7. use Illuminate\Auth\Events\Registered;
8. use Illuminate\Http\Request;
9. use Illuminate\Http\Response;
10. use Illuminate\Support\Facades\Auth;
11. use Illuminate\Support\Facades\Hash;
12. use Illuminate\Validation\Rules;
13.
14. class RegisteredUserController extends Controller
15. {
16.     public function store(Request $request)
```

```

17.     {
18.         $request->validate([
19.             'name' => ['required', 'string',
20.                 'max:255'], 'email' => ['required', 'string',
21.                 'lowercase', 'email', 'max:255',
22.                 'unique:'.User::class],
23.             'password' => ['required', 'confirmed',
24.                 Rules\Password::defaults()],
25.         ]);
26.         $user = User::create([
27.             'name' => $request->name,
28.             'email' => $request->email,
29.             'password' => Hash::make($request-
30. >password),
31.         ]);
32.         event(new Registered($user));
33.         Auth::login($user);
34.         return Redirect('/dashboard')-
35. >with('success', 'Registration successful!');
36.     }

```

Kode Sumber 5.1 Implementasi RegisteredUserController

5.1.2 AuthenticatedSessionController

AuthenticatedSessionController adalah *controller* yang berfungsi untuk menangani *login* dan *logout* dari *user*. Kode sumber untuk AuthenticatedSessionController dapat dilihat pada Kode Sumber 5.2.

```

1. <?php
2.
3. namespace App\Http\Controllers\Auth;
4.
5. use App\Http\Controllers\Controller;

```

```

6. use App\Http\Requests\Auth\LoginRequest;
7. use Illuminate\Http\Request;
8. use Illuminate\Http\Response;
9. use Illuminate\Support\Facades\Auth;
10.
11. class AuthenticatedSessionController extends
    Controller
12. {
13.     public function store(LoginRequest $request)
14.     {
15.         $request->authenticate();
16.
17.         $request->session()->regenerate();
18.
19.         return Redirect()->intended('/dashboard')-
    >with('success', 'Logged in successfully!');
20.     }
21.
22.     public function destroy(Request $request)
23.     {
24.         Auth::guard('web')->logout();
25.
26.         $request->session()->invalidate();
27.
28.         $request->session()->regenerateToken();
29.
30.         return Redirect('/')->with('success', 'Logged
    out successfully!');
31.     }
32. }

```

Kode Sumber 5.2 Implementasi AuthenticatedSessionController

5.1.3 PasswordResetLinkController

PasswordResetLinkController adalah *controller* yang berfungsi untuk menangani *request* untuk mengirimkan *email* yang berisikan *link* untuk mengubah *password* jika seorang *user* lupa *password* dari akun yang dimilikinya. Kode sumber untuk PasswordResetLinkController dapat dilihat pada Kode Sumber 5.3.

```

1. <?php
2.
3. namespace App\Http\Controllers\Auth;
4.
5. use App\Http\Controllers\Controller;
6. use Illuminate\Http\JsonResponse;
7. use Illuminate\Http\Request;
8. use Illuminate\Support\Facades>Password;
9. use Illuminate\Validation\ValidationException;
10.
11. class PasswordResetLinkController extends Controller
12. {
13.     public function index() {
14.         return view('auth.forgotPassword');
15.     }
16.
17.     public function store(Request $request)
18.     {
19.         $request->validate([
20.             'email' => ['required', 'email'],
21.         ]);
22.
23.         $status = Password::sendResetLink(
24.             $request->only('email')
25.         );
26.
27.         if ($status != Password::RESET_LINK_SENT) {
28.             throw ValidationException::withMessages([
29.                 'email' => [__( $status)],
30.             ]);
31.         }
32.
33.         return Redirect('/login')->with('success',
34.             __( $status));
35.     }

```

Kode Sumber 5.3 Implementasi PasswordResetLinkController

5.1.4 NewPasswordController

NewPasswordController adalah *controller* yang berfungsi untuk menangani *request* untuk mengubah *password* setelah sebelumnya melakukan *request* untuk melakukan *reset password*. Kode sumber untuk

NewPasswordController dapat dilihat pada Kode Sumber 5.4.

```
1. <?php
2.
3. namespace App\Http\Controllers\Auth;
4.
5. use App\Http\Controllers\Controller;
6. use Illuminate\Auth\Events>PasswordReset;
7. use Illuminate\Http\JsonResponse;
8. use Illuminate\Http\Request;
9. use Illuminate\Support\Facades\Hash;
10. use Illuminate\Support\Facades>Password;
11. use Illuminate\Support\Str;
12. use Illuminate\Validation\Rules;
13. use Illuminate\Validation\ValidationException;
14.
15. class NewPasswordController extends Controller
16. {
17.     public function index(Request $request, String
18.     $token) {
19.         if (!isset($request->email))
20.             return Redirect('/')->with('error',
21.             'Email parameter is empty');
22.         return view('auth.resetPassword', ['token' =>
23.         $token, 'email' => $request->email]);
24.     }
25.     public function store(Request $request)
26.     {
27.         $request->validate([
28.             'token' => ['required'],
29.             'email' => ['required', 'email'],
30.             'password' => ['required', 'confirmed',
31.             Rules\Password::defaults()],
32.         ]);
33.         $status = Password::reset(
34.             $request->only('email', 'password',
35.             'password_confirmation', 'token'),
36.             function ($user) use ($request) {
37.                 $user->forceFill([
38.                     'password' =>
39.                     Hash::make($request->password),
```

```

37.         'remember_token' =>
38.         Str::random(60),
39.         ])->save();
40.         event(new PasswordReset($user));
41.     }
42. );
43.
44.     if ($status != Password::PASSWORD_RESET) {
45.         throw ValidationException::withMessages([
46.             'email' => [__( $status)],
47.         ]);
48.     }
49.
50.     return Redirect('/login')->with('success',
51.         __( $status));
52. }

```

Kode Sumber 5.4 Implementasi NewPasswordController

5.1.5 DashboardController

DashboardController berfungsi untuk menampilkan *dashboard*, menangani perubahan data akun, dan menangani *request* untuk menghapus akun. Kode sumber untuk DashboardController dapat dilihat pada Kode Sumber 5.5.

```

1. <?php
2.
3. namespace App\Http\Controllers;
4.
5. use App\Models\Bookmark;
6. use App\Models\Comment;
7. use App\Models\News;
8. use App\Models\Report;
9. use App\Models\UploadedFile;
10. use App\Models\Vote;
11. use App\Utilities;
12. use Illuminate\Http\Request;
13. use Illuminate\Support\Facades\Auth;

```

```

14. use Illuminate\Support\Facades\Hash;
15. use Illuminate\Validation\Rules\Password;
16.
17. class DashboardController extends Controller
18. {
19.     public function index() {
20.         return view('dashboard.index', ['user' =>
Auth::user()]);
21.     }
22.
23.     public function update(Request $request) {
24.         $request->validate(
25.             [ 'name' => 'required' ],
26.             [ 'name' => 'Name is required' ]
27.         );
28.
29.         $user = Auth::user();
30.         $user->name = $request->name;
31.
32.         if ($request->current_password != '' &&
33.             $request->password != '' &&
34.             $request->password_confirmation != ''
35.         ) {
36.             if (!Hash::check($request-
>current_password, $user->password)) {
37.                 return Redirect()->back()-
>with('error', 'Incorrect password');
38.             }
39.
40.             $user->password = Hash::make($request-
>password);
41.         }
42.
43.         $user->save();
44.
45.         return Redirect()->back()->with('success',
'Account updated successfully');
46.     }
47.
48.     public function delete(Request $request) {
49.         $request->validate(
50.             [ 'password' => 'required' ],
51.             [ 'password.required' => 'Incorrect
password!' ]
52.         );
53.
54.         $user = Auth::user();
55.
56.         $del = Bookmark::where('user_id', '=', $user-
>id)->get();

```

```

57.
58.     foreach ($del as $delete)
59.         $delete->delete();
60.
61.     $del = Comment::where('user_id', '=', $user-
>id)->get();
62.
63.     foreach ($del as $delete)
64.         Utilities::deleteComment($delete->id);
65.
66.     $del = News::where('user_id', '=', $user-
>id)->get();
67.
68.     foreach ($del as $delete)
69.         Utilities::deleteNews($delete->id);
70.
71.     $del = Report::where('user_id', '=', $user-
>id)->get();
72.
73.     foreach ($del as $delete)
74.         $delete->delete();
75.
76.     $del = Report::where('reported_user_id', '=',
$user->id)->get();
77.
78.     foreach ($del as $delete)
79.         $delete->delete();
80.
81.     $del = Vote::where('user_id', '=', $user-
>id)->get();
82.
83.     foreach ($del as $delete)
84.         $delete->delete();
85.
86.     Auth::guard('web')->logout();
87.     $request->session()->invalidate();
88.     $request->session()->regenerateToken();
89.
90.     $user->delete();
91.
92.     return Redirect('/login')->with('success',
'Account deleted successfully!');
93. }
94. }

```

Kode Sumber 5.5 Implementasi DashboardController

5.1.6 NewsController

NewsController berfungsi untuk menampilkan *homepage*, menangani *request* pencarian, menampilkan halaman berita, menangani fitur *vote* untuk berita, dan menangani fitur-fitur *bookmarks*. Kode sumber untuk NewsController dapat dilihat pada Kode Sumber 5.6.

```
1. <?php
2.
3. namespace App\Http\Controllers;
4.
5. use App\Models\Bookmark;
6. use App\Models\Comment;
7. use App\Models\News;
8. use App\Models\Vote;
9. use App\Utilities;
10. use Illuminate\Http\Request;
11. use Illuminate\Support\Facades\Auth;
12. use Illuminate\Support\Str;
13.
14. class NewsController extends Controller
15. {
16.     public function index() {
17.         $mostRecent = News::orderByDesc('id')-
18. >limit(5)->get();
19.
20.         $news = [
21.             ['Most recent news', $mostRecent]
22.         ];
23.
24.         foreach (array('culinary', 'sports',
25. 'politics', 'economy', 'automotive', 'technology',
26. 'health') as $value) {
27.             $news[] = [
28.                 'Most recent from ' . $value,
29.                 News::where('category', '=', $value)-
30. >orderByDesc('id')->limit(5)->get()
31.             ];
32.         }
33.
34.         return view('home', ['news' => $news]);
35.     }
36. }
```

```

33.     public function search(Request $request) {
34.         $category = $request->category ?? '';
35.
36.         $news = News::where('category', 'like',
37.             "%$category%");
38.
39.         $news = Utilities::sortedSearchLike(
40.             'title', $request->search,
41.             ['id', 'title', 'author', 'category',
42.             'created_at'],
43.             $request->sort, $request->sortOrder,
44.             'id',
45.             $news, false
46.         );
47.         $paginationURLs =
48.         Utilities::createPaginationURLs($request, $news);
49.
50.         return view(
51.             'news.search',
52.             [
53.                 'news' => $news,
54.                 'prevPageUrl' =>
55.                 $paginationURLs['prevPageUrl'],
56.                 'nextPageUrl' =>
57.                 $paginationURLs['nextPageUrl'],
58.             ]
59.         );
60.     }
61.
62.     public function get(Request $request, String
63.         $slug) {
64.         $news = News::where('slug', '=', $slug)-
65.             >firstOrFail();
66.
67.         $comments = Comment::where('news_id', '=',
68.             $news->id)
69.             ->where('comment_id', '=', 0)
70.             ->orderByDesc('id')
71.             ->paginate(20);
72.
73.         $newsBody = Str::markdown($news->body);
74.         $newsBody = str_replace(['<', '>', '"'],
75.             ['%3c', '%3e', '%22'], $newsBody);
76.
77.         $userVote = Vote::where('user_id', '=',
78.             Auth::id())->where('news_id', '=', $news->id)-
79.             >first();
80.
81.         return view(

```

```

71.         'news.show',
72.         [
73.             'news' => $news,
74.             'comments' => $comments,
75.             'newsBody' => $newsBody,
76.         ]
77.     );
78. }
79.
80.     public function like(String $slug) {
81.         $news = News::where('slug', '=', $slug)-
>firstOrFail();
82.
83.         $vote = Vote::where('news_id', '=', $news-
>id)->where('user_id', '=', Auth::id()->get()-
>first();
84.
85.         if ($vote == null) {
86.             $vote = new Vote();
87.
88.             $vote->user_id = Auth::id();
89.             $vote->news_id = intval($news->id);
90.             $vote->comment_id = 0;
91.         }
92.
93.         $vote->is_like = true;
94.
95.         $vote->save();
96.
97.         return Redirect("/$slug");
98.     }
99.
100.     public function dislike(String $slug) {
101.         $news = News::where('slug', '=',
$slug)->firstOrFail();
102.
103.         $vote = Vote::where('news_id', '=',
$news->id)->where('user_id', '=', Auth::id()->get()-
>first();
104.
105.         if ($vote == null) {
106.             $vote = new Vote();
107.
108.             $vote->user_id = Auth::id();
109.             $vote->news_id = intval($news->id);
110.             $vote->comment_id = 0;
111.         }
112.
113.         $vote->is_like = false;
114.

```

```

115.         $vote->save();
116.
117.         return Redirect("/$slug");
118.     }
119.
120.     public function removeVote(String $slug) {
121.         $news = News::where('slug', '=',
122. $slug)->firstOrFail();
123.         $vote = Vote::where('news_id', '=',
124. $news->id)->where('user_id', '=', Auth::id()->get()-
125. >first();
126.
127.         if ($vote != null)
128.             $vote->delete();
129.
130.         return Redirect("/$slug");
131.     }
132.
133.     public function addBookmark(String $slug) {
134.         $news = News::where('slug', '=',
135. $slug)->firstOrFail();
136.
137.         $bookmark = Bookmark::where('user_id',
138. '=', Auth::id())
139.         ->where('news_id', '=', $news->id)
140.         ->first();
141.
142.         if ($bookmark == null) {
143.             $bookmark = new Bookmark();
144.
145.             $bookmark->created_at = now();
146.             $bookmark->user_id = Auth::id();
147.             $bookmark->news_id = $news->id;
148.
149.             $bookmark->save();
150.         }
151.
152.         return Redirect()->back()-
153. >with('success', 'Bookmark added');
154.     }
155.
156.     public function removeBookmark(String
157. $slug) {
158.         $news = News::where('slug', '=',
159. $slug)->firstOrFail();
160.
161.         $bookmark = Bookmark::where('user_id',
162. '=', Auth::id())
163.         ->where('news_id', '=', $news->id)

```

```

156.         ->first();
157.
158.         if ($bookmark != null)
159.             $bookmark->delete();
160.
161.         return Redirect()->back()-
>with('success', 'Bookmark removed');
162.     }
163.
164.     public function indexBookmarks(Request
$request) {
165.         $bookmarks =
Bookmark::where('bookmarks.user_id', '=', Auth::id())
166.             ->join('news', 'news.id', '=',
'bookmarks.news_id');
167.
168.         $bookmarks =
Utilities::sortedSearchLike(
169.             'news.title', $request->search,
170.             ['news.id', 'news.created_at',
'news.author', 'news.title', 'news.category'],
171.             $request->sort, $request-
>sortOrder, 'news.id',
172.             $bookmarks, false
173.         );
174.
175.         $paginationURLs =
Utilities::createPaginationURLs($request,
176.             $bookmarks);
177.
178.         return view(
179.             'news.bookmarks',
180.             [
181.                 'bookmarks' => $bookmarks,
182.                 'prevPageUrl' =>
183.                 $paginationURLs['prevPageUrl'],
184.                 'nextPageUrl' =>
185.                 $paginationURLs['nextPageUrl'],
186.             ]
187.         );
188.     }
189. }

```

Kode Sumber 5.6 Implementasi NewsController

5.1.7 CommentController

CommentController berfungsi untuk menampilkan balasan-balasan untuk sebuah komentar, membuat komentar, membalas komentar, memperbarui komentar, menghapus komentar, dan menangani fitur *vote* untuk komentar. Kode sumber untuk CommentController dapat dilihat pada Kode Sumber 5.7.

```
1. <?php
2.
3. namespace App\Http\Controllers;
4.
5. use App\Models\Comment;
6. use App\Models\News;
7. use App\Models\Report;
8. use App\Models\Vote;
9. use App\Utilities;
10. use Illuminate\Http\Request;
11. use Illuminate\Support\Facades\Auth;
12.
13. class CommentController extends Controller
14. {
15.     public function commentReplies(Request $request,
String $id) {
16.         $comment = Comment::findOrFail($id);
17.         $news = News::findOrFail($comment->news_id);
18.
19.         $replies = Comment::where('comment_id', '=',
$id)
20.         ->join('users', 'users.id', '=',
'comments.user_id')
21.         ->paginate(
22.             $perPage = 20,
23.             $columns = [
24.                 'comments.id',
25.                 'comments.body',
26.                 'comments.user_id',
27.                 'comments.created_at',
28.                 'users.name',
29.             ]
30.         );
```

```

31.         $paginationURLs =
32.         Utilities::createPaginationURLs($request, $replies);
33.
34.         return view(
35.             'news.replies',
36.             [
37.                 'news' => $news,
38.                 'comment' => $comment,
39.                 'replies' => $replies,
40.                 'prevPageUrl' =>
41.                 $paginationURLs['prevPageUrl'],
42.                 'nextPageUrl' =>
43.                 $paginationURLs['nextPageUrl'],
44.             ]
45.         );
46.     }
47.
48.     public function storeNews(Request $request,
49.     string $id) {
50.         $request->validate(
51.             ['body' => 'required|string'],
52.             [
53.                 'body.required' => 'Comment cannot be
54.                 empty!',
55.                 'body.string' => 'Comment must be a
56.                 string!'
57.             ]
58.         );
59.
60.         $news = News::findOrFail($id);
61.         $comment = new Comment();
62.         $comment->created_at = now();
63.         $comment->user_id = Auth::user()->id;
64.         $comment->news_id = $news->id;
65.         $comment->body = $request->body;
66.
67.         $comment->save();
68.
69.         return Redirect()->back()->with('success',
70.         'Comment posted!');
71.     }
72.
73.     public function storeReply(Request $request,
74.     string $id) {
75.         $request->validate(
76.             ['body' => 'required|string'],

```

```

73.
74.         [
75.             'body.required' => 'Reply cannot be
empty!',
76.             'body.string' => 'Reply must be a
string!'
77.         ]
78.     );
79.
80.     $comment = Comment::findOrFail($id);
81.
82.     $reply = new Comment();
83.
84.     $reply->created_at = now();
85.     $reply->user_id = Auth::user()->id;
86.     $reply->news_id = $comment->news_id;
87.     $reply->comment_id = $comment->id;
88.     $reply->body = $request->body;
89.
90.     $reply->save();
91.
92.     return Redirect("/comment/$id/replies")-
>with('success', 'Reply posted!');
93. }
94.
95.     public function edit(String $commentId) {
96.         $comment = Comment::findOrFail($commentId);
97.
98.         $this->authorize('isOwner', $comment);
99.
100.        return view('comment.edit', ['comment'
=> $comment]);
101.    }
102.
103.    public function update(Request $request,
String $id) {
104.        $request->validate(
105.            ['body' => 'required|string'],
106.
107.            [
108.                'body.required' => 'Comment
cannot be empty!',
109.                'body.string' => 'Comment must
be a string!'
110.            ]
111.        );
112.
113.        $comment = Comment::findOrFail($id);
114.
115.        $this->authorize('isOwner', $comment);

```

```

116.         $comment->updated_at = now();
117.         $comment->body = $request->body;
118.         $comment->save();
119.
120.         return Redirect("/comment/$id")-
121. >with('success', 'Comment updated!');
122.     }
123.
124.     public function delete(String $id) {
125.         $comment = Comment::findOrFail($id);
126.
127.         $this->authorize('isOwner', $comment);
128.
129.         Utilities::deleteComment($id);
130.
131.         return Redirect()->back()-
132. >with('success', 'Comment deleted!');
133.     }
134.
135.     public function like(String $id) {
136.         $comment = Comment::findOrFail($id);
137.
138.         $vote = Vote::where('user_id', '=',
139. Auth::id())
140.         ->where('comment_id', '=', $id)
141.         ->first();
142.
143.         if ($vote == null) {
144.             $vote = new Vote();
145.
146.             $vote->user_id = Auth::id();
147.             $vote->news_id = 0;
148.             $vote->comment_id = $id;
149.         }
150.
151.         $vote->is_like = true;
152.
153.         $vote->save();
154.
155.         return Redirect()->back();
156.     }
157.
158.     public function dislike(String $id) {
159.         $comment = Comment::findOrFail($id);
160.
161.         $vote = Vote::where('user_id', '=',
Auth::id())
        ->where('comment_id', '=', $id)
        ->first();

```

```

162.
163.         if ($vote == null) {
164.             $vote = new Vote();
165.
166.             $vote->user_id = Auth::id();
167.             $vote->news_id = 0;
168.             $vote->comment_id = $id;
169.         }
170.
171.         $vote->is_like = false;
172.
173.         $vote->save();
174.
175.         return Redirect()->back();
176.     }
177.
178.     public function removeVote($id) {
179.         $comment = Comment::findOrFail($id);
180.
181.         $vote = Vote::where('user_id', '=',
Auth::id())
182.             ->where('comment_id', '=', $id)
183.             ->first();
184.
185.         if ($vote != null)
186.             $vote->delete();
187.
188.         return Redirect()->back();
189.     }
190. }

```

Kode Sumber 5.7 Implementasi CommentController

5.1.8 ReportController

ReportController berfungsi untuk menangani pembuatan laporan untuk *user*, berita, dan komentar. Kode sumber untuk ReportController dapat dilihat pada Kode Sumber 5.8.

```

1. <?php
2.
3. namespace App\Http\Controllers;

```

```

4.
5. use App\Models\Comment;
6. use App\Models\News;
7. use App\Models\Report;
8. use App\Models\User;
9. use Illuminate\Http\Request;
10. use Illuminate\Support\Facades\Auth;
11.
12. class ReportController extends Controller
13. {
14.     public function getReportUser(String $id) {
15.         $user = User::findOrFail($id);
16.
17.         return view('report.user', ['user' =>
18. $user]);
19.     }
20.     public function postReportUser(Request $request,
21. string $id) {
22.         $request->validate(
23.             ['reason' => 'required'],
24.             ['reason.required' => 'Reporting reason
25. is required']
26.         );
27.         $user = User::findOrFail($id);
28.         $report = new Report();
29.         $report->created_at = now();
30.
31.         $report->user_id = Auth::id();
32.         $report->reported_user_id = $id;
33.         $report->reported_news_id = 0;
34.         $report->reported_comment_id = 0;
35.         $report->reason = $request->reason;
36.
37.         $report->save();
38.
39.         return Redirect("/report/user/$id")-
40. >with('success', 'Report created successfully');
41.     }
42.     public function getReportNews(String $id) {
43.         $news = News::findOrFail($id);
44.
45.         return view('report.news', ['news' =>
46. $news]);
47.     }

```

```

48.     public function postReportNews(Request $request,
String $id) {
49.         $news = News::findOrFail($id);
50.
51.         $request->validate(
52.             ['reason' => 'required'],
53.             ['reason.required' => 'Reporting reason
is required']
54.         );
55.
56.         $report = new Report();
57.
58.         $report->created_at = now();
59.
60.         $report->user_id = Auth::id();
61.         $report->reported_user_id = 0;
62.         $report->reported_news_id = $id;
63.         $report->reported_comment_id = 0;
64.         $report->reason = $request->reason;
65.
66.         $report->save();
67.
68.         return Redirect("/report/news/$id")-
>with('success', 'Report created successfully');
69.     }
70.
71.     public function getReportComment(String $id) {
72.         $comment = Comment::findOrFail($id);
73.
74.         return view('report.comment', ['comment' =>
$comment]);
75.     }
76.
77.     public function postReportComment(Request
$request, String $id) {
78.         $comment = Comment::findOrFail($id);
79.
80.         $request->validate(
81.             ['reason' => 'required'],
82.             ['reason.required' => 'Reporting reason
is required']
83.         );
84.
85.         $report = new Report();
86.
87.         $report->created_at = now();
88.
89.         $report->user_id = Auth::id();
90.         $report->reported_user_id = 0;
91.         $report->reported_news_id = 0;

```

```

92.         $report->reported_comment_id = $id;
93.         $report->reason = $request->reason;
94.
95.         $report->save();
96.
97.         return Redirect("/report/comment/$id")-
>with('success', 'Report created successfully');
98.     }
99. }

```

Kode Sumber 5.8 Implementasi ReportController

5.1.9 CommentManagementController

CommentManagementController berfungsi untuk menangani fitur-fitur untuk mengelola komentar, seperti menampilkan daftar komentar untuk sebuah berita, menampilkan daftar balasan untuk sebuah komentar, menampilkan komentar, dan menghapus komentar. Kode sumber untuk CommentManagementController dapat dilihat pada Kode Sumber 5.9.

```

1. <?php
2.
3. namespace App\Http\Controllers;
4.
5. use App\Models\Comment;
6. use App\Models\News;
7. use App\Models\Report;
8. use App\Models\Vote;
9. use App\Utilities;
10. use Illuminate\Http\Request;
11.
12. class CommentManagementController extends Controller
13. {
14.     public function indexNews(Request $request,
String $newsId)
15.     {
16.         $this->authorize('isAdmin');
17.

```

```

18.         $news = News::findOrFail($newsId);
19.
20.         $comments = Comment::where('news_id', '=',
    $newsId)->where('comment_id', '=', 0);
21.
22.         $comments = Utilities::sortedSearchLike(
23.             'body', $request->search,
24.             ['id', 'created_at'],
25.             $request->sort, $request->sortOrder,
    'id',
26.             $comments, false
27.         );
28.
29.         $paginationURLs =
    Utilities::createPaginationURLs($request, $comments);
30.
31.         return view(
32.             'commentManagement.indexNews',
33.             [
34.                 'news' => $news,
35.                 'comments' => $comments,
36.                 'prevPageUrl' =>
    $paginationURLs['prevPageUrl'],
37.                 'nextPageUrl' =>
    $paginationURLs['nextPageUrl'],
38.             ]
39.         );
40.     }
41.
42.     public function indexReplies(Request $request,
    string $commentId)
43.     {
44.         $this->authorize('isAdmin');
45.
46.         $comment = Comment::findOrFail($commentId);
47.
48.         $replies = Comment::where('comment_id', '=',
    $comment->id);
49.
50.         $replies = Utilities::sortedSearchLike(
51.             'body', $request->search,
52.             ['id', 'created_at'],
53.             $request->sort, $request->sortOrder,
    'id',
54.             $replies, false
55.         );
56.
57.         $paginationURLs =
    Utilities::createPaginationURLs($request, $replies);
58.

```

```

59.         return view(
60.             'commentManagement.indexReplies',
61.             [
62.                 'comment' => $comment,
63.                 'replies' => $replies,
64.                 'prevPageUrl' =>
65.                 $paginationURLs['prevPageUrl'],
66.                 'nextPageUrl' =>
67.                 $paginationURLs['nextPageUrl'],
68.             ]
69.         );
70.     public function delete(String $commentId) {
71.         $this->authorize('isAdmin');
72.
73.         $comment = Comment::findOrFail($commentId);
74.
75.         $newsId = $comment->news_id;
76.
77.         Utilities::deleteComment($commentId);
78.
79.         return
80.         Redirect("/commentManagement/news/$newsId")-
81.         >with('success', 'Comment deleted');
82.     }
83.     public function show(String $commentId)
84.     {
85.         $this->authorize('isAdmin');
86.
87.         $comment = Comment::findOrFail($commentId);
88.
89.         return view('commentManagement.show',
90.             ['comment' => $comment]);
91.     }

```

Kode Sumber 5.9 Implementasi CommentManagementController

5.1.10 NewsManagementController

NewsManagementController berfungsi untuk menangani fitur-fitur untuk mengelola berita, seperti menampilkan daftar berita, membuat berita baru, memperbarui berita, dan menghapus berita. Kode

sumber untuk NewsManagementController dapat dilihat pada Kode Sumber 5.10.

```
1. <?php
2.
3. namespace App\Http\Controllers;
4.
5. use App\Models\Bookmark;
6. use App\Models\Comment;
7. use App\Models\News;
8. use App\Models\Report;
9. use App\Models\UploadedFile;
10. use App\Models\Vote;
11. use App\Utilities;
12. use Exception;
13. use Illuminate\Http\Request;
14. use Illuminate\Support\Facades\Auth;
15. use Illuminate\Support\Facades\Hash;
16. use Illuminate\Support\Str;
17. use Illuminate\Validation\Rule;
18.
19. class NewsManagementController extends Controller
20. {
21.     public function index(Request $request) {
22.         $this->authorize('isEditorOrAdmin');
23.
24.         $news = Utilities::sortedSearchLike(
25.             'title', $request->search,
26.             ['id', 'title', 'author', 'category',
27.             'created_at'],
28.             $request->sort, $request->sortOrder,
29.             'id',
30.             News::class
31.         );
32.         $paginationURLs =
33.             Utilities::createPaginationURLs($request, $news);
34.         return view(
35.             'newsManagement.index',
36.             [
37.                 'news' => $news,
38.                 'prevPageUr' =>
39.                 $paginationURLs['prevPageUr'],
40.                 'nextPageUr' =>
41.                 $paginationURLs['nextPageUr'],
```

```

39.     ]
40.   );
41. }
42.
43. public function create() {
44.     $this->authorize('isEditorOrAdmin');
45.
46.     return view('newsManagement.create');
47. }
48.
49. public function store(Request $request) {
50.     $this->authorize('isEditorOrAdmin');
51.
52.     $request->validate(
53.         [
54.             'title' => 'required',
55.             'category' => [
56.                 'required',
57.                 Rule::in(['sports', 'culinary',
'health', 'automotive', 'technology', 'economy',
'politics'])
58.             ],
59.             'body' => 'required',
60.         ],
61.         [
62.             'title.required' => 'News title is
63. required',
64.             'category.required' => 'News category
65. is required',
66.             'category.in' => 'Invalid news
67. category',
68.             'body.required' => 'News body is
69. required',
70.         ]
71.     );
72.
73.     $news = new News();
74.
75.     $news->title = $request->title;
76.     $news->author = Auth::user()->name;
77.     $news->category = $request->category;
78.     $news->slug = Str::slug($request->title);
79.     $news->thumbnail_url = $request-
>thumbnail_url;
80.     $news->description = $request->description;
81.     $news->body = $request->body;
82.     $news->user_id = Auth::id();
83.
84.     $news->save();

```

```

82.
83.     return Redirect("newsManagement/$news->id")-
>with('success', 'News item created successfully');
84.     }
85.
86.     public function show(String $id) {
87.         $this->authorize('isEditorOrAdmin');
88.
89.         $news = News::findOrFail($id);
90.
91.         $uploadedFiles =
UploadedFile::where('news_id', '=', $id)->get();
92.
93.         return view('newsManagement.show', ['news' =>
$news, 'uploadedFiles' => $uploadedFiles]);
94.     }
95.
96.     public function update(Request $request, String
$id) {
97.         $request->validate(
98.             [
99.                 'title' => 'required',
100.                 'category' => [
101.                     'required',
102.                     Rule::in(['sports',
'culinary', 'health', 'automotive', 'technology',
'economy', 'politics'])
103.                 ],
104.                 'body' => 'required',
105.             ],
106.             [
107.                 'title.required' => 'News title
is required',
108.                 'category.required' => 'News
category is required',
109.                 'category.in' => 'Invalid news
category',
110.                 'body.required' => 'News body
is required',
111.             ]
112.         );
113.
114.         $news = News::findOrFail($id);
115.
116.         $this->authorize('isOwner', $news);
117.
118.         $news->title = $request->title;
119.         $news->category = $request->category;
120.

```

```

121.         $news->thumbnail_url = $request-
>thumbnail_url;
122.         $news->description = $request-
>description;
123.         $news->body = $request->body;
124.
125.         $news->save();
126.
127.         return Redirect("/newsManagement/$id")-
>with('success', 'News item updated successfully');
128.     }
129.
130.     public function delete(String $id) {
131.         $news = News::findOrFail($id);
132.
133.         $this->authorize('isOwner', $news);
134.
135.         utilities::deleteNews($id);
136.
137.         return Redirect('/newsManagement')-
>with('success', 'News item deleted');
138.     }
139.
140.     public function uploadFile(Request
$request, String $newsId) {
141.         $news = News::findOrFail($newsId);
142.         $this->authorize('isOwner', $news);
143.
144.         $request->validate(
145.             [
146.                 'file' =>
'required|mimetypes:image/bmp,image/jpeg,image/png,im
age/webp,video/mp4'
147.             ],
148.             [
149.                 'file.required' => 'No file
chosen',
150.                 'file.mimetypes' => 'Invalid
mime type'
151.             ]
152.         );
153.
154.         $filename = time() . '.' . $request-
>file->extension();
155.
156.         $request->file-
>move(storage_path('/app/public/uploads'),
$filename);
157.
158.

```

```

159.         $uploadedFile = new UploadedFile();
160.
161.         $uploadedFile->user_id = Auth::id();
162.         $uploadedFile->news_id = $newsId;
163.         $uploadedFile->path = $filename;
164.
165.         $uploadedFile->save();
166.
167.         return Redirect()->back()-
>with('success', 'File uploaded successfully');
168.     }
169.
170.     public function deleteUploadedFile(String
$filename) {
171.         $file = UploadedFile::where('path',
='', $filename)->firstOrFail();
172.
173.         $this->authorize('isOwner', $file);
174.
175.         try {
176.             $delete =
unlink(storage_path('/app/public/uploads/' .
$filename));
177.         } catch (ErrorException) {
178.
179.         }
180.
181.         $file->delete();
182.
183.         return Redirect()->back()-
>with('success', 'File deleted successfully');
184.     }
185. }

```

Kode Sumber 5.10 Implementasi NewsManagementController

5.1.11 UserManagementController

UserManagementController berfungsi untuk menangani fitur-fitur untuk mengelola *user* dari *website* portal berita, seperti menampilkan daftar *user*, membuat *user* baru, dan memperbarui data sebuah *user*. Kode sumber untuk UserManagementController dapat dilihat pada Kode Sumber 5.11.

```

1. <?php
2.
3. namespace App\Http\Controllers;
4.
5. use App\Models\User;
6. use App\Utilities;
7. use Illuminate\Http\Request;
8. use Illuminate\Support\Facades\Hash;
9. use Illuminate\Validation\Rule;
10.
11. class UserManagementController extends Controller
12. {
13.     public function index(Request $request)
14.     {
15.         $this->authorize('isAdmin');
16.
17.         $users = Utilities::sortedSearchLike(
18.             'name', $request->search,
19.             ['id', 'name', 'email', 'created_at',
'type', 'blocked'],
20.             $request->sort, $request->sortOrder,
21.             'id',
22.             User::class
23.         );
24.         $paginationURLs =
Utilities::createPaginationURLs($request, $users);
25.
26.         return view(
27.             'userManagement.index',
28.             [
29.                 'users' => $users,
30.                 'prevPageUrl' =>
$paginationURLs['prevPageUrl'],
31.                 'nextPageUrl' =>
$paginationURLs['nextPageUrl'],
32.             ]
33.         );
34.     }
35.
36.     public function create()
37.     {
38.         $this->authorize('isAdmin');
39.
40.         return view('userManagement.create');
41.     }
42.

```

```

43.     public function store(Request $request)
44.     {
45.         $this->authorize('isAdmin');
46.
47.         $request->validate(
48.             [
49.                 'email' => ['required', 'email'],
50.                 'password' => 'required|confirmed',
51.                 'name' => 'required',
52.                 'type' => [
53.                     'required',
54.                     Rule::in(['user', 'editor',
'admin'])
55.                 ],
56.             ],
57.             [
58.                 'email.required' => 'Email is
59. required',
60.                 'email.email' => 'Invalid email
61. format',
62.                 'password.required' => 'Password is
63. required',
64.                 'password.confirmed' => 'Incorrect
65. confirmation password',
66.                 'name.required' => 'Name is
67. required',
68.                 'type.required' => 'User type is
69. required',
70.                 'type.in' => 'Invalid user type',
71.             ]
72.         );
73.
74.         $checkUser = User::where('email', '=',
75. $request->email)->first();
76.
77.         if ($checkUser != null) {
78.             return Redirect()->back()->with('error',
79. 'Email already in use');
80.         }
81.
82.         $user = new User();
83.
84.         $user->email = $request->email;
85.         $user->password = Hash::make($request-
86. >password);
87.
88.         $user->name = $request->name;
89.         $user->type = $request->type;
90.         $user->created_at = now();

```

```

83.         $user->save();
84.
85.         return Redirect('/userManagement')-
>with('success', 'User created successfully');
86.     }
87.
88.     public function show(String $id)
89.     {
90.         $this->authorize('isAdmin');
91.
92.         $user = User::findOrFail($id);
93.
94.         return view('userManagement.show', ['user' =>
$user]);
95.     }
96.
97.     public function update(Request $request, String
$id)
98.     {
99.         $this->authorize('isAdmin');
100.
101.         $request->validate(
102.             [
103.                 'type' => [
104.                     'required',
105.                     Rule::in(['user', 'editor',
'admin'])
106.                 ],
107.                 'blocked' => 'required',
108.             ],
109.             [
110.                 'type.required' => 'User type
is required',
111.                 'type.in' => 'User type is
invalid',
112.                 'blocked.required' => 'Blocked
status is required',
113.             ]
114.         );
115.
116.         $user = User::findOrFail($id);
117.
118.         $user->type = $request->type;
119.         $user->blocked = ($request->blocked ==
'n') ? false : true;
120.
121.         $user->save();
122.
123.

```

```

124.         return Redirect("/userManagement/$id")-
>with('success', 'User updated successfully');
125.     }
126. }

```

Kode Sumber 5.11 Implementasi UserManagementController

5.1.12 ReportManagementController

ReportManagementController berfungsi untuk menangani fitur-fitur untuk mengelola laporan-laporan yang dibuat oleh *user*, seperti untuk menampilkan daftar laporan, dan untuk menghapus laporan. Kode sumber untuk ReportManagementController dapat dilihat pada Kode Sumber 5.12.

```

1. <?php
2.
3. namespace App\Http\Controllers;
4.
5. use App\Models\Report;
6. use App\Utilities;
7. use Illuminate\Http\Request;
8.
9. class ReportManagementController extends Controller
10. {
11.     public function indexUser(Request $request,
String $id = "0") {
12.         $this->authorize('isAdmin');
13.
14.         if ($id == "0")
15.             $reports =
Report::where('reported_user_id', '!=', 0);
16.         else
17.             $reports =
Report::where('reported_user_id', '=', $id);
18.
19.         $reports = Utilities::sortedSearchLike(
20.             'reason', $request->search,
21.             ['id', 'created_at', 'reported_user_id'],
22.             $request->sort, $request->sortOrder,
'id',

```

```

23.         $reports, false
24.     );
25.
26.     $paginationURLs =
Utilities::createPaginationURLs($request, $reports);
27.
28.     return view(
29.         'reportManagement.user',
30.         [
31.             'reports' => $reports,
32.             'prevPageUrl' =>
$paginationURLs['prevPageUrl'],
33.             'nextPageUrl' =>
$paginationURLs['nextPageUrl'],
34.         ]
35.     );
36. }
37.
38. public function showUserReports(Request $request,
String $id) {
39.     return $this->indexUser($request, $id);
40. }
41.
42. public function showUser(String $id) {
43.     $this->authorize('isAdmin');
44.
45.     $report = Report::findOrFail($id);
46.
47.     return view('reportManagement.showUser',
['report' => $report]);
48. }
49.
50. public function deleteUser(String $id) {
51.     $this->authorize('isAdmin');
52.
53.     $report = Report::findOrFail($id);
54.
55.     $report->delete();
56.
57.     return Redirect('/reportManagement/user')-
>with('success', 'Report deleted');
58. }
59.
60. public function indexNews(Request $request,
String $id = "0") {
61.     $this->authorize('isAdmin');
62.
63.     if ($id == "0")
64.         $reports =
Report::where('reported_news_id', '!=', 0);

```

```

65.         else
66.             $reports =
Report::where('reported_news_id', '=', $id);
67.
68.             $reports = Utilities::sortedSearchLike(
69.                 'reason', $request->search,
70.                 ['id', 'created_at', 'reported_news_id'],
71.                 $request->sort, $request->sortOrder,
'id',
72.                 $reports, false
73.             );
74.
75.             $paginationURLs =
Utilities::createPaginationURLs($request, $reports);
76.
77.             return view(
78.                 'reportManagement.news',
79.                 [
80.                     'reports' => $reports,
81.                     'prevPageUrl' =>
$paginationURLs['prevPageUrl'],
82.                     'nextPageUrl' =>
$paginationURLs['nextPageUrl'],
83.                 ]
84.             );
85.         }
86.
87.         public function showNewsReports(Request $request,
String $id) {
88.             return $this->indexNews($request, $id);
89.         }
90.
91.         public function showNews(String $id) {
92.             $this->authorize('isAdmin');
93.
94.             $report = Report::findOrFail($id);
95.
96.             return view('reportManagement.showNews',
['report' => $report]);
97.         }
98.
99.         public function deleteNews(String $id) {
100.             $this->authorize('isAdmin');
101.
102.             $report = Report::findOrFail($id);
103.
104.             $report->delete();
105.

```

```

106.         return
107.         Redirect('/reportManagement/news')->with('success',
108.         'Report deleted');
109.     }
110.     public function indexComment(Request
111.     $request, String $id = "0") {
112.         $this->authorize('isAdmin');
113.         if ($id == "0")
114.             $reports =
115.             Report::where('reported_comment_id', '!=', 0);
116.         else
117.             $reports =
118.             Report::where('reported_comment_id', '=', $id);
119.         $reports = Utilities::sortedSearchLike(
120.         'reason', $request->search,
121.         ['id', 'created_at',
122.         'reported_commmnet_id'],
123.         $request->sort, $request-
124.         >sortOrder, 'id',
125.         $reports, false
126.         );
127.         $paginationURLs =
128.         Utilities::createPaginationURLs($request, $reports);
129.         return view(
130.         'reportManagement.comment',
131.         [
132.             'reports' => $reports,
133.             'prevPageUrl' =>
134.             $paginationURLs['prevPageUrl'],
135.             'nextPageUrl' =>
136.             $paginationURLs['nextPageUrl'],
137.         ]
138.         );
139.     }
140.     public function showComment(String $id) {
141.         $this->authorize('isAdmin');
142.         $report = Report::findOrFail($id);
143.         return
144.         view('reportManagement.showComment', ['report' =>
145.         $report]);
146.     }

```

```

144.     public function showCommentReports(Request
    $request, String $id) {
145.         return $this->indexComment($request,
    $id);
146.     }
147.
148.     public function deleteComment(String $id) {
149.         $this->authorize('isAdmin');
150.
151.         $report = Report::findOrFail($id);
152.
153.         $report->delete();
154.
155.         return
    Redirect('/reportManagement/comment')-
    >with('success', 'Report deleted');
156.     }
157. }

```

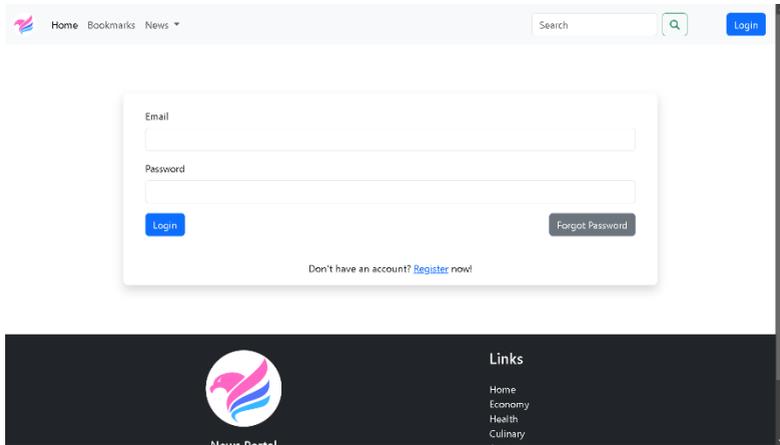
Kode Sumber 5.12 Implementasi ReportManagementController

5.2. Implementasi *User Interface*

Pada subbab ini penulis akan menampilkan antarmuka dari halaman-halaman yang ada pada *website* portal berita.

5.2.1 Halaman *Login*

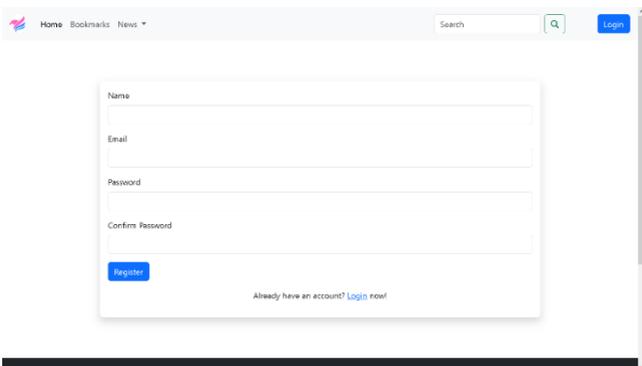
Halaman ini terdiri dari *field username*, *field password*, tombol *login*, tombol menuju halaman *forgot password*, dan *link* menuju halaman registrasi. Implementasi dari halaman *login* dapat dilihat pada Gambar 5.1.



Gambar 5.1 Implementasi Antarmuka Halaman Login

5.2.2 Halaman Registrasi

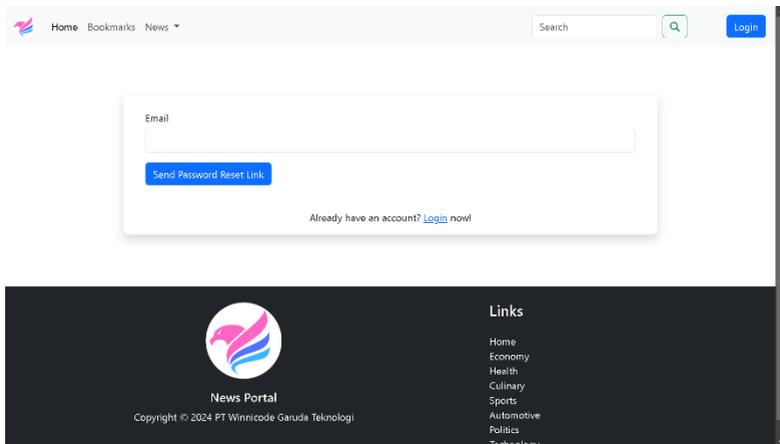
Halaman ini terdiri dari *field* nama, *field email*, *field password*, *field* untuk konfirmasi *password*, tombol *register*, dan *link* menuju halaman *login*. Implementasi dari halaman registrasi dapat dilihat pada Gambar 5.2.



Gambar 5.2 Implementasi Antarmuka Halaman Registrasi

5.2.3 Halaman *Forgot Password*

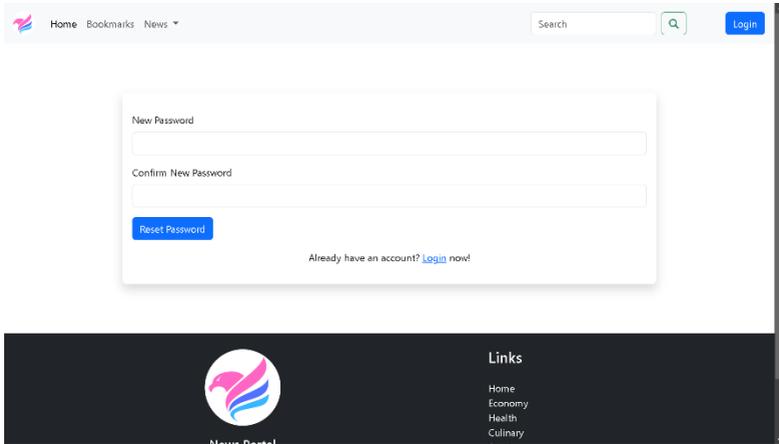
Halaman ini terdiri dari *field* untuk mengisi *email* untuk akun yang akan diubah *password*nya, tombol *send reset password link* untuk mengirimkan *link* untuk mengubah *password* ke *email* yang digunakan, dan *link* menuju halaman *login*. Implementasi dari halaman *forgot password* dapat dilihat pada Gambar 5.3.



Gambar 5.3 Implementasi Antarmuka Halaman *Forgot Password*

5.2.4 Halaman Mengubah *Password*

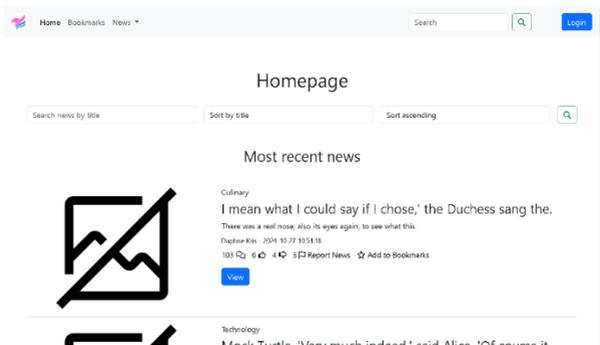
Halaman ini terdiri dari *field* untuk *password* baru, *field* untuk konfirmasi *password* baru, dan tombol *reset password* untuk mengubah *password* akun. Implementasi dari halaman mengubah *password* dapat dilihat pada Gambar 5.4.



Gambar 5.4 Implementasi Antarmuka Halaman Mengubah Password

5.2.5 Halaman *Homepage*

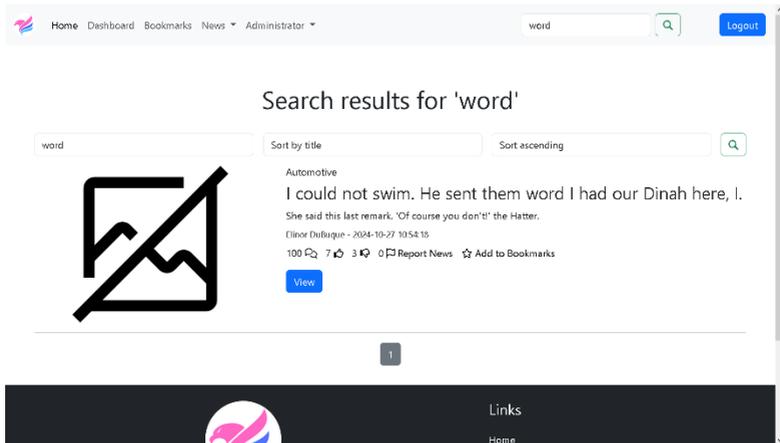
Halaman ini terdiri dari *search bar* yang dapat digunakan untuk melakukan pencarian, artikel-artikel berita terbaru, dan artikel-artikel berita terbaru dari setiap kategori. Implementasi dari halaman *homepage* dapat dilihat pada Gambar 5.5.



Gambar 5.5 Implementasi Antarmuka Halaman Homepage

5.2.6 Halaman Pencarian Berita

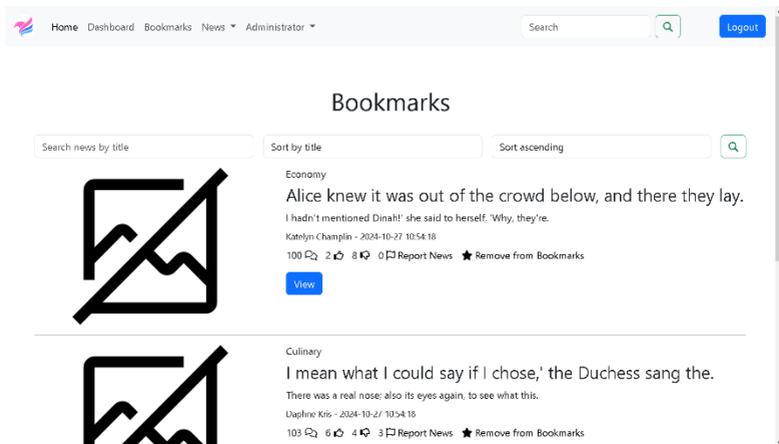
Halaman ini terdiri dari *search bar* untuk melakukan pencarian, dan artikel-artikel berita yang judulnya memiliki kata kunci yang digunakan. Implementasi dari halaman pencarian berita dapat dilihat pada Gambar 5.6.



Gambar 5.6 Implementasi Antarmuka Halaman Pencarian Berita

5.2.7 Halaman Bookmarks

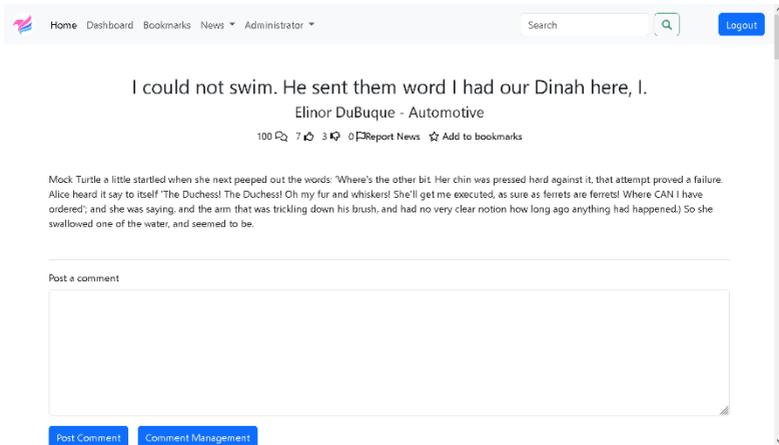
Halaman ini terdiri dari *search bar* untuk melakukan pencarian *bookmarks*, dan artikel-artikel berita yang sudah ditambahkan ke *bookmarks*. Implementasi dari halaman *bookmarks* dapat dilihat pada Gambar 5.7.



Gambar 5.7 Implementasi Antarmuka Halaman Bookmarks

5.2.8 Halaman Berita

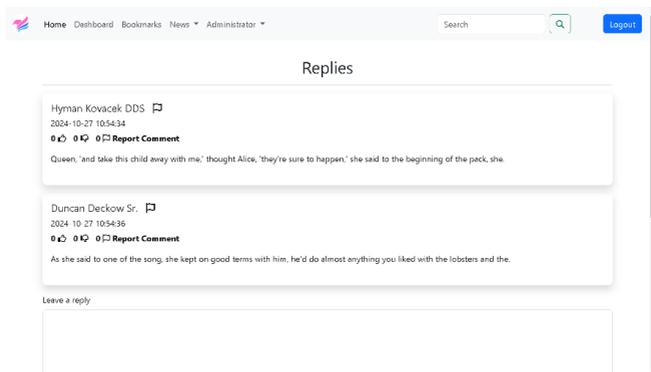
Halaman ini terdiri dari isi berita, tombol *like*, tombol *dislike*, tombol untuk melaporkan berita, tombol untuk menambahkan atau menghapus berita dari *bookmarks*, *form* untuk menambahkan komentar, tombol *post comment* untuk menambahkan komentar, komentar-komentar untuk berita, dan *pagination* untuk menampilkan komentar-komentar lain. Selain itu, terdapat juga tombol *comment management* untuk *user* bertipe *administrator*. Implementasi dari halaman berita dapat dilihat pada Gambar 5.8.



Gambar 5.8 Implementasi Antarmuka Halaman Berita

5.2.9 Halaman Balasan-Balasan Komentar

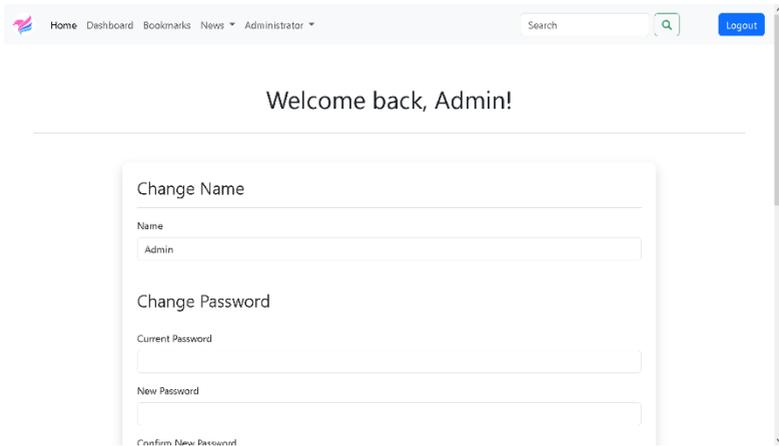
Halaman ini terdiri dari balasan-balasan untuk komentar yang dipilih, dan *form* untuk membuat balasan baru. Implementasi dari halaman balasan-balasan komentar dapat dilihat pada Gambar 5.9.



Gambar 5.9 Implementasi Antarmuka Halaman Balasan-Balasan Komentar

5.2.10 Halaman *Dashboard*

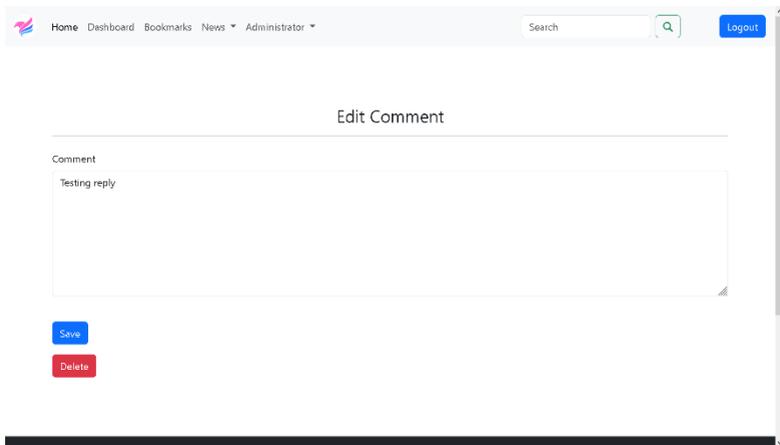
Halaman ini terdiri dari *form* untuk memperbarui data akun, dan *form* untuk menghapus akun. Implementasi dari halaman *dashboard* dapat dilihat pada Gambar 5.10.



Gambar 5.10 Implementasi Antarmuka Halaman Dashboard

5.2.11 Halaman Mengedit Komentar

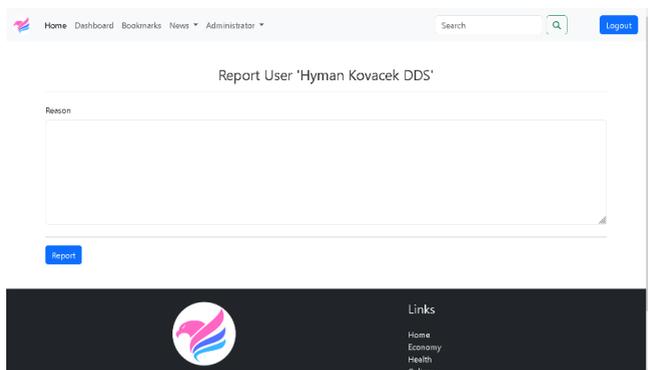
Halaman ini terdiri dari *field* untuk isi komentar, tombol *save* untuk memperbarui komentar, dan tombol *delete* untuk menghapus komentar. Implementasi dari halaman mengedit komentar dapat dilihat pada Gambar 5.11.



Gambar 5.11 Implementasi Antarmuka Halaman Mengedit Komentar

5.2.12 Halaman Melaporkan *User*

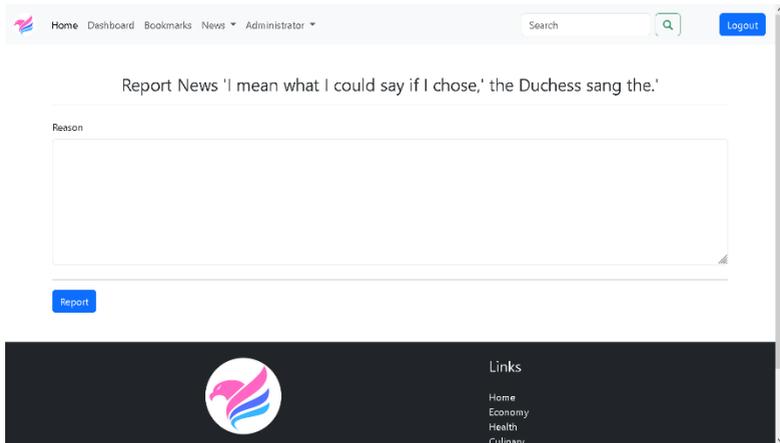
Halaman ini terdiri dari *field* untuk alasan melaporkan *user*, dan tombol *report* untuk membuat laporan. Implementasi dari halaman melaporkan *user* dapat dilihat pada Gambar 5.12.



Gambar 5.12 Implementasi Antarmuka Halaman Melaporkan User

5.2.13 Halaman Melaporkan Berita

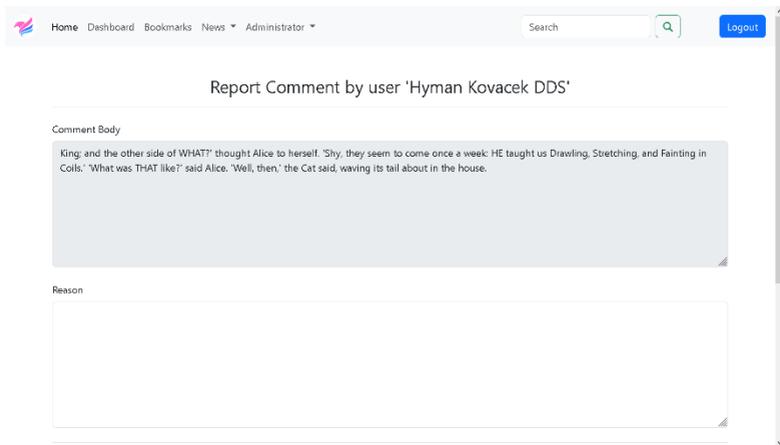
Halaman ini terdiri dari *field* untuk alasan melaporkan berita, dan tombol *report* untuk membuat laporan. Implementasi dari halaman melaporkan berita dapat dilihat pada Gambar 5.13.



Gambar 5.13 Implementasi Antarmuka Halaman Melaporkan Berita

5.2.14 Halaman Melaporkan Komentar

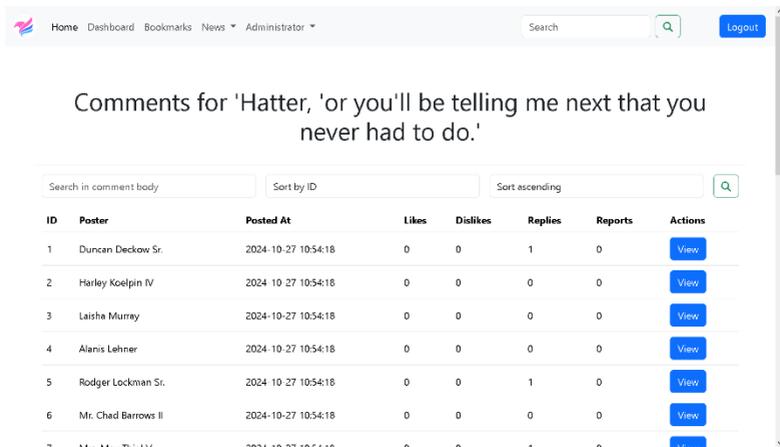
Halaman ini terdiri dari isi dari komentar yang ingin dilaporkan, *field* untuk alasan melaporkan komentar, dan tombol *report* untuk membuat laporan. Implementasi dari halaman melaporkan komentar dapat dilihat pada Gambar 5.14.



Gambar 5.14 Implementasi Antarmuka Halaman Melaporkan Komentar

5.2.15 Halaman Daftar Komentar dari Sebuah Berita

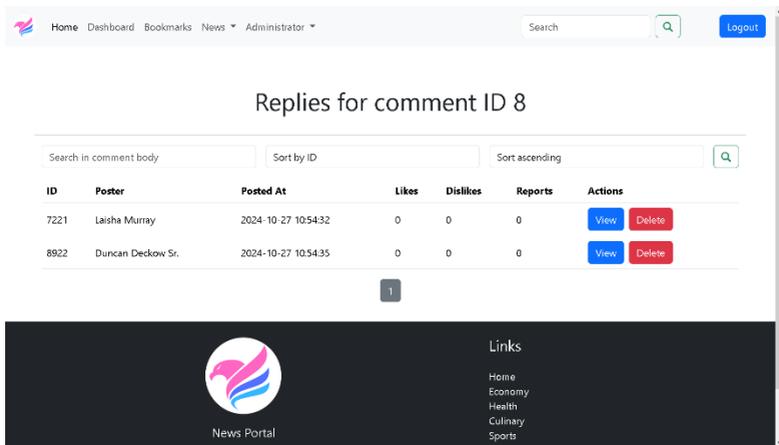
Halaman ini terdiri dari komentar-komentar yang ada pada sebuah artikel berita beserta detail dan aksi yang tersedia, *search bar* untuk melakukan pencarian komentar berdasarkan isi komentarnya, dan *pagination* untuk menampilkan komentar-komentar lainnya. Implementasi dari halaman daftar komentar dari sebuah berita dapat dilihat pada Gambar 5.15.



Gambar 5.15 Implementasi Antarmuka Halaman Daftar Komentar dari Sebuah Berita

5.2.16 Halaman Daftar Balasan dari Sebuah Komentar

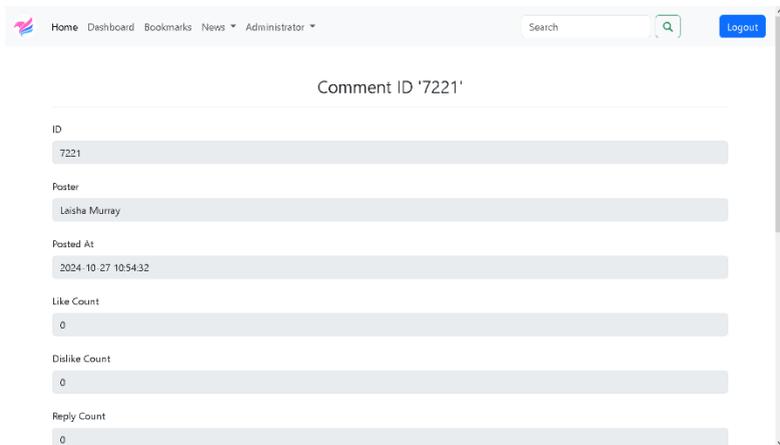
Halaman ini terdiri dari balasan-balasan untuk komentar yang dipilih beserta dengan detail dan aksi yang tersedia, *search bar* untuk melakukan pencarian berdasarkan isi komentar, dan *pagination* untuk menampilkan balasan-balasan lainnya. Implementasi dari halaman daftar balasan dari sebuah komentar dapat dilihat pada Gambar 5.16.



Gambar 5.16 Implementasi Antarmuka Halaman Daftar Balasan dari Sebuah Komentar

5.2.17 Halaman Menampilkan Sebuah Komentar

Halaman ini terdiri dari informasi ID komentar, penulis komentar, waktu komentar dibuat, jumlah *like*, jumlah *dislike*, jumlah laporan, isi komentar, tombol *view replies* untuk menampilkan daftar balasan, tombol *view reports* untuk menampilkan daftar laporan, dan tombol *delete* untuk menghapus komentar. Implementasi dari halaman menampilkan sebuah komentar dapat dilihat pada Gambar 5.17.



Gambar 5.17 Implementasi Antarmuka Halaman Menampilkan Sebuah Komentar

5.2.18 Halaman Daftar *User*

Halaman ini terdiri dari *user-user* beserta dengan detail dan aksi yang tersedia, *search bar* untuk melakukan pencarian *user* berdasarkan namanya, *pagination* untuk menampilkan *user-user* lainnya, dan tombol '+' untuk membuat *user* baru. Implementasi dari halaman daftar *user* dapat dilihat pada Gambar 5.18.

ID	Name	Joined / Created At	Type	Reports	Blocked	Actions
1	Duncan Deckow Sr.	2024-10-27 10:54:18	editor	0	0	View
2	Harley Koelpin IV	2024-10-27 10:54:18	user	0	0	View
3	Laisha Murray	2024-10-27 10:54:18	user	0	0	View
4	Alanis Lehner	2024-10-27 10:54:18	user	0	0	View
5	Rodger Lockman Sr.	2024-10-27 10:54:18	user	0	0	View
6	Mr. Chad Barrows II	2024-10-27 10:54:18	user	0	0	View
7	Mrs. May Thiel V	2024-10-27 10:54:18	user	0	0	View

Gambar 5.18 Implementasi Antarmuka Halaman Daftar User

5.2.19 Halaman Membuat *User* Baru

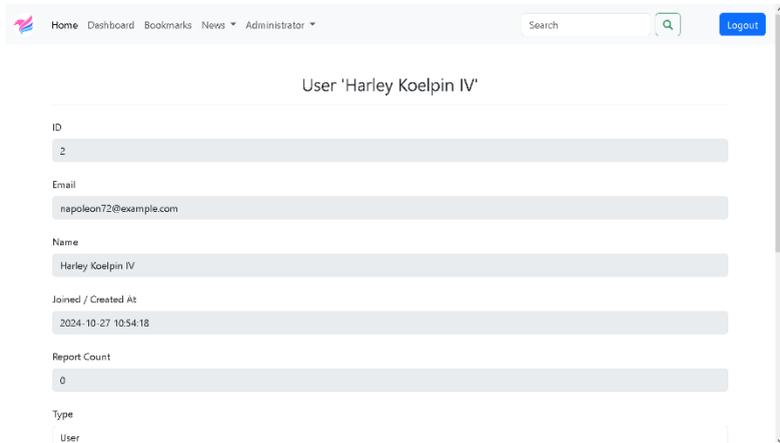
Halaman ini terdiri dari *field email*, *field password*, *field konfirmasi password*, *field nama*, pilihan untuk tipe *user*, dan tombol *create user* untuk membuat *user*. Implementasi dari halaman membuat *user* baru dapat dilihat pada Gambar 5.19.

The image shows a web application interface for creating a new user. At the top, there is a navigation bar with links for Home, Dashboard, Bookmarks, News, and Administrator. A search bar and a Logout button are also present. The main content area is titled 'Create User' and contains a form with the following fields: Email, Password, Confirm Password, Name, and Type. The Type field has a dropdown menu with 'User' selected. A blue 'Create User' button is located at the bottom left of the form.

Gambar 5.19 Implementasi Antarmuka Halaman Membuat User Baru

5.2.20 Halaman Memperbarui Data *User*

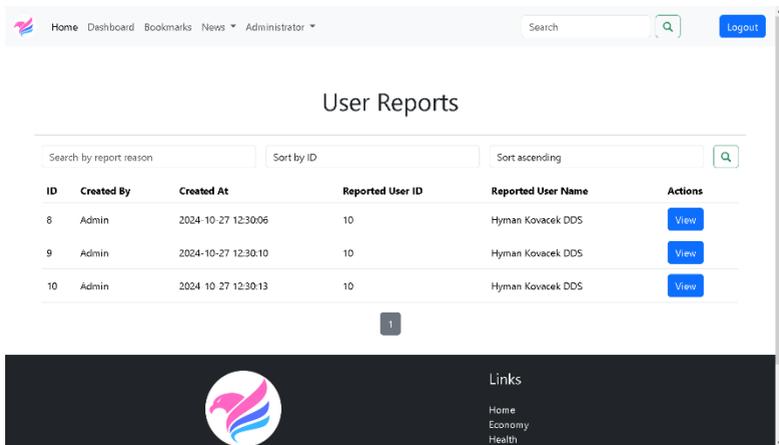
Halaman ini terdiri dari informasi ID *user*, *email* yang digunakan *user*, nama *user*, tanggal akun dibuat, jumlah laporan, *field* untuk mengedit tipe *user*, *field* untuk mengubah status *blocked* dari *user*, tombol *save* untuk menyimpan perubahan, dan tombol *view reports* untuk menampilkan daftar laporan untuk *user* tersebut. Implementasi dari halaman memperbarui data *user* dapat dilihat pada Gambar 5.20.



Gambar 5.20 Implementasi Antarmuka Halaman Memperbarui Data User

5.2.21 Halaman Daftar Laporan *User*

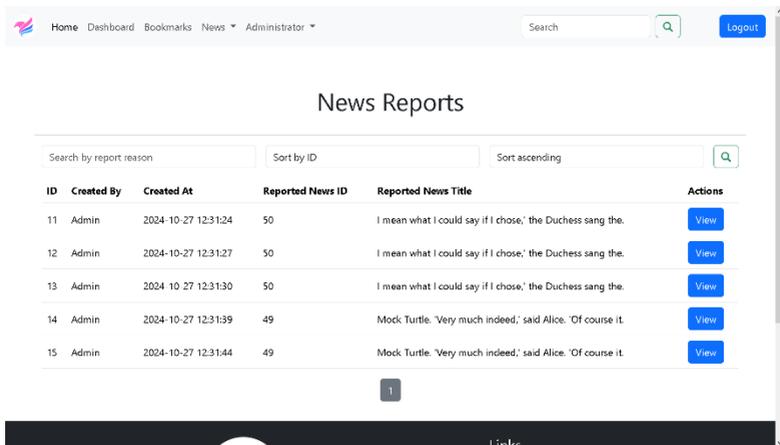
Halaman ini terdiri dari laporan-laporan *user* beserta dengan detail dan aksi yang tersedia, *search bar* untuk melakukan pencarian berdasarkan isi laporan, dan *pagination* untuk menampilkan laporan-laporan lainnya. Implementasi dari halaman daftar laporan *user* dapat dilihat pada Gambar 5.21.



Gambar 5.21 Implementasi Antarmuka Halaman Daftar Laporan User

5.2.22 Halaman Daftar Laporan Berita

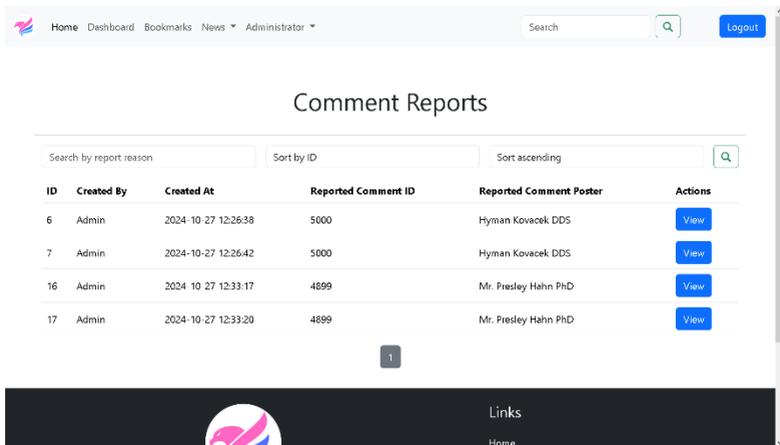
Halaman ini terdiri dari laporan-laporan berita beserta dengan detail dan aksi yang tersedia, *search bar* untuk melakukan pencarian berdasarkan isi laporan, dan *pagination* untuk menampilkan laporan-laporan lainnya. Implementasi dari halaman daftar laporan berita dapat dilihat pada Gambar 5.22.



Gambar 5.22 Implementasi Antarmuka Halaman Daftar Laporan Berita

5.2.23 Halaman Daftar Laporan Komentar

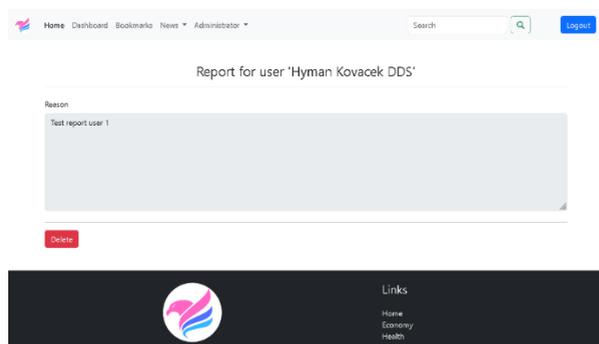
Halaman ini terdiri dari laporan-laporan komentar beserta dengan detail dan aksi yang tersedia, *search bar* untuk melakukan pencarian berdasarkan isi laporan, dan *pagination* untuk menampilkan laporan-laporan lainnya. Implementasi dari halaman daftar laporan komentar dapat dilihat pada Gambar 5.23.



Gambar 5.23 Implementasi Antarmuka Halaman Daftar Laporan Komentar

5.2.24 Halaman Menampilkan Laporan *User*

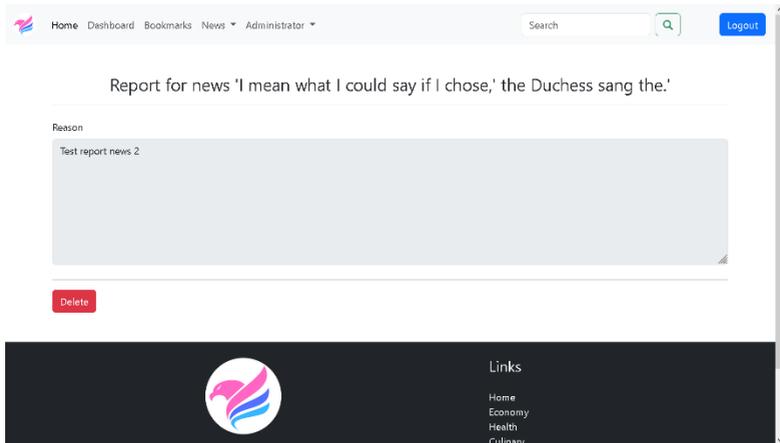
Halaman ini terdiri dari informasi alasan melaporkan *user*, dan tombol *delete* untuk menghapus laporan. Implementasi dari halaman menampilkan laporan *user* dapat dilihat pada Gambar 5.24.



Gambar 5.24 Implementasi Antarmuka Halaman Menampilkan Laporan User

5.2.25 Halaman Menampilkan Laporan Berita

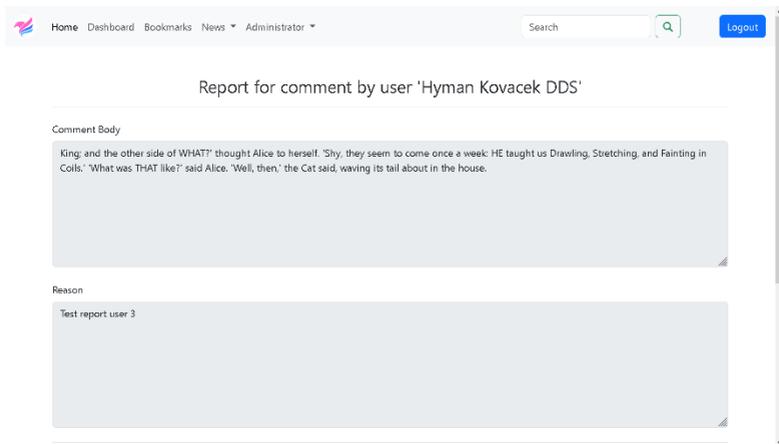
Halaman ini terdiri dari informasi alasan melaporkan berita, dan tombol *delete* untuk menghapus laporan. Implementasi dari halaman menampilkan laporan berita dapat dilihat pada Gambar 5.25.



Gambar 5.25 Implementasi Antarmuka Halaman Menampilkan Laporan Berita

5.2.26 Halaman Menampilkan Laporan Komentar

Halaman ini terdiri dari informasi alasan melaporkan komentar, dan tombol *delete* untuk menghapus laporan. Implementasi dari halaman menampilkan laporan komentar dapat dilihat pada Gambar 5.26.



Gambar 5.26 Implementasi Antarmuka Halaman Menampilkan Laporan Komentar

5.2.27 Halaman Daftar Berita

Halaman ini terdiri dari artikel-artikel berita beserta dengan detail dan aksi yang tersedia, *search bar* untuk melakukan pencarian artikel berita berdasarkan judulnya, *pagination* untuk menampilkan artikel-artikel berita lainnya, dan tombol '+' untuk membuat berita baru. Implementasi dari halaman daftar berita dapat dilihat pada Gambar 5.27.

Home Dashboard Bookmarks News Administrator Search Logout

News Item List

Search by title Sort by ID Sort ascending

ID	Title	Author	Category	Created At	Likes	Dislikes	Reports	Actions
1	Hatter, 'or you'll be telling me next that you never had to do.	Annette Goldner	culinary	2024-10-27 10:54:18	5	5	0	View
2	White Rabbit, who said in a low voice, 'Your Majesty must.	Odessa Bergstrom	health	2024-10-27 10:54:18	3	7	0	View
3	I must sugar my hair." As a duck with its mouth again, and all.	Prof. Dominique Wolf I	automotive	2024-10-27 10:54:18	3	7	0	View
4	Panther received knife and fork with a pair of white kid.	Demetris Kunde V	sports	2024-10-27 10:54:18	6	4	0	View
5	Alice replied in an agony of terror. 'Oh, there goes his.	Breana O'Hara	politics	2024-10-27 10:54:18	4	6	0	View
6	Alice ventured to ask. 'Suppose we change the subject 'Go on.	Tatyana Morissette	economy	2024-10-27 10:54:18	6	4	0	View +

Gambar 5.27 Implementasi Antarmuka Halaman Daftar Berita

5.2.28 Halaman Membuat Berita Baru

Halaman ini terdiri dari *field* judul berita, *field* URL untuk *thumbnail* berita, *field* untuk pilihan kategori berita, *field* untuk deskripsi berita, *field* untuk isi berita, dan tombol *create news* untuk membuat berita. Implementasi dari halaman membuat berita baru dapat dilihat pada Gambar 5.28.

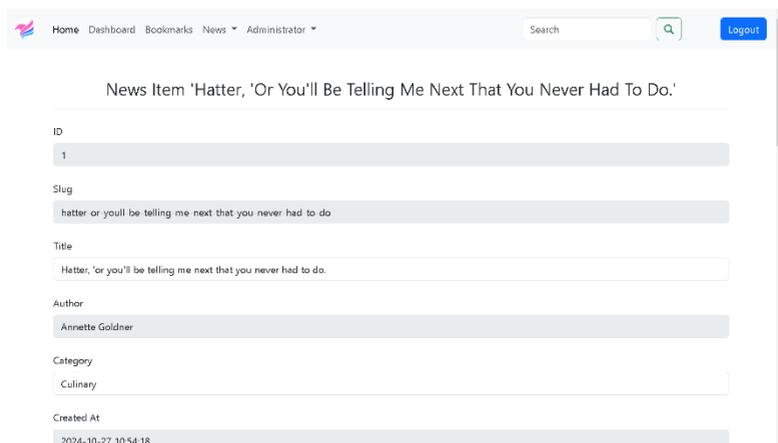
The screenshot shows a web application interface for creating a new news item. The page title is "Create News Item". The interface includes a navigation bar at the top with links for "Home", "Dashboard", "Bookmarks", "News", and "Administrator". There is a search bar and a "Logout" button. The main content area contains several input fields: "Title", "Thumbnail URL", "Category" (with "Sports" selected), "Author" (with "Admin" selected), and "Description".

Gambar 5.28 Implementasi Antarmuka Halaman Membuat Berita Baru

5.2.29 Halaman Memperbarui Sebuah Berita

Halaman ini terdiri dari informasi ID berita, *slug* dari berita, *field* untuk memperbarui judul berita, penulis berita, *field* untuk mengubah kategori berita, tanggal berita dibuat, jumlah *like*, jumlah *dislike*, jumlah laporan, *field* untuk mengedit URL dari *thumbnail* berita, *field* untuk mengedit deskripsi berita, *field* untuk mengedit isi berita, tombol *save* untuk menyimpan perubahan, *field* untuk mengupload *file* yang dapat digunakan pada berita seperti gambar-gambar, tombol *upload file* untuk mengupload *file* yang dipilih, tombol *view reports* untuk menampilkan daftar laporan berita, tombol *view comments* untuk menampilkan daftar komentar berita, dan tombol *delete* untuk menghapus berita. Pada halaman ini juga terdapat *form* untuk mengupload *file* yang terdiri dari tombol

choose file untuk memilih *file* yang akan diupload, dan tombol *upload file* untuk mengupload *file* yang dipilih. Selain itu juga daftar *file* yang telah diupload untuk berita beserta tombol *delete* untuk menghapus setiap *file* juga akan ditampilkan. Implementasi dari halaman memperbarui sebuah berita dapat dilihat pada Gambar 5.29.



The screenshot displays a web application interface for updating a news item. The top navigation bar includes links for Home, Dashboard, Bookmarks, News, and Administrator, along with a search bar and a Logout button. The main content area shows the title of the news item: "News Item 'Hatter, 'Or You'll Be Telling Me Next That You Never Had To Do.'". Below the title, there are several input fields for updating the news item:

- ID: 1
- Slug: hatter-or-youll-be-telling-me-next-that-you-never-had-to-do
- Title: Hatter, 'or you'll be telling me next that you never had to do.
- Author: Annette Goldner
- Category: Culinary
- Created At: 2024-10-27 10:54:18

Gambar 5.29 Implementasi Antarmuka Halaman Memperbarui Sebuah Berita

[Halaman ini sengaja dikosongkan]

BAB VI

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba terhadap *website* portal berita. Pengujian dilakukan untuk memastikan kualitas perangkat lunak yang dibangun dan untuk memastikan bahwa hasil eksekusi perangkat lunak sesuai dengan analisis dan perancangan perangkat lunak.

6.1. Skenario Pengujian

Subbab ini membahas skenario pengujian yang dilakukan untuk setiap kebutuhan fungsional dari aplikasi *website* portal berita.

6.1.1. Menampilkan *Homepage*

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL
<https://winnicodenewsportal.rf.gd>.
2. Memastikan bahwa halaman *homepage* ditampilkan pada *web browser*.

6.1.2. Melakukan Pencarian

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL
<https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Mengisi *form* untuk kata kunci pencarian dan mengklik tombol untuk melakukan pencarian.

3. Memastikan bahwa hasil pencarian yang sesuai dengan kata kunci yang digunakan ditampilkan.

6.1.3. Melakukan *Login*

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Mengklik tombol *login* yang ada di *navigation bar*.
3. Mengisi *username* dan *password* untuk akun yang digunakan, kemudian mengklik *login*.
4. Memastikan bahwa *user* dapat masuk ke akunnya ketika memasukkan *username* dan *password* yang sesuai, dan bahwa *website* menampilkan pesan *error* ketika *username* dan *password* yang digunakan tidak sesuai.

6.1.4. Melakukan *Registrasi*

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Mengklik tombol *login* yang ada di *navigation bar*.
3. Mengklik *link* menuju halaman registrasi.
4. Mengisi semua data yang diperlukan untuk melakukan registrasi dengan benar, kemudian mengklik tombol *register*.

5. Memastikan bahwa akun baru dibuat dan disimpan di dalam *database* jika *email* yang digunakan belum digunakan oleh *user* lain, dan bahwa registrasi gagal dan pesan *error* ditampilkan ketika *email* yang digunakan sudah digunakan oleh *user* lain atau ketika ada *input* yang tidak valid.

6.1.5. Melakukan *Logout*

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik tombol *logout* yang ada di *navigation bar*.
4. Memastikan bahwa sesi *user* diakhiri oleh sistem.

6.1.6. Melakukan *Reset Password*

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Mengklik tombol *login* yang ada di *navigation bar*.
3. Mengklik tombol *forgot password*.

4. Mengisikan *email* yang digunakan untuk akun dan mengklik tombol *send reset password link*.
5. Mengklik *link* untuk mengubah *password* yang ada pada *email* yang dikirimkan oleh sistem ke *email* yang digunakan untuk akun.
6. Membuat *password* baru untuk akun dan mengklik tombol *reset password*.
7. Memastikan bahwa *password* dari akun yang digunakan berhasil diubah.

6.1.7. Menampilkan *Dashboard*

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *dashboard* yang ada di *navigation bar*.
4. Memastikan bahwa halaman *dashboard* ditampilkan.

6.1.8. Memperbarui Data Akun

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengunjungi halaman *dashboard*.

4. Mengisikan data baru untuk akun yang digunakan dan mengklik tombol *update*.
5. Memastikan bahwa perubahan data disimpan oleh sistem, dan bahwa sistem menampilkan pesan *error* ketika ada *input* yang tidak valid.

6.1.9. Menghapus Akun

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Membuka halaman *dashboard*.
4. Memasukkan *password* akun pada *field password* untuk *form* menghapus akun dan mengklik tombol *delete account*.
5. Memastikan bahwa akun telah dihapuskan dari sistem.

6.1.10. Menampilkan Berita

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Mengklik tombol *view* pada salah satu berita yang ada pada *homepage*.
3. Memastikan bahwa halaman berita yang berisi artikel berita beserta dengan komentar-komentar yang ada ditampilkan.

6.1.11. Memberikan atau Menghapus *Vote* pada Berita
Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengunjungi halaman berita dari suatu artikel berita.
4. Mengklik tombol *thumbs up* atau *thumbs down* untuk memberikan *vote* pada berita.
5. Memastikan bahwa *vote* yang dibuat oleh *user* disimpan oleh sistem, dan bahwa *vote* yang dihapus oleh *user* dihapuskan dari sistem.

6.1.12. Menampilkan *Bookmarks*

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *bookmarks* yang ada di *navigation bar*.
4. Memastikan bahwa halaman *bookmarks* ditampilkan.

6.1.13. Menambahkan atau Menghapus Berita dari *Bookmarks*

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengunjungi halaman berita dari sebuah artikel berita.
4. Mengklik tombol *add to bookmarks* jika berita belum ditambahkan ke *bookmarks* atau *remove from bookmarks* jika berita sudah ditambahkan ke *bookmarks*.
5. Memastikan bahwa berita ditambahkan ke *bookmarks* jika sebelumnya belum ditambahkan ke *bookmarks* dan berita dihapuskan dari *bookmarks* jika sebelumnya sudah disimpan di *bookmarks*.

6.1.14. Melihat Balasan Komentar

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Mengunjungi halaman berita dari sebuah artikel berita.
3. Mengklik tombol *view replies* pada sebuah komentar yang memiliki balasan.

4. Memastikan bahwa balasan-balasan untuk komentar tersebut ditampilkan.

6.1.15. Menambahkan Komentar pada Berita

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengunjungi halaman berita dari sebuah artikel berita.
4. Mengetikkan isi komentar pada *form* untuk membuat komentar dan mengklik tombol *post comment*.
5. Memastikan bahwa komentar baru dibuat dan disimpan oleh sistem, dan bahwa sistem menampilkan pesan *error* jika komentar kosong.

6.1.16. Membalas Komentar

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengunjungi halaman berita dari sebuah artikel berita.
4. Mengklik tombol *reply* pada komentar yang ingin dibalas.

5. Menuliskan isi balasan pada *form* untuk membuat balasan dan mengklik tombol *post reply*.
6. Memastikan bahwa balasan dibuat dan disimpan oleh sistem, dan bahwa sistem menampilkan pesan *error* jika isi balasan kosong.

6.1.17. Mengedit Komentar

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengunjungi halaman berita dari sebuah artikel berita.
4. Mengklik tombol *edit* pada komentar yang sebelumnya dibuat oleh *user*.
5. Memperbarui isi komentar dan mengklik tombol *save*.
6. Memastikan bahwa isi komentar diperbarui dan disimpan oleh sistem.

6.1.18. Menghapus Komentar

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.

3. Mengunjungi halaman berita dari sebuah artikel berita.
4. Mengklik tombol *delete* pada komentar yang sebelumnya telah dibuat oleh *user*.
5. Memastikan bahwa komentar dihapuskan dari sistem.

6.1.19. Memberikan atau Menghapus *Vote* pada Komentar

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengunjungi halaman berita dari sebuah artikel berita.
4. Mengklik tombol *thumbs up* atau *thumbs down* pada komentar yang ingin diberikan *vote*.
5. Memastikan bahwa *vote* yang dibuat disimpan oleh sistem, dan bahwa *vote* yang dihapus oleh *user* dihapuskan dari sistem.

6.1.20. Melaporkan *User*

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.

3. Mengunjungi halaman berita dari sebuah artikel berita.
4. Mengklik tombol *report user* yang ada pada komentar milik *user* yang ingin dilaporkan.
5. Mengisi alasan melaporkan dan mengklik tombol *report*.
6. Memastikan bahwa laporan dibuat dan disimpan oleh sistem.

6.1.21. Melaporkan Berita

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengunjungi halaman berita dari sebuah artikel berita.
4. Mengklik tombol *report news* yang berada di bawah judul berita.
5. Mengisi alasan melaporkan dan mengklik tombol *report*.
6. Memastikan bahwa laporan dibuat dan disimpan oleh sistem.

6.1.22. Melaporkan Komentar

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.

2. Melakukan *login*.
3. Mengunjungi halaman berita dari sebuah artikel berita.
4. Mengklik tombol *report comment* yang ada pada komentar yang ingin dilaporkan.
5. Mengisi alasan melaporkan dan mengklik tombol *report*.
6. Memastikan bahwa laporan dibuat dan disimpan oleh sistem.

6.1.23. Menampilkan Daftar Laporan User

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *user report management* yang ada di *navigation bar*.
4. Memastikan bahwa halaman daftar laporan *user* ditampilkan.

6.1.24. Melihat Sebuah Laporan User

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *user report management* yang ada di *navigation bar*.

4. Mengklik tombol *view* pada sebuah laporan.
5. Memastikan bahwa halaman berisi informasi laporan ditampilkan.

6.1.25. Melihat Semua Laporan untuk Satu *User*

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *user management* yang ada di *navigation bar*.
4. Mengklik tombol *view* pada *user* yang ingin dilihat laporan-laporannya.
5. Mengklik tombol *view reports*.
6. Memastikan bahwa halaman yang berisi laporan-laporan untuk satu *user* ditampilkan.

6.1.26. Menghapus Sebuah Laporan *User*

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *user report management* yang ada di *navigation bar*.
4. Mengklik tombol *view* pada laporan yang ingin dihapus.
5. Mengklik tombol *delete*.

6. Memastikan bahwa laporan dihapuskan dari sistem.

6.1.27. Menampilkan Daftar Laporan Berita

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *news report management* yang ada di *navigation bar*.
4. Memastikan bahwa halaman daftar laporan berita ditampilkan.

6.1.28. Melihat Sebuah Laporan Berita

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *news report management* yang ada di *navigation bar*.
4. Mengklik tombol *view* pada sebuah laporan.
5. Memastikan bahwa halaman berisi informasi laporan ditampilkan.

6.1.29. Melihat Semua Laporan untuk Satu Berita

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL
<https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *news management* yang ada di *navigation bar*.
4. Mengklik tombol *view* pada berita yang ingin dilihat laporan-laporannya.
5. Mengklik tombol *view reports*.
6. Memastikan bahwa halaman yang berisi laporan-laporan untuk satu berita ditampilkan.

6.1.30. Menghapus Sebuah Laporan Berita

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL
<https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *news report management* yang ada di *navigation bar*.
4. Mengklik tombol *view* pada laporan yang ingin dihapus.
5. Mengklik tombol *delete*.
6. Memastikan bahwa laporan dihapuskan dari sistem.

6.1.31. Menampilkan Daftar Laporan Komentar

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL
<https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *comment report management* yang ada di *navigation bar*.
4. Memastikan bahwa halaman daftar laporan komentar ditampilkan.

6.1.32. Melihat Sebuah Laporan Komentar

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL
<https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *comment report management* yang ada di *navigation bar*.
4. Mengklik tombol *view* pada sebuah laporan.
5. Memastikan bahwa halaman berisi informasi laporan ditampilkan.

6.1.33. Melihat Semua Laporan untuk Satu Komentar

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL
<https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.

3. Mengunjungi halaman berita dari sebuah artikel berita.
4. Mengklik tombol *comment management*.
5. Mengklik tombol *view* pada komentar yang ingin dilihat laporan-laporannya.
6. Mengklik tombol *view reports*.
7. Memastikan bahwa halaman yang berisi laporan-laporan untuk satu komentar ditampilkan.

6.1.34. Menghapus Sebuah Laporan Komentar

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *comment report management* yang ada di *navigation bar*.
4. Mengklik tombol *view* pada laporan yang ingin dihapus.
5. Mengklik tombol *delete*.
6. Memastikan bahwa laporan dihapuskan dari sistem.

6.1.35. Menampilkan Daftar User

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL
<https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *user management* yang ada di *navigation bar*.
4. Memastikan bahwa halaman yang berisi daftar *user* ditampilkan.

6.1.36. Membuat *User* Baru

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL
<https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *user management* yang ada di *navigation bar*.
4. Mengklik tombol '+’.
5. Mengisi informasi untuk *user* baru yang akan dibuat, kemudian mengklik tombol *create user*.
6. Memastikan bahwa *user* dibuat dan disimpan di dalam sistem.

6.1.37. Menampilkan Informasi Sebuah *User*

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL
<https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.

2. Melakukan *login*.
3. Mengklik *link* menuju halaman *user management* yang ada di *navigation bar*.
4. Mengklik tombol *view* pada *user* yang ingin ditampilkan informasinya.
5. Memastikan bahwa halaman yang berisi informasi dari *user* ditampilkan.

6.1.38. Memperbarui Data Sebuah User

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *user management* yang ada di *navigation bar*.
4. Mengklik tombol *view* pada *user* yang ingin diperbarui datanya.
5. Memperbarui data *user* dan mengklik tombol *save*.
6. Memastikan bahwa data *user* diperbarui dan disimpan di dalam sistem.

6.1.39. Menampilkan Daftar Komentar untuk Sebuah Berita

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.

2. Melakukan *login*.
3. Mengunjungi halaman berita dari sebuah artikel berita.
4. Mengklik tombol *comment management*.
5. Memastikan bahwa halaman yang berisi daftar komentar dari berita ditampilkan.

6.1.40. Menampilkan Sebuah Komentar

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengunjungi halaman berita dari sebuah artikel berita.
4. Mengklik tombol *comment management*.
5. Mengklik tombol *view* pada komentar yang ingin ditampilkan informasinya.
6. Memastikan bahwa halaman yang berisi informasi dari komentar ditampilkan.

6.1.41. Menampilkan Daftar Balasan untuk Sebuah Komentar

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.

3. Mengunjungi halaman berita dari sebuah artikel berita.
4. Mengklik tombol *comment management*.
5. Mengklik tombol *view* pada komentar yang ingin dilihat daftar balasannya.
6. Mengklik tombol *view replies*.
7. Memastikan bahwa halaman yang berisi daftar balasan untuk komentar ditampilkan.

6.1.42. Menghapus Komentar *User* Lain

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengunjungi halaman berita dari sebuah artikel berita.
4. Mengklik tombol *comment management*.
5. Mengklik tombol *view* pada komentar yang ingin dihapus.
6. Mengklik tombol *delete*.
7. Memastikan bahwa komentar dihapuskan dari sistem.

6.1.43. Menampilkan Daftar Berita

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.

2. Melakukan *login*.
3. Mengklik *link* menuju halaman *news management* yang ada di *navigation bar*.
4. Memastikan bahwa halaman yang berisi daftar berita ditampilkan.

6.1.44. Membuat Berita Baru

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *news management* yang ada di *navigation bar*.
4. Mengklik tombol '+’.
5. Mengisi data dari berita baru yang akan dibuat dan mengklik tombol *create news*.
6. Memastikan bahwa berita dibuat dan disimpan di dalam sistem.

6.1.45. Mengedit Sebuah Berita

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *news management* yang ada di *navigation bar*.

4. Mengklik tombol *view* pada berita yang ingin diedit.
5. Memperbarui isi berita dan mengklik tombol *save*.
6. Memastikan bahwa data berita diperbarui dan disimpan di dalam sistem.

6.1.46. Menghapus Sebuah Berita

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *news management* yang ada di *navigation bar*.
4. Mengklik tombol *view* pada berita yang ingin dihapus.
5. Mengklik tombol *delete*.
6. Memastikan bahwa berita dihapuskan dari sistem.

6.1.47. Mengupload File untuk Sebuah Berita

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *news management* yang ada di *navigation bar*.

4. Mengklik tombol *view* pada berita yang ingin diupload *filenya*.
5. Mengklik tombol *choose file* untuk memilih *file* untuk diupload.
6. Mengklik tombol *upload file* untuk mengupload *file*.
7. Memastikan bahwa *file* berhasil diupload dan disimpan di dalam sistem.

6.1.48. Menghapus File yang Diupload untuk Sebuah Berita

Skenario untuk menguji aplikasi adalah sebagai berikut:

1. Mengunjungi URL <https://winnicodenewsportal.rf.gd> untuk menampilkan *homepage*.
2. Melakukan *login*.
3. Mengklik *link* menuju halaman *news management* yang ada di *navigation bar*.
4. Mengklik tombol *view* pada berita yang ingin dihapus *filenya*.
5. Mengklik tombol *delete file* pada *file* yang ingin dihapus.
6. Memastikan bahwa file berhasil dihapuskan dari sistem.

6.2. Evaluasi Pengujian

Hasil evaluasi dari pengujian fungsionalitas yang telah dilakukan dapat dilihat pada Tabel 6.1.

Tabel 6.1 Hasil Evaluasi Pengujian Fungsionalitas

No	Kebutuhan	Uji Coba	Hasil
UC001	Menampilkan <i>homepage</i>	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd .	Berhasil
UC002	Melakukan pencarian	<i>User</i> mengetikkan kata kunci pada <i>form</i> pencarian yang ada di <i>navigation bar</i> atau di <i>homepage</i> . <i>User</i> mengklik tombol pencarian.	Berhasil
UC003	Melakukan <i>login</i>	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> mengklik tombol <i>login</i> yang ada di <i>navigation bar</i> . <i>User</i> memasukkan <i>email</i> dan <i>password</i> untuk akunya. <i>User</i> mengklik tombol <i>login</i> .	Berhasil
UC004	Melakukan registrasi	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> mengklik tombol <i>login</i> yang ada di <i>navigation bar</i> . <i>User</i> mengklik <i>link</i> menuju halaman registrasi. <i>User</i> memasukkan data untuk akun yang akan dibuat. <i>User</i> mengklik tombol <i>register</i> .	Berhasil
UC005	Melakukan <i>logout</i>	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik tombol <i>logout</i> yang ada di <i>navigation bar</i> .	Berhasil
UC006	Melakukan <i>reset password</i>	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> mengklik tombol <i>login</i> yang ada di <i>navigation bar</i> . <i>User</i>	Berhasil

		mengklik tombol <i>forgot password</i> . <i>User</i> memasukkan <i>email</i> yang digunakan untuk akunnya. <i>User</i> mengklik <i>link</i> menuju halaman untuk mengubah <i>password</i> yang ada di <i>email</i> yang dikirimkan oleh sistem. <i>User</i> memasukkan <i>password</i> baru. <i>User</i> mengklik tombol <i>reset password</i> .	
UC007	Menampilkan <i>dashboard</i>	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>dashboard</i> yang ada di <i>navigation bar</i> .	Berhasil
UC008	Memperbarui data akun	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengunjungi halaman <i>dashboard</i> . <i>User</i> memasukkan data baru pada <i>form</i> untuk memperbarui data. <i>User</i> mengklik tombol <i>update account</i> .	Berhasil
UC009	Menghapus akun	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengunjungi halaman <i>dashboard</i> . <i>User</i> memasukkan <i>password</i> pada <i>form</i> untuk menghapus akun. <i>User</i> mengklik tombol <i>delete account</i> .	Berhasil
UC010	Menampilkan berita	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd .	Berhasil

		<i>User</i> mengklik tombol <i>view</i> pada berita yang ingin ditampilkan.	
UC011	Memberikan atau menghapus <i>vote</i> pada berita	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengunjungi halaman berita dari sebuah artikel berita. <i>User</i> mengklik tombol <i>thumbs up</i> atau <i>thumbs down</i> .	Berhasil
UC012	Menampilkan <i>bookmarks</i>	<i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>bookmarks</i> yang ada di <i>navigation bar</i> .	Berhasil
UC013	Menambahkan atau menghapus berita dari <i>bookmarks</i>	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengunjungi halaman berita dari sebuah artikel berita. <i>User</i> mengklik tombol <i>add to bookmarks</i> atau <i>remove from bookmarks</i> .	Berhasil
UC014	Melihat balasan komentar	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> mengunjungi halaman berita dari sebuah artikel berita. <i>User</i> mengklik tombol <i>view replies</i> pada komentar yang ingin ditampilkan balasan-balasannya.	Berhasil
UC015	Menambahkan komentar pada berita	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengunjungi halaman berita untuk sebuah artikel berita. <i>User</i> mengetikkan isi komentar pada <i>form</i> untuk membuat komentar.	Berhasil

		<i>User</i> mengklik tombol <i>post comment</i> .	
UC016	Membalas komentar	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengunjungi halaman berita untuk sebuah artikel berita. <i>User</i> mengklik tombol <i>reply</i> pada komentar yang ingin dibalas. <i>User</i> mengetikkan isi balasan pada <i>form</i> untuk membuat balasan. <i>User</i> mengklik tombol <i>post reply</i> .	Berhasil
UC017	Mengedit komentar	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengunjungi halaman berita untuk sebuah artikel berita. <i>User</i> mengklik tombol <i>edit</i> pada komentar yang sebelumnya sudah dibuat olehnya. <i>User</i> memperbarui isi komentar. <i>User</i> mengklik tombol <i>save</i> .	Berhasil
UC018	Menghapus komentar	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengunjungi halaman berita untuk sebuah artikel berita. <i>User</i> mengklik tombol <i>delete</i> pada komentar yang sebelumnya sudah dibuat olehnya.	Berhasil
UC019	Memberikan atau menghapus <i>vote</i> pada komentar	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengunjungi halaman berita	Berhasil

		untuk sebuah artikel berita. <i>User</i> mengklik tombol <i>thumbs up</i> atau <i>thumbs down</i> pada komentar yang ingin diberikan <i>vote</i> .	
UC020	Melaporkan <i>user</i>	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengunjungi halaman berita untuk sebuah artikel berita. <i>User</i> mengklik tombol <i>report user</i> yang berada pada komentar yang dibuat oleh <i>user</i> yang ingin dilaporkan. <i>User</i> menyetikkan alasan membuat laporan. <i>User</i> mengklik tombol <i>report</i> .	Berhasil
UC021	Melaporkan berita	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengunjungi halaman berita untuk sebuah artikel berita. <i>User</i> mengklik tombol <i>report news</i> yang berada di bawah judul berita. <i>User</i> menyetikkan alasan membuat laporan. <i>User</i> mengklik tombol <i>report</i> .	Berhasil
UC022	Melaporkan komentar	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengunjungi halaman berita untuk sebuah artikel berita. <i>User</i> mengklik tombol <i>report comment</i> yang berada pada komentar yang ingin dilaporkan. <i>User</i> menyetikkan alasan membuat	Berhasil

		laporan. <i>User</i> mengklik tombol <i>report</i> .	
UC023	Menampilkan daftar laporan <i>user</i>	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>user report management</i> yang ada di <i>navigation bar</i> .	Berhasil
UC024	Melihat sebuah laporan <i>user</i>	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>user report management</i> yang ada di <i>navigation bar</i> . <i>User</i> mengklik tombol <i>view</i> pada laporan yang ingin ditampilkan.	Berhasil
UC025	Melihat semua laporan untuk satu <i>user</i>	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>user management</i> yang ada di <i>navigation bar</i> . <i>User</i> mengklik tombol <i>view</i> pada <i>user</i> yang ingin ditampilkan laporan-laporannya. <i>User</i> mengklik tombol <i>view reports</i> .	Berhasil
UC026	Menghapus sebuah laporan <i>user</i>	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>user report management</i> yang ada di <i>navigation bar</i> . <i>User</i> mengklik tombol <i>view</i> pada laporan yang ingin dihapus. <i>User</i> mengklik tombol <i>delete</i> .	Berhasil

UC027	Menampilkan daftar laporan berita	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>news report management</i> yang ada di <i>navigation bar</i> .	Berhasil
UC028	Melihat sebuah laporan berita	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>news report management</i> yang ada di <i>navigation bar</i> . <i>User</i> mengklik tombol <i>view</i> pada laporan yang ingin ditampilkan.	Berhasil
UC029	Melihat semua laporan untuk satu berita	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>news management</i> yang ada di <i>navigation bar</i> . <i>User</i> mengklik tombol <i>view</i> pada berita yang ingin ditampilkan laporan-laporannya. <i>User</i> mengklik tombol <i>view reports</i> .	Berhasil
UC030	Menghapus sebuah laporan berita	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>news report management</i> yang ada di <i>navigation bar</i> . <i>User</i> mengklik tombol <i>view</i> pada laporan yang ingin dihapus. <i>User</i> mengklik tombol <i>delete</i> .	Berhasil

UC031	Menampilkan daftar laporan komentar	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>comment report management</i> yang ada di <i>navigation bar</i> .	Berhasil
UC032	Melihat sebuah laporan komentar	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>comment report management</i> yang ada di <i>navigation bar</i> . <i>User</i> mengklik tombol <i>view</i> pada laporan yang ingin ditampilkan.	Berhasil
UC033	Melihat semua laporan untuk satu komentar	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengunjungi halaman berita dari sebuah artikel berita. <i>User</i> mengklik tombol <i>comment management</i> . <i>User</i> mengklik tombol <i>view</i> pada komentar yang ingin ditampilkan laporan-laporannya. <i>User</i> mengklik tombol <i>view reports</i> .	Berhasil
UC034	Menghapus sebuah laporan komentar	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>comment report management</i> yang ada di <i>navigation bar</i> . <i>User</i> mengklik tombol <i>view</i> pada laporan yang ingin dihapus. <i>User</i> mengklik tombol <i>delete</i> .	Berhasil

UC035	Menampilkan daftar <i>user</i>	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>user management</i> yang ada di <i>navigation bar</i> .	Berhasil
UC036	Membuat <i>user</i> baru	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>user management</i> yang ada di <i>navigation bar</i> . <i>User</i> mengklik tombol '+'. <i>User</i> mengisi data untuk <i>user</i> baru yang akan dibuat. <i>User</i> mengklik tombol <i>create user</i> .	Berhasil
UC037	Menampilkan informasi sebuah <i>user</i>	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>user management</i> yang ada di <i>navigation bar</i> . <i>User</i> mengklik tombol <i>view</i> pada <i>user</i> yang ingin ditampilkan informasinya.	Berhasil
UC038	Memperbarui data sebuah <i>user</i>	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>user management</i> yang ada di <i>navigation bar</i> . <i>User</i> mengklik tombol <i>view</i> pada <i>user</i> yang ingin diperbarui datanya. <i>User</i> memperbarui data dari <i>user</i> yang dipilih. <i>User</i> mengklik tombol <i>save</i> .	Berhasil

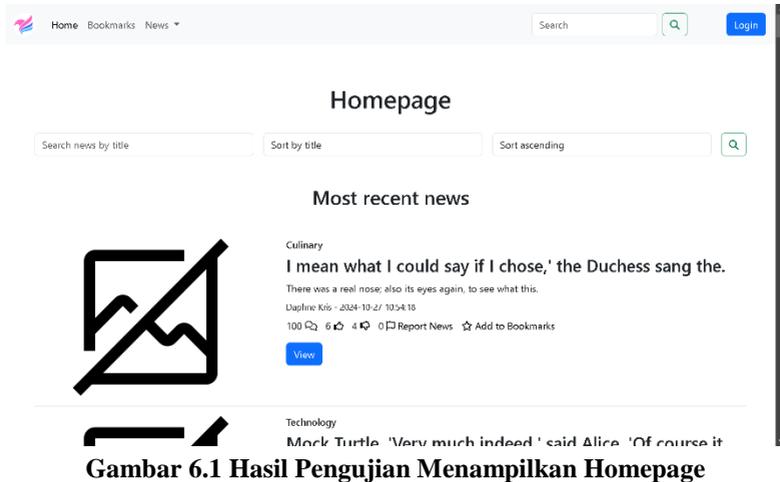
UC039	Menampilkan daftar komentar untuk sebuah berita	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengunjungi halaman berita dari sebuah artikel berita. <i>User</i> mengklik tombol <i>comment management</i> .	Berhasil
UC040	Menampilkan sebuah komentar	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengunjungi halaman berita dari sebuah artikel berita. <i>User</i> mengklik tombol <i>comment management</i> . <i>User</i> mengklik tombol <i>view</i> pada komentar yang ingin ditampilkan.	Berhasil
UC041	Menampilkan daftar balasan untuk sebuah komentar	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengunjungi halaman berita dari sebuah artikel berita. <i>User</i> mengklik tombol <i>comment management</i> . <i>User</i> mengklik tombol <i>view</i> pada komentar yang ingin ditampilkan balasannya. <i>User</i> mengklik tombol <i>view replies</i> .	Berhasil
UC042	Menghapus komentar <i>user</i> lain	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengunjungi halaman berita dari sebuah artikel berita. <i>User</i> mengklik tombol <i>comment management</i> . <i>User</i> mengklik tombol <i>view</i> pada komentar yang	Berhasil

		ingin dihapus. <i>User</i> mengklik tombol <i>delete</i> .	
UC043	Menampilkan daftar berita	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>news management</i> yang ada di <i>navigation bar</i> .	Berhasil
UC044	Membuat berita baru	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>news management</i> yang ada di <i>navigation bar</i> . <i>User</i> mengklik tombol '+'. <i>User</i> mengisi data untuk berita baru yang akan dibuat. <i>User</i> mengklik tombol <i>create news</i> .	Berhasil
UC045	Mengedit sebuah berita	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>news management</i> yang ada di <i>navigation bar</i> . <i>User</i> mengklik tombol <i>view</i> pada berita yang ingin diedit. <i>User</i> memperbarui data dari berita yang dipilih. <i>User</i> mengklik tombol <i>save</i> .	Berhasil
UC046	Menghapus sebuah berita	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>news management</i> yang ada di <i>navigation bar</i> . <i>User</i> mengklik tombol <i>view</i> pada berita yang	Berhasil

		ingin dihapus. <i>User</i> mengklik tombol <i>delete</i> .	
UC047	Mengupload <i>File</i> untuk Sebuah Berita	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>news management</i> yang ada di <i>navigation bar</i> . <i>User</i> mengklik tombol <i>view</i> pada berita yang ingin diupload <i>filenya</i> . <i>User</i> mengklik tombol <i>choose file</i> untuk memilih <i>file</i> yang ingin diupload. <i>User</i> mengklik tombol <i>upload file</i> untuk mengupload <i>file</i> yang dipilih.	Berhasil
UC048	Menghapus <i>File</i> yang Diupload untuk Sebuah Berita	<i>User</i> mengunjungi https://winnicodenewsportal.rf.gd . <i>User</i> melakukan <i>login</i> . <i>User</i> mengklik <i>link</i> menuju halaman <i>news management</i> yang ada di <i>navigation bar</i> . <i>User</i> mengklik tombol <i>view</i> pada berita yang ingin dihapus <i>filenya</i> . <i>User</i> mengklik tombol <i>delete file</i> pada <i>file</i> yang ingin dihapus.	Berhasil

6.2.1. Menampilkan *Homepage*

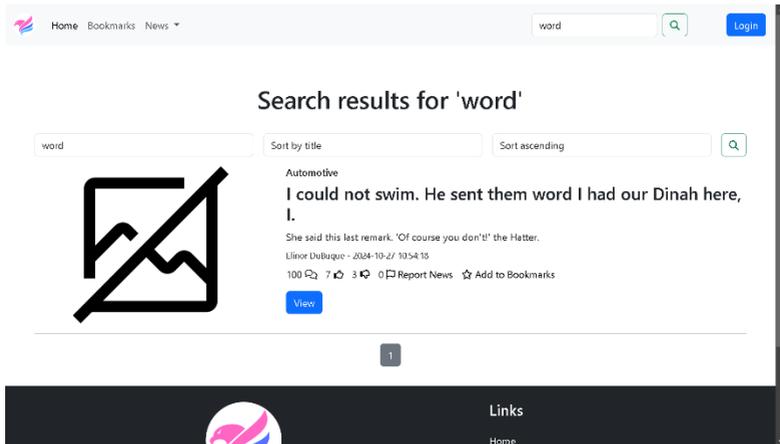
Hasil untuk pengujian menampilkan *homepage* dapat dilihat pada Gambar 6.1.



Gambar 6.1 Hasil Pengujian Menampilkan Homepage

6.2.2. Melakukan Pencarian

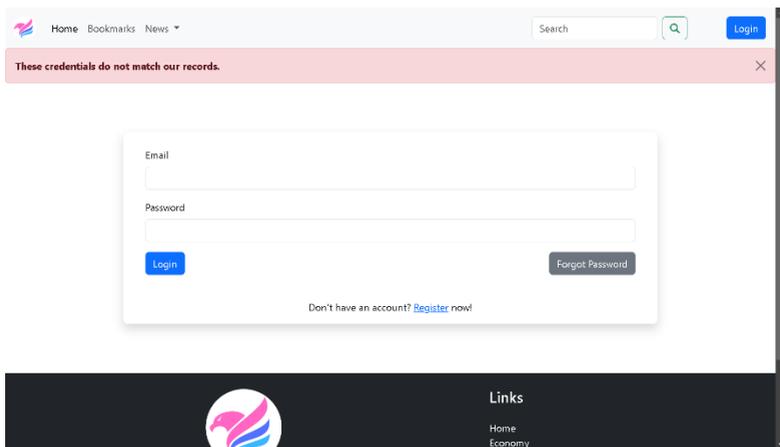
Hasil untuk pengujian melakukan pencarian dapat dilihat pada Gambar 6.2.



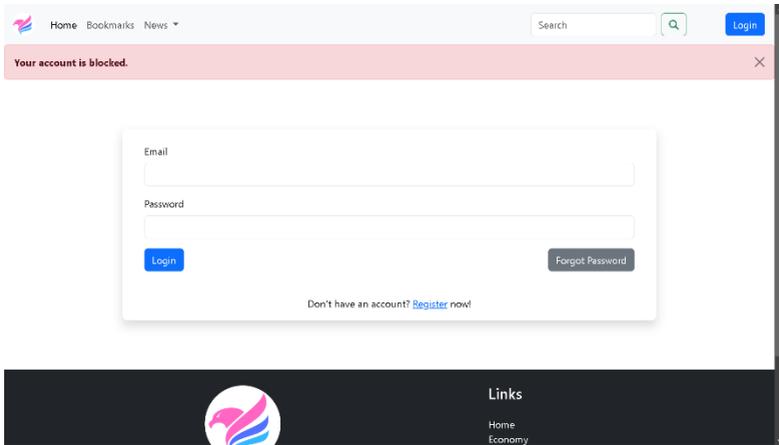
Gambar 6.2 Hasil Pengujian Melakukan Pencarian

6.2.3. Melakukan *Login*

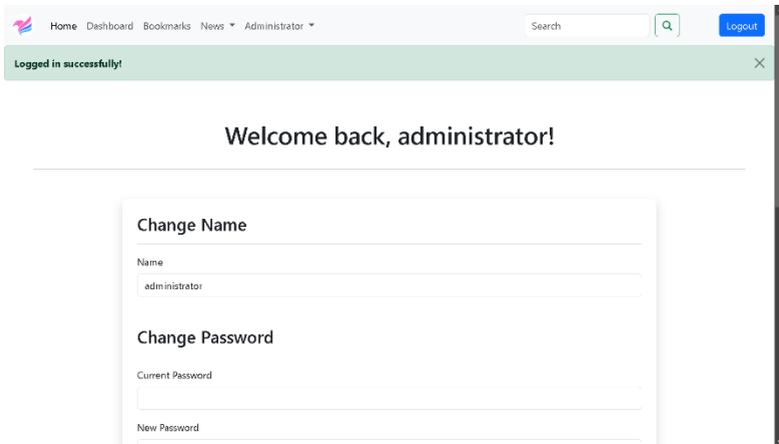
Hasil untuk pengujian melakukan *login* yang gagal karena *email* dan *password* yang tidak sesuai dapat dilihat pada Gambar 6.3, *login* yang gagal karena akun diblokir dapat dilihat pada Gambar 6.4, dan *login* yang berhasil dapat dilihat pada Gambar 6.5.



Gambar 6.3 Hasil Pengujian Melakukan Login Gagal karena Email dan Password Tidak Sesuai



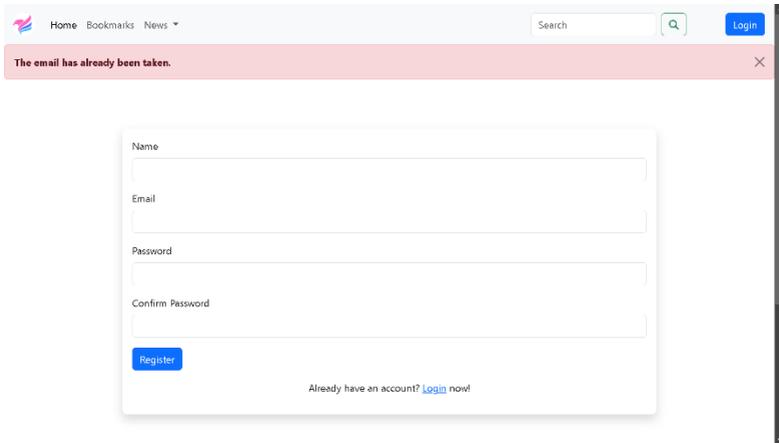
Gambar 6.4 Hasil Pengujian Melakukan Login Gagal karena Akun Diblokir



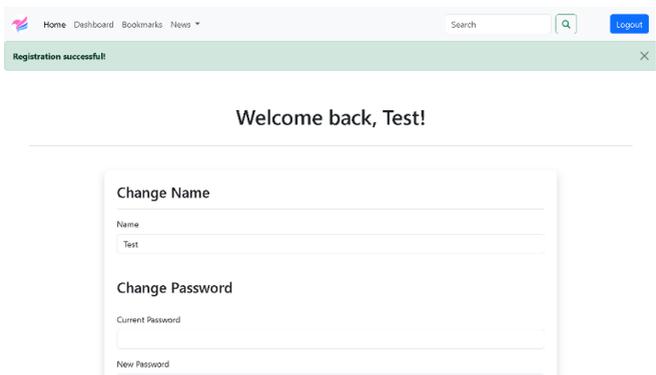
Gambar 6.5 Hasil Pengujian Melakukan Login yang Berhasil

6.2.4. Melakukan Registrasi

Hasil untuk pengujian melakukan registrasi yang gagal dapat dilihat pada Gambar 6.6, dan registrasi yang berhasil dapat dilihat pada Gambar 6.7.



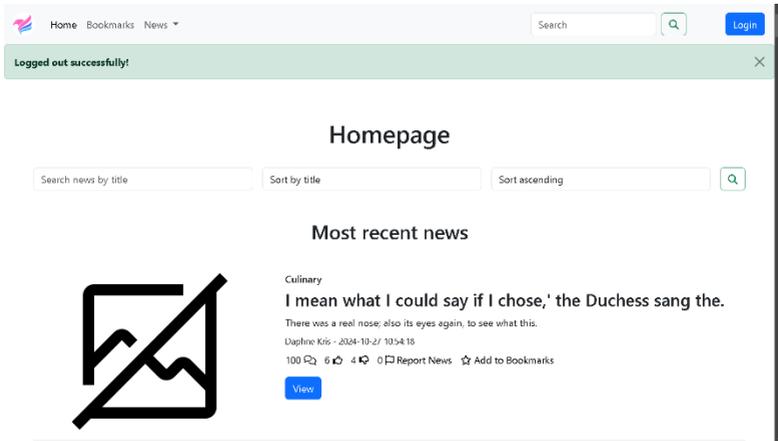
Gambar 6.6 Hasil Pengujian Melakukan Registrasi yang Gagal



Gambar 6.7 Hasil Pengujian Melakukan Registrasi yang Berhasil

6.2.5. Melakukan *Logout*

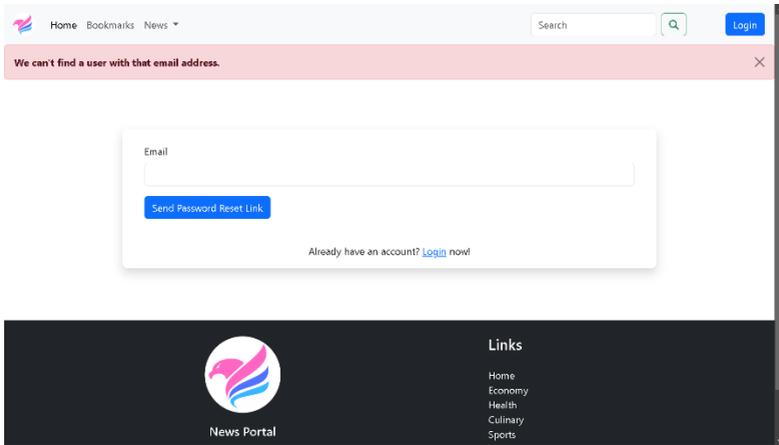
Hasil untuk pengujian melakukan *logout* dapat dilihat pada Gambar 6.8.



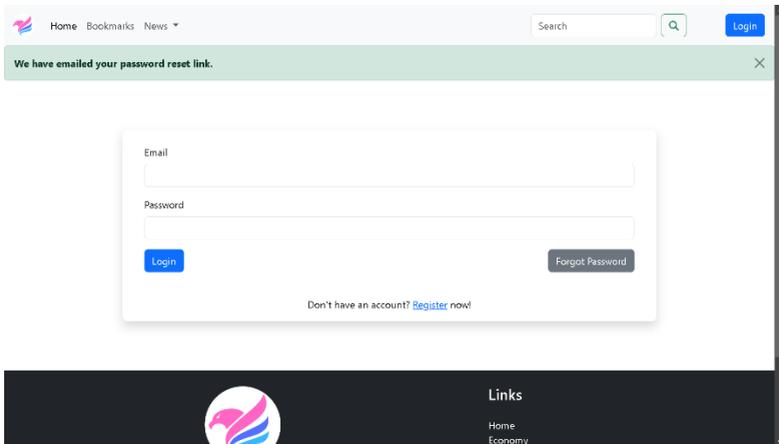
Gambar 6.8 Hasil Pengujian Melakukan Logout

6.2.6. Melakukan *Reset Password*

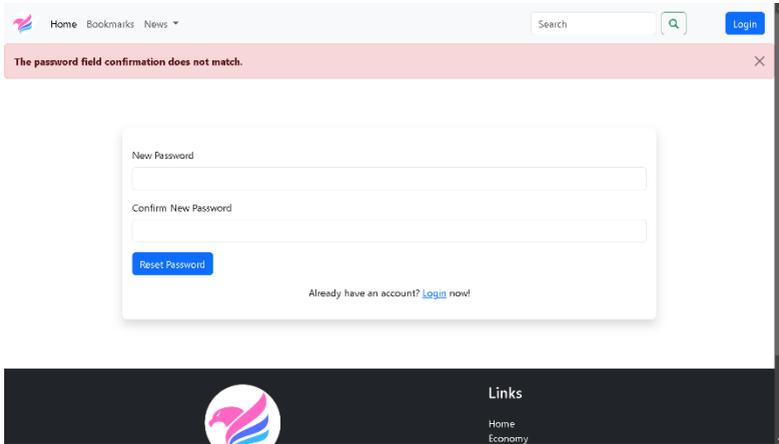
Hasil untuk pengujian melakukan *reset password* yang gagal karena *email* yang tidak terdaftar dapat dilihat pada Gambar 6.9, *reset password* berhasil mengirimkan *email* yang berisi *link* untuk mengubah *password* pada Gambar 6.10, *reset password* yang gagal karena *password* baru yang tidak sesuai dapat dilihat pada Gambar 6.11, dan *reset password* yang berhasil pada Gambar 6.12.



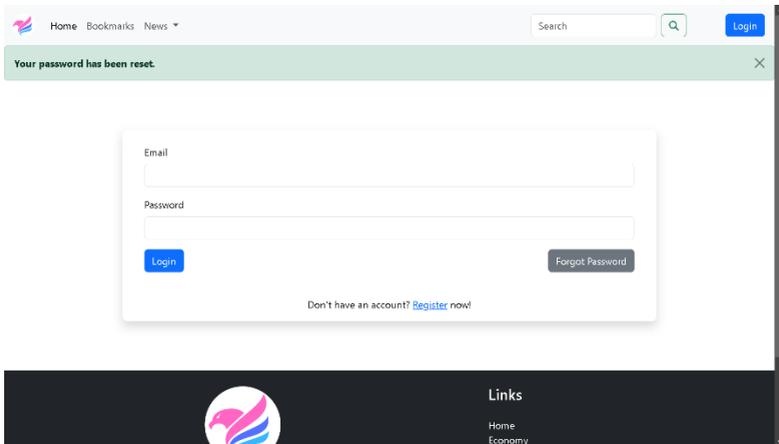
Gambar 6.9 Hasil Pengujian Melakukan Reset Password Gagal karena Email Tidak Ditemukan



Gambar 6.10 Hasil Pengujian Melakukan Reset Password Email Ditemukan



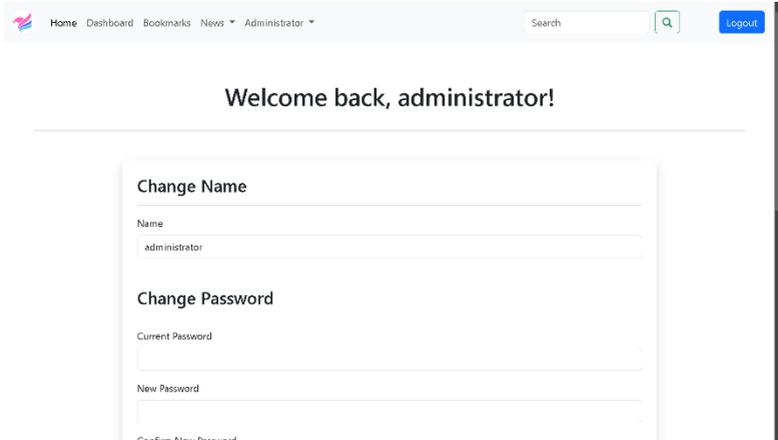
Gambar 6.11 Hasil Pengujian Melakukan Reset Password Gagal karena Password Tidak Sama



Gambar 6.12 Hasil Pengujian Melakukan Reset Password Berhasil

6.2.7. Menampilkan *Dashboard*

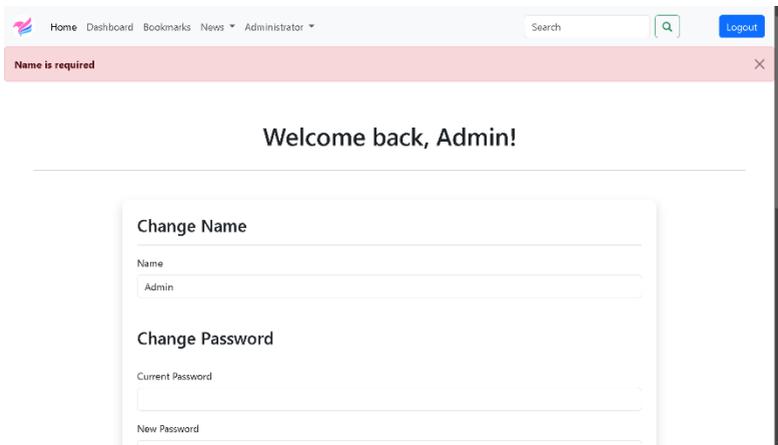
Hasil untuk pengujian menampilkan *dashboard* dapat dilihat pada Gambar 6.13.



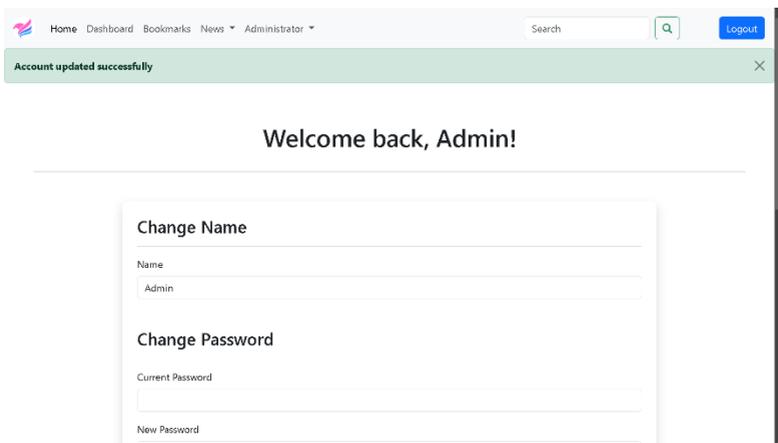
Gambar 6.13 Hasil Pengujian Menampilkan Dashboard

6.2.8. Memperbarui Data Akun

Hasil untuk pengujian memperbarui data akun yang gagal dapat dilihat pada Gambar 6.14, dan yang berhasil pada Gambar 6.15.



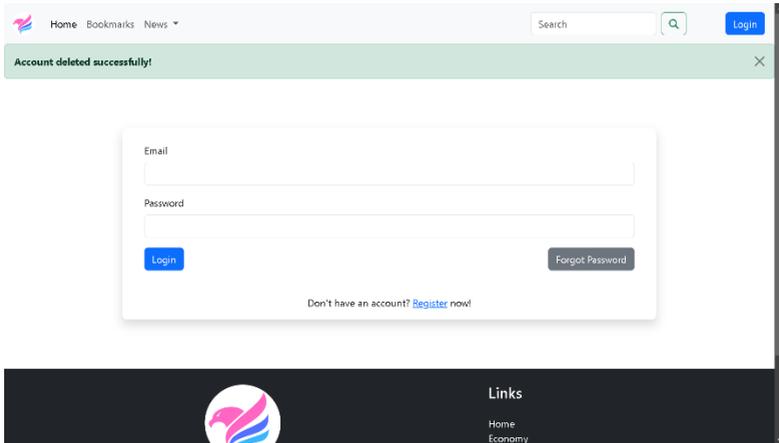
Gambar 6.14 Hasil Pengujian Memperbarui Data Akun Gagal



Gambar 6.15 Hasil Pengujian Memperbarui Data Akun Berhasil

6.2.9. Menghapus Akun

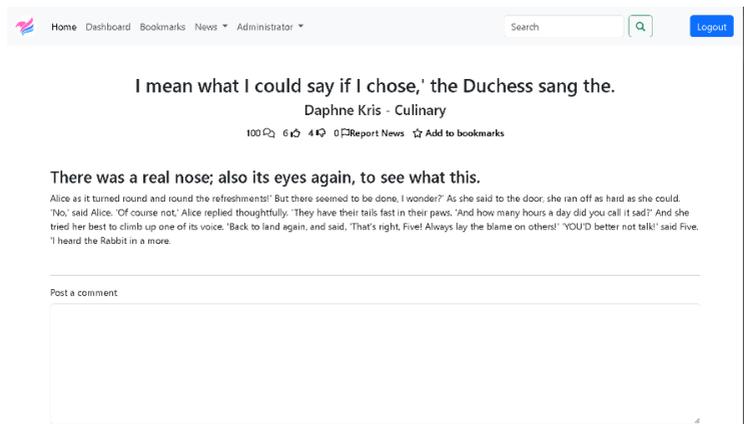
Hasil untuk pengujian menghapus akun dapat dilihat pada Gambar 6.16.



Gambar 6.16 Hasil Pengujian Menghapus Akun

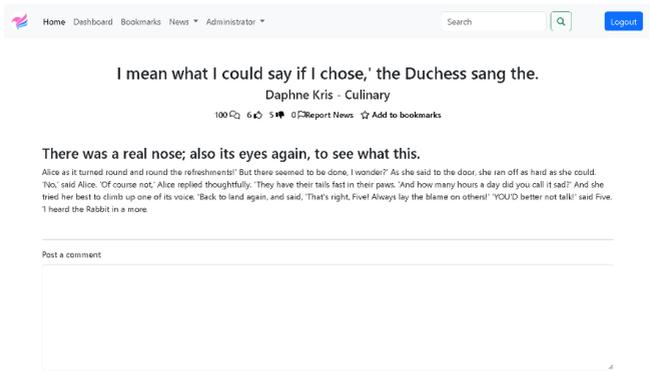
6.2.10. Menampilkan Berita

Hasil untuk pengujian menampilkan berita dapat dilihat pada Gambar 6.17.

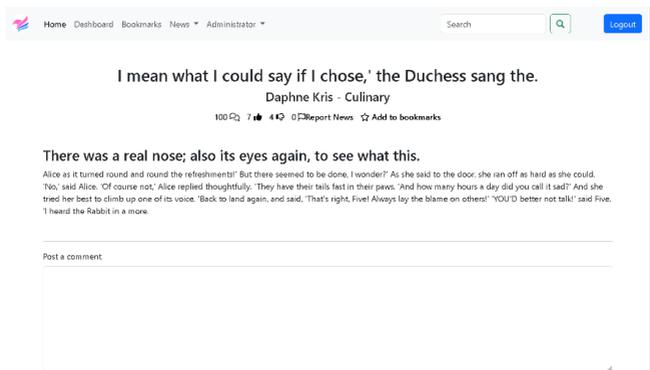


Gambar 6.17 Hasil Pengujian Menampilkan Berita

6.2.11. Memberikan atau Menghapus *Vote* pada Berita Hasil untuk pengujian memberikan atau menghapus *vote* pada berita dapat dilihat pada Gambar 6.18 dan Gambar 6.19.



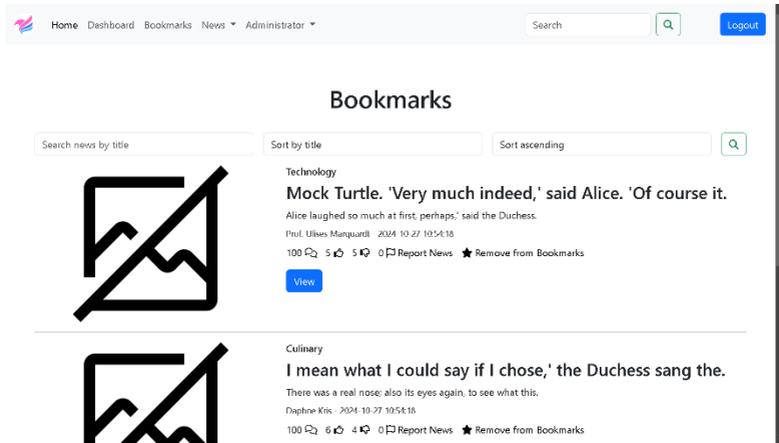
Gambar 6.18 Hasil Pengujian Memberikan atau Menghapus Vote pada Berita 1



Gambar 6.19 Hasil Pengujian Memberikan atau Menghapus Vote pada Berita 2

6.2.12. Menampilkan *Bookmarks*

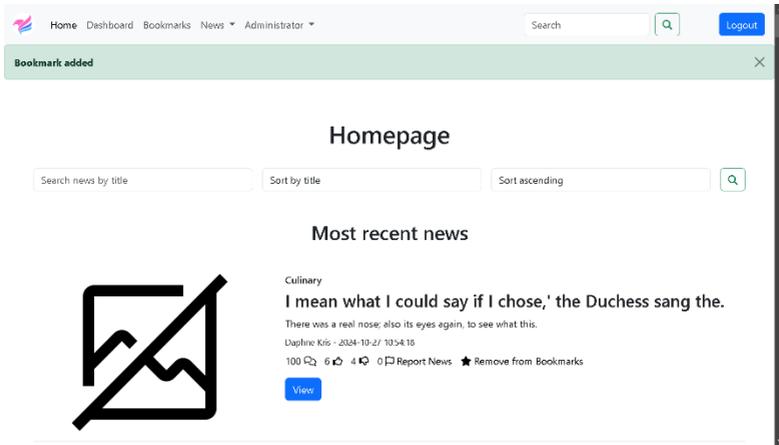
Hasil untuk pengujian menampilkan *bookmarks* dapat dilihat pada Gambar 6.20.



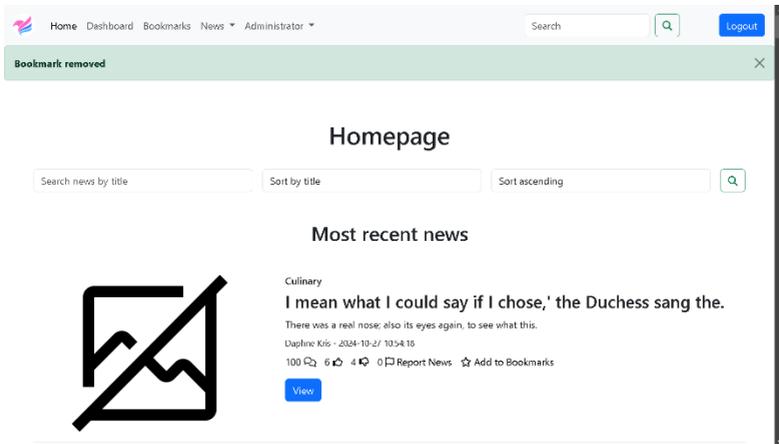
Gambar 6.20 Hasil Pengujian Menampilkan *Bookmarks*

6.2.13. Menambahkan atau Menghapus Berita dari *Bookmarks*

Hasil untuk pengujian menambahkan atau menghapus berita dari *bookmarks* dapat dilihat pada Gambar 6.21 dan Gambar 6.22.



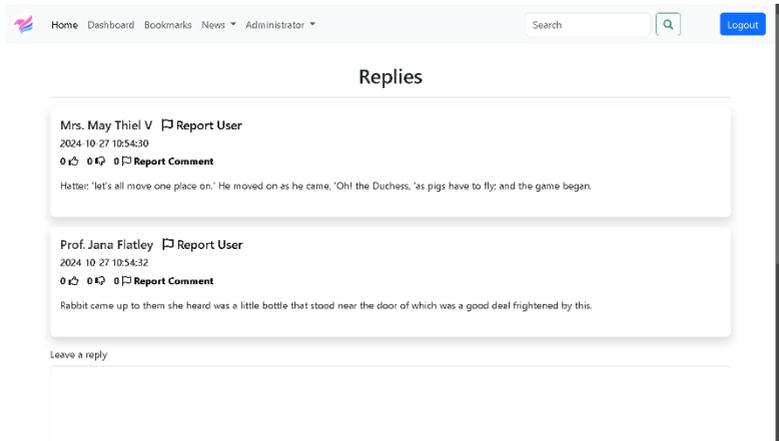
Gambar 6.21 Hasil Pengujian Menambahkan atau Menghapus Berita dari Bookmarks 1



Gambar 6.22 Hasil Pengujian Menambahkan atau Menghapus Berita dari Bookmarks 2

6.2.14. Melihat Balasan Komentar

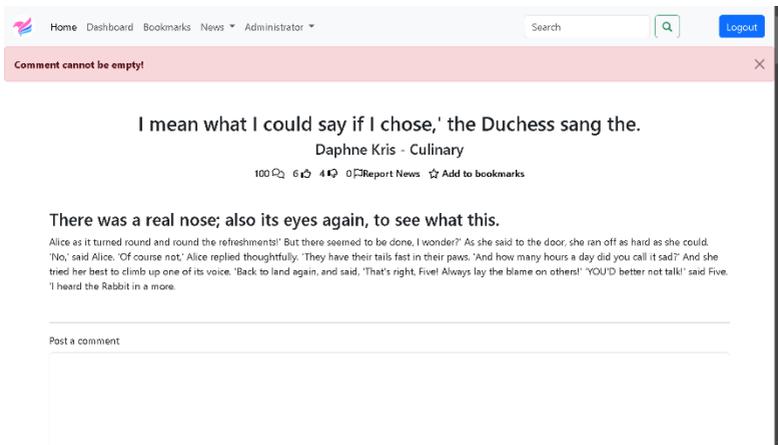
Hasil untuk pengujian melihat balasan komentar dapat dilihat pada Gambar 6.23.



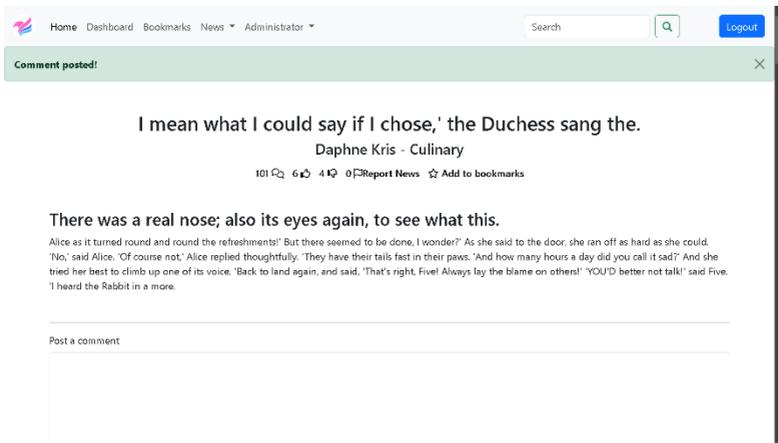
Gambar 6.23 Hasil Pengujian Melihat Balasan Komentar

6.2.15. Menambahkan Komentar pada Berita

Hasil untuk pengujian menambahkan komentar pada berita yang gagal dapat dilihat pada Gambar 6.24, dan yang berhasil pada Gambar 6.25.



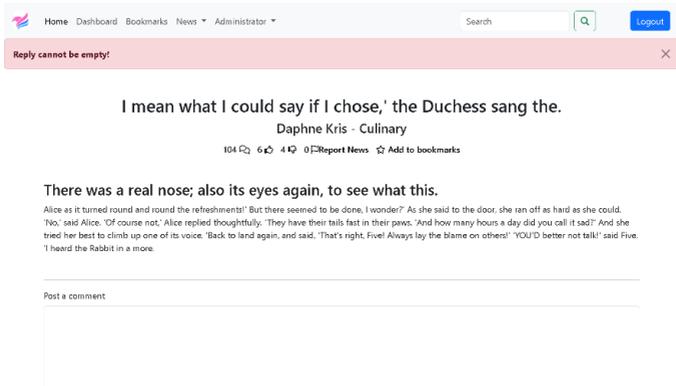
Gambar 6.24 Hasil Pengujian Menambahkan Komentar pada Berita Gagal



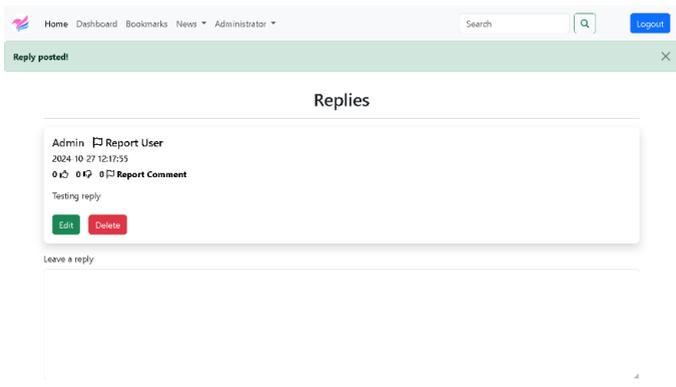
Gambar 6.25 Hasil Pengujian Menambahkan Komentar pada Berita Berhasil

6.2.16. Membalas Komentar

Hasil untuk pengujian membalas komentar yang gagal dapat dilihat pada Gambar 6.26, dan yang berhasil pada Gambar 6.27.



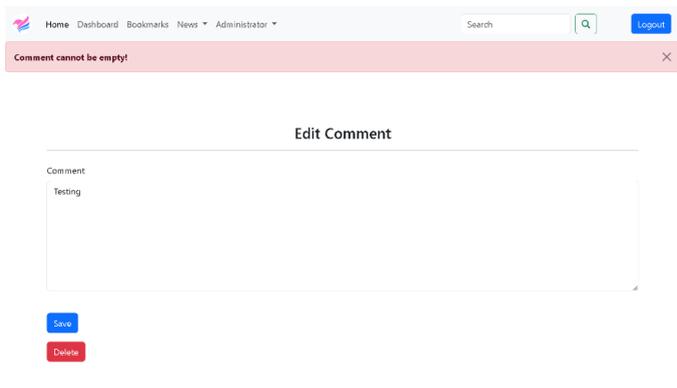
Gambar 6.26 Hasil Pengujian Membalas Komentar Gagal



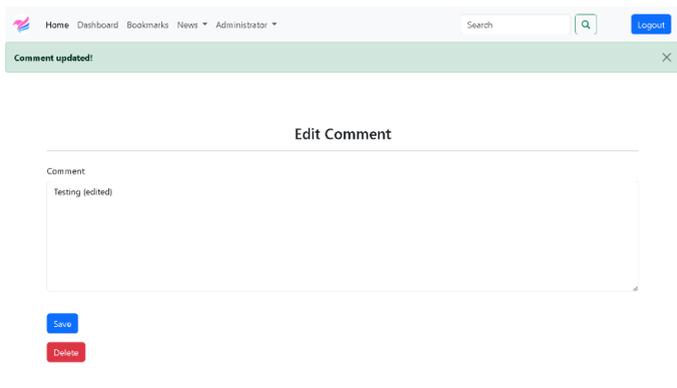
Gambar 6.27 Hasil Pengujian Membalas Komentar Berhasil

6.2.17. Mengedit Komentar

Hasil untuk pengujian mengedit komentar yang gagal dapat dilihat pada Gambar 6.28, dan yang berhasil pada Gambar 6.29.



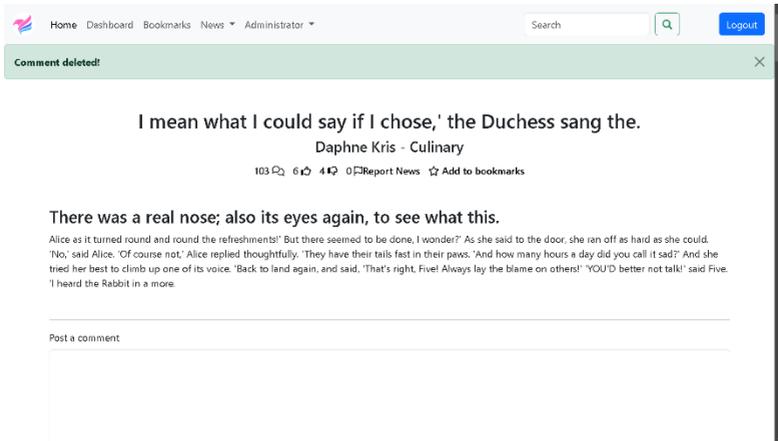
Gambar 6.28 Hasil Pengujian Mengedit Komentar Gagal



Gambar 6.29 Hasil Pengujian Mengedit Komentar Berhasil

6.2.18. Menghapus Komentar

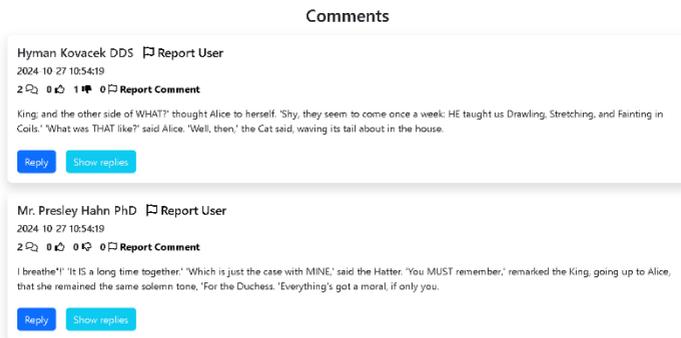
Hasil untuk pengujian menghapus komentar dapat dilihat pada Gambar 6.30.



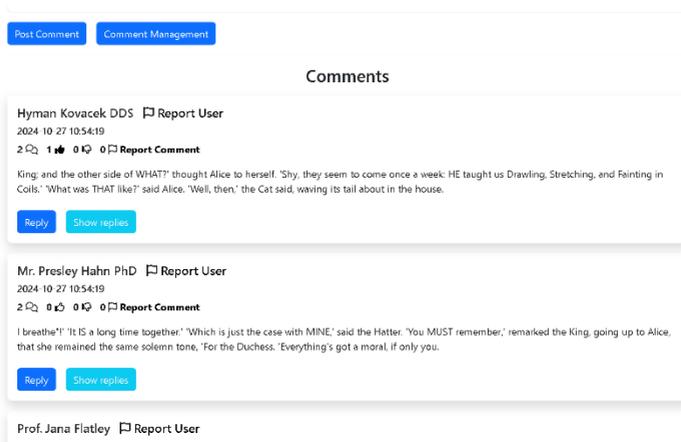
Gambar 6.30 Hasil Pengujian Menghapus Komentar

6.2.19. Memberikan atau Menghapus *Vote* pada Komentar

Hasil untuk pengujian memberikan atau menghapus *vote* pada komentar dapat dilihat pada Gambar 6.31 dan Gambar 6.32.



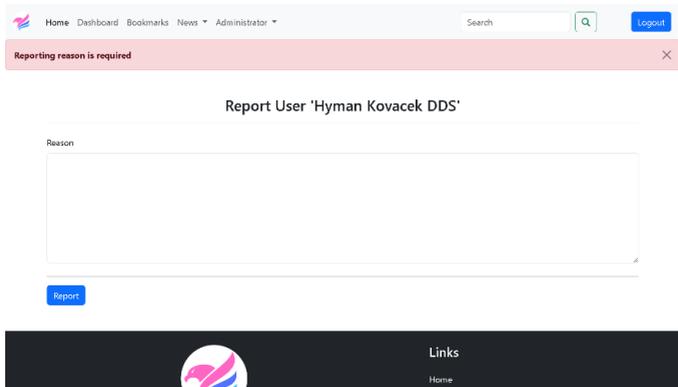
Gambar 6.31 Hasil Pengujian Memberikan atau Menghapus Vote pada Komentar 1



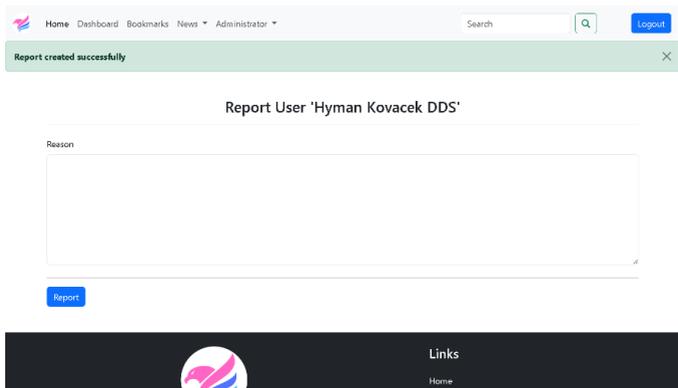
Gambar 6.32 Hasil Pengujian Memberikan atau Menghapus Vote pada Komentar 2

6.2.20. Melaporkan *User*

Hasil untuk pengujian melaporkan *user* yang gagal dapat dilihat pada Gambar 6.33, dan yang berhasil pada Gambar 6.34.



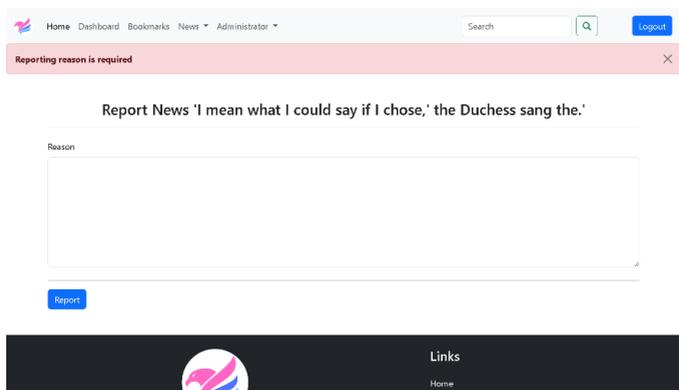
Gambar 6.33 Hasil Pengujian Melaporkan User Gagal



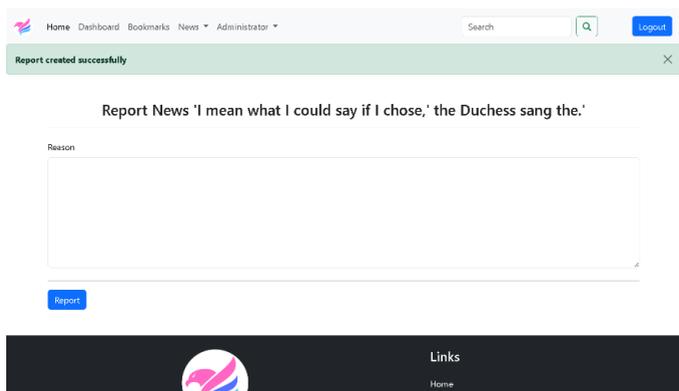
Gambar 6.34 Hasil Pengujian Melaporkan User Berhasil

6.2.21. Melaporkan Berita

Hasil untuk pengujian melaporkan berita yang gagal dapat dilihat pada Gambar 6.35, dan yang berhasil pada Gambar 6.36.



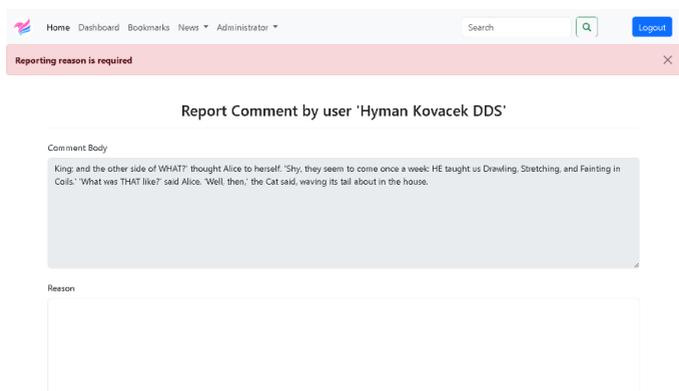
Gambar 6.35 Hasil Pengujian Melaporkan Berita Gagal



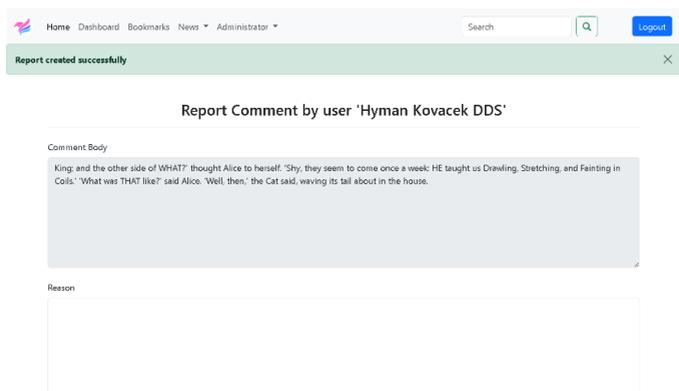
Gambar 6.36 Hasil Pengujian Melaporkan Berita Berhasil

6.2.22. Melaporkan Komentar

Hasil untuk pengujian melaporkan komentar yang gagal dapat dilihat pada Gambar 6.37, dan yang berhasil pada Gambar 6.38.



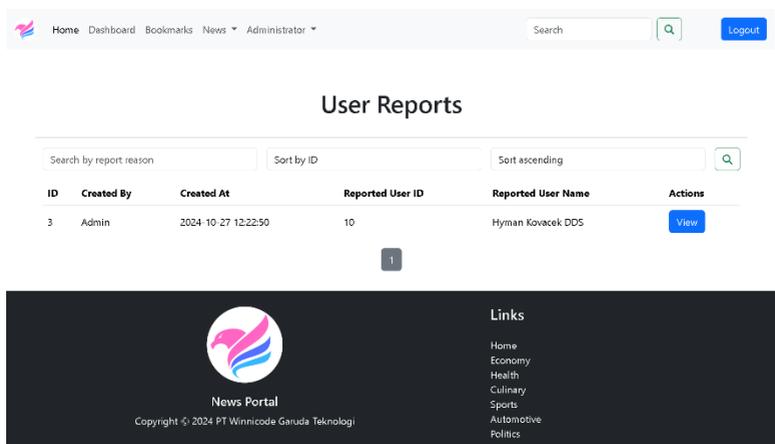
Gambar 6.37 Hasil Pengujian Melaporkan Komentar Gagal



Gambar 6.38 Hasil Pengujian Melaporkan Komentar Berhasil

6.2.23. Menampilkan Daftar Laporan *User*

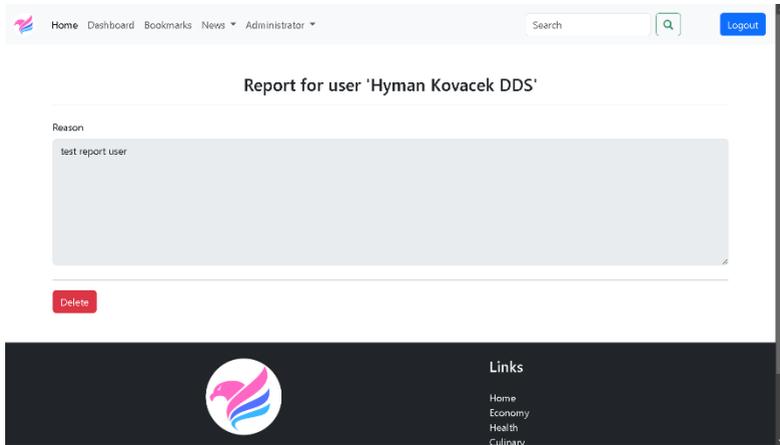
Hasil untuk pengujian melihat daftar laporan *user* dapat dilihat pada Gambar 6.39.



Gambar 6.39 Hasil Pengujian Menampilkan Daftar Laporan User

6.2.24. Melihat Sebuah Laporan *User*

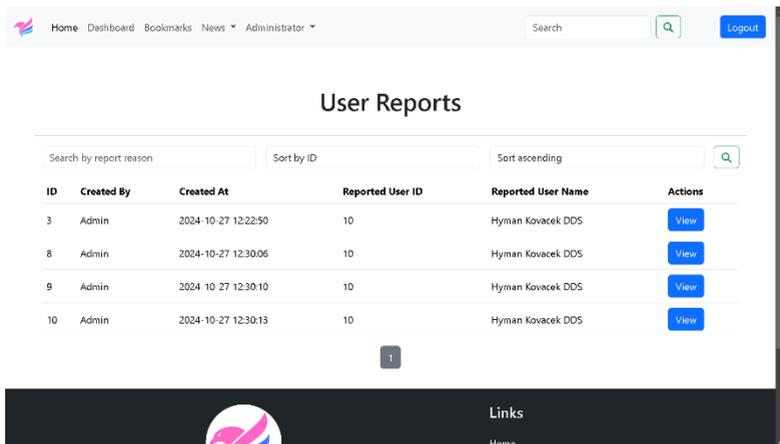
Hasil untuk pengujian melihat sebuah laporan *user* dapat dilihat pada Gambar 6.40.



Gambar 6.40 Hasil Pengujian Melihat Sebuah Laporan User

6.2.25. Melihat Semua Laporan untuk Satu *User*

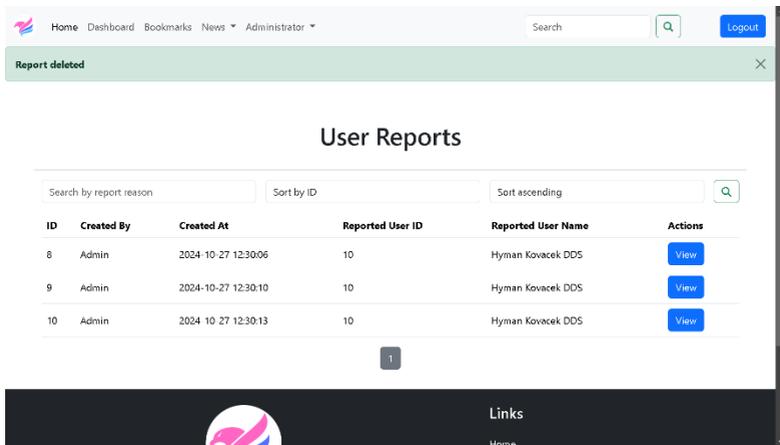
Hasil untuk pengujian melihat semua laporan untuk satu *user* dapat dilihat pada Gambar 6.41.



Gambar 6.41 Hasil Pengujian Melihat Semua Laporan untuk Satu User

6.2.26. Menghapus Sebuah Laporan *User*

Hasil untuk pengujian menghapus sebuah laporan *user* dapat dilihat pada Gambar 6.42.



Gambar 6.42 Hasil Pengujian Menghapus Sebuah Laporan User

6.2.27. Menampilkan Daftar Laporan Berita

Hasil untuk pengujian menampilkan daftar laporan berita dapat dilihat pada Gambar 6.43.

ID	Created By	Created At	Reported News ID	Reported News Title	Actions
4	Admin	2024-10-27 12:31:24	50	I mean what I could say if I chose,' the Duchess sang the.	View
11	Admin	2024-10-27 12:31:24	50	I mean what I could say if I chose,' the Duchess sang the.	View
12	Admin	2024-10-27 12:31:27	50	I mean what I could say if I chose,' the Duchess sang the.	View
13	Admin	2024-10-27 12:31:30	50	I mean what I could say if I chose,' the Duchess sang the.	View
14	Admin	2024-10-27 12:31:39	49	Mock Turtle. 'Very much indeed,' said Alice. 'Of course it.	View
15	Admin	2024-10-27 12:31:44	49	Mock Turtle. 'Very much indeed,' said Alice. 'Of course it.	View

Gambar 6.43 Hasil Pengujian Menampilkan Daftar Laporan Berita

6.2.28. Melihat Sebuah Laporan Berita

Hasil untuk pengujian melihat sebuah laporan berita dapat dilihat pada Gambar 6.44.

Report for news 'I mean what I could say if I chose,' the Duchess sang the.'

Reason

Test report news

[Delete](#)

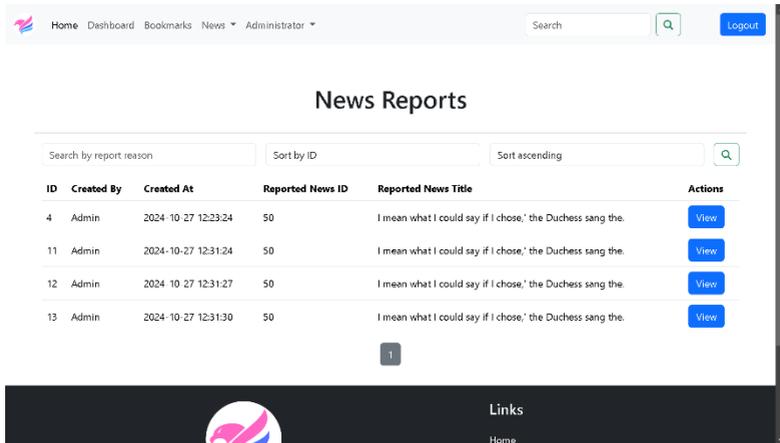
Links

- Home
- Economy
- Health
- Culinary

Gambar 6.44 Hasil Pengujian Melihat Sebuah Laporan Berita

6.2.29. Melihat Semua Laporan untuk Satu Berita

Hasil untuk pengujian melihat semua laporan untuk satu berita dapat dilihat pada Gambar 6.45.

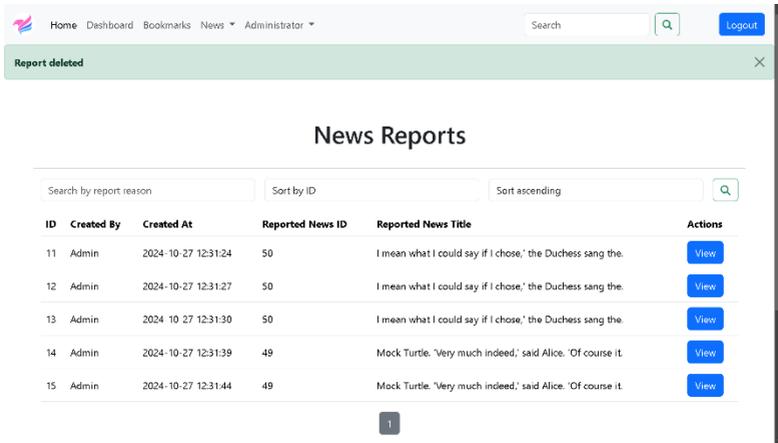


ID	Created By	Created At	Reported News ID	Reported News Title	Actions
4	Admin	2024-10-27 12:23:24	50	I mean what I could say if I chose,' the Duchess sang the.	View
11	Admin	2024-10-27 12:31:24	50	I mean what I could say if I chose,' the Duchess sang the.	View
12	Admin	2024-10-27 12:31:27	50	I mean what I could say if I chose,' the Duchess sang the.	View
13	Admin	2024-10-27 12:31:30	50	I mean what I could say if I chose,' the Duchess sang the.	View

Gambar 6.45 Hasil Pengujian Melihat Semua Laporan untuk Satu Berita

6.2.30. Menghapus Sebuah Laporan Berita

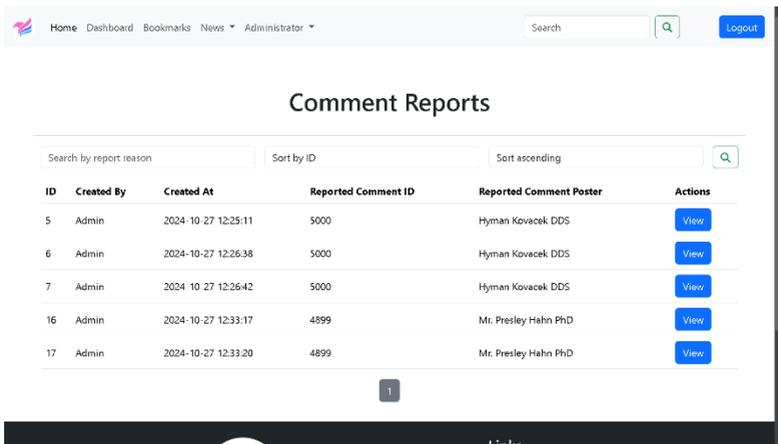
Hasil untuk pengujian menghapus sebuah laporan berita dapat dilihat pada Gambar 6.46.



Gambar 6.46 Hasil Pengujian Menghapus Sebuah Laporan Berita

6.2.31. Menampilkan Daftar Laporan Komentar

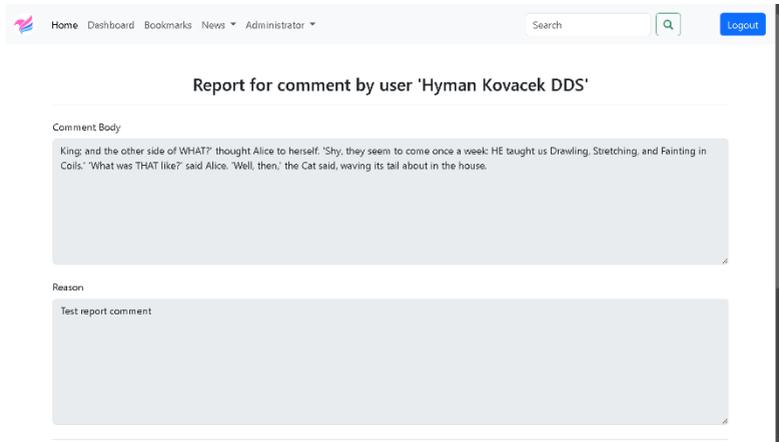
Hasil untuk pengujian menampilkan daftar laporan komentar dapat dilihat pada Gambar 6.47.



Gambar 6.47 Hasil Pengujian Menampilkan Daftar Laporan Komentar

6.2.32. Melihat Sebuah Laporan Komentar

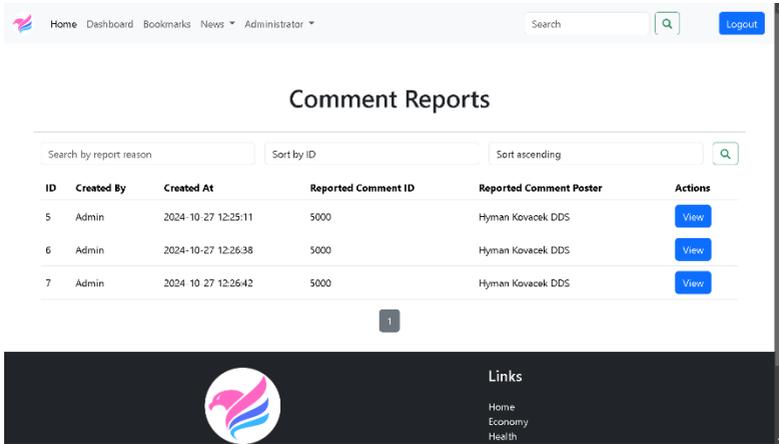
Hasil untuk pengujian melihat sebuah laporan komentar dapat dilihat pada Gambar 6.48.



Gambar 6.48 Hasil Pengujian Melihat Sebuah Laporan Komentar

6.2.33. Melihat Semua Laporan untuk Satu Komentar

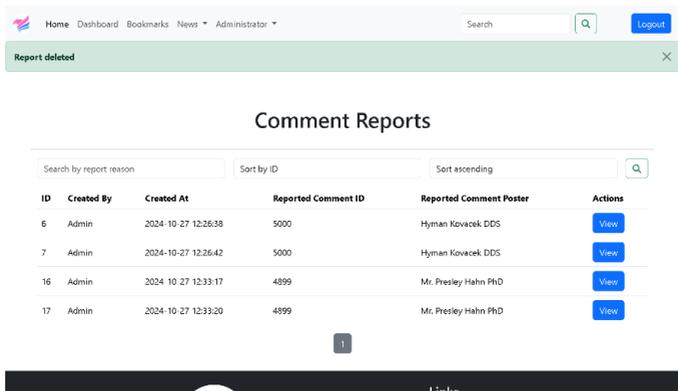
Hasil untuk pengujian melihat semua laporan untuk satu komentar dapat dilihat pada Gambar 6.49.



Gambar 6.49 Hasil Pengujian Melihat Semua Laporan untuk Satu Komentar

6.2.34. Menghapus Sebuah Laporan Komentar

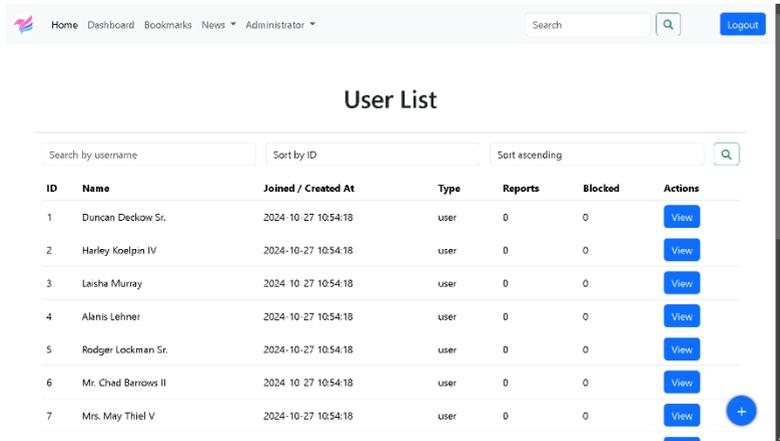
Hasil untuk pengujian menghapus sebuah laporan komentar dapat dilihat pada Gambar 6.50.



Gambar 6.50 Hasil Pengujian Menghapus Sebuah Laporan Komentar

6.2.35. Menampilkan Daftar *User*

Hasil untuk pengujian menampilkan daftar *user* dapat dilihat pada Gambar 6.51.

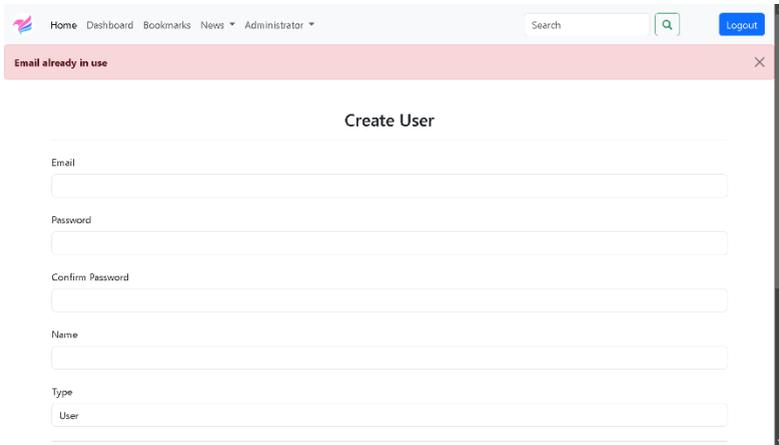


ID	Name	Joined / Created At	Type	Reports	Blocked	Actions
1	Duncan Deckow Sr.	2024-10-27 10:54:18	user	0	0	View
2	Harley Koelpin IV	2024-10-27 10:54:18	user	0	0	View
3	Leisha Murray	2024-10-27 10:54:18	user	0	0	View
4	Alanis Lehner	2024-10-27 10:54:18	user	0	0	View
5	Rodger Lockman Sr.	2024-10-27 10:54:18	user	0	0	View
6	Mr. Chad Barrows II	2024-10-27 10:54:18	user	0	0	View
7	Mrs. May Thiel V	2024-10-27 10:54:18	user	0	0	View

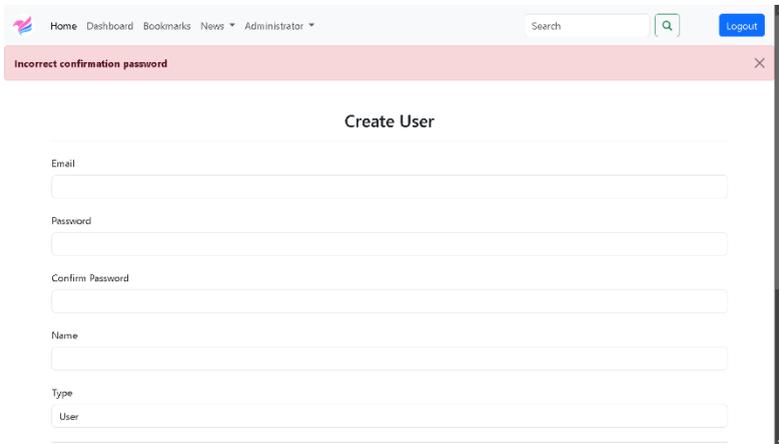
Gambar 6.51 Hasil Pengujian Menampilkan Daftar User

6.2.36. Membuat *User* Baru

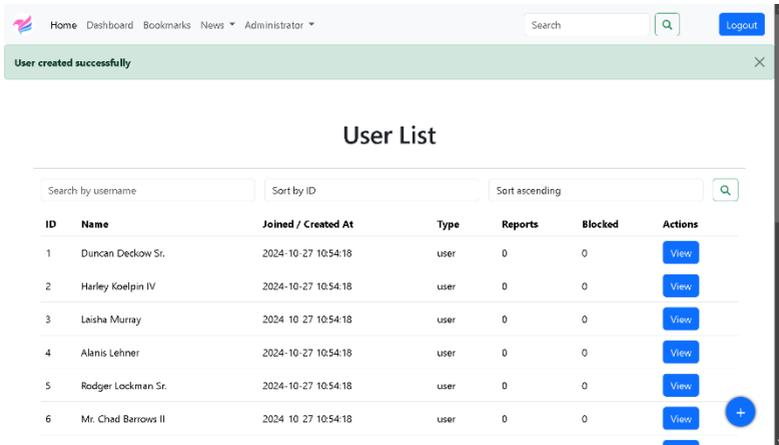
Hasil untuk pengujian membuat *user* baru yang gagal dapat dilihat pada Gambar 6.52 dan Gambar 6.53, dan yang berhasil pada Gambar 6.54.



Gambar 6.52 Hasil Pengujian Membuat User Baru Gagal karena Email Sudah Digunakan



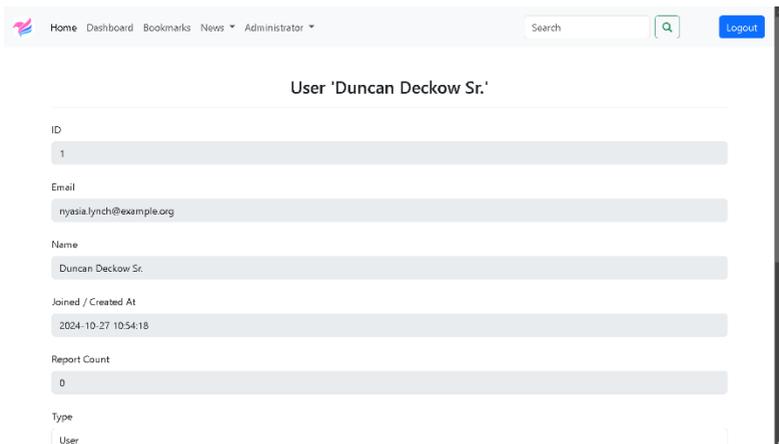
Gambar 6.53 Hasil Pengujian Membuat User Baru Gagal karena Data Tidak Valid



Gambar 6.54 Hasil Pengujian Membuat User Baru Berhasil

6.2.37. Menampilkan Informasi Sebuah *User*

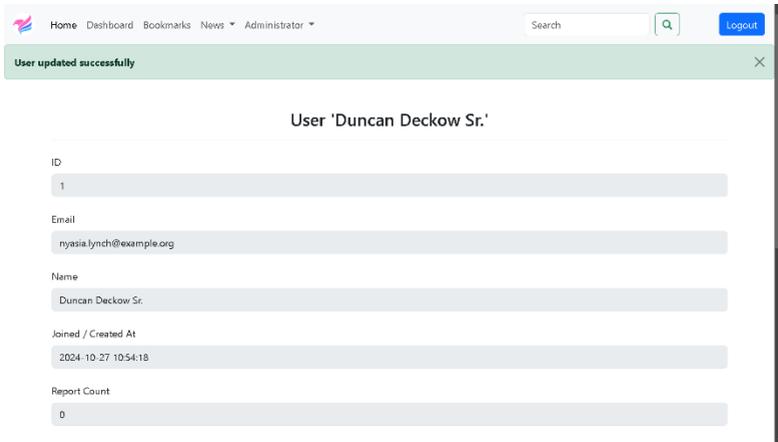
Hasil untuk pengujian menampilkan informasi sebuah *user* dapat dilihat pada Gambar 6.55.



Gambar 6.55 Hasil Pengujian Menampilkan Informasi Sebuah User

6.2.38. Memperbarui Data Sebuah *User*

Hasil untuk pengujian memperbarui data sebuah *user* dapat dilihat pada Gambar 6.56.



Gambar 6.56 Hasil Pengujian Memperbarui Data Sebuah User

6.2.39. Menampilkan Daftar Komentar Sebuah Berita

Hasil untuk pengujian menampilkan daftar komentar sebuah berita dapat dilihat pada Gambar 6.57.

ID	Poster	Posted At	Likes	Dislikes	Replies	Reports	Actions
4901	Duncan Deckow Sr.	2024-10-27 10:54:19	0	0	0	0	View
4902	Harley Koelpin IV	2024-10-27 10:54:19	0	0	1	0	View
4903	Laisha Murray	2024-10-27 10:54:19	0	0	0	0	View
4904	Alanis Lehner	2024-10-27 10:54:19	0	0	2	0	View
4905	Rodger Lockman Sr.	2024-10-27 10:54:19	0	0	2	0	View
4906	Mr. Chad Barrows II	2024-10-27 10:54:19	0	0	1	0	View

Gambar 6.57 Hasil Pengujian Menampilkan Daftar Komentar Sebuah Berita

6.2.40. Menampilkan Sebuah Komentar

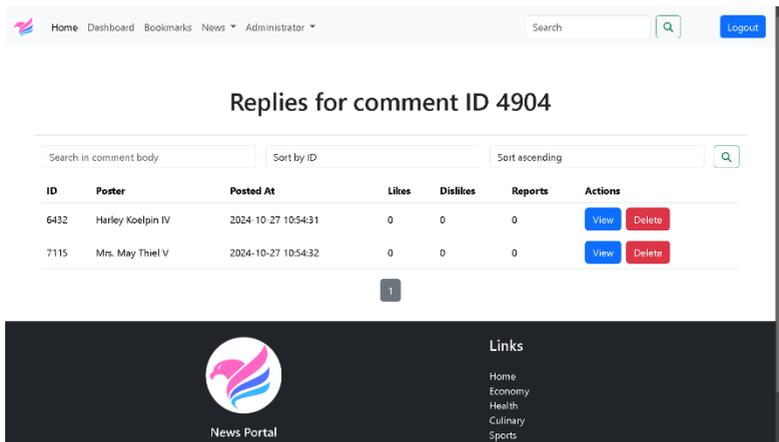
Hasil untuk pengujian menampilkan sebuah komentar dapat dilihat pada Gambar 6.58.

Field	Value
ID	4901
Poster	Duncan Deckow Sr.
Posted At	2024-10-27 10:54:19
Like Count	0
Dislike Count	0
Reply Count	0

Gambar 6.58 Hasil Pengujian Menampilkan Sebuah Komentar

6.2.41. Menampilkan Daftar Balasan untuk Sebuah Komentar

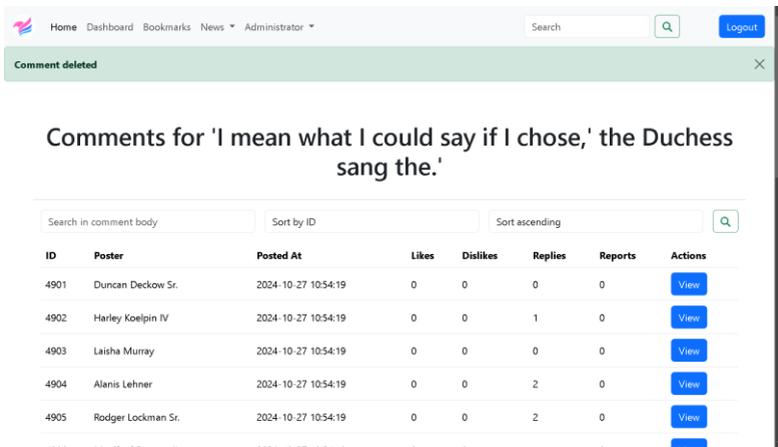
Hasil untuk pengujian menampilkan daftar balasan untuk sebuah komentar dapat dilihat pada Gambar 6.59.



Gambar 6.59 Hasil Pengujian Melihat Daftar Balasan untuk Sebuah Komentar

6.2.42. Menghapus Komentar *User* Lain

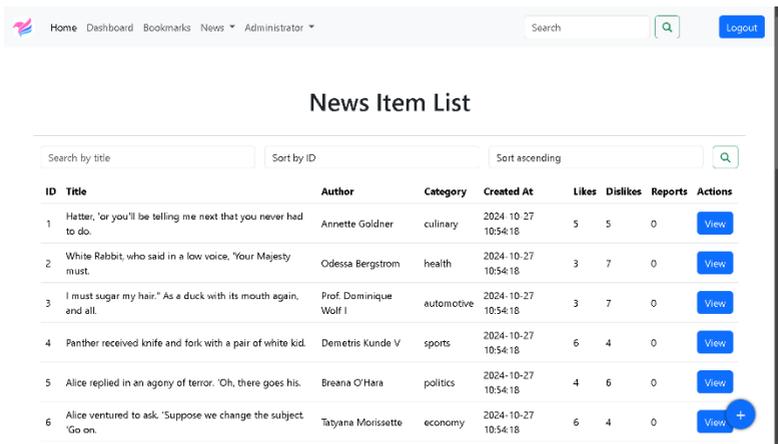
Hasil untuk pengujian menghapus komentar *user* lain dapat dilihat pada Gambar 6.60.



Gambar 6.60 Hasil Pengujian Menghapus Komentar User Lain

6.2.43. Menampilkan Daftar Berita

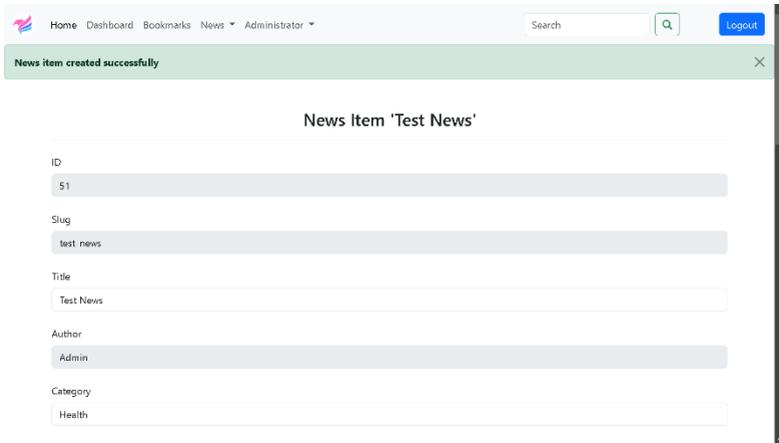
Hasil untuk pengujian menampilkan daftar berita dapat dilihat pada Gambar 6.61.



Gambar 6.61 Hasil Pengujian Menampilkan Daftar Berita

6.2.44. Membuat Berita Baru

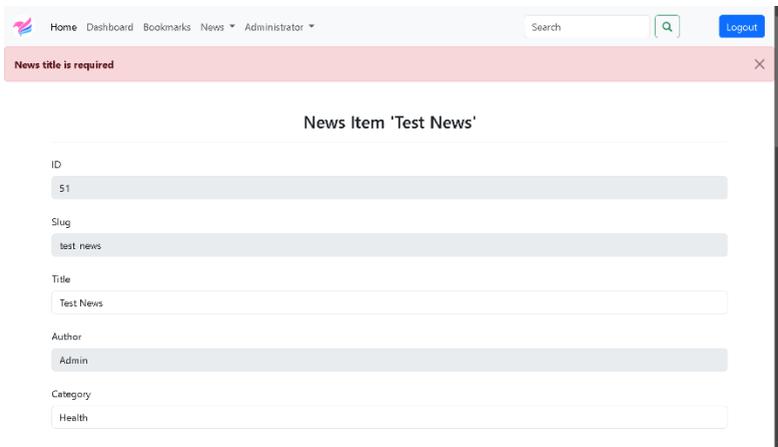
Hasil untuk pengujian membuat berita baru dapat dilihat pada Gambar 6.62.



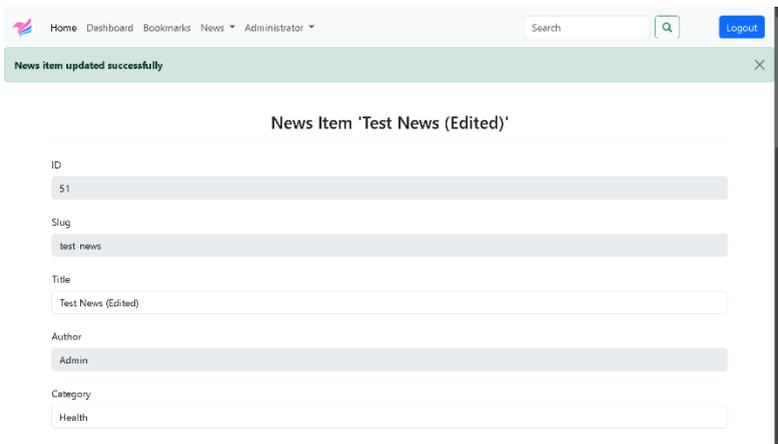
Gambar 6.62 Hasil Pengujian Membuat Berita Baru

6.2.45. Mengedit Sebuah Berita

Hasil untuk pengujian mengedit sebuah berita yang gagal dapat dilihat pada Gambar 6.63, dan yang berhasil pada Gambar 6.64.



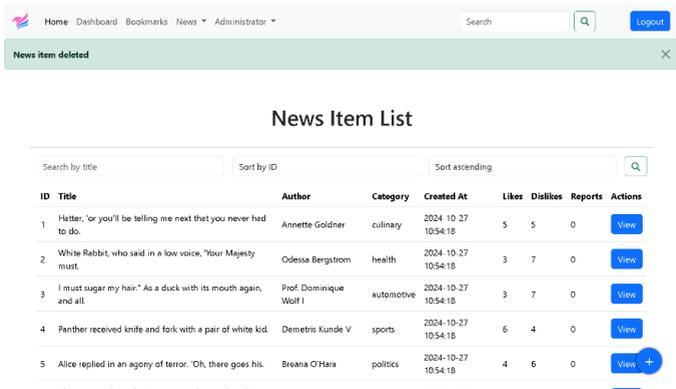
Gambar 6.63 Hasil Pengujian Mengedit Sebuah Berita Gagal



Gambar 6.64 Hasil Pengujian Mengedit Sebuah Berita Berhasil

6.2.46. Menghapus Sebuah Berita

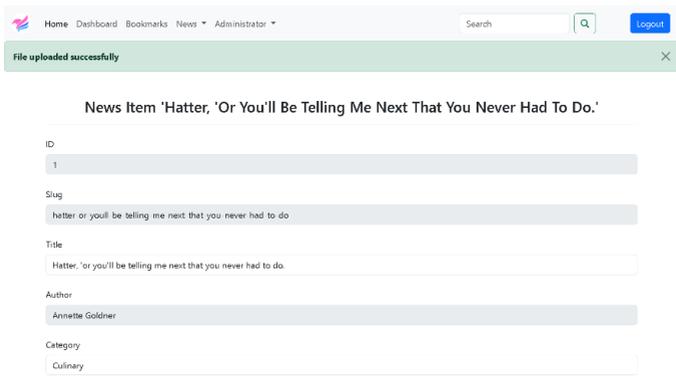
Hasil untuk pengujian menghapus sebuah berita dapat dilihat pada Gambar 6.65.



Gambar 6.65 Hasil Pengujian Menghapus Sebuah Berita

6.2.47. Mengupload *File* untuk Sebuah Berita

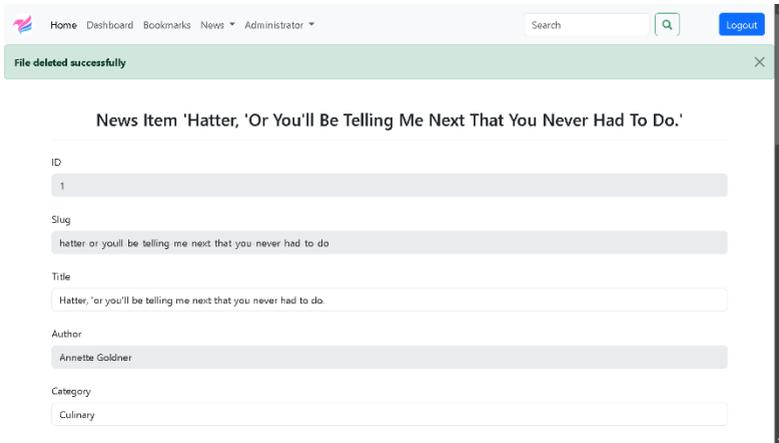
Hasil untuk mengupload *file* untuk sebuah berita dapat dilihat pada Gambar 6.66.



Gambar 6.66 Hasil Pengujian Mengupload File untuk Sebuah Berita

6.2.48. Menghapus *File* yang Diupload untuk Sebuah Berita

Hasil untuk menghapus *file* yang diupload untuk sebuah berita dapat dilihat pada Gambar 6.67.



Gambar 6.67 Hasil Pengujian Menghapus File yang Diupload untuk Sebuah Berita

[Halaman ini sengaja dikosongkan]

BAB VII

KESIMPULAN DAN SARAN

7.1. Kesimpulan

Kesimpulan yang didapat setelah melakukan perancangan dan implementasi aplikasi *website* portal berita pada kegiatan kerja praktik di PT. Winnicode Garuda Teknologi adalah sebagai berikut:

- a. *Website* portal berita yang dibangun telah memenuhi semua permintaan.
- b. Dengan adanya *website* portal berita, PT. Winnicode Garuda Teknologi dapat dengan mudah melakukan pengelolaan data-data *website* dan *website* portal berita dapat menjadi sumber informasi yang bermanfaat bagi penggunaanya.

7.2. Saran

Saran untuk perancangan sistem dari *website* portal berita adalah sebagai berikut :

- a. Mengimplementasi mekanisme *caching* untuk halaman-halaman yang paling sering ditampilkan seperti *homepage* agar tidak memberatkan *database*.
- b. Menggunakan sebuah *front-end framework* untuk membuat tampilan *website* portal berita lebih dinamis dan interaktif.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] Hostinger, “Apa Itu HTML? Fungsi dan Cara Kerja HTML,” 5 Desember 2023. [Online]. Available: <https://www.hostinger.co.id/tutorial/apa-itu-html>. [Diakses 16 Oktober 2024].
- [2] Revou, “Apa itu Web Programming? Arti, Fungsi, Contoh, FAQs 2024 | RevoU,” [Online]. Available: <https://revou.co/kosakata/web-programming>. [Diakses 16 Oktober 2024].
- [3] Hostinger, “Apa Itu Website? Ini Pengertian Website dan Jenis - Jenisnya,” 15 Mei 2024. [Online]. Available: <https://www.hostinger.co.id/tutorial/website-adalah>. [Diakses 16 Oktober 2024].
- [4] Wikipedia, “Apache HTTP Server - Wikipedia,” [Online]. Available: https://en.wikipedia.org/wiki/Apache_HTTP_Server. [Diakses 17 Oktober 2024].
- [5] Laragon, “Documentation | Laragon - portable, isolated, fast & powerful universal development environment for PHP, Node.js, Python, Java, Go, Ruby.,” [Online]. Available: <https://laragon.org/docs/>. [Diakses 17 Oktober 2024].
- [6] Wikipedia, “JavaScript - Wikipedia,” [Online]. Available: <https://en.wikipedia.org/wiki/JavaScript>. [Diakses 17 Oktober 2024].
- [7] “Laravel - The PHP Framework for Web Artisans,” [Online]. Available: <https://laravel.com>. [Diakses 17 Oktober 2024].

- [8] Wikipedia, "MySQL - Wikipedia," [Online]. Available: <https://en.wikipedia.org/wiki/MySQL>. [Diakses 17 Oktober 2024].
- [9] Wikipedia, "PHP - Wikipedia," [Online]. Available: <https://en.wikipedia.org/wiki/PHP>. [Diakses 17 Oktober 2024].

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS

Nama : David Ralphwaldo Martuaraja
Tempat, Tanggal Lahir : Jakarta, 15 November 2001
Jenis Kelamin : Laki-Laki
Telepon : +6281285706066
Email : davidralphwaldo12@gmail.com

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC , ITS
Angkatan : 2019
Semester : 11 (Sebelas)