

**PENGEMBANGAN MODUL
MANAJEMEN LABORATORIUM DAN PENGELOLAAN AKUN
MENGUNAKAN FRAMEWORK LARAVEL PADA SISTEM
INFORMASI CHEMISTRY PROGRAM DATA CENTER**

(Skripsi)

Oleh

**PUTU PUTRA EKA PERSADA
2057051016**



**JURUSAN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

**PENGEMBANGAN MODUL
MANAJEMEN LABORATORIUM DAN PENGELOLAAN AKUN
MENGUNAKAN FRAMEWORK LARAVEL PADA SISTEM
INFORMASI CHEMISTRY PROGRAM DATA CENTER**

Oleh

**PUTU PUTRA EKA PERSADA
2057051016**

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA KOMPUTER**

Pada

**Jurusan Ilmu Komputer
Fakultas Matematika Dan Ilmu Pengetahuan Alam
Universitas Lampung**



**JURUSAN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

ABSTRAK

PENGEMBANGAN MODUL MANAJEMEN LABORATORIUM DAN PENGELOLAAN AKUN MENGUNAKAN FRAMEWORK LARAVEL PADA SISTEM INFORMASI CHEMISTRY PROGRAM DATA CENTER

Oleh

Putu Putra Eka Persada

Dalam era digital saat ini, pengelolaan laboratorium dan manajemen akun menjadi salah satu aspek krusial dalam operasional program studi kimia. Skripsi ini bertujuan untuk mengembangkan modul manajemen laboratorium dan pengelolaan akun menggunakan *framework* Laravel pada sistem informasi Chemistry Program Data Center. Penggunaan *framework* Laravel dipilih karena kehandalan dan kemudahan penggunaannya dalam pengembangan aplikasi web. Modul yang dikembangkan meliputi fitur-fitur seperti pencatatan inventaris laboratorium, pengaturan jadwal penggunaan laboratorium, serta pengelolaan akun pengguna. Metode pengembangan yang digunakan adalah metode *Waterfall* yang meliputi tahap analisis kebutuhan, desain sistem, implementasi, pengujian, dan pemeliharaan. Hasil dari pengembangan ini diharapkan dapat meningkatkan efisiensi dan efektivitas pengelolaan laboratorium serta manajemen akun di program studi kimia.

Kata kunci: Manajemen Laboratorium, Pengelolaan Akun, Laravel, Sistem Informasi, Chemistry Program Data Center.

ABSTRACT

DEVELOPMENT OF LABORATORY MANAGEMENT AND ACCOUNT MANAGEMENT MODULE USING LARAVEL FRAMEWORK IN CHEMISTRY PROGRAM DATA CENTER INFORMATION SYSTEM

By

Putu Putra Eka Persada

In the current digital era, laboratory management and account management are crucial aspects in the operations of chemistry study programs. This thesis aims to develop a laboratory management and account management module using the Laravel framework in the Chemistry Program Data Center information system. The Laravel framework is chosen due to its reliability and ease of use in web application development. The developed module includes features such as laboratory inventory recording, laboratory usage scheduling, and user account management. The development method used is the Waterfall method, which includes the stages of requirements analysis, system design, implementation, testing, and maintenance. The result of this development is expected to improve the efficiency and effectiveness of laboratory management and account management in the chemistry study program.

Keywords: Laboratory Management, Account Management, Laravel, Information System, Chemistry Program Data Center.

Judul Skripsi

: **PENGEMBANGAN MODUL MANAJEMEN
LABORATORIUM DAN PENGELOLAAN
AKUN MENGGUNAKAN FRAMEWORK
LARAVEL PADA SISTEM INFORMASI
CHEMISTRY PROGRAM DATA CENTER**

Nama Mahasiswa

: **Putu Putra Eka Persada**

Nomor Pokok Mahasiswa

: 2057051016

Program Studi

: S1 Ilmu Komputer

Fakultas

: Matematika dan Ilmu Pengetahuan Alam



MENYETUJUI
1. Komisi Pembimbing

Febi Eka Febriansyah, M.T
NIP. 198002192006041001

Mulyono, S.Si., M.Si., Ph.D.
NIP.197406112000031002

2. Ketua Jurusan Ilmu Komputer

Dwi Sakethi, S.Si., M.Kom.
NIP. 196806111998021001

MENGESAHKAN

1. Tim Penguji

Ketua

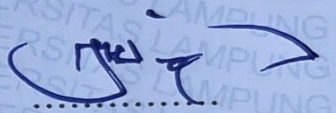
: **Febi Eka Febriansyah, M.T**



Penguji I

Sekretaris

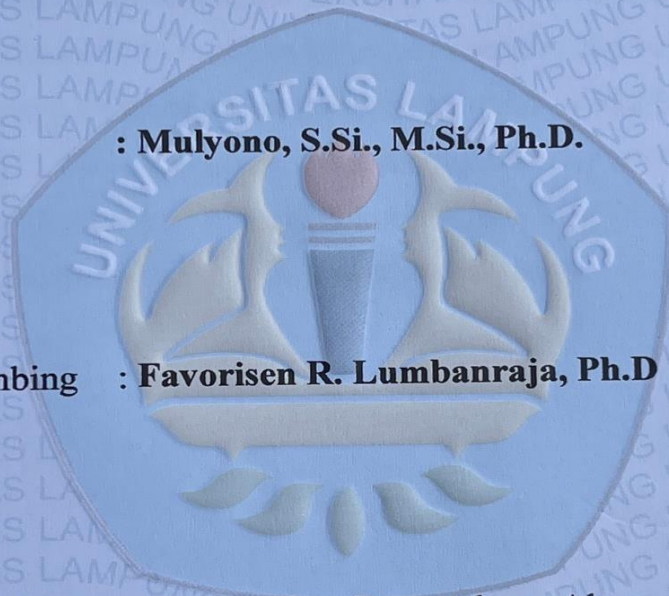
: **Mulyono, S.Si., M.Si., Ph.D.**



Penguji II

Bukan Pembimbing

: **Favorisen R. Lumbanraja, Ph.D**



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Eng. Heri Satria, S.Si., M.Si.

NIP. 19711001 200501 1 002

Tanggal Lulus Ujian Skripsi : **1 Agustus 2024**

PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya dengan judul **“Pengembangan Modul Manajemen Laboratorium dan Pengelolaan Akun Menggunakan Framework Laravel Pada Sistem Informasi Chemistry Program Data Center”** merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 6 Agustus 2024



Putu Putra Eka Persada

NPM. 2057051016

RIWAYAT HIDUP



Lahir pada 16 Oktober 2001 di Kota Metro, Provinsi Lampung. Anak Pertama dari pasangan Bapak I Nyoman Wesnawa Mulyadi dan Ibu Ni Komang Sulistyani. Telah menyelesaikan pendidikan dasar pada tahun 2014 di SDN 2 Rama Dewa, Kecamatan Seputih Raman, Kabupaten Lampung Tengah. Kemudian Menyelesaikan pendidikan menengah pertama pada tahun 2017 di SMPN 2 Kotagajah, dan menyelesaikan pendidikan Menengah Atas pada tahun 2020 di SMAN 1 Kotagajah. Pada tahun 2020 kemudian melanjutkan Pendidikan di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui Jalur SMMPTN (Seleksi Mandiri Masuk Perguruan Tinggi Negeri). Berikut kegiatan yang pernah dilakukan selama menjadi mahasiswa.

1. Menjadi Asisten Dosen di Jurusan Ilmu Komputer pada Tahun 2021 hingga 2023.
2. Menjadi anggota Unit Kegiatan Mahasiswa Hindu Universitas Lampung pada Tahun 2020 hingga 2024.
3. Menjadi wakil ketua bidang informasi dan komunikasi Perhimpunan Pemuda Hindu Indonesia (PERADAH) Kota Bandarlampung pada Tahun 2022 hingga 2024.
4. Menjadi anggota Kesatuan Mahasiswa Hindu Dharma Indonesia (KMHDHI) Kota Bandarlampung pada Tahun 2020 hingga 2024.
5. Mengikuti Kegiatan Magang Bersama Kampus Merdeka pada Bulan Februari 2023 di Jurusan Kimia Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

MOTTO

Uddhared ātmanātmānaṁ nātmānam avasādayet; ātmaiva hy ātmano bandhur
ātmaiva ripur ātmanah

Seseorang harus mengangkat dirinya sendiri dengan usahanya sendiri dan tidak boleh merendahkan dirinya sendiri; karena diri sendiri adalah teman bagi dirinya sendiri dan diri sendiri adalah musuh bagi dirinya sendiri

(Bhagavad-gītā Adhyaya 6 Sloka 5)

"Jika kamu tidak ingin menyerah, jangan menyerah."
(Rock Lee)

"Seseorang yang tidak bisa menghargai dirinya sendiri tidak akan pernah bisa menghargai orang lain."
(Gaara)

“Mereka yang melanggar aturan memang sampah, tapi mereka yang meninggalkan teman-temannya lebih buruk dari sampah."
(Obito Uchiha)

"Kegagalan tidak membuatku malu. Yang membuatku malu adalah tidak belajar dari kegalanku."
(Jiraiya)

PERSEMBAHAN

Kepada Ayah dan Ibu, Rasa terima kasih yang tak terhingga saya sampaikan kepada Ayah dan Ibu yang telah menjadi sumber inspirasi, dukungan, dan cahaya sepanjang perjalanan akademik ini. Kata-kata tidak cukup untuk menggambarkan betapa dalam rasa syukur dan penghargaan atas semangat dan cinta yang selalu Ayah dan Ibu berikan.

Ayah dan Ibu adalah pilar kokoh dalam hidup. Dalam setiap langkah, memberikan dorongan, nasihat, dan kasih sayang tanpa batas. Terima kasih atas segala pengorbanan yang Ayah dan Ibu lakukan untuk membantu saya meraih tujuan.

Doa dan dukungan tiada henti dari Ayah dan Ibu menjadi penopang dalam menghadapi setiap tantangan. Saya berjanji untuk terus menghormati dan menghargai segala yang telah Ayah dan Ibu berikan.

Semoga skripsi ini dapat menjadi tanda kecil dari pengabdian kepada Ayah dan Ibu, serta menjadi kebanggaan. Terima kasih atas segala-galanya, Ayah dan Ibu. Semoga kebaikan dan kasih sayang Ayah dan Ibu selalu menyertai langkah-langkah ke depan.

SANWACANA

Puji syukur ke hadirat Tuhan Yang Maha Esa atas rahmat dan berkat-Nya, sehingga dapat menyelesaikan skripsi yang berjudul “Pengembangan Modul Manajemen Laboratorium dan Pengelolaan Akun Menggunakan Framework Laravel Pada Sistem Informasi Chemistry Program Data Center” dengan baik tepat waktu. Dalam melaksanakan penelitian dan pembuatan skripsi ini, banyak mendapatkan bimbingan dan dukungan dari berbagai pihak, sehingga pada kesempatan ini menyampaikan ungkapan terima kasih ini kepada:

1. Kedua orang tua yang telah memberikan dukungan dan motivasi untuk menyelesaikan skripsi dengan baik.
2. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
3. Bapak Dwi Sakethi, S.Si., M.Kom. selaku Ketua Jurusan Ilmu Komputer Universitas Lampung.
4. Ibu Anie Rose Irawati, S.T., M.Cs. selaku Sekretaris Jurusan Ilmu Komputer Universitas Lampung.
5. Bapak Febi Eka Febriansyah, M.T selaku pembimbing utama atas bimbingan, saran dan kritik dalam proses menyelesaikan skripsi.
6. Bapak Mulyono, S.Si., M.Si., Ph.D. selaku pembimbing kedua atas bimbingan, saran dan kritik dalam proses menyelesaikan skripsi.
7. Bapak Favorisen R. Lumbanraja, Ph.D selaku pembahas atas masukan dan saran dalam proses menyelesaikan skripsi.
8. Bapak dan Ibu Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan ilmu serta pengalaman semasa perkuliahan.

9. Pihak Jurusan Kimia yang telah membantu dalam menyelesaikan skripsi ini dengan baik.
10. Ibu Ade Nora Maela, Bang Zainuddin, dan Mas Naufal yang telah membantu dalam hal administrasi di Jurusan Ilmu Komputer.
11. Angkatan 2020 Ilmu Komputer selaku teman seperjuangan yang telah memberikan pengalaman yang baik semasa kuliah.

Bandarlampung, 6 Agustus 2024

Putu Putra Eka Persada
NPM. 2057051016

DAFTAR ISI

	Halaman
DAFTAR ISI	i
DAFTAR GAMBAR.....	iii
DAFTAR TABEL	viii
DAFTAR LAMPIRAN	x
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
II. TINJAUAN PUSTAKA	5
2.1 Penelitian Terdahulu	5
2.2 Sistem Informasi	6
2.3 <i>Framework</i> Laravel.....	7
2.4 <i>Website</i>	7
2.5 <i>Model-View-Controller</i> (MVC).....	8
2.6 <i>Hypertext Preprocessor</i> (PHP).....	9
2.7 <i>Unified Modelling Language</i> (UML)	10
2.8 <i>Metode Pengembangan Sistem</i>	16
2.9 <i>Desain Sistem</i>	18
2.10 <i>Metode Pengujian Sistem</i>	18
III. METODE PENELITIAN.....	21
3.1 Tempat dan Waktu Penelitian	21
3.2 Alat Penelitian.....	23

3.2.1 Perangkat Keras (<i>Hardware</i>)	23
3.2.2 Perangkat Lunak (<i>Software</i>)	23
3.3 Tahapan Penelitian	26
3.3.1 Kerangka Penelitian.....	26
3.3.2 Metode Pengumpulan Data.....	28
3.4 Analisis Kebutuhan Sistem	28
3.5 Uraian Tahapan Pengembangan Sistem	31
3.5.1 <i>Communication</i>	31
3.5.2 <i>Planning</i>	33
3.6 Desain Sistem dan <i>Activity Diagram</i>	35
3.7 Penulisan Kode Program	87
3.8 Pengujian Sistem.....	87
3.8.1 Skenario <i>Black Box Testing</i>	87
IV. HASIL DAN PEMBAHASAN	94
4.1 Hasil.....	94
4.2 Pembahasan.....	95
4.2.1 Tahapan Pembahasan	95
4.2.2 Hasil Implementasi	97
4.2.3 Pengujian Sistem	122
V. PENUTUP.....	169
5.1 Simpulan	169
5.2 Saran	169
DAFTAR PUSTAKA	

DAFTAR GAMBAR

Gambar	Halaman
1. Metode <i>Waterfall</i>	16
2. Alur Penelitian.	27
3. <i>Use Case Diagram</i> Admin dan Super Admin Pada Website CPDC.....	30
4. Desain Antarmuka Masuk Sistem.	36
5. Desain Antarmuka Manajemen Akun Dosen.....	37
6. <i>Activity Diagram</i> Tambah Akun Dosen.	37
7. Desain Antarmuka Tambah Akun Dosen.	38
8. <i>Activity Diagram</i> Edit Akun Dosen.	38
9. Desain Antarmuka Edit Akun Dosen.	39
10. <i>Activity Diagram</i> Hapus Akun Dosen.....	39
11. Desain Antarmuka Hapus Akun Dosen	40
12. Desain Antarmuka Manajemen Akun Mahasiswa.	41
13. <i>Activity Diagram</i> Edit Akun Mahasiswa.	41
14. Desain Antarmuka Edit Akun Mahasiswa.	42
15. <i>Activity Diagram</i> Hapus Akun Mahasiswa.	42
16. Desain Antarmuka Hapus Akun Mahasiswa.	43
17. Desain Antarmuka Manajemen Akun Admin.....	44
18. <i>Activity Diagram</i> Tambah Akun Admin.	44
19. Desain Antarmuka Tambah Akun Admin.	45
20. <i>Activity Diagram</i> Hapus Akun Admin.	45
21. Desain Antarmuka Hapus Akun Admin.	46
22. <i>Activity Diagram</i> Edit Akun Admin.	46
23. Desain Antarmuka Edit Akun Admin.	47
24. Desain Antarmuka Halaman Kategori.	48
25. <i>Activity Diagram</i> Tambah Kategori.	48
26. Desain Antarmuka Tambah Kategori.....	49

27. <i>Activity Diagram</i> Edit Kategori.	49
28. Desain Antarmuka Hapus Kategori. Desain Antarmuka Edit Kategori.....	50
29. <i>Activity Diagram</i> Unduh Kategori.	50
30. Desain Antarmuka Unduh Kategori.....	51
31. <i>Activity Diagram</i> Hapus Kategori.....	51
32. Desain Antarmuka Hapus Kategori.	52
33. Desain Antarmuka Halaman Model.....	53
34. Desain Antarmuka Edit Model. <i>Activity Diagram</i> Edit Model.....	53
35. Desain Antarmuka Edit Model.	54
36. <i>Activity Diagram</i> Tambah Model.....	54
37. Desain Antarmuka Tambah Model.	55
38. <i>Activity Diagram</i> Hapus Model.	55
39. Desain Antarmuka Hapus Model.	56
40. Desain Antarmuka Manajemen Peran Akun Kepala Laboratorium.	57
41. <i>Activity Diagram</i> Edit Kepala Laboratorium.	57
43. <i>Activity Diagram</i> Unduh Data Kepala Laboratorium.	58
42. Desain Antarmuka Edit Data Kepala Laboratorium.	58
44. Desain Antarmuka Unduh Data Kepala Laboratorium.	59
45. Desain Antarmuka Manajemen Peran Akun Admin Jurusan.	60
46. <i>Activity Diagram</i> Edit Data Admin Jurusan.	60
47. Desain Antarmuka Edit Data Admin Jurusan.	61
48. <i>Activity Diagram</i> Unduh Data Admin Jurusan.	61
49. Desain Antarmuka Unduh Data Admin Jurusan.....	62
50. Desain Antarmuka NPM Mahasiswa.....	63
51. <i>Activity Diagram</i> Tambah Data NPM Mahasiswa.....	63
52. Desain Antarmuka Tambah Data NPM Mahasiswa.	64
53. <i>Activity Diagram</i> Import Data NPM Mahasiswa.	64
54. Desain Antarmuka Import Data NPM Mahasiswa.	65
55. <i>Activity Diagram</i> Edit NPM Mahasiswa.....	65
56. Desain Antarmuka Edit Data NPM Mahasiswa.....	66
57. <i>Activity Diagram</i> Hapus NPM Mahasiswa.	66
58. Desain Antarmuka Hapus Data NPM Mahasiswa.	67

59. Desain Antarmuka Aktivitas Laboratorium.	68
60. <i>Activity Diagram</i> Tambah Aktivitas Laboratorium.	68
61. Desain Antarmuka Tambah Aktivitas Laboratorium.	69
62. <i>Activity Diagram</i> Edit Aktivitas Laboratorium.	69
63. Desain Antarmuka Edit Aktivitas Laboratorium.	70
64. <i>Activity Diagram</i> Hapus Aktivitas Laboratorium.	70
65. Desain Antarmuka Hapus Aktivitas Laboratorium.	71
66. <i>Activity Diagram</i> Lihat Data Aktivitas Laboratorium.	71
67. Desain Antarmuka Lihat Data Aktivitas Laboratorium.	72
68. Desain Antarmuka SOP Laboratorium.	73
69. <i>Activity Diagram</i> Tambah SOP Laboratorium.	73
70. Desain Antarmuka Tambah SOP Laboratorium.	74
71. <i>Activity Diagram</i> Edit SOP Laboratorium.	74
72. Desain Antarmuka Edit SOP Laboratorium.	75
73. <i>Activity Diagram</i> Unduh SOP Laboratorium.	75
74. Desain Antarmuka Unduh Data SOP Laboratorium.	76
75. <i>Activity Diagram</i> Hapus SOP Laboratorium.	76
76. Desain Antarmuka Hapus SOP Laboratorium.	77
77. <i>Activity Diagram</i> Lihat SOP Laboratorium.	77
78. Desain Antarmuka Lihat SOP Laboratorium.	78
79. Desain Antarmuka Inventaris Laboratorium.	79
80. <i>Activity Diagram</i> Tambah Data Inventaris Laboratorium.	79
81. Desain Antarmuka Tambah Data Inventaris Laboratorium.	80
82. <i>Activity Diagram</i> Edit Inventaris Laboratorium.	80
84. Desain Antarmuka Edit Data Inventaris Laboratorium.	81
83. <i>Activity Diagram</i> Unduh Data Inventaris Laboratorium.	81
85. Desain Antarmuka Unduh Data Inventaris Laboratorium.	82
86. <i>Activity Diagram</i> Lihat Inventaris Laboratorium.	82
87. Desain Antarmuka Lihat Data Inventaris Laboratorium.	83
88. <i>Activity Diagram</i> Hapus Inventaris Laboratorium.	83
89. Desain Antarmuka Hapus Data Inventaris Laboratorium.	84
90. Desain Antarmuka Presensi Aktivitas Laboratorium Mahasiswa S2.	85

91. Desain Antarmuka Mahasiswa Sudah Melakukan Presensi.	85
92. <i>Activity Diagram</i> Presensi Aktivitas Laboratorium Mahasiswa S2.	86
93. Tampilan <i>Login</i>	97
94. Tampilan Data Akun Mahasiswa.	98
95. Tampilan Hapus Akun Mahasiswa.	98
96. Tampilan Edit Akun Mahasiswa.	99
97. Tampilan Data Akun Dosen.	99
98. Tampilan Hapus Akun Dosen.	100
99. Tampilan Tambah Akun Dosen.	100
100. Tampilan Edit Akun Dosen.	101
101. Tampilan Data Akun Admin.	101
102. Tampilan Tambah Akun Admin.	102
103. Tampilan Edit Akun Admin.	102
104. Tampilan Hapus Akun Admin.	103
105. Tampilan Kategori inventaris Laboratorium.	103
106. Tampilan Tambah Kategori.	104
107. Tampilan Edit Kategori.	104
108. Tampilan Hapus Kategori.	105
109. Tampilan Model Inventaris Laboratorium.	105
110. Tampilan Tambah Model.	106
111. Tampilan Edit Model.	106
112. Tampilan Hapus Model.	107
113. Tampilan Data NPM Mahasiswa.	107
114. Tampilan Data NPM Mahasiswa.	108
115. Tampilan <i>Import</i> NPM Mahasiswa.	108
116. Tampilan Hapus NPM Mahasiswa.	109
117. Tampilan Manajemen Akun Kepala Laboratorium.	109
118. Tampilan Unduh Atau Cetak Data Akun Kepala Laboratorium.	110
119. Tampilan Edit Data Akun Kepala Laboratorium.	110
120. Tampilan Manajemen Akun Admin Jurusan.	111
121. Tampilan Unduh Atau Cetak Data Admin Jurusan.	111
122. Tampilan Edit Akun Admin Jurusan.	112

123. Tampilan Aktivitas Laboratorium.....	112
124. Tampilan Tambah Aktivitas Laboratorium.....	113
125. Tampilan Lihat Aktivitas Laboratorium.	113
126. Tampilan Hapus Aktivitas Laboratorium.	114
127. Tampilan Edit Aktivitas Laboratorium.	114
128. Tampilan SOP Laboratorium.	115
129. Tampilan Tambah SOP Laboratorium.	115
130. Tampilan Lihat SOP Laboratorium.....	116
131. Tampilan Edit SOP Laboratorium.	116
132. Tampilan Hapus SOP Laboratorium.....	117
133. Tampilan Manajemen Inventaris Laboratorium.....	117
134. Tampilan Unduh Atau Cetak Inventaris Laboratorium.	118
135. Tampilan Lihat Inventaris Laboratorium.	118
136. Tampilan Tambah Data Inventaris Laboratorium.....	119
137. Tampilan Hapus Inventaris Laboratorium.	119
138. Tampilan Edit Inventaris Laboratorium.....	120
139. Tampilan Pendataan <i>Peer Group</i> Laboratorium Sebelum Presensi.....	120
140. Tampilan Presensi Aktivitas Laboratorium TA S2 Sebelum Dikirim.	121
141. Tampilan Presensi Aktivitas Laboratorium TA S2 Sesudah Dikirim.....	121

DAFTAR TABEL

Tabel	Halaman
1. Akun Admin Jurusan.	5
2. Simbol <i>Use Case Diagram</i>	11
3. Simbol <i>Class Diagram</i>	13
4. Simbol <i>Sequence Diagram</i>	14
5. Simbol <i>Activity Diagram</i>	15
6. Perencanaan Penelitian.....	22
7. Rencana Kerja Proyek.....	34
8. Skenario <i>Black Box Testing</i>	87
9. Hasil Pengujian Halaman Kategori Laboratorium.	122
10. Hasil Pengujian Halaman Model Laboratorium.	123
11. Hasil Pengujian Halaman Manajemen Akun Mahasiswa.	124
12. Hasil Pengujian Halaman Manajemen Akun Mahasiswa.	124
13. Hasil Pengujian Halaman Akun Admin.	125
14. Hasil Pengujian Halaman Data NPM Mahasiswa.....	126
15. Hasil Pengujian Halaman Manajemen Akun Kepala Laboratorium.	127
16. Hasil Pengujian Halaman Manajemen Akun Admin Laboratorium.	128
17. Hasil Pengujian Pada Halaman Aktivitas Laboratorium.	128
18. Hasil Pengujian Pada Halaman SOP Laboratorium.....	129
19. Hasil Pengujian Halaman Inventaris Laboratorium.	130
20. Skala Likert.	132
21. Kuesioner Pengujian Halaman Aktivitas Laboratorium.	133
22. Hasil Pengujian Halaman Aktivitas Laboratorium.	135
23. Kuesioner Pengujian Halaman SOP Laboratorium.....	137
24. Hasil Pengujian Halaman SOP Laboratorium.....	138
25. Kuesioner Pengujian Halaman Inventaris Laboratorium.	140
26. Hasil Pengujian Halaman Inventaris Laboratorium.	142
27 Kuesioner Pengujian Pada Halaman Akun Mahasiswa.	144

28. Hasil Pengujian Halaman Akun Mahasiswa.....	145
29. Kuesioner Pengujian Pada Halaman Akun Dosen.....	146
30. Hasil Pengujian Halaman Akun Dosen.....	148
31. Kuesioner Pengujian Pada Halaman Akun Admin.....	149
32. Hasil Pengujian Halaman Admin.....	151
33. Kuesioner Pengujian Pada Halaman Kategori.....	152
34. Hasil Pengujian Halaman Kategori.....	154
35. Kuesioner Pengujian Pada Halaman Model.....	155
36. Hasil Pengujian Halaman Model.....	157
37. Kuesioner Pengujian Pada Halaman Data NPM Mahasiswa.....	159
38. Hasil Pengujian Halaman Data NPM Mahasiswa.....	160
39. Kuesioner Halaman Akun Kepala Laboratorium.....	162
40. Hasil pengujian Halaman Akun Kepala Laboratorium.....	163
41. Kuesioner Pengujian Pada Halaman Akun Admin Jurusan.....	165
42. Hasil Pengujian Pada Halaman Akun Admin Jurusan.....	166

DAFTAR LAMPIRAN

Lampiran	Halaman
1. <i>Entity Relationship Diagram</i> Dosen.	174
2. <i>Entity Relationship Diagram</i> Admin dan User.....	174
3. <i>Entity Relationship Diagram</i> Mahasiswa dan Manajemen Aset.....	175
4. <i>Class Diagram</i> Mahasiswa, User dan Admin.	176
5. <i>Class Diagram</i> Dosen.	177

I. PENDAHULUAN

1.1 Latar Belakang

Di era digital ini, *website* menjadi sarana utama bagi suatu organisasi atau lembaga untuk berkomunikasi, berinteraksi, dan penyedia informasi kepada penggunanya. Dalam hal ini *website* Chemistry Program Data Center di Jurusan Kimia yang mana jurusan ini telah resmi berdiri melalui Surat Keputusan DIKTI No.12/DIKTI/KEP/1999. Jurusan Kimia merupakan bagian yang terintegrasi dari Fakultas Matematika dan Ilmu Pengetahuan Alam di Universitas Lampung.

Sangat penting untuk terus meningkatkan kinerja dan manajemen situs web Chemistry Program Data Center ini seiring dengan kemajuan teknologi informasi dan kebutuhan akan akses informasi yang lebih cepat dan efisien. Manajemen laboratorium dan pengelolaan akun adalah komponen yang memerlukan perhatian khusus.

Manajemen laboratorium yang efektif sangat penting untuk mendukung proses pembelajaran dan penelitian di Jurusan Kimia. Dengan perkembangan teknologi laboratorium dan jumlah mahasiswa yang terus meningkat, modul yang dapat membantu mengelola dan menggunakan laboratorium secara efektif diperlukan. Pelaporan hasil eksperimen, pemantauan inventaris, dan penjadwalan penggunaan laboratorium menjadi lebih mudah dengan sistem manajemen laboratorium yang terintegrasi dengan *website* ini.

Fokus utama lainnya adalah mengelola akun di situs web Chemistry Program Data Center. Dengan banyaknya pengguna yang memiliki hak akses yang berbeda, sistem yang memastikan bahwa setiap pengguna memiliki hak akses yang sesuai dengan perannya dalam Jurusan Kimia diperlukan. Selain itu, pengelolaan akun yang baik juga dapat meminimalkan risiko keamanan data dan informasi yang disimpan di *website*.

Selain modul manajemen laboratorium dan pengelolaan akun, pengembangan sistem pendukung keputusan (DSS - *Decision Support System*) juga menjadi aspek penting dalam optimalisasi website Chemistry Program Data Center. DSS dapat memberikan dukungan tambahan bagi pengambilan keputusan di tingkat manajerial dan akademis di Jurusan Kimia. Dengan integrasi DSS, pengguna website dapat memperoleh informasi yang lebih terstruktur dan analisis yang mendalam terkait dengan data laboratorium, perkembangan riset, dan kinerja mahasiswa. Sistem ini dapat menyajikan data dalam bentuk visualisasi yang mudah dipahami, membantu staf pengajar dan peneliti untuk membuat keputusan yang lebih informasional dan tepat. Penerapan DSS dalam website ini dapat mencakup berbagai aspek, seperti analisis trend hasil eksperimen, prediksi kebutuhan laboratorium berdasarkan sejarah penggunaan, dan evaluasi kinerja mahasiswa. Dengan demikian, DSS dapat menjadi alat yang sangat berharga untuk meningkatkan efisiensi, produktivitas, dan kualitas keputusan di Jurusan Kimia Universitas Lampung melalui Chemistry Program Data Center.

Oleh karena itu, penelitian ini mempelajari pengembangan modul manajemen laboratorium dan pengelolaan akun menggunakan framework Laravel untuk membantu pengembangan dan pengoptimalan website Chemistry Program Data Center. Dengan adanya modul ini, diharapkan website tersebut dapat lebih efisien dalam mendukung kegiatan pembelajaran, penelitian, dan administrasi di Jurusan Kimia Universitas Lampung.

1.2 Rumusan Masalah

Penelitian ini mempelajari beberapa masalah utama yang muncul saat mengembangkan modul manajemen laboratorium dan pengelolaan akun di website Chemistry Program Data Center yang menggunakan framework Laravel. Rumusan masalahnya adalah sebagai berikut.

1. Bagaimana dampak dari implementasi modul manajemen laboratorium dan pengelolaan akun menggunakan *framework* Laravel terhadap efisiensi dan produktivitas staf di Jurusan Kimia, Universitas Lampung.
2. Apa strategi yang dapat diimplementasikan untuk mengatasi tantangan dan hambatan yang mungkin muncul selama proses pengembangan dan implementasi modul ini.

1.3 Batasan Masalah

Dalam penelitian ini terdapat batasan masalah untuk membatasi permasalahan pokok dalam suatu penelitian. Adapun batasan masalah tersebut antara lain:

1. Penelitian ini hanya berfokus pada bagian Jurusan Kimia di Universitas Lampung. Berdasarkan kebutuhan Jurusan Kimia, terdapat pengembangan dan implementasi modul yang khusus untuk manajemen laboratorium dan pengelolaan akun.
2. Penelitian ini berfokus pada manfaat baik yang diberikan oleh modul yang dibuat untuk meningkatkan efisiensi dan produktivitas staf di Jurusan Kimia. Penelitian tidak menyelidiki secara detail tentang pengaruh terhadap mahasiswa atau dosen.
3. Penelitian ini menggunakan *framework* Laravel sebagai alat utama untuk mengembangkan modul. Penggunaan *framework* lainnya tidak dibahas.

Dengan mempertimbangkan batasan masalah yang ada, penelitian ini lebih terfokus pada konsekuensi, strategi mengatasi hambatan, dan penerapan modul yang lebih efektif dan produktif di lingkungan Jurusan Kimia, Universitas Lampung.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk mengembangkan modul manajemen laboratorium dan pengelolaan akun pada *website* Chemistry Program Data Center Universitas Lampung menggunakan *framework* Laravel.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Dengan menerapkan modul manajemen laboratorium dan pengelolaan akun yang efektif, staf Jurusan Kimia dapat menjadi lebih efisien dan produktif. Ini memungkinkan mereka untuk berkonsentrasi pada tugas-tugas yang lebih strategis dan mendukung kemajuan program studi.
2. Modul ini dapat meningkatkan manajemen laboratorium, termasuk penjadwalan penggunaan laboratorium, pengawasan inventaris, dan pelaporan hasil eksperimen. Akibatnya, pendidikan dan penelitian di Jurusan Kimia dapat ditingkatkan.
3. Modul ini mempermudah akses dosen, mahasiswa, dan pihak terkait ke informasi laboratorium, pengelolaan akun, dan aktivitas lainnya.
4. Manajemen akun yang lebih baik dapat meningkatkan keamanan data dan informasi yang tersimpan dalam pusat data program kimia. Ini dapat membantu melindungi data sensitif dari potensi ancaman keamanan.
5. Modul yang dihasilkan dapat memperbaiki pengalaman pengguna dengan menyediakan alat yang lebih produktif dan ramah pengguna dalam mengatur kegiatan laboratorium dan akun.
6. Penelitian ini bisa berkontribusi pada bidang pengembangan sistem informasi yang berbasis web dan aplikasi yang berbasis *framework* Laravel. Selain itu, penelitian ini juga dapat dijadikan contoh tentang bagaimana teknologi informasi bisa diterapkan dalam konteks pendidikan tinggi.
7. Hasil riset ini bisa menjadi acuan dan panduan bagi riset berikutnya yang berkaitan dengan pengembangan modul dan sistem serupa di institusi pendidikan lainnya.

II. TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Berikut adalah beberapa penelitian yang telah dilakukan terkait pengembangan sistem pada penelitian ini.

Tabel 1. Penelitian Terdahulu.

No	Judul Penelitian	Metode	Hasil
1	The Future of Laboratory Chemistry Learning and Teaching Must be Accesible (Egambaram, O., Hilton, K., Leigh, J., Richardson, R., Sarju, J., Slater, A., & Turner, B. 2022).	Penelitian Studi Literatur	Aksesibilitas dalam pembelajaran kimia berbasis laboratorium masih menjadi masalah besar, namun banyak cara untuk meningkatkannya dan hal ini bermanfaat bagi semua siswa.
2	Learning Beyond the Laboratory: A Web Application Framework for Development of Interactive Postlaboratory Exercises Kimia Moozeh (Farmer, J., Tihanyi, D., & Evans, G. J. 2020).	Penelitian Studi Literatur	Aplikasi web interaktif untuk latihan pasca-laboratorium kimia meningkatkan pemahaman dan keterlibatan siswa. Uji coba menunjukkan peningkatan signifikan dalam hasil tes dan umpan balik positif terkait kemudahan penggunaan dan efektivitas aplikasi. Temuan ini menegaskan bahwa teknologi interaktif dapat memperkaya pembelajaran kimia dan meningkatkan hasil akademik siswa.

Lanjutan **tabel 1.** Penelitian Terdahulu.

3	Gallardo-Williams, L. R. (2020). We Should Keep Developing Digital Laboratory Resources in the Postpandemic Era. <i>Journal of Chemical Education</i>	Penelitian Studi Literatur	Pentingnya melanjutkan penggunaan sumber daya laboratorium digital pasca-pandemi. Manfaat yang disoroti meliputi fleksibilitas, aksesibilitas, dan peningkatan keterlibatan siswa, serta menyarankan bahwa alat-alat digital ini harus terus dikembangkan untuk keuntungan jangka panjang dalam pendidikan kimia.
---	---	----------------------------	---

2.2 Sistem Informasi

Sistem informasi adalah sekumpulan komponen yang saling berhubungan yang mengumpulkan, memproses, menyimpan, dan menyebarkan data dan informasi; memberikan umpan balik yang membantu organisasi mencapai tujuan mereka (Stair & Reynolds, 2015).

Tujuan sistem informasi terdiri dari kegunaan (*usefulness*), ekonomi (*economic*), keandalan (*reliability*), pelayanan pelanggan (*customer service*), kesederhanaan (*simplicity*) dan fleksibilitas (*flexibility*).

1) Kegunaan (*Usefulness*)

Sistem harus menghasilkan informasi yang akurat, tepat waktu dan relevan untuk pengambilan keputusan manajemen dan personil operasi di dalam organisasi.

2) Ekonomi (*Economic*)

Semua bagian komponen sistem termasuk laporan, pengendalian dan mesin harus menyumbang suatu nilai manfaat setidaknya sebesar biaya yang dibutuhkan.

3) Keandalan (*Reliability*)

Keluaran sistem harus mampu beroperasi secara efektif baik pada waktu komponen manusia tidak hadir atau saat bagian-bagian mesin tidak beroperasi secara temporer.

4) Pelayanan Langgan (*Customer Service*)

Memberikan pelayanan yang baik atau tamah kepada para pelanggan. Sehingga sistem tersebut dapat dipahami oleh para pelanggannya.

5) Kesederhanaan (*Simplicity*)

Kesederhanaan sistem dapat dicapai dengan menggunakan bahasa yang mudah dipahami dan tidak menggunakan terlalu banyak fitur (Budiharto, dkk 2018).

6) Fleksibilitas (*Flexibility*)

Fleksibel dalam menangani perubahan yang terjadi, kepentingannya dapat menjadi alasan dengan kondisi dimana sistem operasi diwajibkan oleh organisasi.

2.3 Framework Laravel

Laravel adalah *framework* aplikasi web dengan sintaks yang ekspresif dan elegan yang telah menyediakan fondasi yang kokoh untuk ide besar Anda, membebaskan Anda untuk berkreasi tanpa terhambat oleh hal-hal kecil. (Taylor Otwell, 2011).

2.4 Website

Sebuah *website* adalah kumpulan halaman digital yang berisi berbagai informasi yang dapat diakses oleh pengguna. Jenis informasi ini dapat mencakup teks, gambar, animasi, suara, atau kombinasi berbagai elemen multimedia. Baik bentuk statis, di mana konten tetap tidak berubah, maupun bentuk dinamis, di mana pengguna dapat berinteraksi dengan elemen tertentu. Dengan perkembangan pesat dunia internet saat ini, pengelolaan konten menjadi komponen penting dari sebuah *website*. ini mencakup pembuatan, penyuntingan, dan pengaturan konten secara efisien agar memenuhi kebutuhan dan tujuan pengguna. Pengelolaan konten yang baik tidak hanya memastikan kualitas informasi yang tersedia tetapi juga meningkatkan pengalaman pengguna dan dapat membantu kesuksesan keseluruhan *website*, terutama dalam menjaga relevansi dan daya tarik kontes.

2.5 Model-View-Controller (MVC)

Model-View-Controller (MVC) ialah kerangka desain perangkat lunak yang difungsikan untuk mengisolasi elemen-elemen inti dalam pengembangan aplikasi. MVC memiliki tujuan untuk meningkatkan pemahaman pada kode, kemudahan dalam mengembangkan perangkat lunak, dan kolaborasi antar individu. Selain itu, MVC juga bertujuan untuk mendorong skalabilitas dan mempermudah pemeliharaan perangkat lunak. Model, *View*, dan *Controller* merupakan tiga elemen penting dalam MVC. Setiap individu memiliki peran dan tanggung jawab yang berbeda-beda. Di bawah ini adalah pengertian yang lebih rinci mengenai setiap komponen:

1) Model

- a) Model merupakan komponen yang bertugas mengatur data dan logika bisnis dalam suatu aplikasi.
- b) Model menyajikan bentuk data dan pola pikir perangkat lunak yang menjadi dasar aplikasi.
- c) Model memiliki kemampuan untuk berinteraksi dengan basis data, melaksanakan tugasnya dalam pengolahan data, serta menyampaikan hasilnya pada *Controller*.
- d) Model umumnya tidak terpengaruh oleh tampilan atau antarmuka pengguna dan beroperasi secara mandiri dari elemen arsitektur lainnya.

2) View

- a) Komponen *view* bertugas mengelola tampilan serta menyajikan data kepada pengguna.
- b) *View* menggambarkan bagaimana informasi disajikan kepada pengguna, seperti susunan, elemen visual, dan antarmuka pengguna.
- c) *View* tidak memiliki keterikatan dengan logika bisnis, tetapi hanya menunjukkan informasi yang disediakan oleh Model atau diatur oleh *Controller*.
- d) Dalam banyak situasi, suatu aplikasi bisa memiliki beberapa tampilan yang berbeda untuk memperlihatkan data dengan cara yang berbeda.

3) Controller

- a) Controller memiliki peran untuk menghubungkan antara Model dan View.
- b) *Controller* bertugas mengatur jalur informasi antara model dan view serta mengatur interaksi pengguna.
- c) *Controller* memberikan respon masukan pengguna dan menentukan tindakan yang perlu dilakukan, kemudian mengkoordinasikan Model untuk melaksanakan tugas yang sesuai dan memperbaharui tampilan saat diperlukan.
- d) *Controller* juga bertugas untuk mengendalikan alur navigasi dan mengontrol arus aplikasi.

Keuntungan utama menggunakan MVC adalah pemisahan yang jelas antara tampilan, data, dan logika bisnis. Hal ini membuat proses pengembangan lebih terstruktur, mempermudah pemeliharaan dan patching, serta memungkinkan tim yang berbeda untuk menguji dan mengembangkan setiap komponen secara paralel. Dengan mengikuti prinsip-prinsip MVC, aplikasi dapat menjadi lebih modular, lebih mudah dikelola, dan lebih efisien dalam jangka panjang.

2.6 Hypertext Preprocessor (PHP)

Rasmus Lerdorf, pencipta PHP, mendefinisikan PHP sebagai bahasa pemrograman "*Hypertext Preprocessor*" yang "mudah dipelajari dan digunakan" dengan tujuan utama "membuat halaman web dinamis". Diperkenalkan pertama kali pada tahun 1994, PHP awalnya merupakan sekumpulan skrip Perl kecil untuk melacak pengunjung situs web pribadinya. Seiring perkembangannya, PHP terus disempurnakan dan menjadi bahasa pemrograman yang lengkap dan mandiri seperti sekarang. Rasmus Lerdorf menekankan kemudahan penggunaan dan fleksibilitas PHP. Menurutnya, PHP dirancang untuk membuat halaman web dinamis tanpa memerlukan keahlian pemrograman yang rumit. Fleksibilitasnya memungkinkan PHP digunakan untuk berbagai macam tugas, mulai dari membangun situs web sederhana hingga aplikasi web kompleks dan bahkan skrip baris perintah. Hingga saat ini, Rasmus Lerdorf masih aktif dalam pengembangan PHP dan diakui sebagai

salah satu tokoh berpengaruh dalam dunia pemrograman web. PHP terus berkembang dan menjadi pilihan populer bagi developer yang ingin membangun situs web dan aplikasi web modern.

2.7 Unified Modelling Language (UML)

Unified Modelling Language (UML) tidak memiliki pencipta tunggal, melainkan dikembangkan oleh sekelompok pakar industri dan akademisi yang dipimpin oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson. Ketiganya, yang dikenal sebagai "*The Three Musketeers*" atau "*The Gang of Three*", memiliki peran penting dalam menyatukan berbagai notasi pemodelan yang ada pada masanya menjadi satu bahasa pemodelan yang universal dan mudah dimengerti. Grady Booch, seorang penulis dan pembicara terkenal di bidang pemrograman berorientasi objek, fokus pada aspek desain berorientasi objek dalam UML. James Rumbaugh, pakar pemodelan bisnis dan analisis sistem, berkontribusi dalam pengembangan notasi untuk pemodelan kasus penggunaan dan proses bisnis. Sedangkan Ivar Jacobson, seorang pelopor dalam metodologi *Use Case Driven Development* (UCDD), memfokuskan diri pada pengembangan notasi untuk memodelkan interaksi antara pengguna dan sistem.

Kerja sama mereka bertiga menghasilkan UML versi 1.0 pada tahun 1997, yang kemudian terus dikembangkan dan disempurnakan hingga versi terbaru saat ini, yaitu UML 2.5. Menurut Grady Booch, UML adalah "bahasa pemodelan visual untuk memvisualisasikan, menspesifikasikan, membangun, dan mendokumentasikan arsitektur perangkat lunak". James Rumbaugh menambahkan bahwa UML "memberikan cara standar untuk memvisualisasikan sistem perangkat lunak, termasuk struktur, perilaku, dan interaksi". Ivar Jacobson menekankan manfaat UML dalam komunikasi, "UML memungkinkan tim pengembangan untuk berkomunikasi secara efektif tentang desain sistem perangkat lunak". Secara keseluruhan, UML diciptakan untuk mengatasi kompleksitas sistem perangkat lunak yang semakin

meningkat dan membantu para developer dalam mendesain, membangun, dan memelihara sistem yang lebih baik.

UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan kebutuhan sistem, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Dalam penelitian ini menggunakan beberapa diagram diantaranya sebagai berikut.:

1) *Use Case Diagram*

Grady Booch (1994) memperkenalkan konsep "*use case*" sebagai cara untuk menggambarkan interaksi antara pengguna dan sistem dalam bukunya "*Object-Oriented Analysis and Design with Applications*". Konsep ini kemudian dikembangkan oleh James Rumbaugh (1998) dalam bukunya "*UML: The Unified Modeling Language*" dengan mendefinisikan notasi untuk diagram *use case* dan elemen-elemennya, seperti aktor, *use case*, dan hubungan di antaranya.

Tabel 2. Simbol *Use Case Diagram*.

No	Gambar	Nama	Keterangan
1		<i>Use Case</i>	Deskripsi dari urutan aksi- aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
2		<i>Actor</i>	Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .
3		<i>Association</i>	Sebagai penghubung antara aktor dengan <i>use case</i> yang saling berinteraksi.

Lanjutan **tabel 2**. Simbol *Use Case Diagram*.

4		<i>Generalization</i>	hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
5		<i>Extend</i>	Menspesifikasikan bahwa use case target adalah perluasan perilaku dari <i>use case</i> sumbernya di suatu titik.
6		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
7		<i>Include</i>	Menspesifikasikan bahwa <i>Use case</i> sumber secara eksplisit.
8		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) dapat mempengaruhi segala elemen yang bergantung di elemen yang tidak mandiri

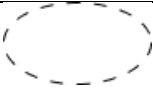
2) *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram ini sangat penting dalam fase desain sebuah sistem karena membantu dalam merinci atribut dan metode yang akan dimiliki oleh setiap kelas, serta hubungan antara kelas-kelas tersebut seperti asosiasi, agregasi, dan pewarisan.

Grady Booch (1994) memperkenalkan konsep "*class*" dan "*relationship*" sebagai elemen dasar pemrograman berorientasi objek dalam bukunya "*Object-Oriented Analysis and Design with Applications*". James Rumbaugh (1998) kemudian mengembangkan notasi untuk diagram *class*

dalam bukunya "UML: *The Unified Modeling Language*", mendefinisikan simbol untuk kelas, atribut, metode, dan hubungan antar kelas.

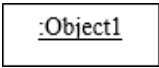
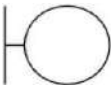
Tabel 3. Simbol *Class Diagram*.

No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Menunjukkan hubungan antara objek anak (<i>descendent</i>) yang berbagi perilaku dan struktur data dengan objek induk (<i>ancestor</i>).
2		<i>Message</i>	Himpunan dari objek yang berbagi atribut dan operasi yang sama.
3		<i>Collaboration</i>	Menunjukkan interaksi aturan dan elemen yang saling bersatu
4		<i>Realization</i>	Menunjukkan operasi yang dilakukan oleh objek.
5		<i>Dependency</i>	Menunjukkan hubungan yang terjadi pada elemen mandiri mempengaruhi elemen yang bergantung padanya.
6		<i>Association</i>	Menghubungkan antara objek satu dengan objek lainnya.

3) *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Ivar Jacobson (1994), pelopor *Use Case Driven Development* (UCDD), memperkenalkan konsep "*sequence diagram*" untuk menggambarkan interaksi antar objek dalam sistem. Gamma, Helm, Johnson, dan Vlissides (1995) kemudian mempopulerkan penggunaan *sequence diagram* dalam buku mereka "*Design Patterns: Elements of Reusable Object-Oriented Software*" untuk memvisualisasikan urutan pesan dan kolaborasi antar objek.

Tabel 4. Simbol *Sequence Diagram*.

No	Gambar	Nama	Keterangan
1		<i>Object</i>	Instance dari sebuah class yang dituliskan secara horizontal.
2		<i>Actor</i>	Spesifikasi peran yang digunakan user untuk berinteraksi dengan use case.
3		<i>Lifeline</i>	Menunjukkan indikasi sebuah objek dalam basis waktu.
4		<i>Activation</i>	Menunjukkan indikasi dari suatu objek yang akan melakukan <i>action</i> .
5		<i>Boundary</i>	Penggambaran dari form. Biasanya berupa tepi dari sistem dan alat yang berada dan berinteraksi dalam sistem.
6		<i>Control</i>	Penggambaran penghubung antara boundary dengan tabel.
7		<i>Entity</i>	Penggambaran untuk menangani informasi yang akan disimpan secara permanen
8		<i>Message</i>	Penghubung komunikasi antar objek.
9		<i>Self-Message</i>	Menunjukkan komunikasi kembali ke dalam sebuah objek itu sendiri.
10		<i>Loop</i>	Menunjukkan eksekusi terjadi berulang kali.

4) *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Sukamto & Shalahuddin, 2018).

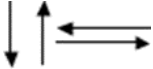
David Harel (1987) memperkenalkan konsep "*statechart*" sebagai model untuk menggambarkan perilaku sistem yang menjadi dasar bagi diagram

activity UML. Konsep *statechart* ini memperkenalkan cara yang lebih dinamis untuk merepresentasikan perubahan keadaan dalam sebuah sistem, yang sangat berguna dalam mengidentifikasi dan memodelkan aliran kontrol yang kompleks. Michael Jesse Siatkowski (1999) kemudian mendefinisikan notasi untuk diagram *activity* dalam bukunya "*Modeling and Executing Business Processes with UML*". Dalam bukunya, Siatkowski menguraikan bagaimana diagram *activity* dapat digunakan untuk memodelkan proses bisnis secara rinci, termasuk penggunaan *swimlanes* untuk memisahkan tanggung jawab berbagai aktor atau unit organisasi, serta penggunaan token untuk menunjukkan aliran proses dan aktivitas dalam sistem. Pendekatan ini memberikan cara yang lebih visual dan intuitif untuk memahami bagaimana berbagai aktivitas dalam sistem saling berinteraksi dan mengalir dari satu tahap ke tahap berikutnya.

Tabel 5. Simbol *Activity Diagram*.

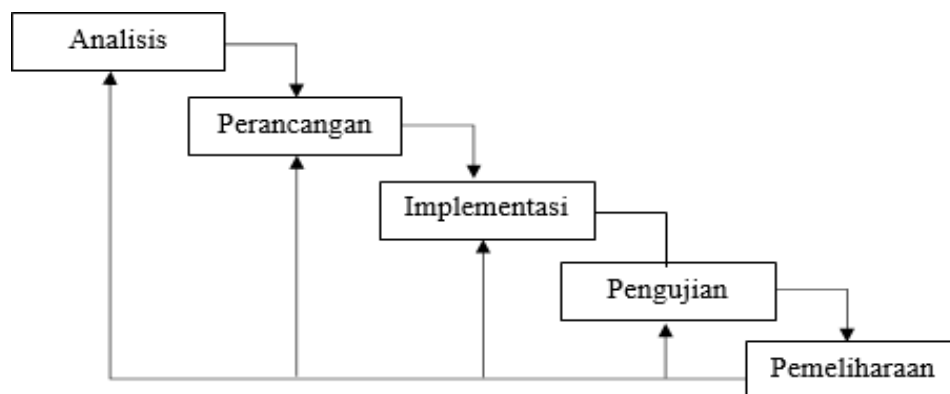
No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Menunjukkan bagaimana interaksi antara masing – masing kelas.
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Initial Node</i>	Merupakan tanda tentang bagaimana objek dibentuk atau dimulai.
4		<i>Activity Final Node</i>	Simbol bagaimana objek dibentuk atau diakhiri.
5		<i>Decision</i>	Menampilkan suatu keputusan yang harus diambil pada kondisi tertentu.

Lanjutan **tabel 5**. Simbol *Activity Diagram*.

6		<i>Line Connector</i>	Penghubung antara satu simbol dengan yang lainnya.
---	---	-----------------------	--

2.8 Metode Pengembangan Sistem

Metode air terjun atau metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), nama model ini sebenarnya adalah “*Linear Sequential Model*”, di mana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna (*communication*) lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), pemodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem ke para pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan (Pressman dan Maxim, 2020). Tahapan metode *waterfall* dapat dilihat dari Gambar 1.



Gambar 1. Metode *Waterfall*.

Berikut penjelasan dari tahapan metode *waterfall* pada Gambar 1:

1. Analisis dan Definisi Persyaratan

Tujuan dari tahapan ini adalah untuk mendapatkan pemahaman tentang bagaimana suatu sistem beroperasi dan mengidentifikasi masalahnya. Setelah itu, seseorang dapat menentukan kebutuhan sistem secara garis besar sebagai persiapan untuk tahapan selanjutnya, yaitu perancangan.

Wawancara atau diskusi dengan pihak yang membutuhkan software atau stakeholder dapat dilakukan untuk mendapatkan informasi tentang pembuatan *software*.

2. Perancangan

Tahap perancangan ini bertujuan untuk memberikan gambaran umum tentang sistem yang akan dibuat kepada pengguna dan memberikan rancang bangun lengkap sistem kepada pihak yang terlibat dalam pengembangan sistem. Tujuan dari tahap ini adalah untuk memberikan gambaran tentang sistem yang akan dibuat kepada pengguna, yang kemudian akan masuk ke tahap persiapan implementasi.

3. Implementasi

Setelah proses perencanaan sistem selesai, langkah selanjutnya adalah melanjutkan ke tahap implementasi sistem atau tahap penerapan dari rencana yang telah dibuat. Setelah itu, langkah berikutnya adalah merubah rencana sistem menjadi kode-kode bahasa pemrograman yang diinginkan. di tahap ini dilakukan produksi komponen. Elemen-elemen sistem termasuk paket aplikasi, antarmuka, dan basis data.

4. Pengujian

Pada tahap pengujian ini, dilakukan dengan tujuan memverifikasi bahwa perangkat lunak yang dibuat adalah akurat dan cocok dengan keperluan yang telah dijelaskan oleh pengguna.

5. Pemeliharaan

Pada tahap ini, dilakukannya pemeliharaan terhadap perangkat lunak yang telah diserahkan kepada pengguna. Tahapan pemeliharaan melakukan evaluasi terhadap sistem yang baru untuk mengetahui kelemahan dan kelebihan sistem yang telah dibuat dan untuk mengetahui apakah sistem telah memenuhi tujuan yang ingin dicapai. Berdasarkan hasil evaluasi ini dimungkinkan untuk melakukan perubahan yang diperlukan terhadap sistem agar sistem dapat digunakan dengan baik.

2.9 Desain Sistem

Pressman (2019) mendefinisikan desain sistem sebagai proses transformasi spesifikasi yang lebih tinggi menjadi bentuk yang lebih terperinci dan dapat diimplementasikan dalam bentuk sistem komputer. Desain sistem ini mencakup pemilihan struktur data, algoritma, dan implementasi komponen-komponen sistem. Beberapa langkah yang dilakukan pada proses desain sistem yaitu sebagai berikut:

1. Menganalisis masalah dari pengguna berdasarkan sumber pengetahuan yang didapat saat melakukan wawancara. Tujuannya adalah untuk mendapatkan pengetahuan terkait sistem yang akan dibuat yang berdasarkan kebutuhan dari pengguna.
2. Studi kelayakan, yaitu membandingkan satu alternatif terhadap alternatif masalah yang lain untuk menemukan suatu jalan keluar.
3. Perancangan sistem, yaitu membuat usulan terkait pemecahan masalah yang diusulkan kepada pengguna.
4. Detail desain, melakukan pemecahan masalah secara terperinci.
5. Penerapan yaitu memindahkan detail rancangan yang telah dibuat kedalam bentuk kode program yang diinginkan.
6. Pemeliharaan dan evaluasi terkait sistem yang telah diterapkan.

2.10 Metode Pengujian Sistem

Dalam buku "*Modern Systems Analysis and Design*" menjelaskan bahwa metode pengujian sistem mencakup pengujian *alpha* dan *beta*. Pengujian *alpha* dilakukan oleh tim pengembang atau QA di lingkungan internal untuk mendeteksi dan memperbaiki *bug* atau masalah sebelum sistem diluncurkan kepada pengguna eksternal. Pengujian ini memastikan bahwa sistem bekerja sesuai dengan spesifikasi yang ditentukan. Selanjutnya, pengujian *beta* dilakukan oleh sekelompok pengguna akhir di lingkungan operasional nyata. Tujuannya adalah untuk mendapatkan umpan balik tentang kinerja, kegunaan, dan stabilitas sistem dari perspektif pengguna akhir. Pengujian *beta* membantu mengidentifikasi masalah yang mungkin tidak terdeteksi selama pengujian *alpha* dan memastikan bahwa sistem siap digunakan secara luas. Umpan balik

yang diperoleh digunakan untuk melakukan perbaikan akhir sebelum perilisan sistem secara resmi (Valacich, George, dan Hoffer, 2020). Berikut penjelasan dari pengujian *alpha* dan *beta*.

1. Pengujian *Alpha*

Pengujian *alpha* dilakukan untuk mengetahui fungsional dari sistem informasi dengan menggunakan pendekatan *black box testing*. Pengujian *black box* merupakan salah satu teknik metode pengujian sistem yang fokus pada kebutuhan fungsional sistem. Pengujian *black box* dapat menurunkan serangkaian kondisi input yang menyebabkan sepenuhnya melaksanakan fungsional untuk suatu program (Pradipta, 2020). Pengujian *black box* memiliki kemungkinan untuk menemukan kesalahan dalam kategori sebagai berikut (Masripah dan Ramayanti, 2020):

- 1) Kesalahan atau hilangnya fungsi,
- 2) Kesalahan tampilan,
- 3) Kesalahan pada struktur data pada database,
- 4) Kesalahan kinerja, dan
- 5) Inisiasi dan kesalahan terminasi.

2. Pengujian *Beta*

Pengujian *beta* merupakan pengujian yang bersifat langsung di lingkungan yang sebenarnya dengan penyebaran kuesioner yang akan dihitung untuk dapat diambil kesimpulan terhadap penilaian aplikasi yang dibangun (Suandi dkk, 2017) dalam (Masripah & Ramayanti, 2020). Pengujian *beta* dilakukan pengguna sistem tanpa kehadiran pihak pembangun aplikasi. Pengujian *beta* merupakan pengujian yang bersifat langsung di lingkungan sebenarnya. Kuesioner pengujian *beta* merupakan media yang digunakan pengguna sistem untuk memberikan penilaian terhadap sistem yang dibangun. Berdasarkan kuesioner tersebut dilakukan penilaian penerapan sistem yang dibangun (Rosano, 2019).

Setelah dilakukan pengisian kuesioner oleh pengguna, dilakukan perhitungan dengan skala likert. Skala likert adalah suatu skala

psikometrik yang umum digunakan dalam kuesioner dan merupakan skala yang paling banyak digunakan dalam riset berupa survei (Pradipta, 2020). Skala likert digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Skala Likert dapat menjabarkan variabel yang diukur menjadi indikator variabel (Nasution, 2018).

III. METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian ini merupakan suatu upaya yang dilakukan di Jurusan Kimia Universitas Lampung, yang berlokasi di Jalan Prof. Dr. Ir. Sumantri Brojonegoro No.1, Gedong Meneng, Kecamatan Rajabasa, Kota Bandar Lampung. Adapun periode penelitian, dimulai dari tanggal 20 Februari 2023 hingga 10 Agustus 2023, mencakup fase signifikan di mana peneliti terlibat dalam serangkaian kegiatan, pengumpulan data, serta analisis. Rentang waktu yang panjang memberikan kesempatan untuk menyelidiki secara menyeluruh dan mendalam, serta memastikan bahwa setiap aspek penelitian dapat ditangani secara cermat dan komprehensif.

Selanjutnya, perencanaan penulisan skripsi dimulai pada Oktober 2023 hingga April 2024 merupakan tahapan penting dalam pengembangan penelitian ini. Proses ini melibatkan pengumpulan data tambahan, analisis mendalam, serta penyusunan hasil dan temuan yang diperoleh selama periode penelitian sebelumnya. Informasi lebih lanjut mengenai perencanaan selama penelitian dan penulisan skripsi dapat ditemukan dalam tabel 6.

Tabel 6. Perencanaan Penelitian.

Tahapan	Kegiatan	2023												2024							
		Feb	Mar	Apr	Mei	Jun	Jul	Agu	Sep	Okt	Nov	Des	Jan	Feb	Mar	Apr	Mei	Jun	Jul	Agu	
Penelitian Awal	Analisa kebutuhan pengguna																				
	Pengumpulan data																				
	Perancangan dan pengembangan																				
	Pengujian																				
	Pemahaman studi literatur																				
Penelitian Lanjutan	Penyusunan Bab I - III																				
	Seminar Usul Penelitian																				
	Revisi Bab I-III dan menyusun Bab IV																				
	Seminar Hasil Penelitian																				
Penelitian Akhir	Revisi Draft Skripsi																				
	Sidang Komprehensif																				

3.2 Alat Penelitian

Mengembangkan modul manajemen laboratorium dan pengelolaan akun pada situs web Program Data Center Chemistry membutuhkan sejumlah alat dan teknologi yang mendukung. Untuk mendukung proyek pengembangan ini, berikut adalah beberapa sumber daya yang diperlukan:

3.2.1 Perangkat Keras (*Hardware*)

Hardware yang digunakan dalam penelitian ini adalah sebagai berikut:

- a. Model: ASUS TUF Gaming F15 FX506HC
- b. *Processor*: 11th Gen Intel(R) Core (TM) i5-11400H @ 2.70GHz
- c. *Random Access Memory*: 24GB RAM
- d. *Storage*: 512GB PCIe SSD
- e. *Video Graphics Array (VGA)*: NVIDIA RTX 3050

3.2.2 Perangkat Lunak (*Software*)

Software yang digunakan adalah sebagai berikut:

- a. *Operating System*: Windows 11 Pro 64-Bit
- b. *Tools*

Berikut *tools* yang digunakan selama pengembangan berlangsung.

1. Visual Studio Code

Visual Studio Code (VSCode) adalah editor kode sumber sumber terbuka yang sangat populer dan kuat, dikembangkan oleh Microsoft. VSCode dapat digunakan di Windows, macOS, dan Linux, dan terkenal karena ringan, cepat, serta memiliki ekosistem ekstensi yang luas untuk memperluas fungsionalitasnya. Dengan integrasi Git, fitur *Intellisense*, *debugger* bawaan, terminal terintegrasi, dan dukungan untuk berbagai bahasa pemrograman, VSCode menjadi pilihan utama bagi banyak pengembang. Selain itu, kehadiran komunitas yang besar memberikan akses mudah ke tutorial dan dukungan, sementara tingkat kustomisasi yang tinggi memungkinkan pengguna untuk

menyesuaikan editor sesuai preferensi mereka, menjadikannya alat yang serbaguna dan kuat untuk pengembangan perangkat lunak.

2. Laragon

Laragon adalah lingkungan pengembangan web lokal yang sangat *user-friendly* dan ringan, dirancang untuk memudahkan pengembang dalam mengatur server web, basis data, serta lingkungan pengembangan PHP. Laragon memiliki antarmuka yang intuitif, mendukung banyak bahasa pemrograman, dan menyediakan fitur-fitur seperti penyesuaian konfigurasi dengan cepat, penjalanan proyek dengan satu klik, serta integrasi dengan alat-alat populer seperti Git, Composer, dan Node.js. Dengan kemampuan ini, Laragon menjadi pilihan yang populer untuk pengembangan *website* lokal di Windows.

3. Framework Laravel

Laravel adalah salah satu *framework* pengembangan *website* berbasis PHP yang sangat populer dan kuat. Dikenal dengan sintaks yang elegan dan mudah dipahami, Laravel menyediakan beragam alat dan fitur bawaan untuk memudahkan pengembangan aplikasi web, seperti manajemen otentikasi, *routing* yang kuat, sistem *templating* *Blade*, ORM *Eloquent*, serta dukungan untuk pengujian. Laravel juga memiliki komunitas yang besar dan aktif, serta berbagai ekstensi dan paket tambahan yang membuatnya menjadi pilihan utama bagi pengembang *website* untuk membangun aplikasi yang *scalable* dan *maintainable*.

4. Web Browser

Web browser, atau peramban web, adalah perangkat lunak yang digunakan untuk mengakses dan menjelajahi *World Wide Web*. Fungsinya adalah untuk mengambil dan menampilkan halaman web yang terdiri dari teks, gambar, video, dan elemen-elemen lainnya dari server web. *Browser* modern seperti Chrome, Firefox, Safari, dan Edge memiliki berbagai fitur tambahan, termasuk *bookmarking*, *tabbed browsing*, manajemen kata sandi, serta kemampuan untuk menginstal ekstensi dan plug-in tambahan yang dapat memperluas

fungsionalitasnya, seperti blokir iklan, pengelolaan kata sandi, dan banyak lagi.

5. *Database Management System (DBMS)*

Sistem Manajemen Basis Data (DBMS) adalah perangkat lunak yang digunakan untuk mengelola, menyimpan, dan mengakses data dalam basis data. DBMS menyediakan antarmuka yang memungkinkan pengguna untuk menjalankan query, menambahkan, mengubah, dan menghapus data dengan mudah. Contoh-contoh DBMS yang populer termasuk MySQL, PostgreSQL, Microsoft SQL Server, Oracle *Database*, dan MongoDB. Setiap DBMS memiliki karakteristik dan kegunaan yang berbeda, seperti MySQL yang sering digunakan untuk aplikasi web, PostgreSQL yang dikenal karena kehandalannya, SQL Server yang mendukung ekosistem Microsoft, Oracle *Database* yang digunakan dalam bisnis dan perusahaan besar, serta MongoDB yang merupakan basis data NoSQL yang cocok untuk aplikasi yang membutuhkan fleksibilitas dalam menyimpan data semi-struktural atau tidak terstruktur.

6. Trello

Trello adalah platform manajemen proyek berbasis web yang sangat visual dan intuitif, yang memungkinkan individu dan tim untuk mengatur tugas dan proyek mereka dalam bentuk papan dan kartu. Setiap kartu dapat berisi detail tugas, lampiran, daftar tugas yang dapat dicentang, serta komentar, dan dapat dipindahkan di antara kolom-kolom yang mewakili tahapan proyek. Trello memungkinkan kolaborasi yang efisien, pelacakan progres, dan pengelolaan tugas yang mudah dipahami, menjadikannya alat yang populer untuk manajemen proyek, perencanaan, dan pelacakan pekerjaan.

7. Git

Git adalah sistem kontrol versi terdistribusi yang digunakan oleh pengembang perangkat lunak untuk mengelola perubahan dalam kode sumber proyek. Ini memungkinkan tim pengembang untuk bekerja secara kolaboratif, melacak perubahan kode, dan menggabungkan

kode dari berbagai sumber dengan aman. Git memungkinkan pencatatan setiap perubahan dalam sejarah proyek, memudahkan pemulihan jika terjadi kesalahan, dan menyediakan alat yang kuat untuk menggabungkan perubahan antara cabang-cabang berbeda. Git sangat penting dalam pengembangan perangkat lunak modern dan digunakan secara luas dalam proyek-proyek *open-source* maupun komersial.

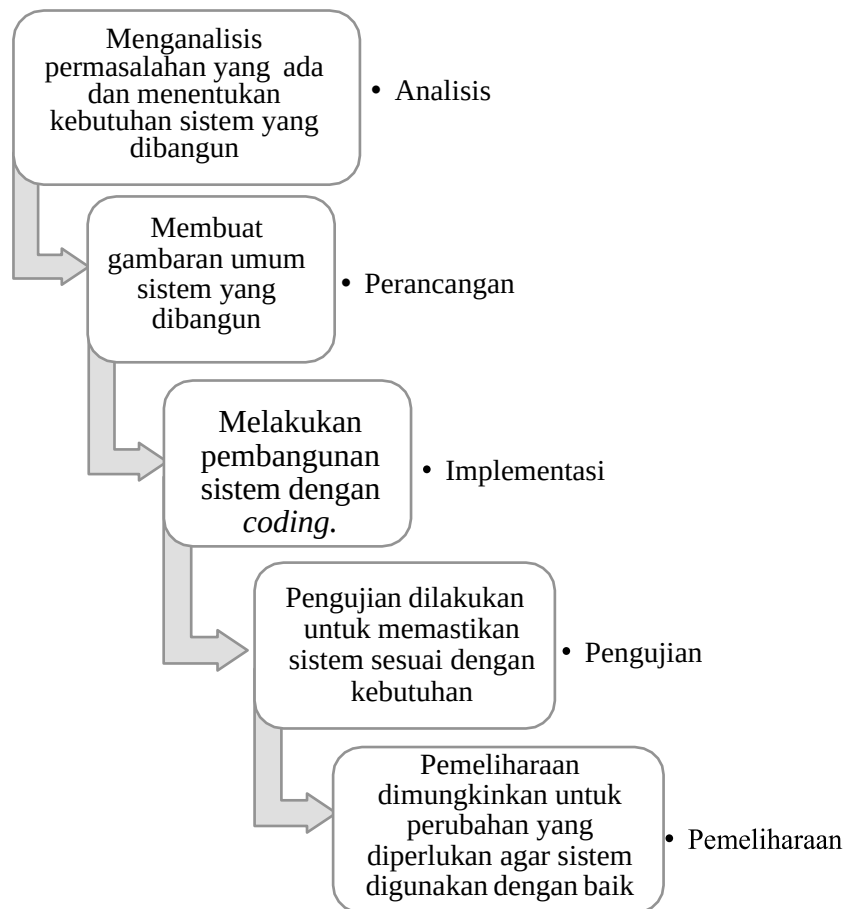
8. Composer

Composer adalah alat manajemen dependensi yang digunakan dalam pengembangan perangkat lunak berbasis PHP. Ini memungkinkan pengembang untuk dengan mudah mengelola dan mengintegrasikan pustaka dan paket-paket eksternal ke dalam proyek mereka. Dengan menggunakan *file* konfigurasi `composer.json`, Composer dapat mengunduh, menginstal, dan memperbarui dependensi PHP secara otomatis, mempermudah pengembangan dan memastikan bahwa proyek PHP selalu menggunakan versi yang sesuai dari pustaka-pustaka yang dibutuhkan. Composer adalah alat yang kritis dalam ekosistem PHP modern dan memfasilitasi manajemen dependensi yang efisien.

3.3 Tahapan Penelitian

3.3.1 Kerangka Penelitian

Penelitian ini bertujuan agar membantu permasalahan yang ada pada Jurusan Kimia terkait dengan pendataan pada seminar mahasiswa hingga *asset* laboratorium. Kerangka penelitian dapat dilihat pada Gambar 2.



Gambar 2. Alur Penelitian.

Penelitian ini dilatarbelakangi karena adanya masalah pada pendataan pada seminar mahasiswa hingga pendataan *asset* laboratorium di Jurusan Kimia FMIPA Universitas Lampung. Pada 20 Februari 2023, telah dilakukan wawancara dengan ketua jurusan Kimia untuk menanyakan terkait permasalahan pada pendataan pada seminar mahasiswa hingga pendataan *asset* laboratorium. Hasil dari wawancara didapatkan bahwa proses pendataan seminar mahasiswa yang masih menggunakan Google Forms, sementara pengecekan secara manual melalui bagian administrasi di Jurusan Kimia. Selain itu, pendataan *asset* laboratorium juga dilakukan secara manual melalui Microsoft Excel dan Google Forms. Dengan adanya hasil wawancara, akhirnya

akan dibuat sistem informasi yang dapat memecahkan permasalahan tersebut, salah satunya dengan membuat sistem dengan metode *waterfall*.

3.3.2 Metode Pengumpulan Data

Metode pengumpulan data dapat dibedakan menjadi dua yaitu:

1. Data Primer

Data primer didapatkan dari hasil wawancara dengan Ketua Jurusan Kimia Universitas Lampung.

2. Data Sekunder

Data sekunder didapatkan dari buku, jurnal, dan *internet* yang menyajikan informasi mengenai pendataan pada seminar mahasiswa hingga pendataan *asset* laboratorium.

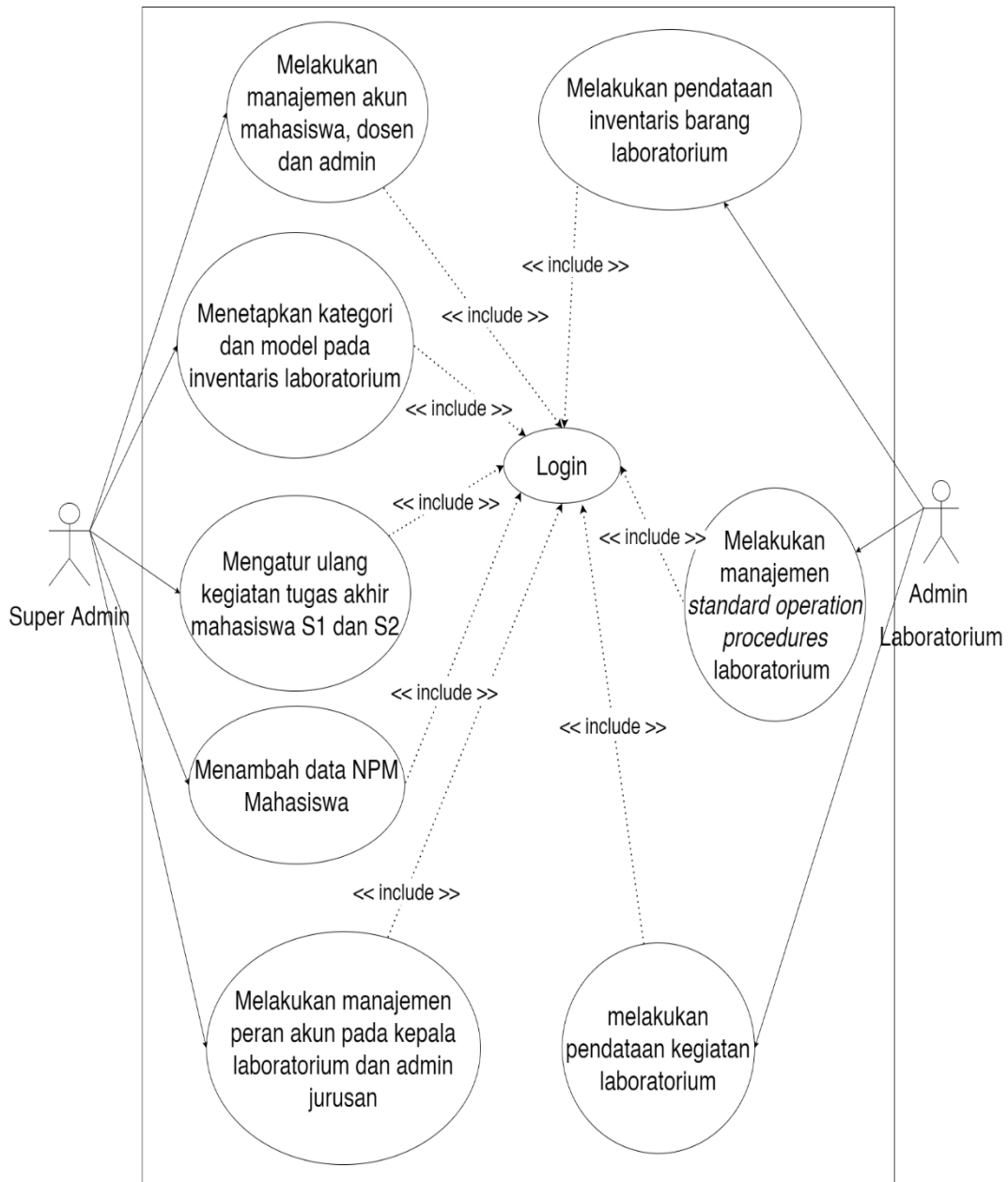
3.4 Analisis Kebutuhan Sistem

Dalam tahap analisis kebutuhan sistem, telah dilakukan perancangan upaya penyelesaian masalah yang terkait dengan pengelolaan data pada seminar mahasiswa dan pendataan aset laboratorium di Jurusan Kimia Universitas Lampung. Langkah ini diambil untuk menciptakan sebuah sistem yang diharapkan mampu memberikan solusi efektif terhadap permasalahan tersebut. Sistem yang dirancang ini diharapkan dapat memberikan bantuan signifikan dalam mengatasi tantangan yang muncul seiring dengan pengelolaan data seminar mahasiswa dan aset laboratorium.

Selanjutnya, kebutuhan pada sistem tersebut telah dijelaskan secara rinci dan terstruktur ke dalam berbagai modul, yang kemudian direpresentasikan dalam bentuk diagram *use case*, sebagaimana terlihat pada Gambar 2. Diagram *use case* ini memberikan gambaran visual mengenai interaksi antara pengguna dan sistem, serta menjelaskan fungsionalitas utama yang diharapkan dari setiap modulnya.

Dalam konteks ini, sistem yang dirancang tidak hanya bertujuan untuk menyelesaikan permasalahan yang timbul, tetapi juga untuk meningkatkan efisiensi dan efektivitas dalam pengelolaan data seminar mahasiswa dan pendataan aset laboratorium di lingkungan Jurusan Kimia. Adanya modul-modul yang telah dirinci dalam diagram *use case* menjadi pedoman dalam pengembangan sistem, memastikan bahwa setiap aspek kebutuhan pengguna tercakup dengan baik.

Diharapkan bahwa dengan adanya implementasi sistem ini, proses pengelolaan data di Jurusan Kimia dapat berjalan lebih lancar dan terorganisir. Selain itu, sistem ini diharapkan dapat memberikan kontribusi positif dalam meningkatkan kualitas manajemen data seminar mahasiswa dan aset laboratorium, sehingga seluruh proses dapat dilakukan secara lebih efisien dan akurat.



Gambar 3. Use Case Diagram Admin dan Super Admin Pada Website CPDC.

Berdasarkan Gambar 3, terdapat 2 level pengguna diantaranya adalah sebagai berikut:

1. Admin Laboratorium

Admin laboratorium dapat melakukan pendataan pada inventaris barang yang ada di laboratorium dan mendata kegiatan yang ada di laboratorium.

2. Super Admin

Super admin memiliki kemampuan untuk manajemen data akun dari mahasiswa hingga dosen, menambahkan kategori dan model pada inventaris laboratorium, mengatur ulang kegiatan tugas akhir pada mahasiswa S1 dan S2, menambahkan data mahasiswa, serta melakukan manajemen akun pada di bagian laboratorium dan admin jurusan.

3.5 Uraian Tahapan Pengembangan Sistem

3.5.1 *Communication*

Tahapan ini dilakukan dengan pengumpulan kebutuhan proyek yang akan dibuat, kebutuhan tersebut dibagi menjadi dua macam, yaitu kebutuhan fungsional dan non-fungsional yang akan diuraikan sebagai berikut.

a. Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang berisi proses yang dilakukan oleh sistem. Kebutuhan ini mencakup proses pengerjaan sistem dengan masukan tertentu. Sistem yang dikembangkan memiliki fitur yang dibutuhkan pada modul pengelolaan akun manajemen laboratorium di Jurusan Kimia Universitas Lampung. Kebutuhan tersebut akan diuraikan sebagai berikut.

1. Akses modul manajemen laboratorium dan pengelolaan akun terlebih dahulu melakukan *login* dengan memasukkan *email* dan *password*.
2. Admin laboratorium dapat mengelola seluruh aset hingga SOP yang ada dalam laboratorium sesuai penugasan dimana admin tersebut ditempatkan.
3. Super admin dapat mengelola user sesuai dengan hak aksesnya dan menentukan kategori dan model pada aset laboratorium agar tidak terjadi redudansi data.

b. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan suatu kebutuhan terkait dengan kemudahan penggunaan sistem. Pada pengembangan sistem ini terdapat

kebutuhan non-fungsional yang terdiri dari *hardware*, *software* dan kemudahan pengguna.

1. *Software*

Software yang digunakan selama pengembangan sistem informasi ini adalah sebagai berikut.

1. Sistem Operasi: Windows 11 Pro 64bit
2. Aplikasi
 - a. Visual Studio Code versi 1.85.1 sebagai teks editor
 - b. Laragon versi 6.0 sebagai *local server*, dengan modul yang digunakan:
 1. Apache sebagai *web server*
 2. MySQL sebagai *database server*
 - c. Chrome versi 121.0.6167.141 sebagai web browser
 - d. Draw.io versi 13.9.9 untuk desain sistem dalam bentuk Unified Model Language (UML)

2. *Hardware*

Hardware yang digunakan selama pengembangan sistem informasi ini adalah sebagai berikut.

1. Model: ASUS TUF Gaming F15 FX506HC
2. Processor: 11th Gen Intel(R) Core (TM) i5-11400H @ 2.70GHz
3. Random Access Memory: 24GB RAM
4. Storage: 512GB PCIe SSD
5. Video Graphics Array (VGA): NVIDIA RTX 3050

3. Kemudahan Pengguna

Kemudahan yang didapatkan pengguna dalam sistem ini adalah sebagai berikut.

1. Sistem mudah diakses
2. Sistem mudah digunakan
3. Sistem menggunakan bahasa indonesia sehingga mudah dimengerti oleh pengguna
4. Sistem dapat membantu memudahkan dalam pengelolaan akun dan manajemen laboratorium

3.5.2 Planning

Pada tahap ini akan dibuat estimasi rencana kerja proyek yang telah diinisiasikan, tahapan ini akan diuraikan kegiatan mulai dari tahapan *modeling, construction, deployment*, hingga testing program. Rencana kerja proyek dapat dilihat pada Tabel 7.

Tabel 7. Rencana Kerja Proyek.

Kegiatan	Detail Pekerjaan	2023						
		Feb	Mar	Apr	Mei	Jun	Jul	Agu
Modelling	<i>Use Case Diagram</i>							
	<i>Activity Diagram</i>							
Construction	Data NPM Mahasiswa, Akun Mahasiwa, Dosen dan Admin							
	Kategori dan Model aset laboratorium							
	Pengelolaan akun admin lab dan jurusan							
	Aktivitas laboratorium							
	SOP							
	Inventaris laboratorium							
Deployment	Hosting							
Testing Program (Black Box)	Pembuatan skenario Pengujian							

Tabel 7 merupakan tahapan rencana kerja proyek yang dilakukan berdasarkan tahapan yang ada pada metode *waterfall*. Pada tabel 7 tahapan dimulai dari kegiatan modeling atau mendesain sistem, dilanjutkan dengan construction untuk membuat implementasi kode program terhadap kegiatan *modeling* yang telah dilakukan. Setelah tahap *construction* selesai lalu dilakukan tahap *deployment* atau menyebarluaskan sistem dengan cara hosting. Jika tahap *deployment* selesai maka dilakukan testing program dengan metode *Black Box* untuk menguji fungsionalitas sistem yang dikembangkan.

3.6 Desain Sistem dan Activity Diagram

Setelah tahap analisis selesai, lanjut ke tahap desain sistem untuk mempermudah pengembangan lebih lanjut. Pada tahap ini, fokus diberikan pada merinci bagaimana sistem akan beroperasi, merancang antarmuka yang akan digunakan, dan menggambarkan pendekatan sistem dalam menyelesaikan tantangan yang ada.

1. Masuk Sistem

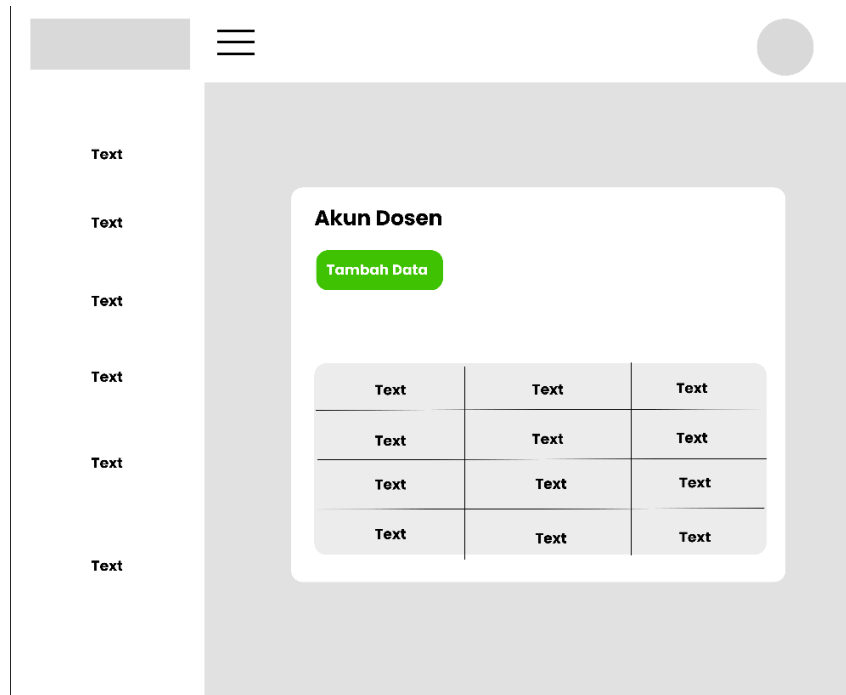
Pada halaman ini digunakan sebagai awal untuk masuk ke dalam sistem. Pada halaman ini, terdapat inputan email dan *password* untuk masuk sesuai dengan hak akses masing-masing pengguna. Desain antarmuka dan *activity diagram* halaman masuk sistem dapat dilihat pada gambar 8.

The image shows a login form titled "Masuk Akun" (Login Account). It is centered on a light gray background. The form itself is a white rounded rectangle. At the top left of the form is a gray circular profile icon. At the top right is a "Kembali" (Back) link. Below the title, there are two input fields: "Email" and "Kata Sandi" (Password). To the right of the password field is a "Lupa Kata Sandi?" (Forgot Password?) link. Below the input fields are two buttons: a solid blue "Masuk" (Login) button and a white "Daftar Akun" (Register Account) button with a blue border.

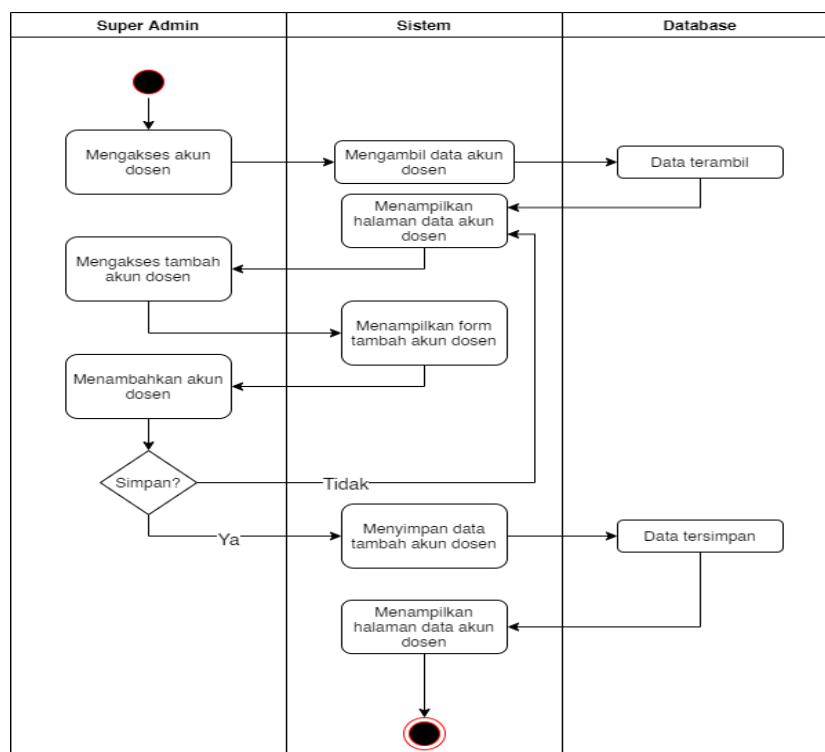
Gambar 4. Desain Antarmuka Masuk Sistem.

2. Manajemen Akun Dosen

Pada halaman akun dosen, super admin dapat menambah, mengedit, dan menghapus data akun dosen. Halaman ini terhubung dengan tabel dosen dalam *database* sistem informasi Chemistry Program Data Center yang terdapat pada lampiran 1. Desain antarmuka dan diagram aktivitas halaman manajemen akun dosen dapat dilihat pada Gambar 5, Gambar 6, Gambar 7, Gambar 8, Gambar 9, Gambar 10, dan Gambar 11.



Gambar 5. Desain Antarmuka Manajemen Akun Dosen.



Gambar 6. Activity Diagram Tambah Akun Dosen.

Tambah Akun Dosen

Email

Kata Sandi

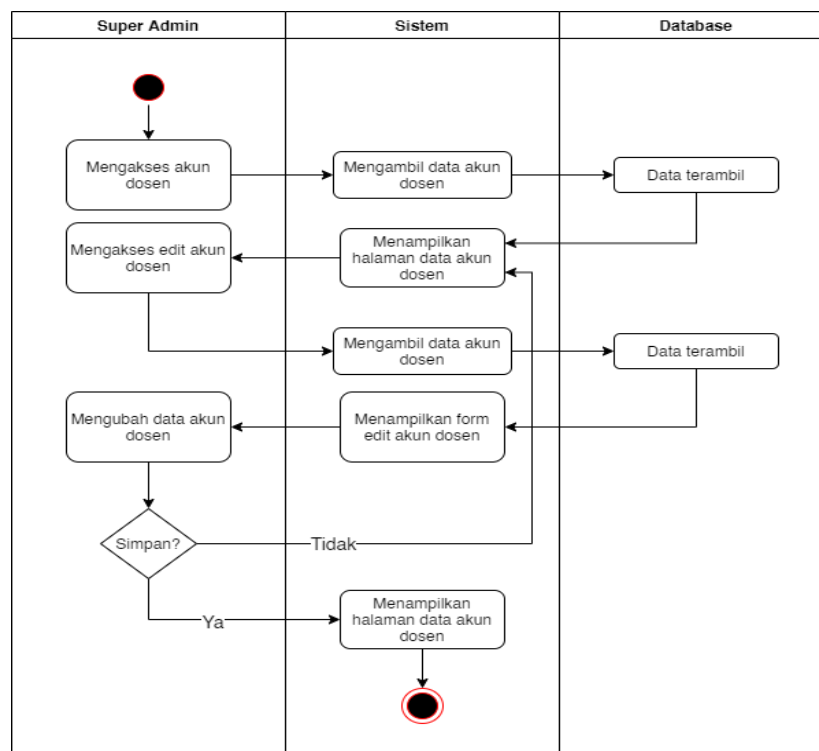
Nama

Role

Kirim

Batal

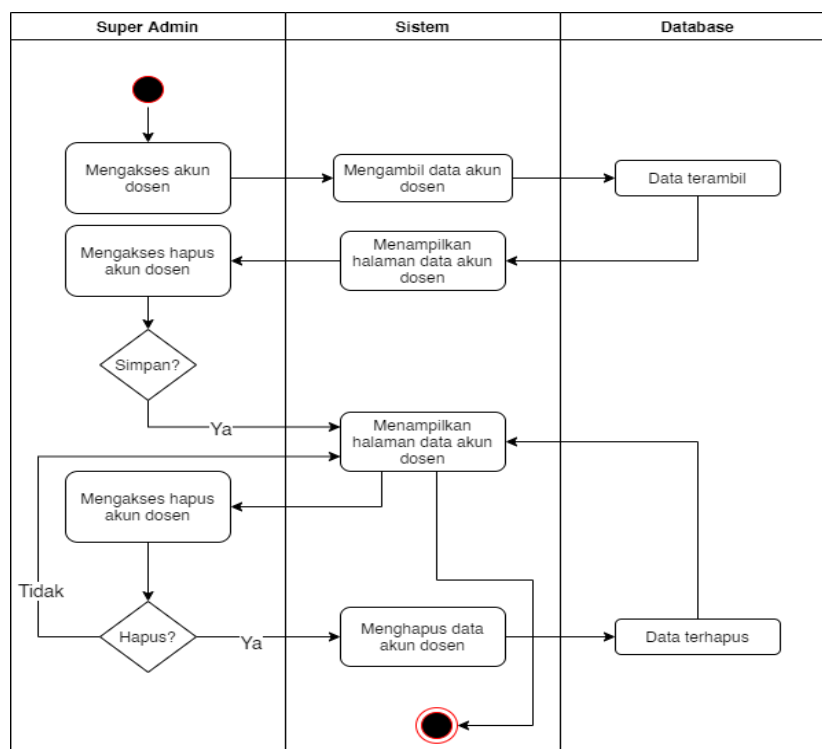
Gambar 7. Desain Antarmuka Tambah Akun Dosen.



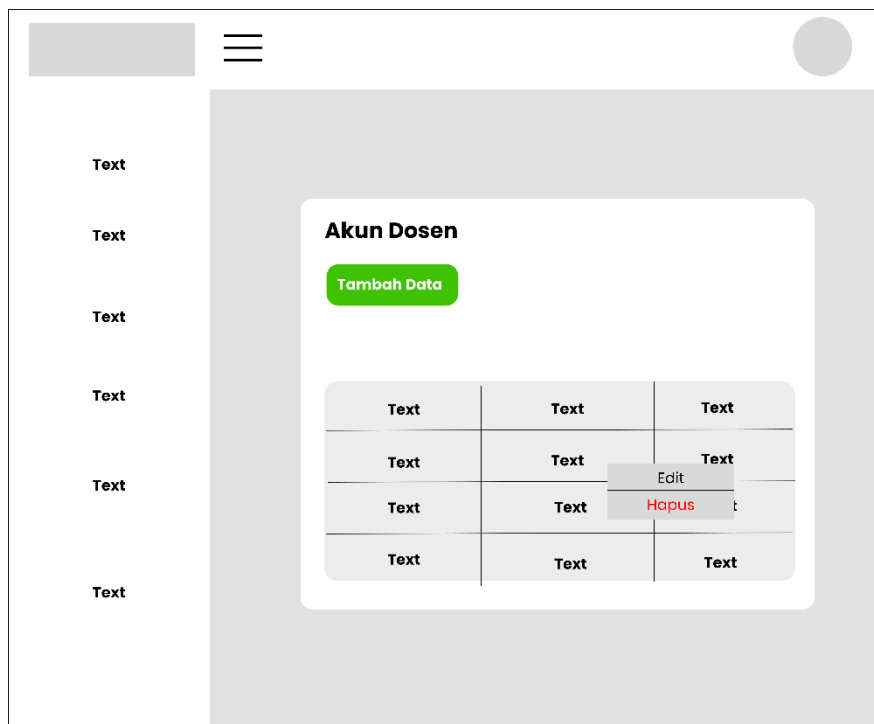
Gambar 8. *Activity Diagram* Edit Akun Dosen.

The image shows a web application interface for editing a lecturer's account. On the left is a sidebar with five 'Text' labels. The main content area features a form titled 'Edit Akun Dosen'. The form contains four input fields: 'Email', 'Kata Sandi' (Password), 'Nama' (Name), and 'Role'. Below these fields are two buttons: 'Kirim' (Send) in blue and 'Batal' (Cancel) in grey.

Gambar 9. Desain Antarmuka Edit Akun Dosen.



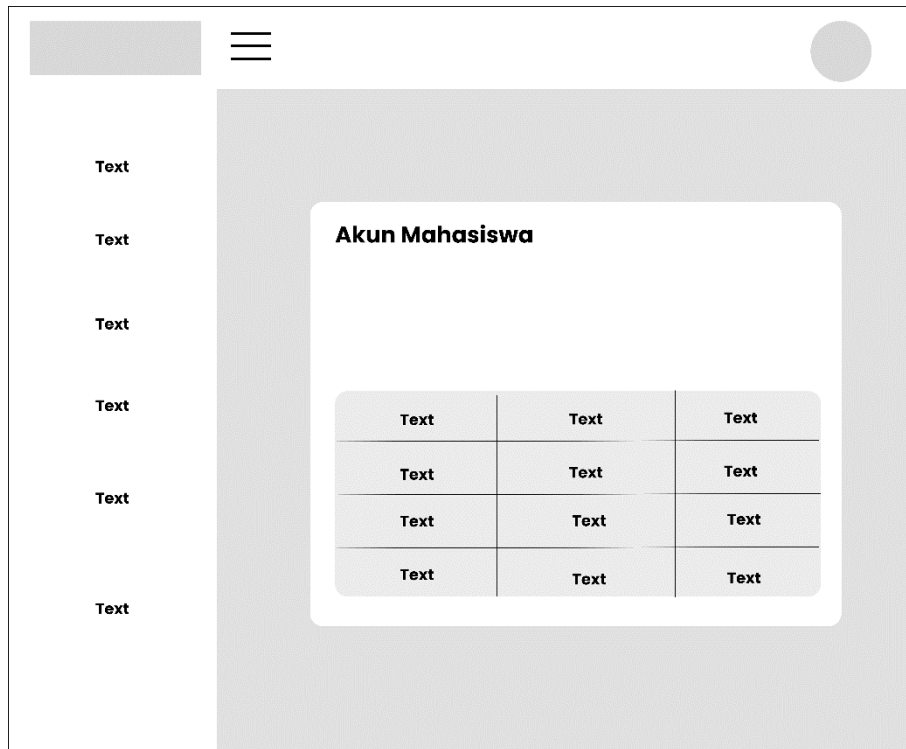
Gambar 10. Activity Diagram Hapus Akun Dosen.



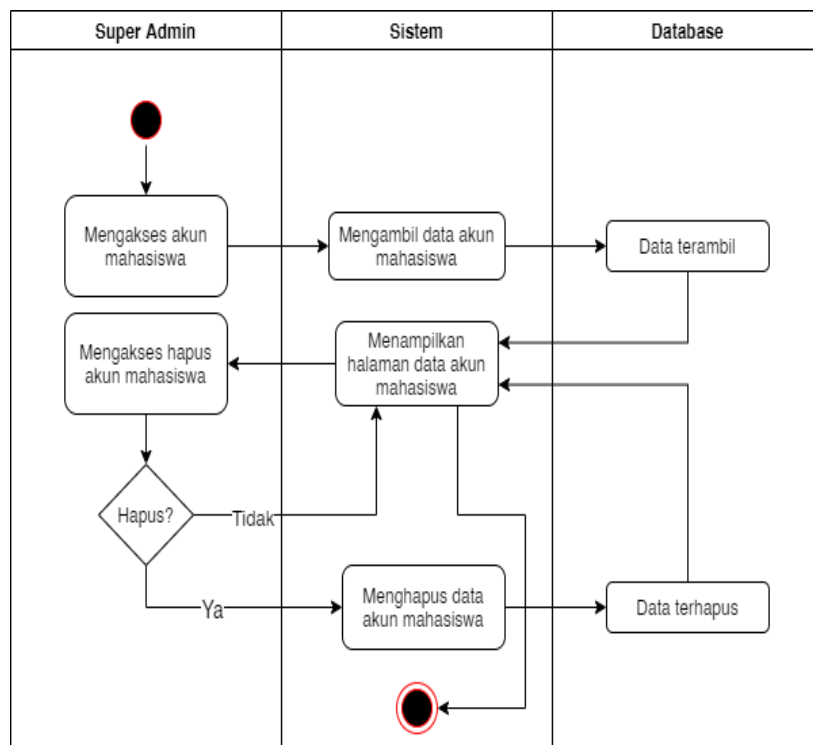
Gambar 11. Desain Antarmuka Hapus Akun Dosen

3. Manajemen Akun Mahasiswa

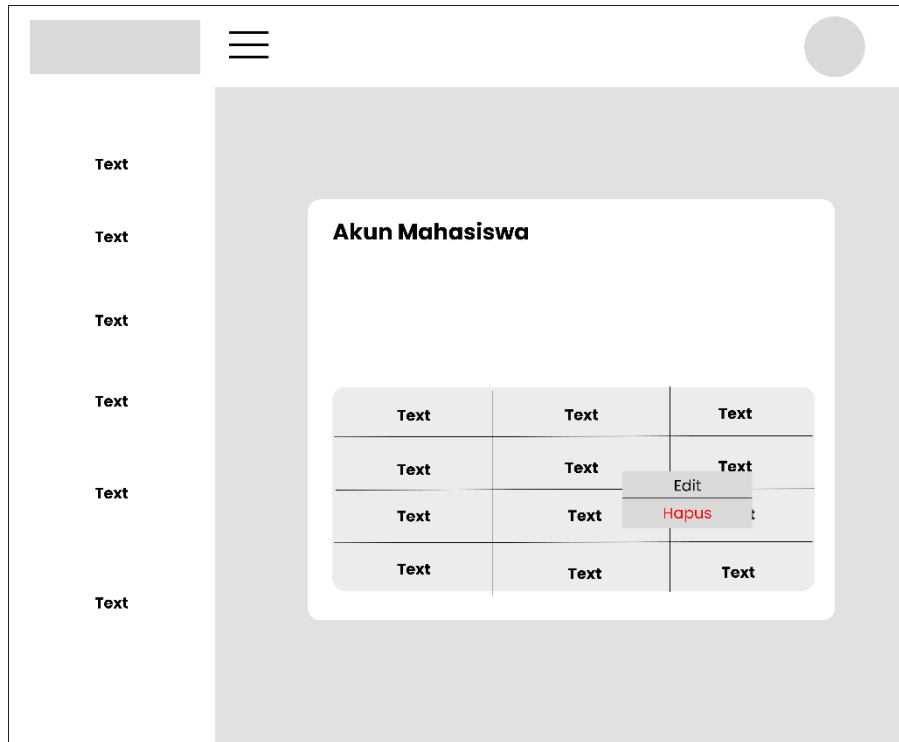
Pada halaman akun Mahasiswa, super admin dapat mengedit, dan menghapus data akun Mahasiswa. Halaman ini terhubung dengan tabel mahasiswa dalam *database* sistem informasi Chemistry Program Data Center yang terdapat pada lampiran 3. Desain antarmuka dan *activity diagram* halaman manajemen akun Mahasiswa dapat dilihat pada Gambar 12, Gambar 13, Gambar 14, Gambar 15 dan Gambar 16.



Gambar 12. Desain Antarmuka Manajemen Akun Mahasiswa.



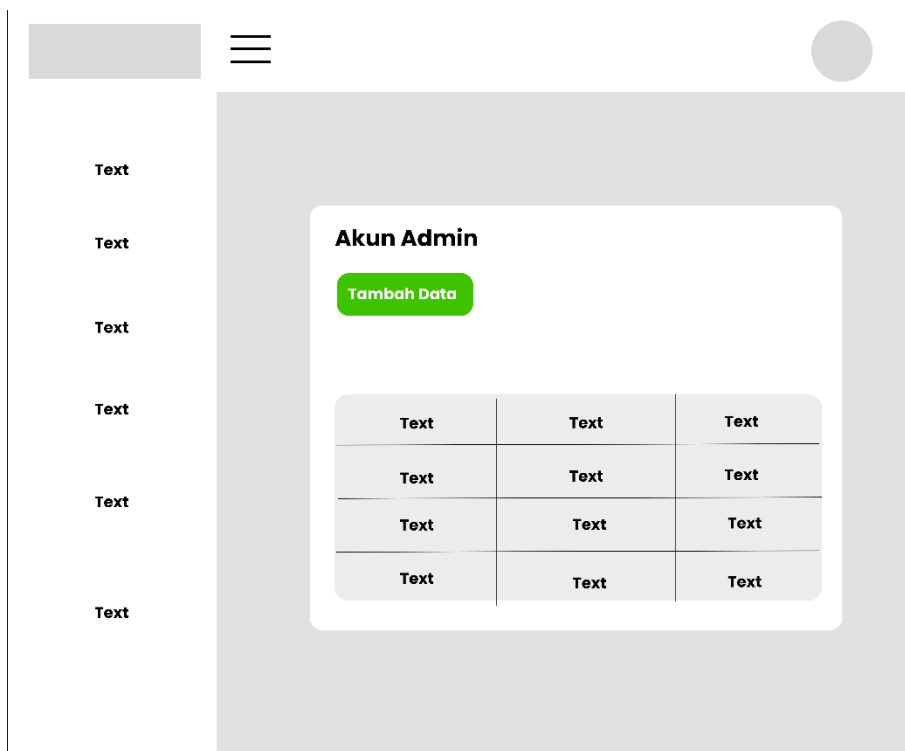
Gambar 13. Activity Diagram Edit Akun Mahasiswa.



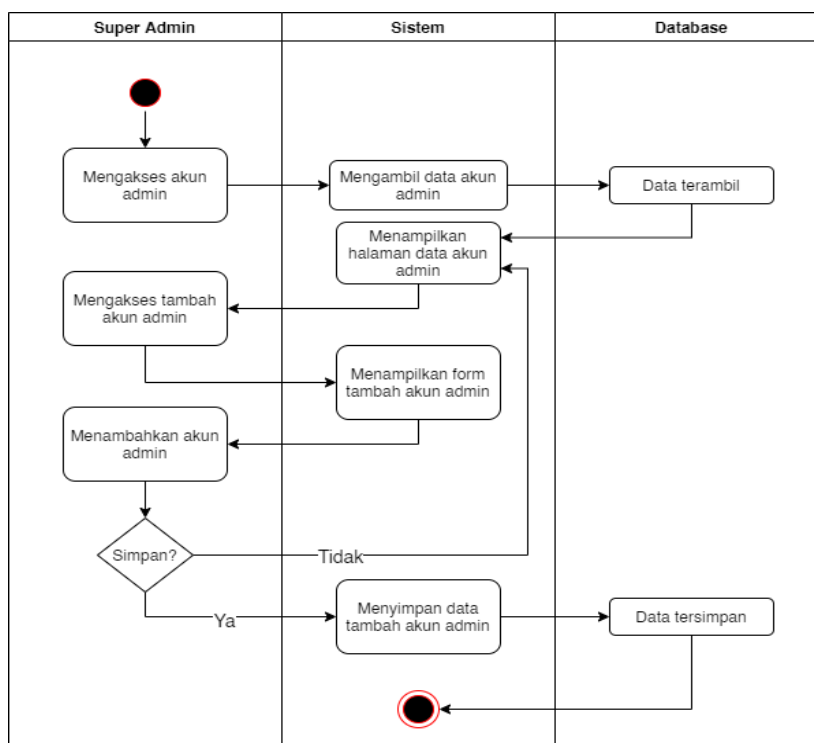
Gambar 16. Desain Antarmuka Hapus Akun Mahasiswa.

4. Manajemen Akun Admin

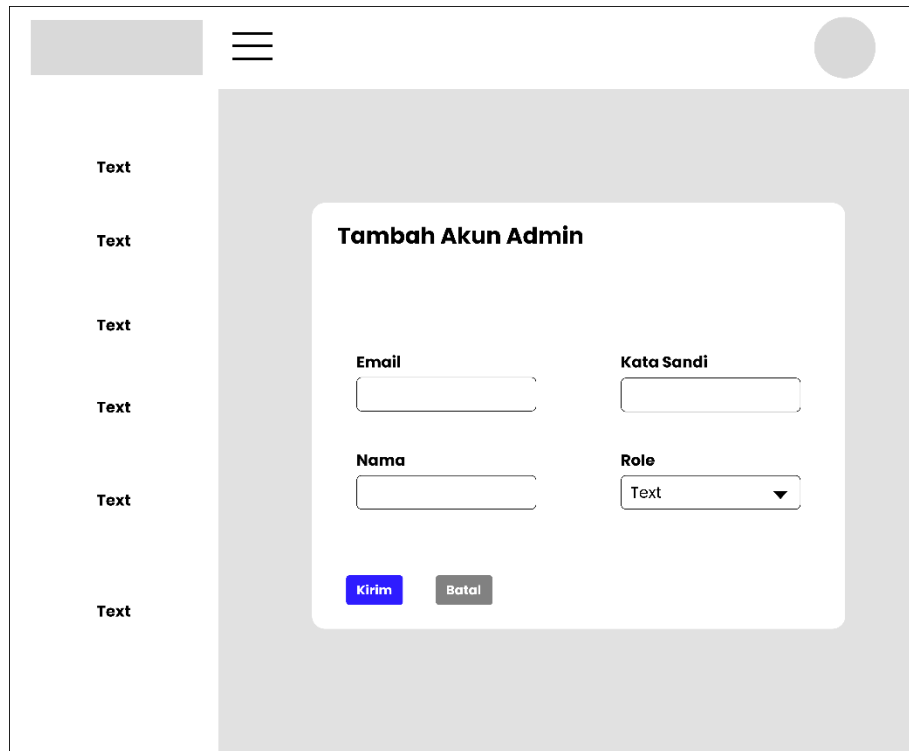
Pada halaman akun Admin, super admin dapat menambah, mengedit, dan menghapus data akun Admin. Halaman ini terhubung dengan tabel admin dalam *database* sistem informasi Chemistry Program Data Center yang terdapat pada lampiran 2. Desain antarmuka dan *activity diagram* halaman manajemen akun Admin dapat dilihat pada Gambar 17, Gambar 18, Gambar 19, Gambar 20, Gambar 21, Gambar 22, Gambar 23 dan Gambar 24.



Gambar 17. Desain Antarmuka Manajemen Akun Admin.

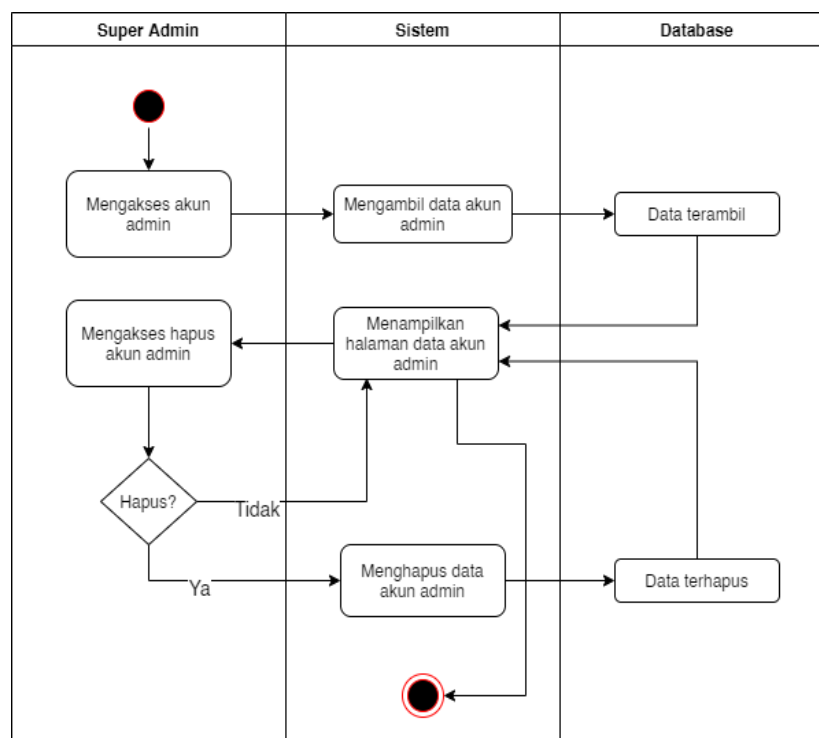


Gambar 18. Activity Diagram Tambah Akun Admin.

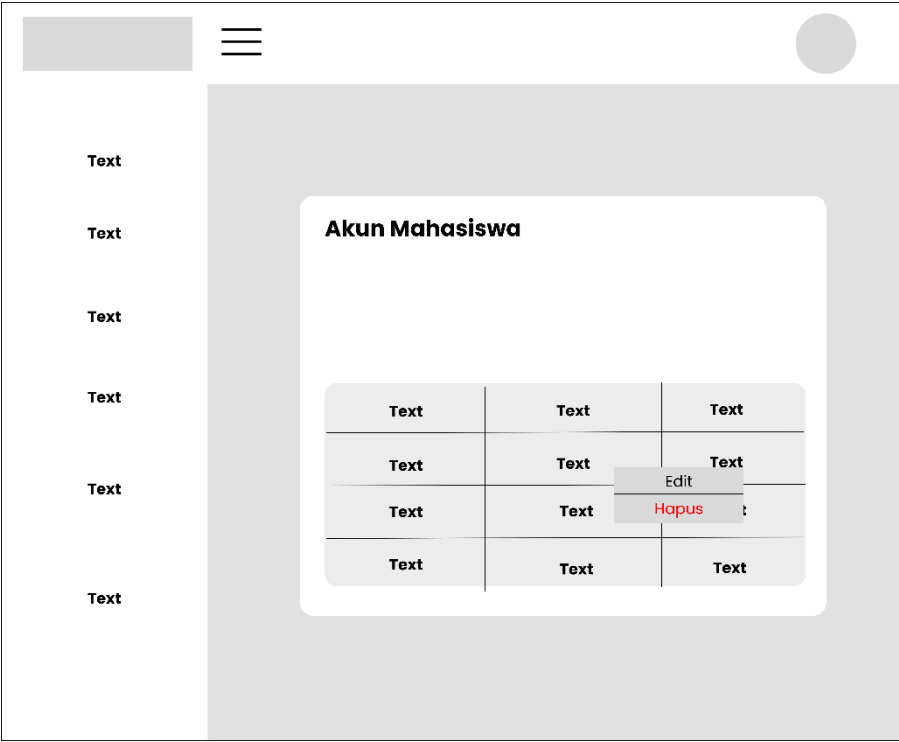


The image shows a web application interface for adding an admin account. On the left is a sidebar with five 'Text' labels. The main content area features a form titled 'Tambah Akun Admin'. The form contains four input fields: 'Email', 'Kata Sandi' (Password), 'Nama' (Name), and 'Role'. The 'Role' field is a dropdown menu currently showing 'Text'. At the bottom of the form are two buttons: 'Kirim' (Send) in blue and 'Batal' (Cancel) in grey.

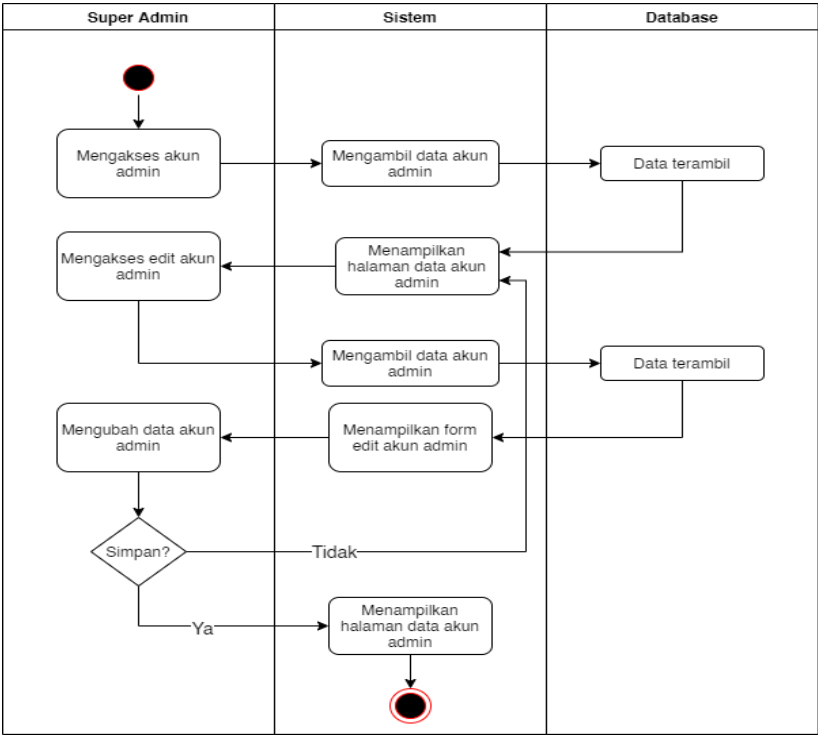
Gambar 19. Desain Antarmuka Tambah Akun Admin.



Gambar 20. Activity Diagram Hapus Akun Admin.



Gambar 21. Desain Antarmuka Hapus Akun Admin.

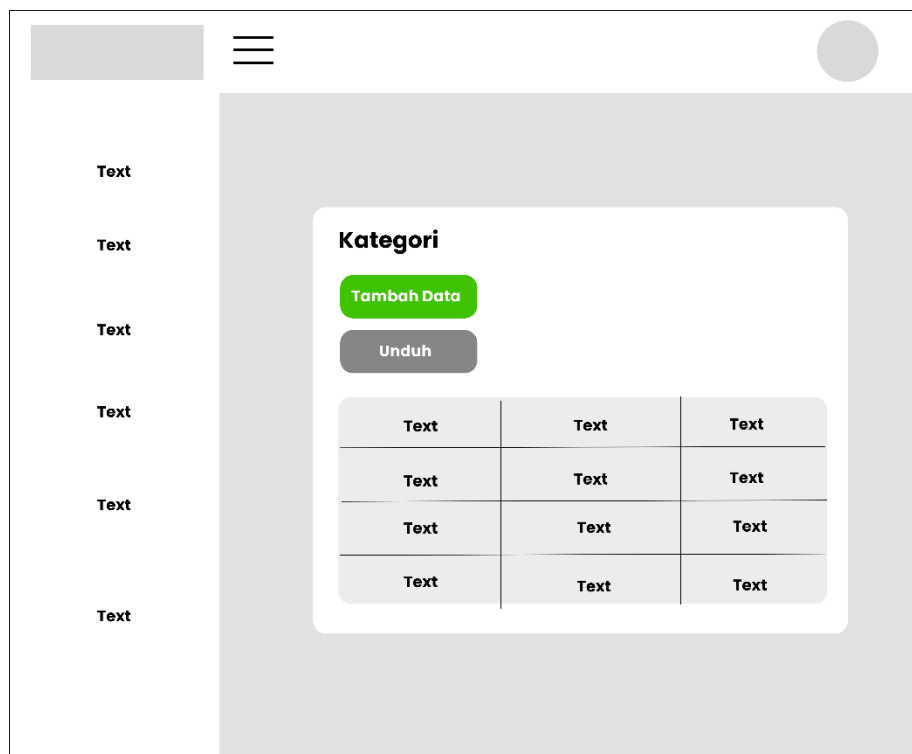


Gambar 22. Activity Diagram Edit Akun Admin.

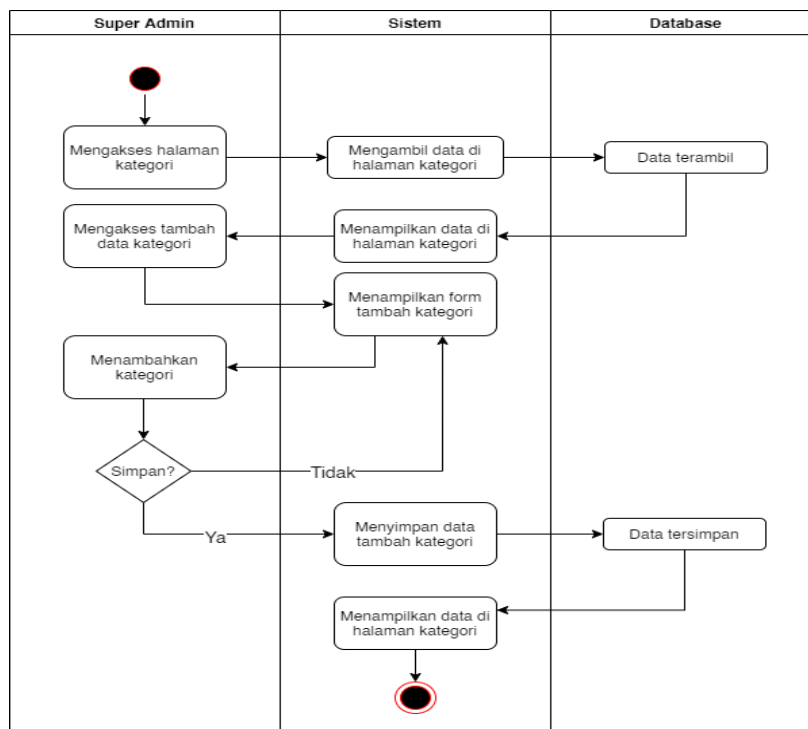
Gambar 23. Desain Antarmuka Edit Akun Admin.

5. Penetapan Kategori pada Inventaris Laboratorium

Pada bagian ini, super admin dapat melakukan menambah, mengubah dan menghapus kategori pada inventaris laboratorium. Halaman ini terhubung dengan tabel barang dalam *database* sistem informasi Chemistry Program Data Center yang terdapat pada lampiran 3. Desain antarmuka dan *activity diagram* penetapan kategori pada inventaris laboratorium dapat dilihat pada Gambar 24, Gambar 25, Gambar 26, Gambar 27, Gambar 28, Gambar 29, Gambar 30, Gambar 31 dan Gambar 32.



Gambar 24. Desain Antarmuka Halaman Kategori.



Gambar 25. Activity Diagram Tambah Kategori.

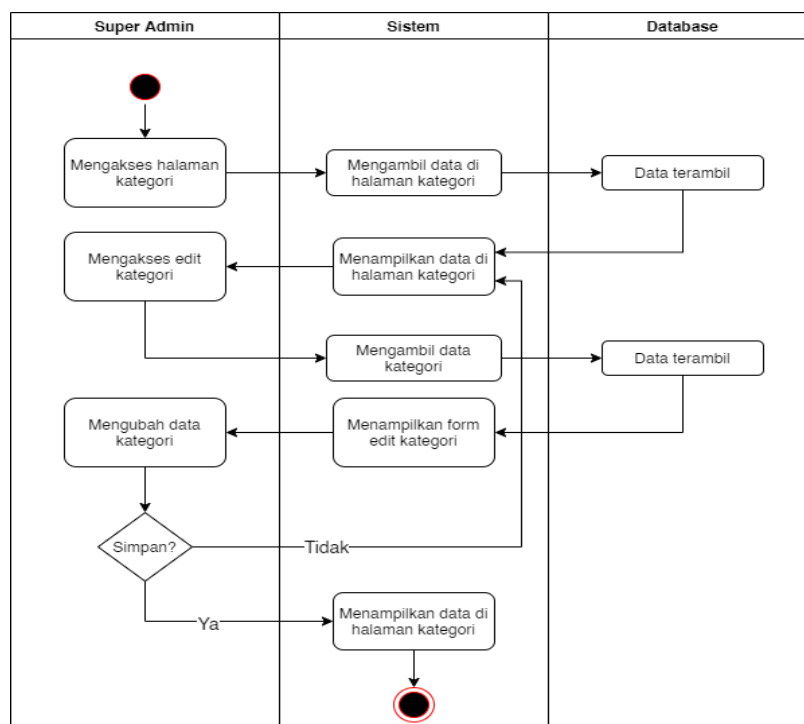
Tambah Kategori

Nama Kategori

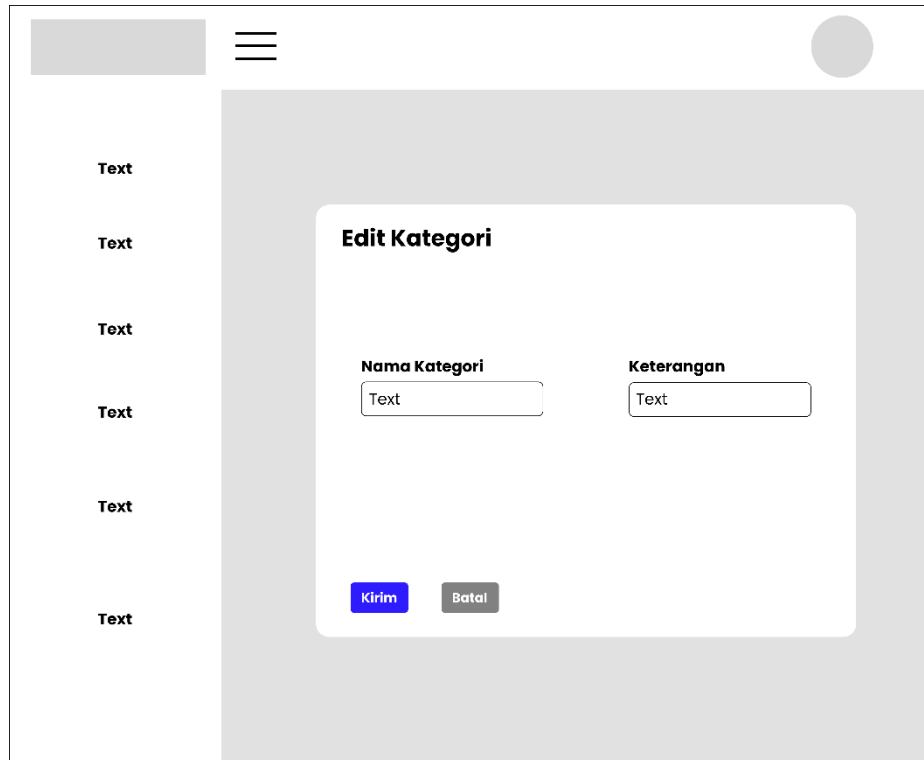
Keterangan

Kirim **Batal**

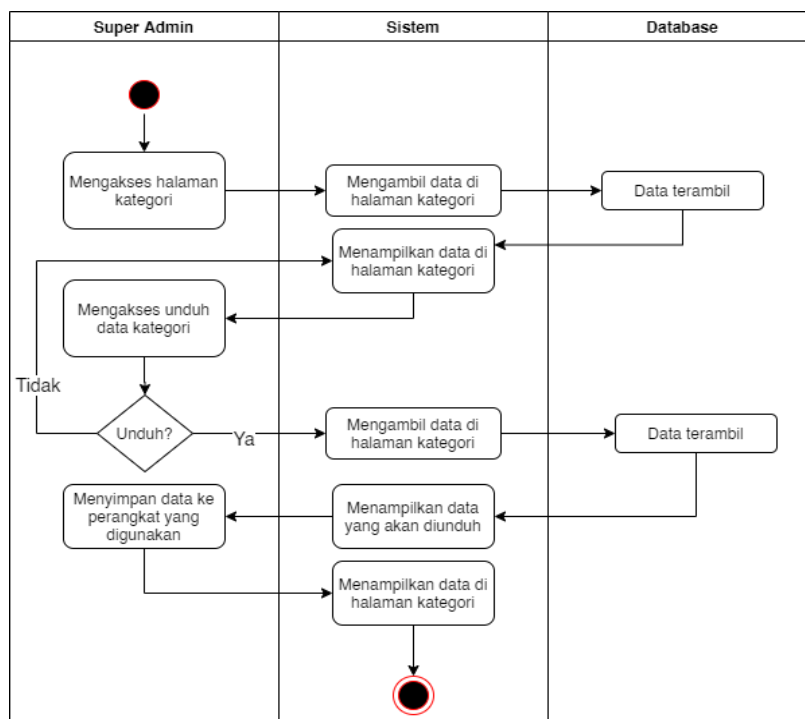
Gambar 26. Desain Antarmuka Tambah Kategori.



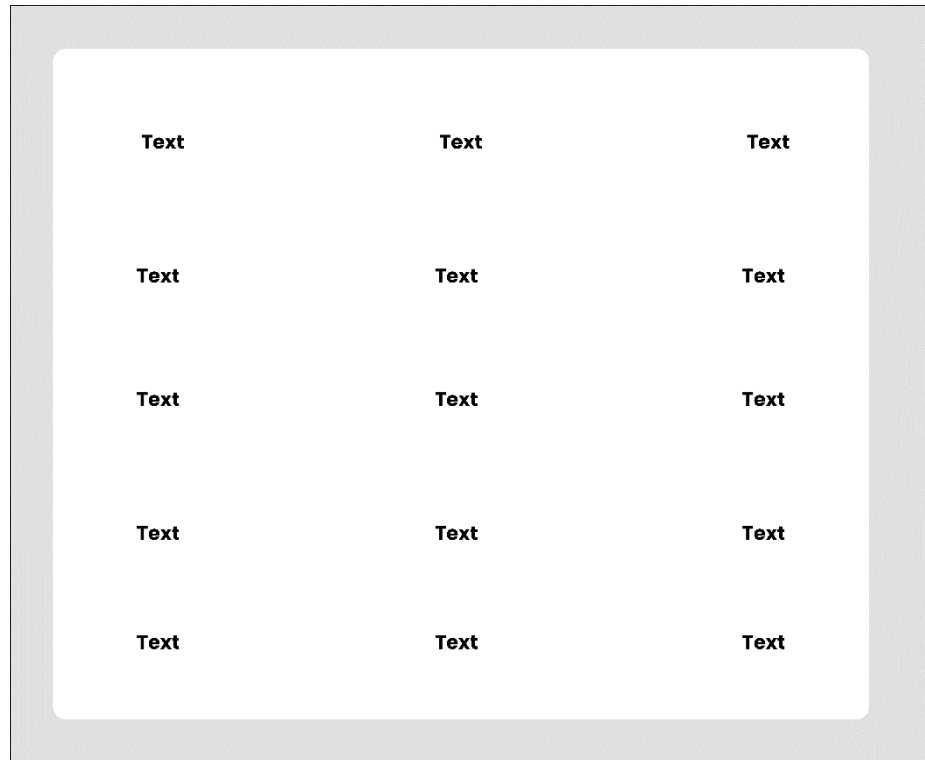
Gambar 27. Activity Diagram Edit Kategori.



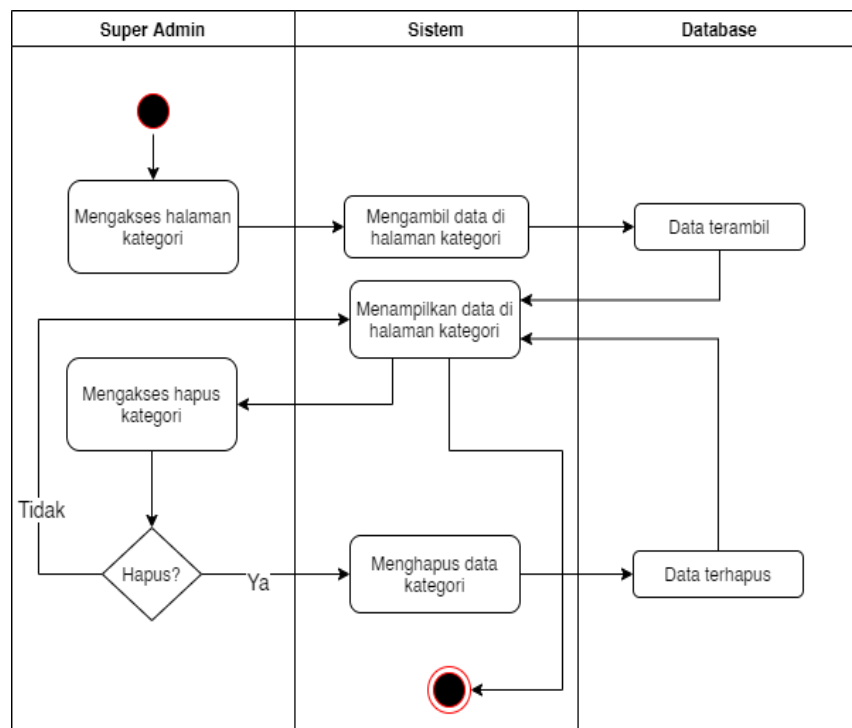
Gambar 28. Desain Antarmuka Hapus Kategori. Desain Antarmuka Edit Kategori.



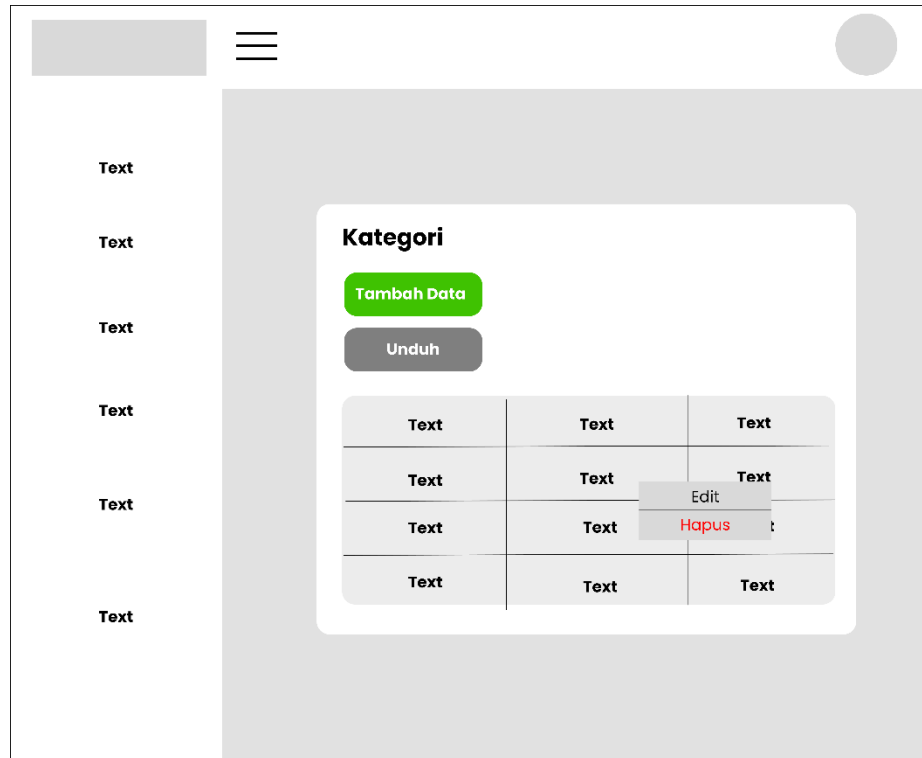
Gambar 29. Activity Diagram Unduh Kategori.



Gambar 30. Desain Antarmuka Unduh Kategori.



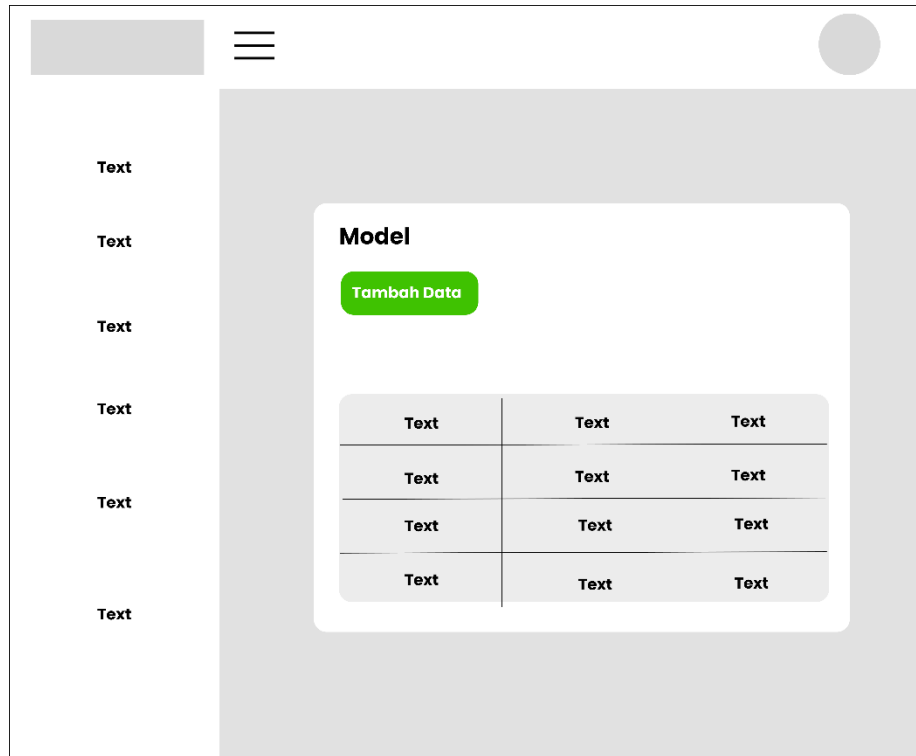
Gambar 31. Activity Diagram Hapus Kategori.



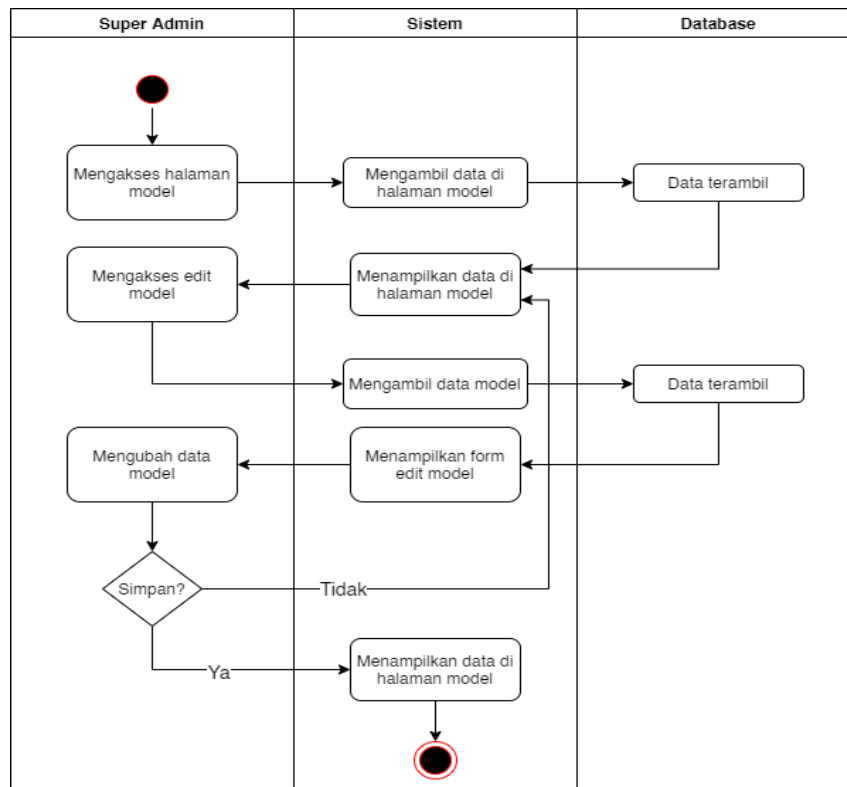
Gambar 32. Desain Antarmuka Hapus Kategori.

6. Penetapan Model pada Inventaris Laboratorium

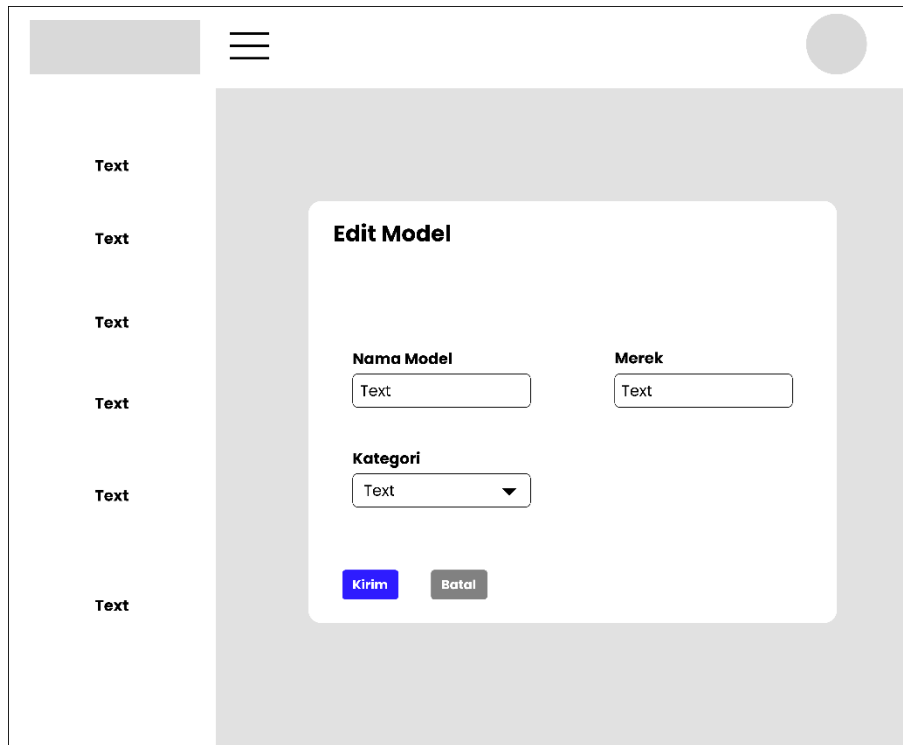
Pada bagian ini, super admin dapat melakukan menambah, mengubah dan menghapus model pada inventaris laboratorium. Halaman ini terhubung dengan tabel kategori dalam *database* sistem informasi Chemistry Program Data Center yang terdapat pada lampiran 3. Desain Antarmuka dan *activity diagram* penetapan model pada inventaris laboratorium dapat dilihat pada Gambar 33, Gambar 34, Gambar 35, Gambar 36, Gambar 37, Gambar 38 dan Gambar 39.



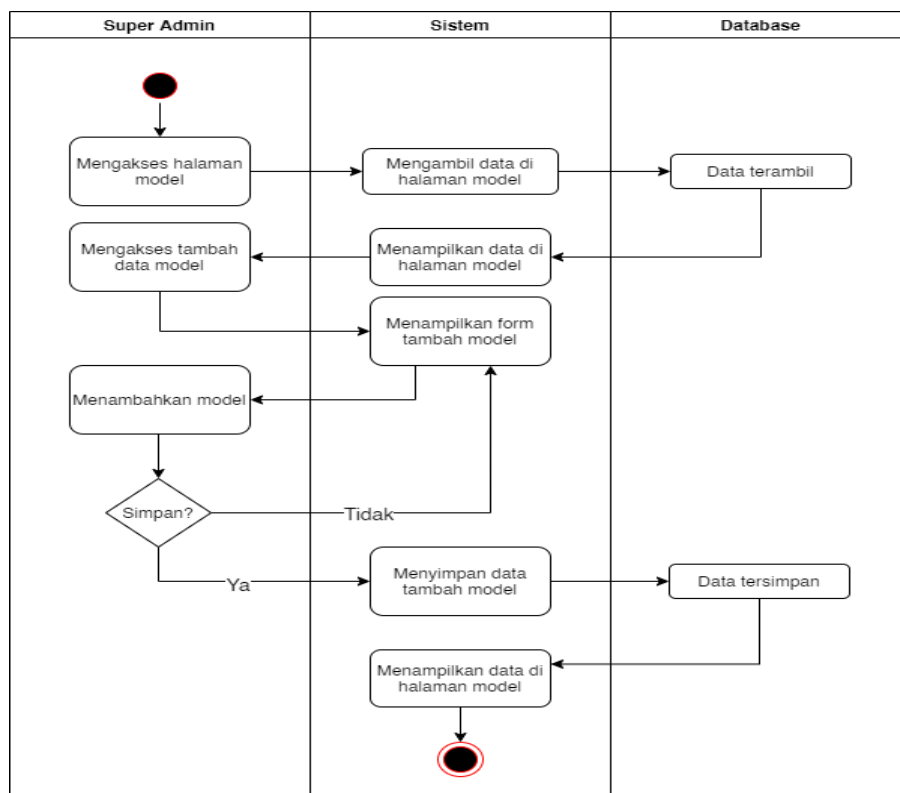
Gambar 33. Desain Antarmuka Halaman Model.



Gambar 34. Activity Diagram Edit Model.

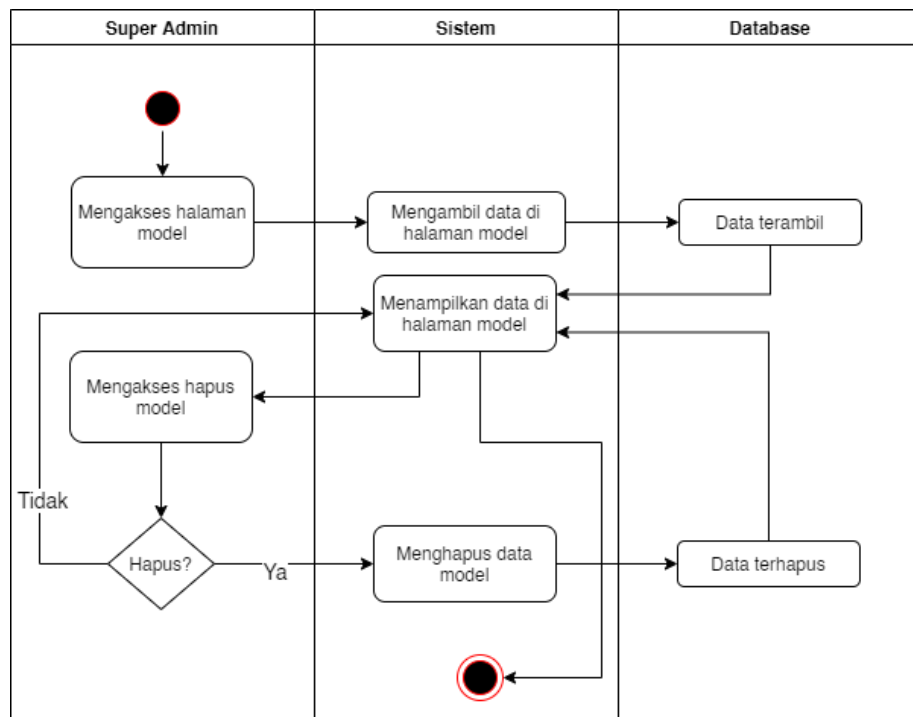


Gambar 35. Desain Antarmuka Edit Model.

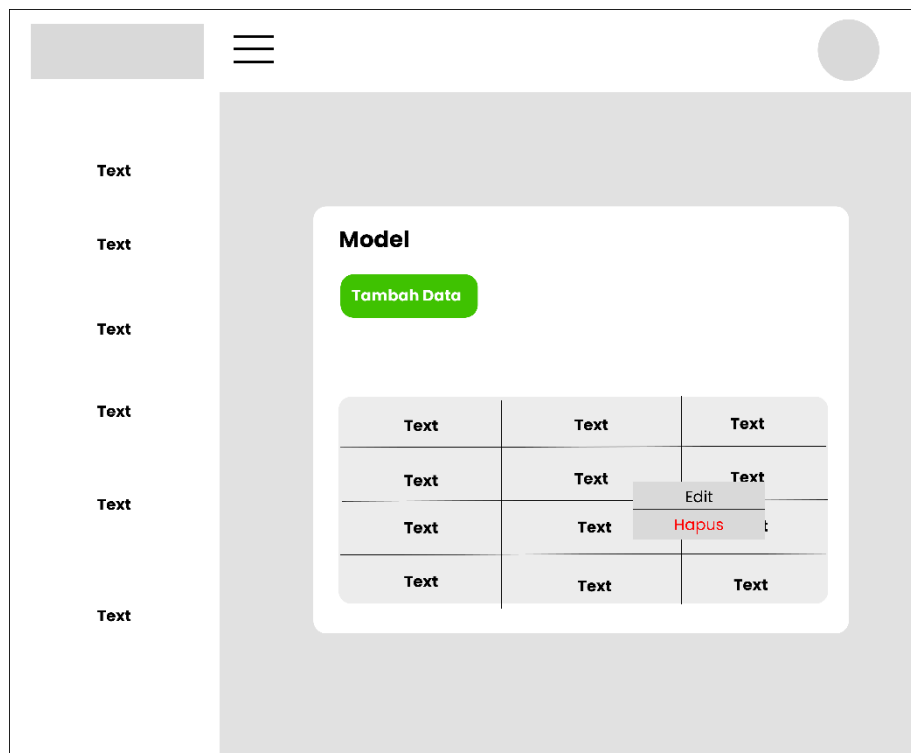


Gambar 36. Activity Diagram Tambah Model.

Gambar 37. Desain Antarmuka Tambah Model.



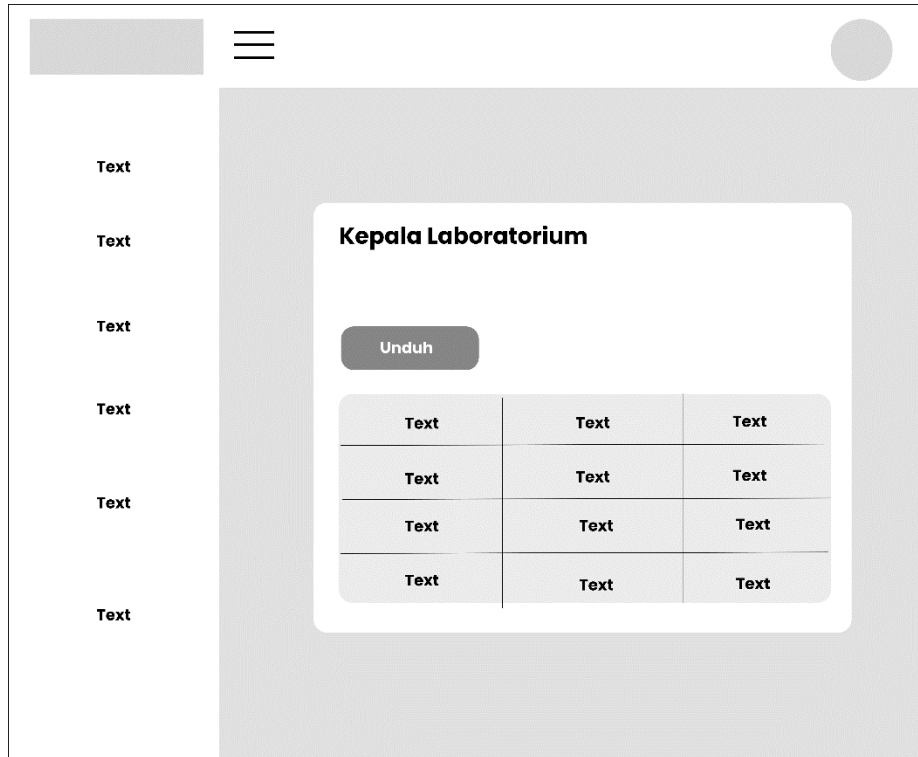
Gambar 38. Activity Diagram Hapus Model.



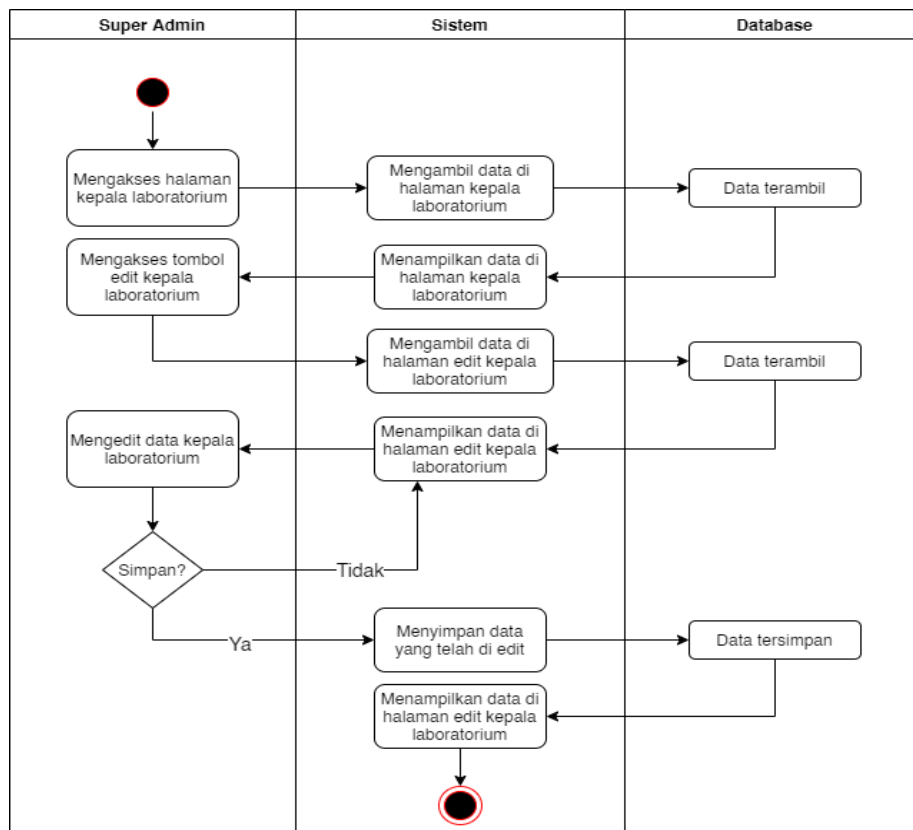
Gambar 39. Desain Antarmuka Hapus Model.

7. Manajemen Peran Akun Pada Kepala Laboratorium

Pada halaman ini, super admin dapat mengubah hak akses akun pada dosen untuk menjadi kepala laboratorium. Selain itu, super admin dapat mengunduh data yang ada di halaman ini. Halaman ini terhubung dengan tabel dosen dalam *database* sistem informasi Chemistry Program Data Center yang terdapat pada lampiran 1. Desain antarmuka dan *activity diagram* halaman manajemen peran pada akun kepala laboratorium dapat dilihat pada Gambar 40, Gambar 41, Gambar 42, Gambar 43 dan Gambar 44.



Gambar 40. Desain Antarmuka Manajemen Peran Akun Kepala Laboratorium.



Gambar 41. Activity Diagram Edit Kepala Laboratorium.

Data Kepala Laboratorium

Text Text Text Text

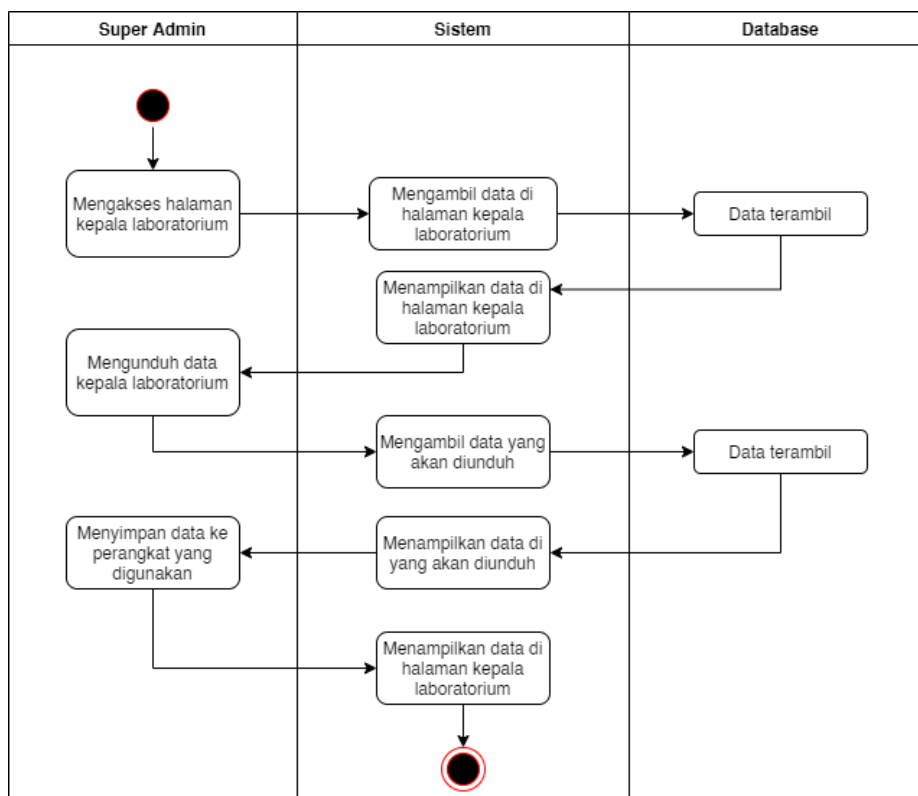
Text Text Text Text

Lokasi

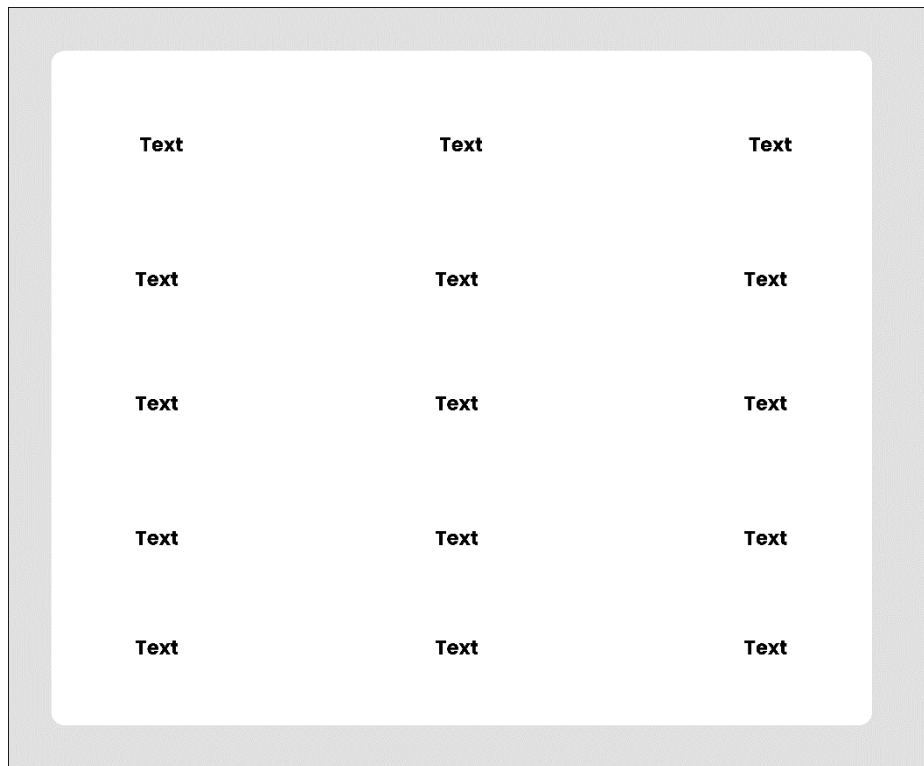
Text ▼

Kirim **Batal**

Gambar 43. Desain Antarmuka Edit Data Kepala Laboratorium.



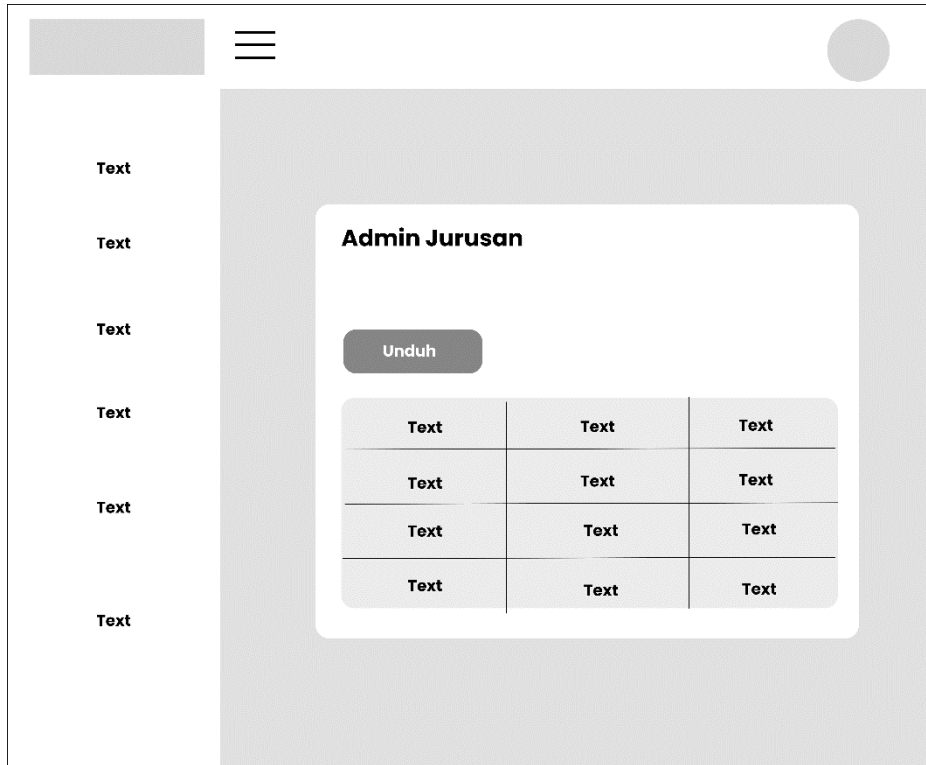
Gambar 42. Activity Diagram Unduh Data Kepala Laboratorium.



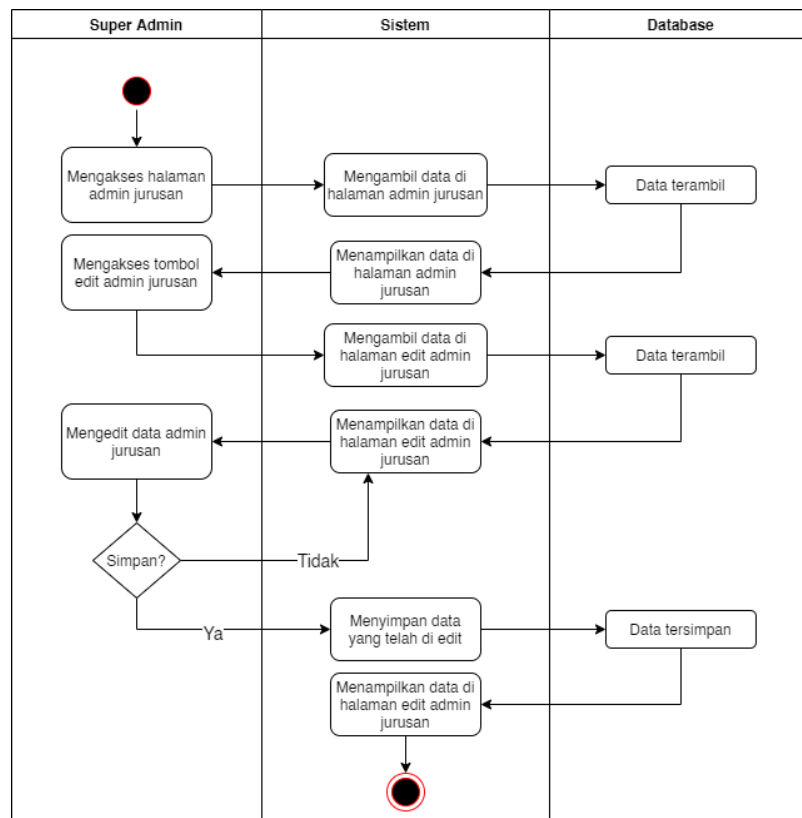
Gambar 44. Desain Antarmuka Unduh Data Kepala Laboratorium.

8. Manajemen Peran Akun Pada Admin Jurusan

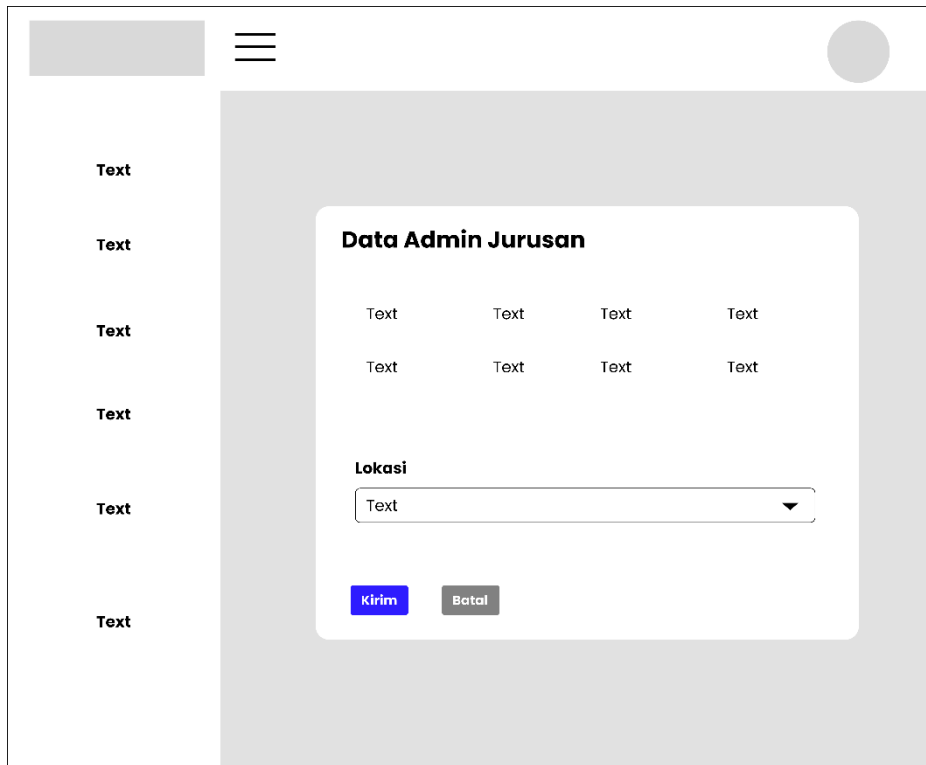
Pada halaman ini, super admin dapat mengubah hak akses akun pada staf untuk menjadi admin jurusan. Selain itu, super admin dapat mengunduh data yang ada di halaman ini. Halaman ini terhubung dengan tabel admin dalam *database* sistem informasi Chemistry Program Data Center yang terdapat pada lampiran 2. Desain antarmuka dan *activity diagram* halaman manajemen peran pada akun admin jurusan dapat dilihat pada Gambar 45, Gambar 46, Gambar 47, Gambar 48 dan Gambar 49.



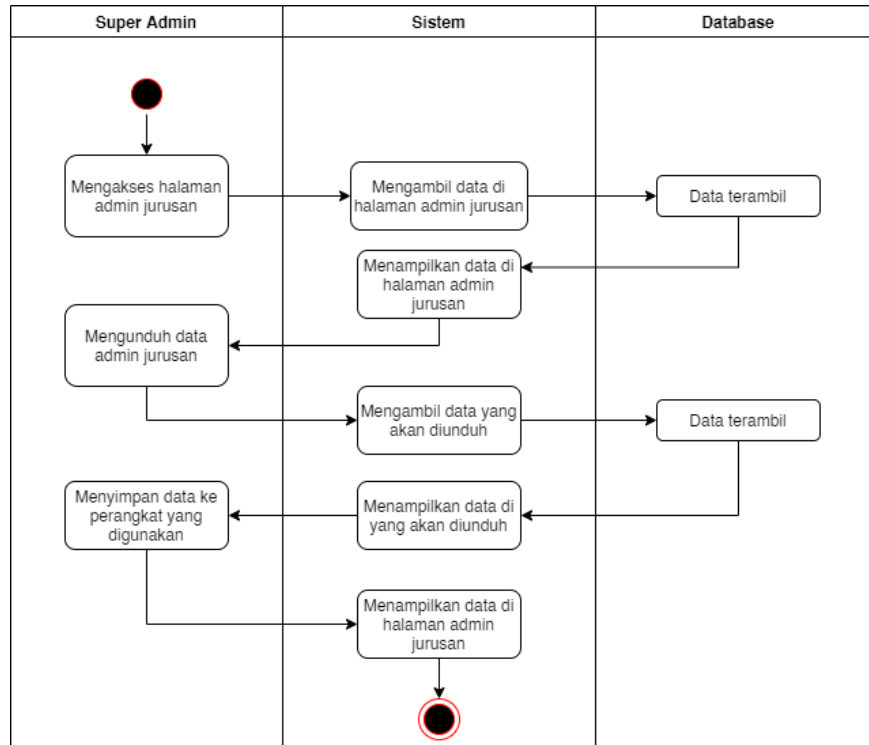
Gambar 45. Desain Antarmuka Manajemen Peran Akun Admin Jurusan.



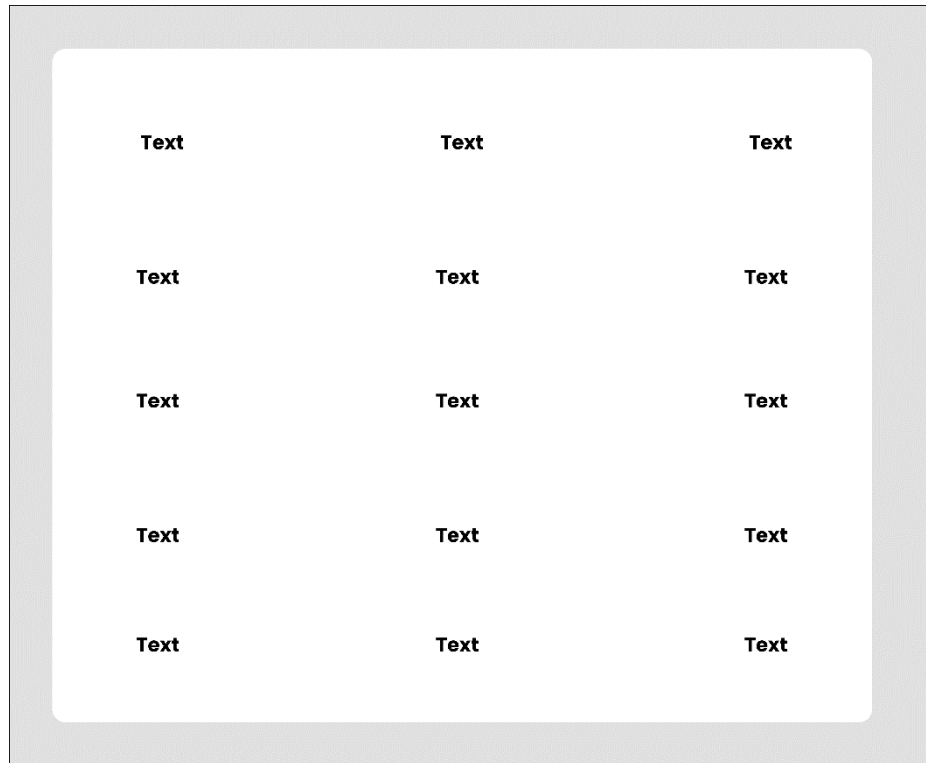
Gambar 46. Activity Diagram Edit Data Admin Jurusan.



Gambar 47. Desain Antarmuka Edit Data Admin Jurusan.



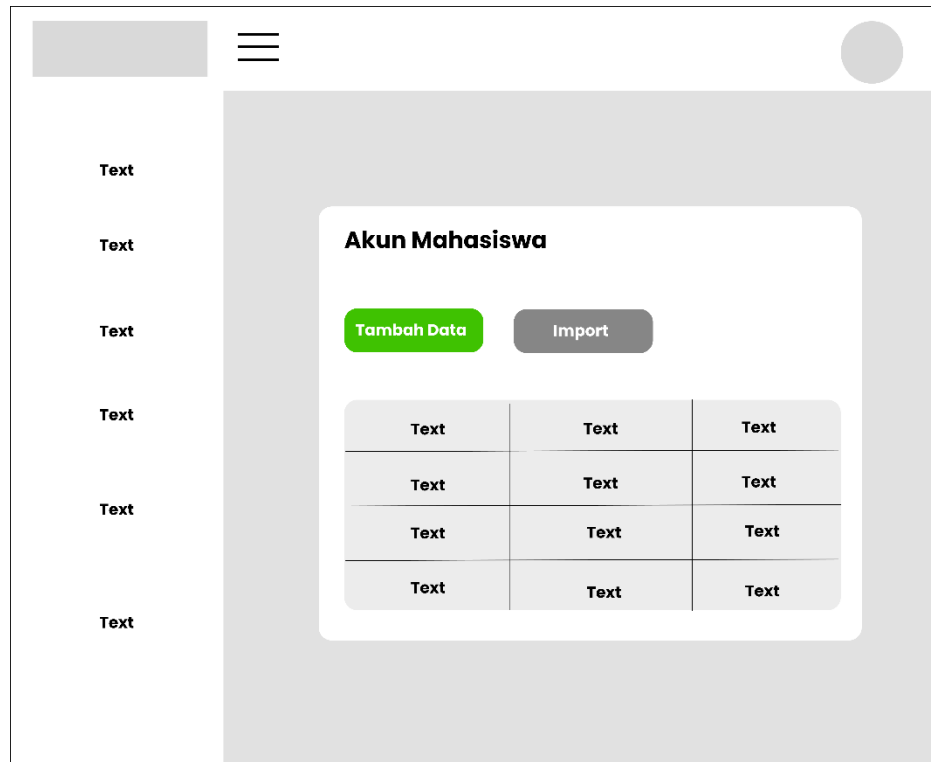
Gambar 48. Activity Diagram Unduh Data Admin Jurusan.



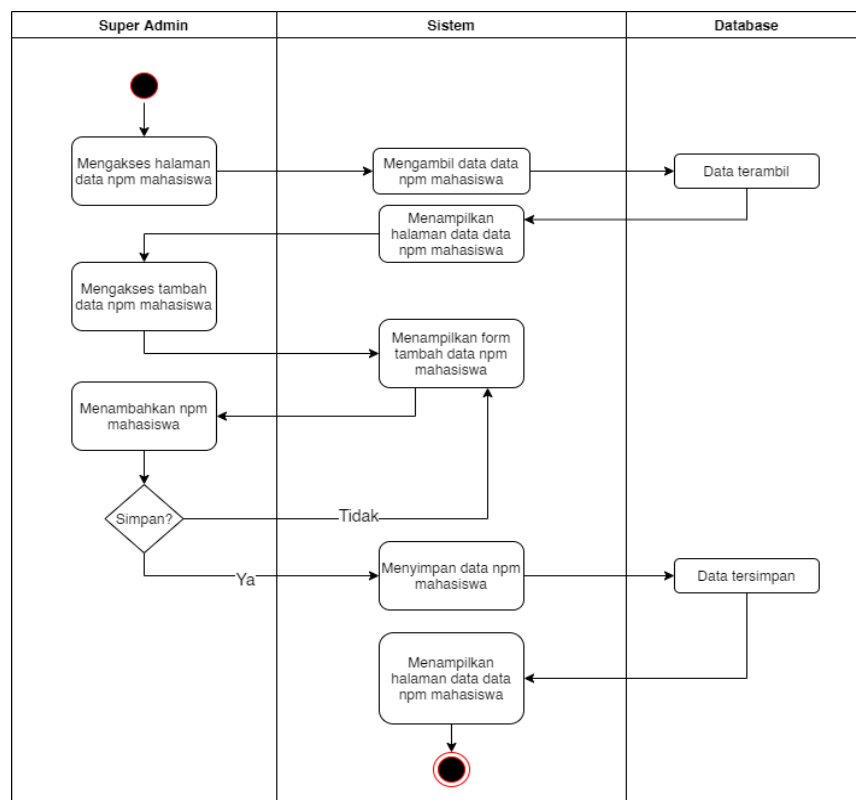
Gambar 49. Desain Antarmuka Unduh Data Admin Jurusan.

9. Data NPM Mahasiswa

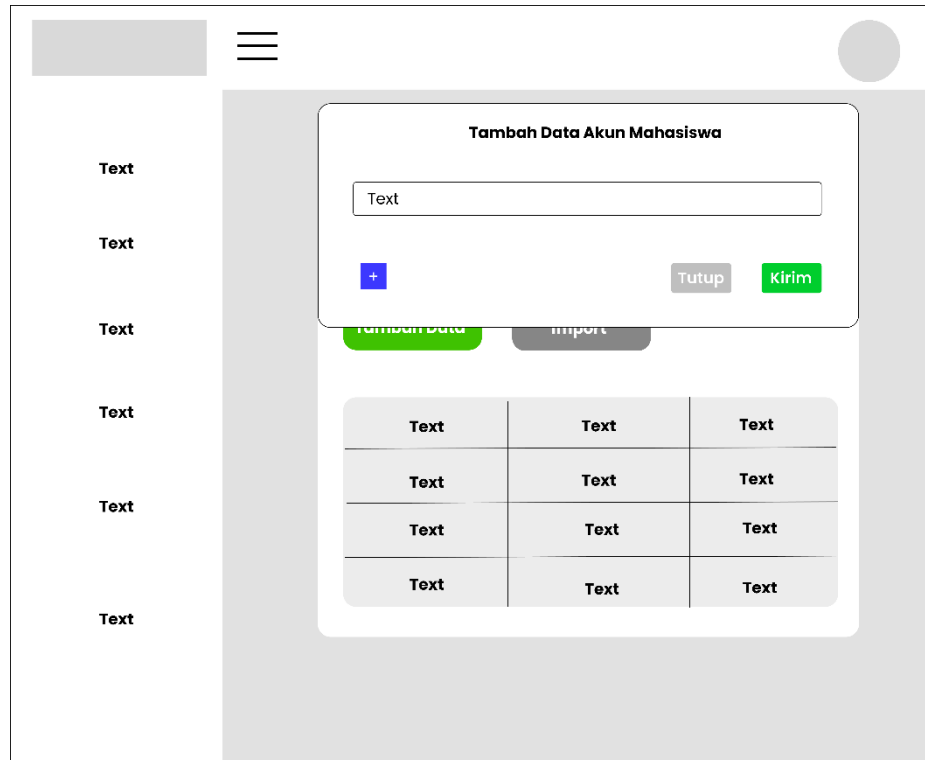
Pada halaman ini, super admin dapat menambah data npm mahasiswa dengan cara klik tombol tambah, kemudian super admin dapat menggunggah data excel dari npm mahasiswa jika jumlahnya banyak dengan klik tombol *import*. Selain itu, super admin juga dapat mengubah dan menghapus data npm mahasiswa. Halaman ini terhubung dengan tabel mahasiswa dalam *database* sistem informasi Chemistry Program Data Center yang terdapat pada lampiran 3. Desain antarmuka dan activity diagram halaman data npm mahasiswa dapat dilihat pada Gambar 50, Gambar 51, Gambar 52, Gambar 53, Gambar 54, Gambar 55, Gambar 56, Gambar 57 dan Gambar 58.



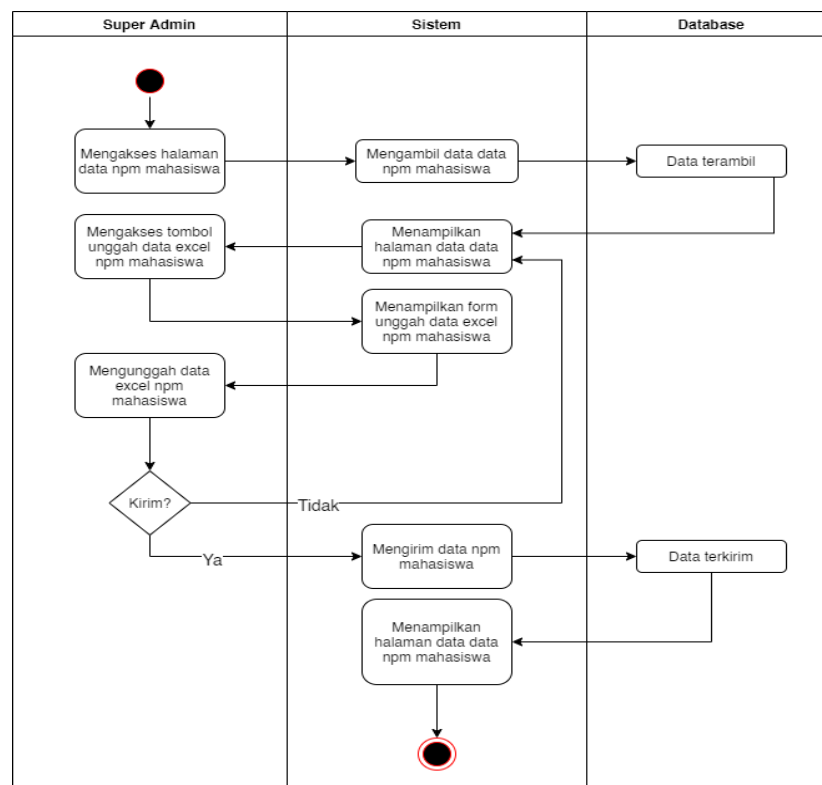
Gambar 50. Desain Antarmuka NPM Mahasiswa.



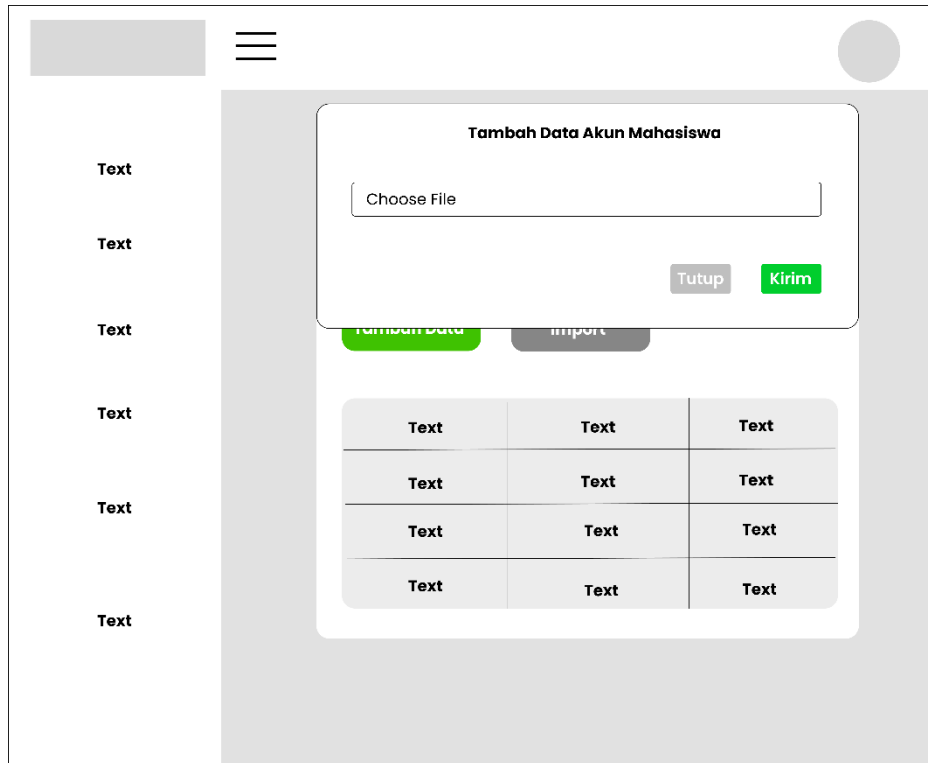
Gambar 51. Activity Diagram Tambah Data NPM Mahasiswa.



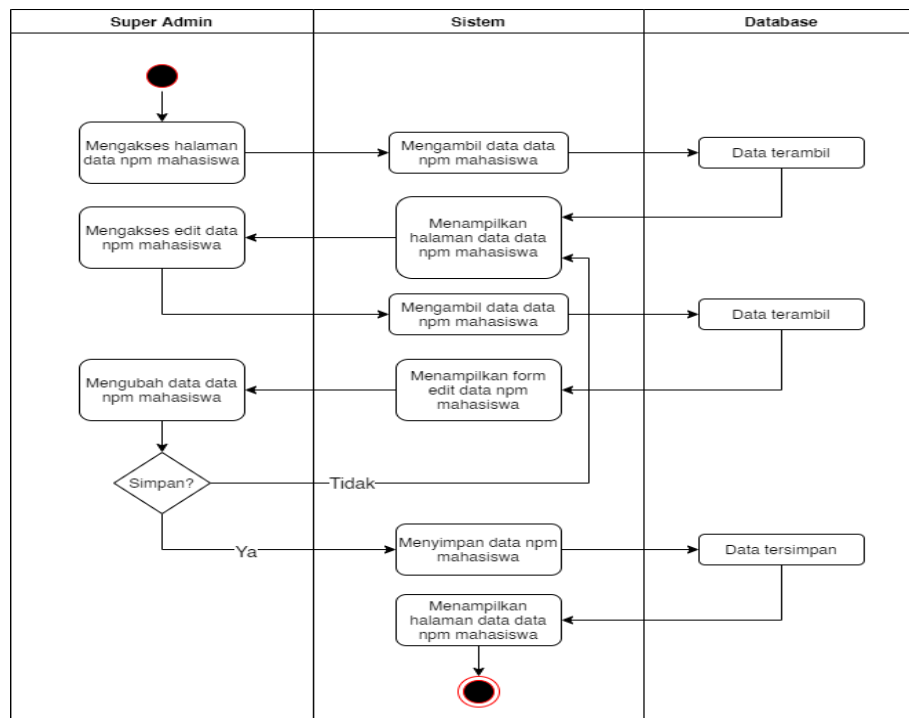
Gambar 52. Desain Antarmuka Tambah Data NPM Mahasiswa.



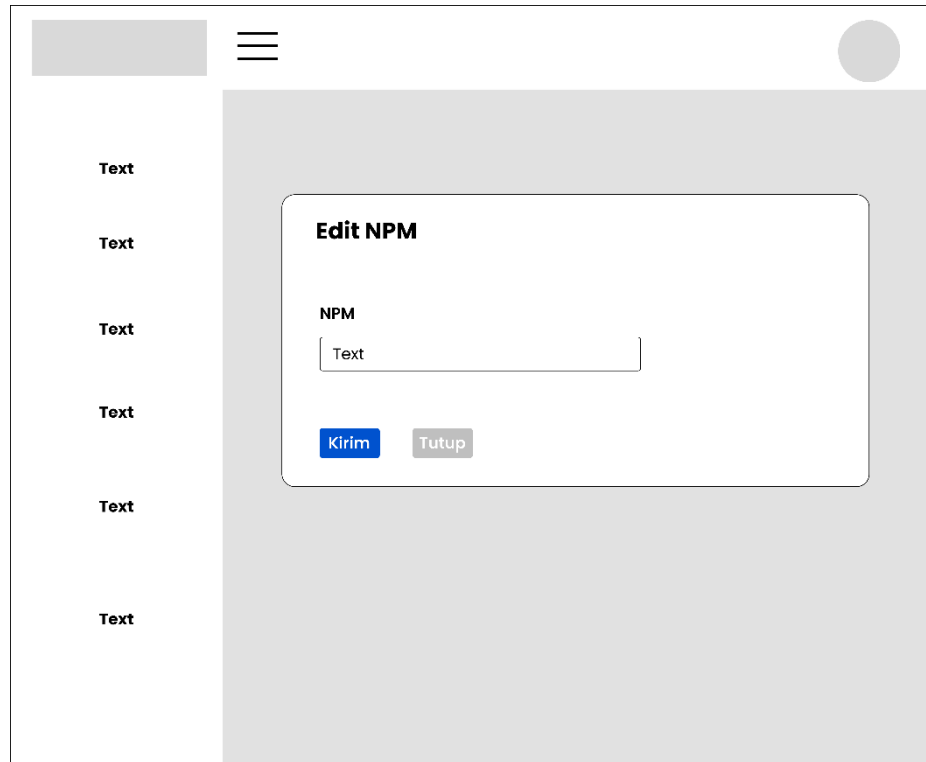
Gambar 53. Activity Diagram Import Data NPM Mahasiswa.



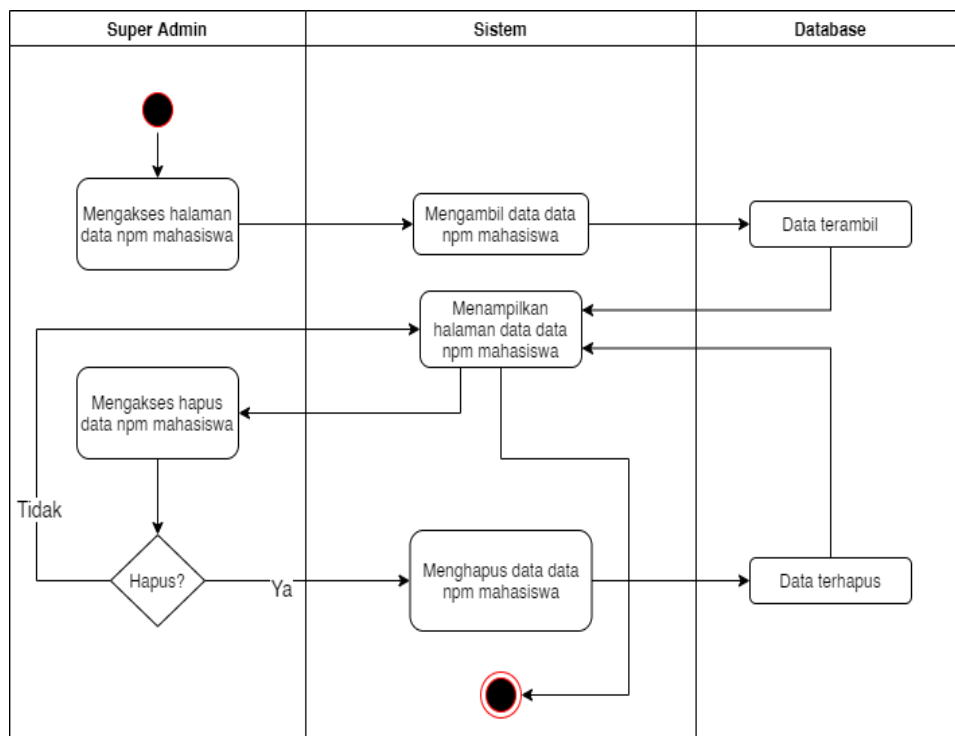
Gambar 54. Desain Antarmuka *Import* Data NPM Mahasiswa.



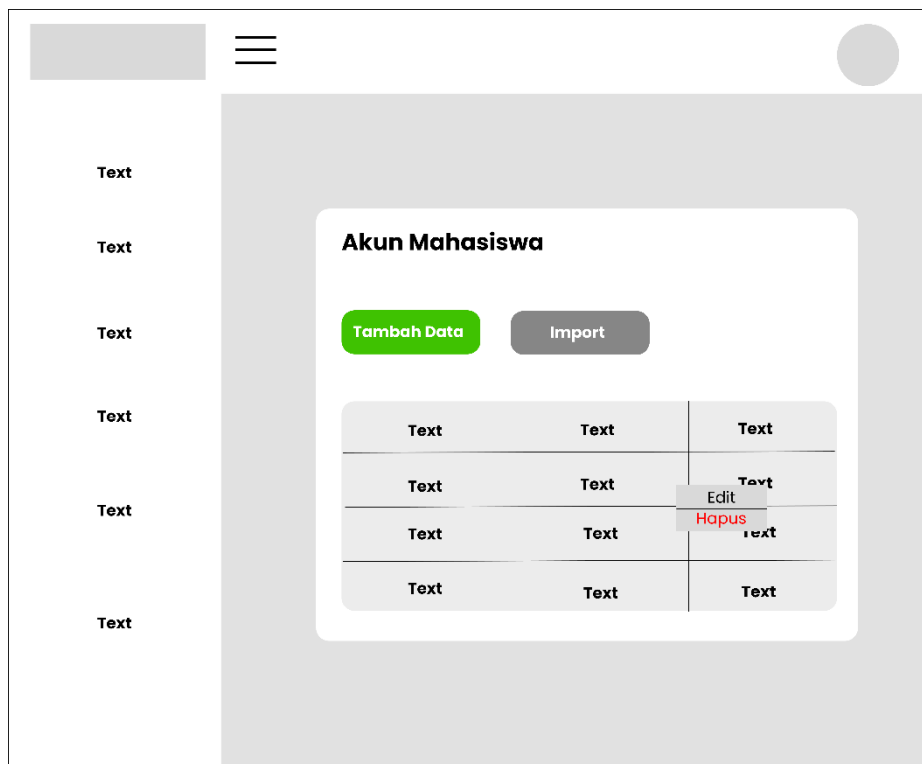
Gambar 55. *Activity Diagram* Edit NPM Mahasiswa.



Gambar 56. Desain Antarmuka Edit Data NPM Mahasiswa.



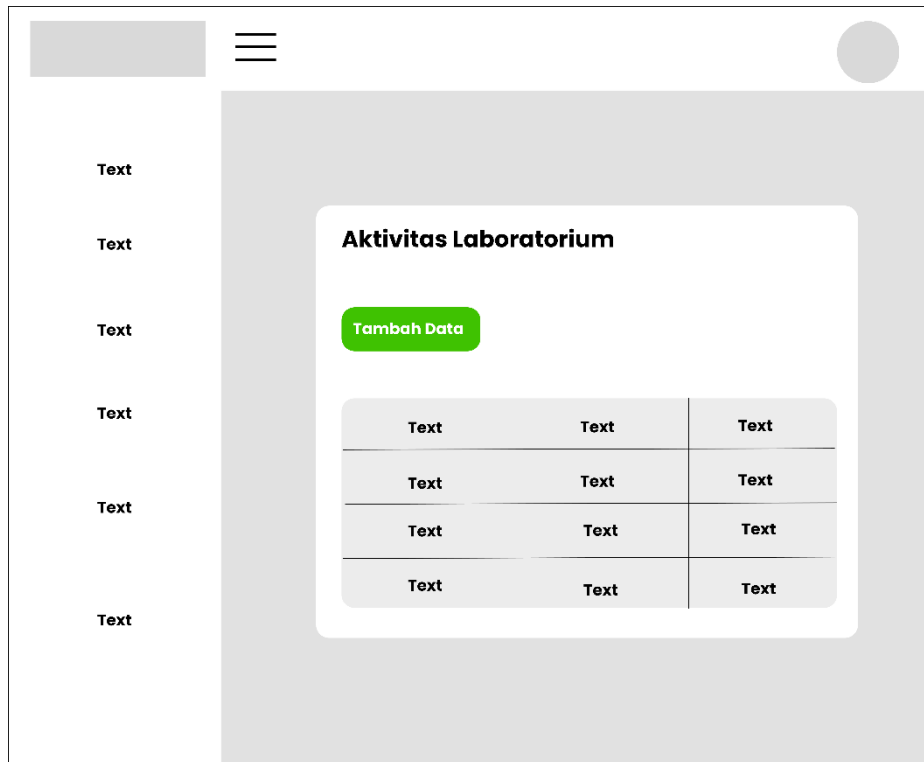
Gambar 57. Activity Diagram Hapus NPM Mahasiswa.



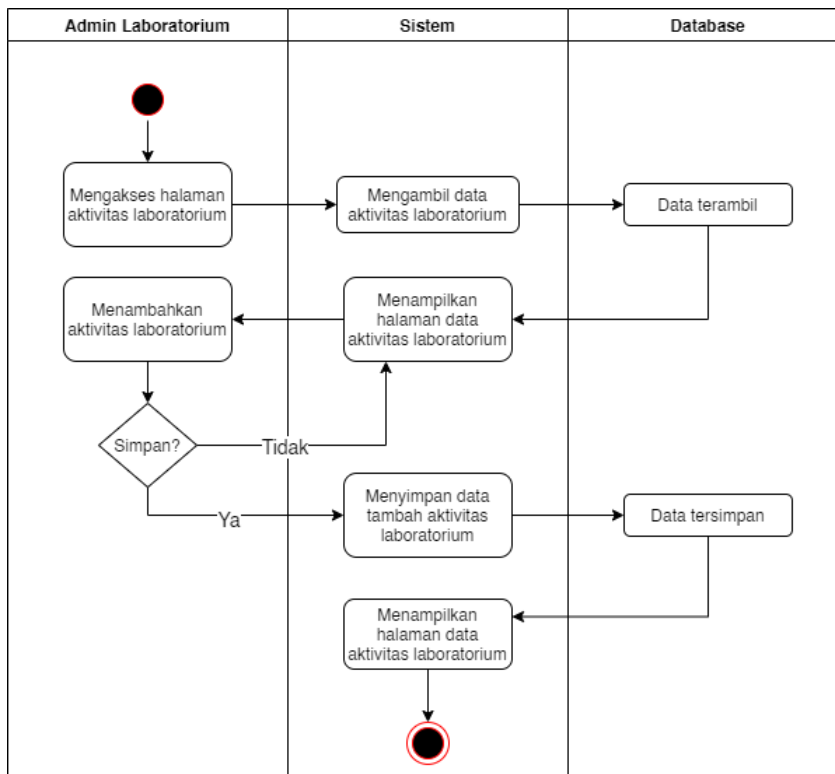
Gambar 58. Desain Antarmuka Hapus Data NPM Mahasiswa.

10. Aktivitas Laboratorium

Pada halaman ini, admin laboratorium dapat menambah, mengedit, melihat dan menghapus aktivitas yang ada di laboratorium sesuai dengan penempatan di laboratorium yang telah ditetapkan oleh super admin. Halaman ini terhubung dengan tabel *activity lab* dalam *database* sistem informasi Chemistry Program Data Center yang terdapat pada lampiran 3. Desain antarmuka dan *activity diagram* halaman aktivitas laboratorium dapat dilihat pada Gambar 59, Gambar 60, Gambar 61, Gambar 62, Gambar 63, Gambar 64, Gambar 65, Gambar 66 dan Gambar 67.

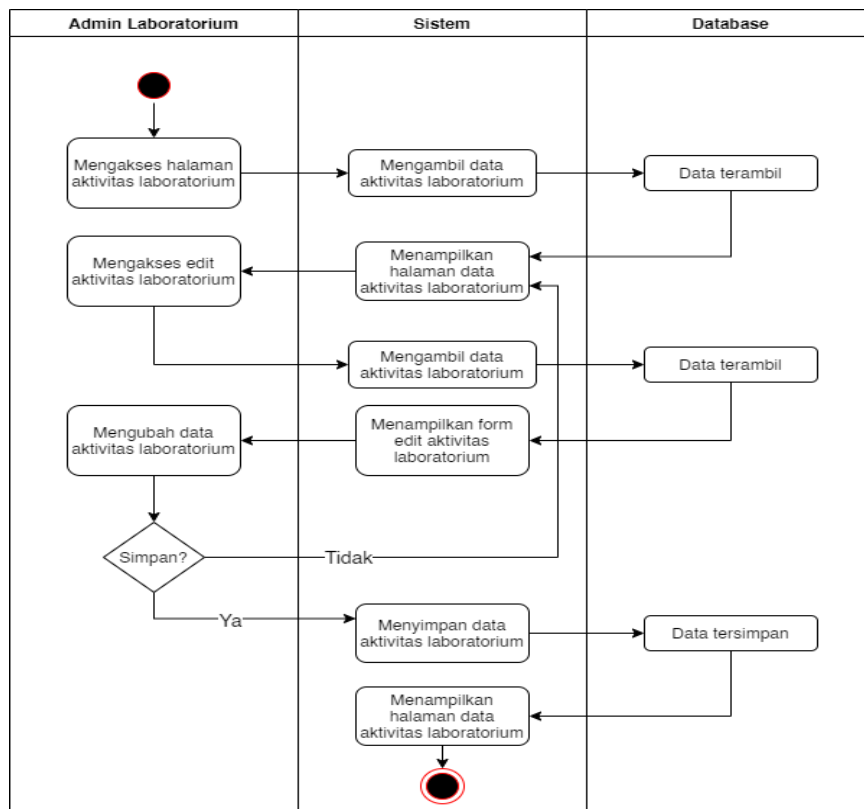


Gambar 59. Desain Antarmuka Aktivitas Laboratorium.

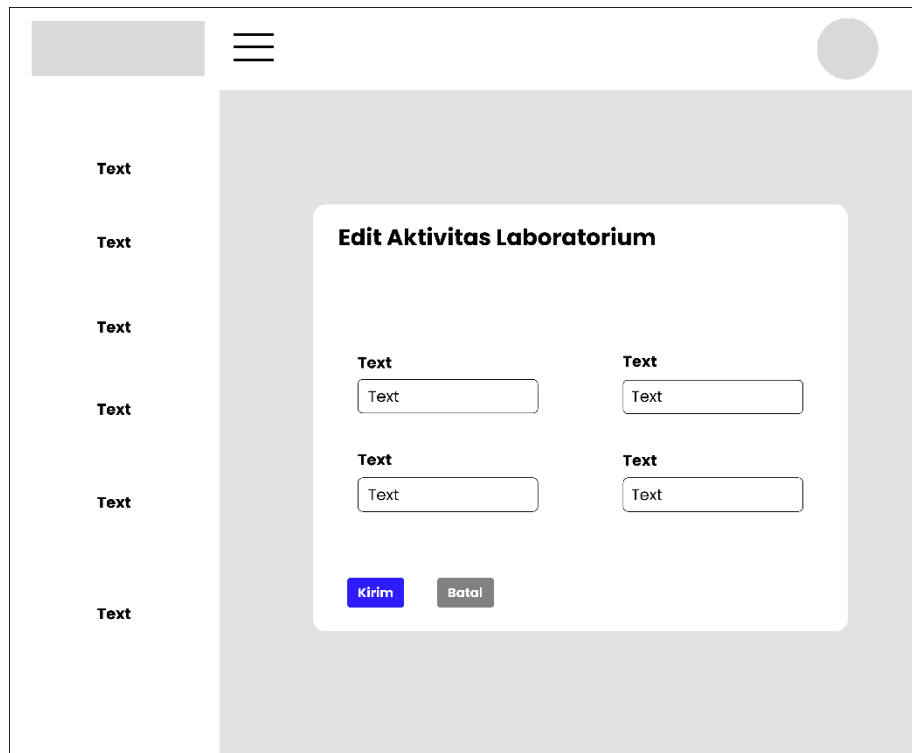


Gambar 60. Activity Diagram Tambah Aktivitas Laboratorium.

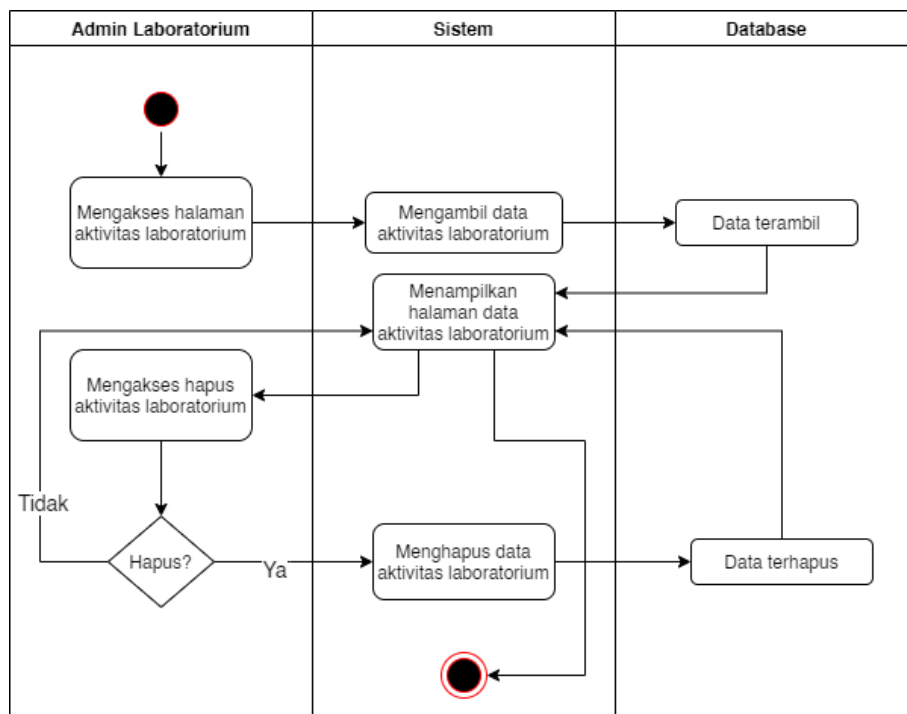
Gambar 61. Desain Antarmuka Tambah Aktivitas Laboratorium.



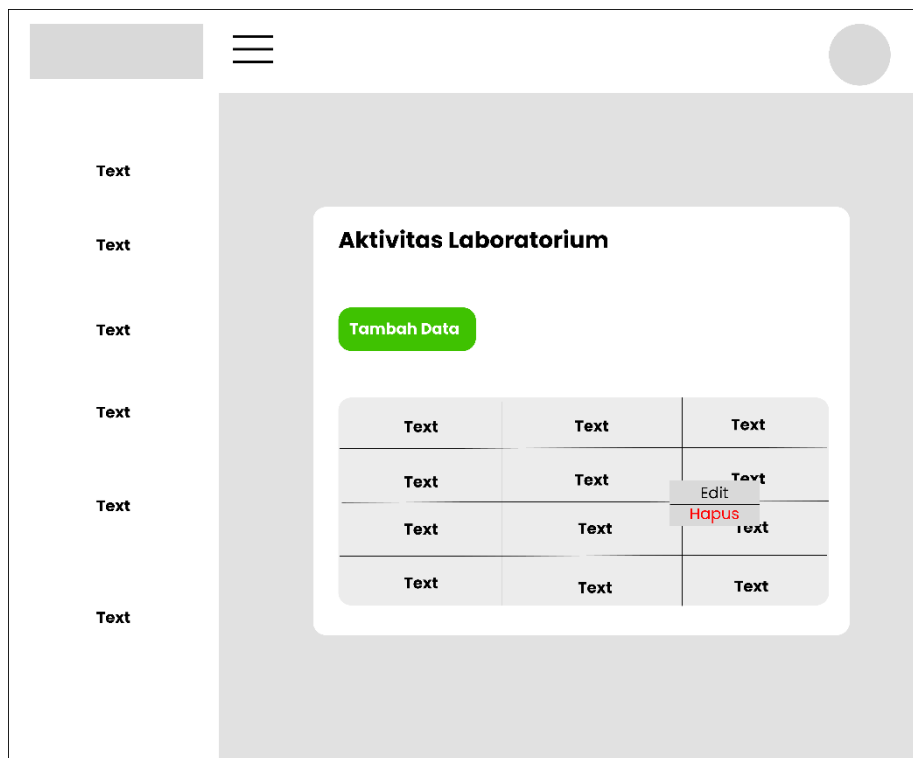
Gambar 62. Activity Diagram Edit Aktivitas Laboratorium.



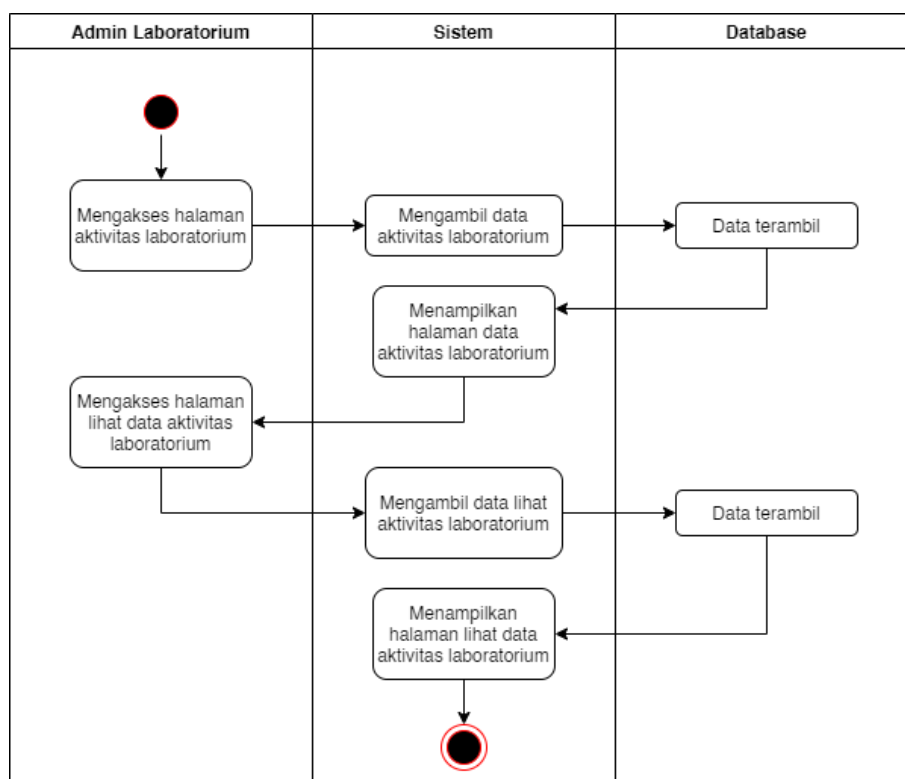
Gambar 63. Desain Antarmuka Edit Aktivitas Laboratorium.



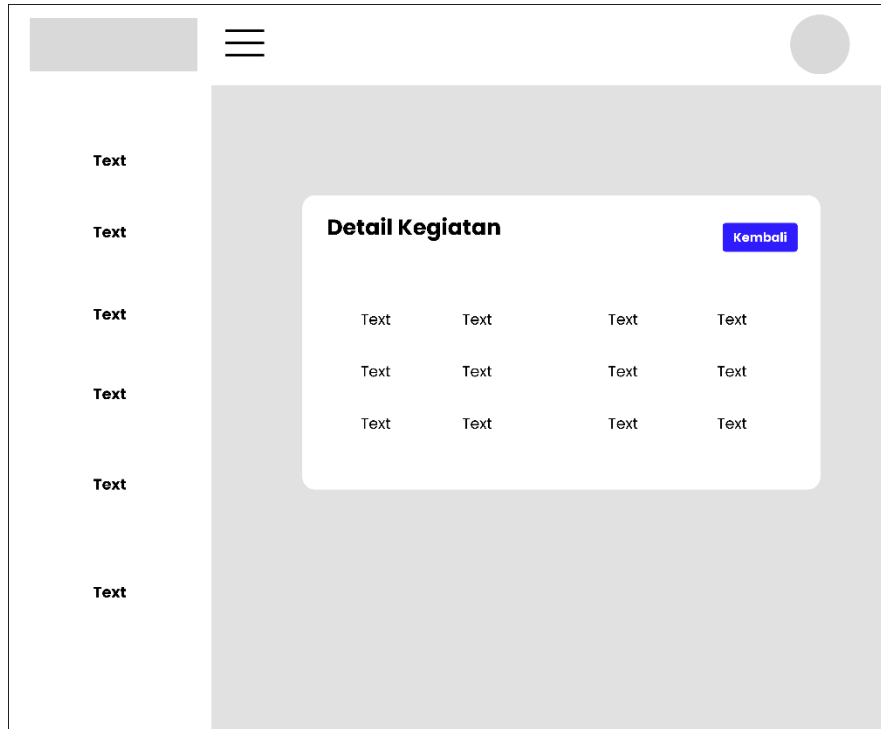
Gambar 64. Activity Diagram Hapus Aktivitas Laboratorium.



Gambar 65. Desain Antarmuka Hapus Aktivitas Laboratorium.



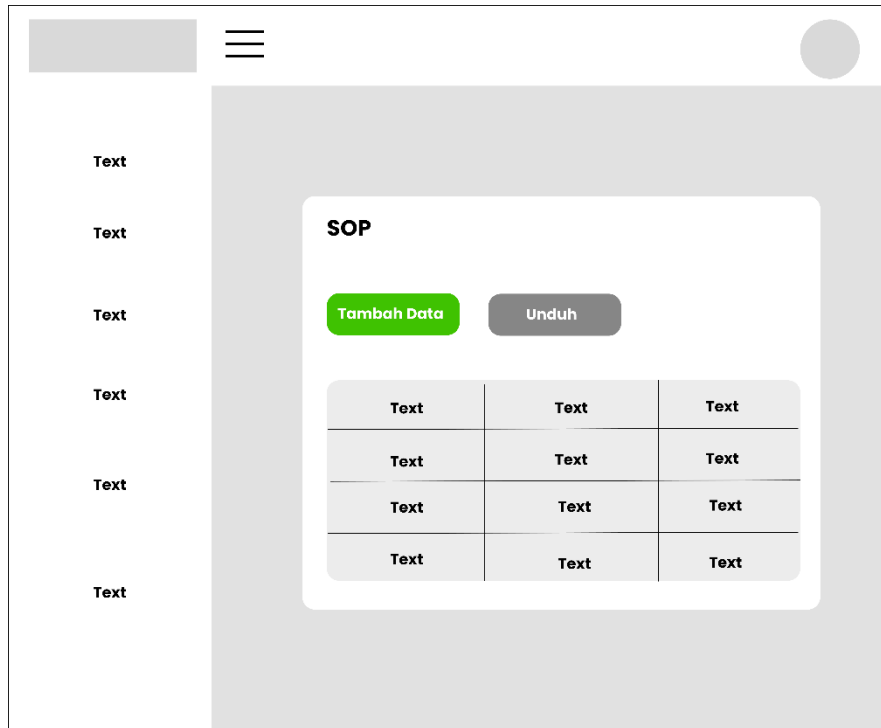
Gambar 66. Activity Diagram Lihat Data Aktivitas Laboratorium.



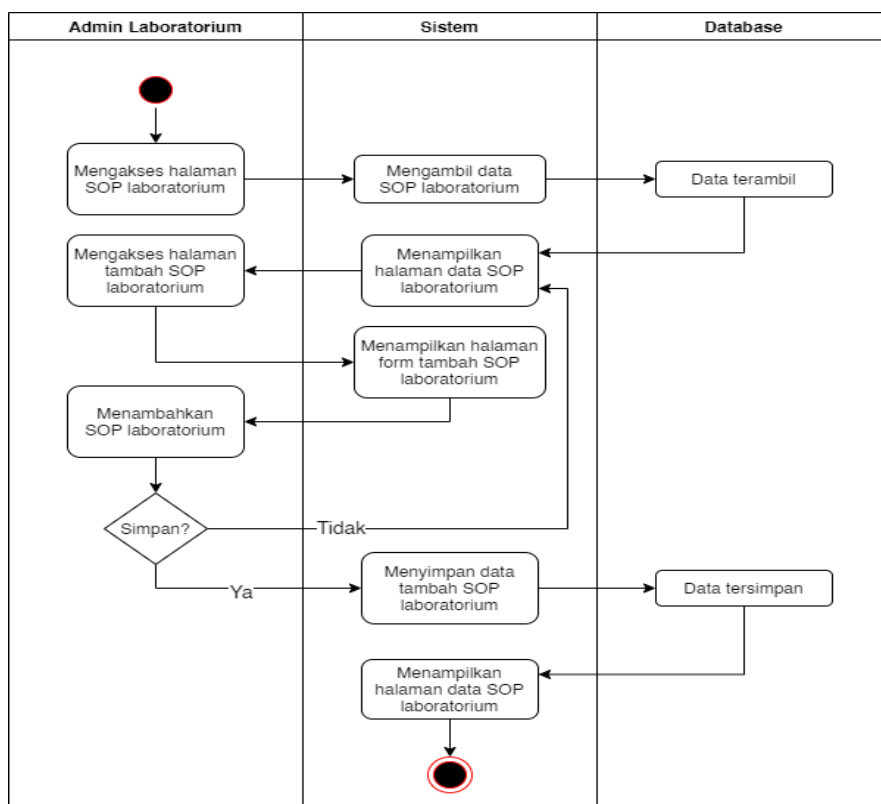
Gambar 67. Desain Antarmuka Lihat Data Aktivitas Laboratorium.

11. *Standard Operating Procedures (SOP) Laboratorium*

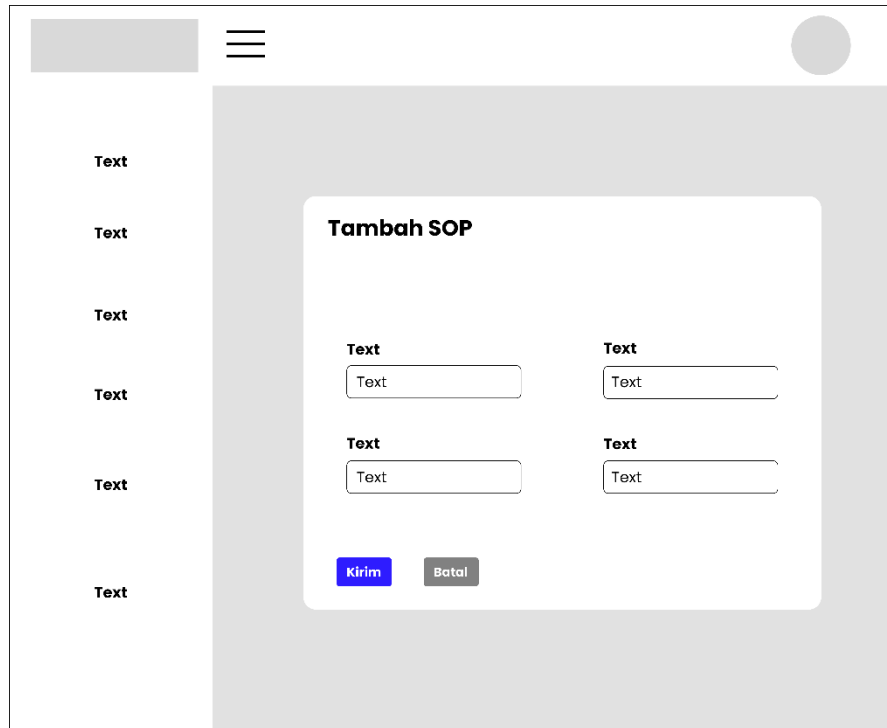
Pada bagian ini, admin laboratorium dapat menambah, mengubah, melihat dan menghapus SOP yang ada pada laboratoriumnya sesuai dengan penempatan laboratoriumnya oleh super admin. Halaman ini terhubung dengan tabel sop lab dalam *database* sistem informasi Chemistry Program Data Center. Desain antarmuka halaman pendataan SOP laboratorium dapat dilihat pada Gambar 68, Gambar 69, Gambar 70, Gambar 71, Gambar 72, Gambar 73, Gambar 74, Gambar 75, Gambar 76, Gambar 77 dan Gambar 78.



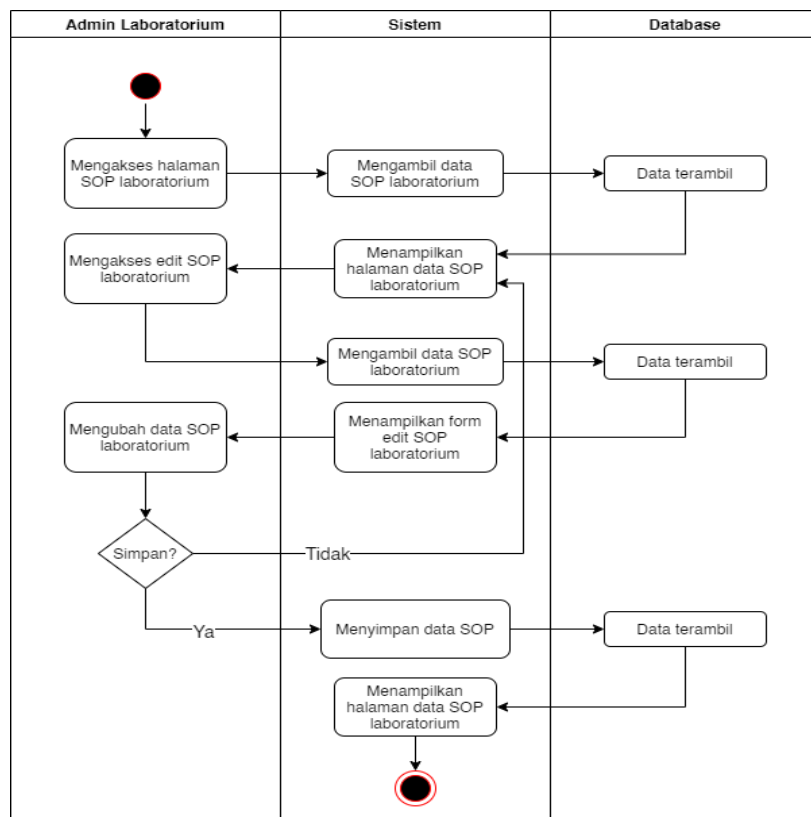
Gambar 68. Desain Antarmuka SOP Laboratorium.



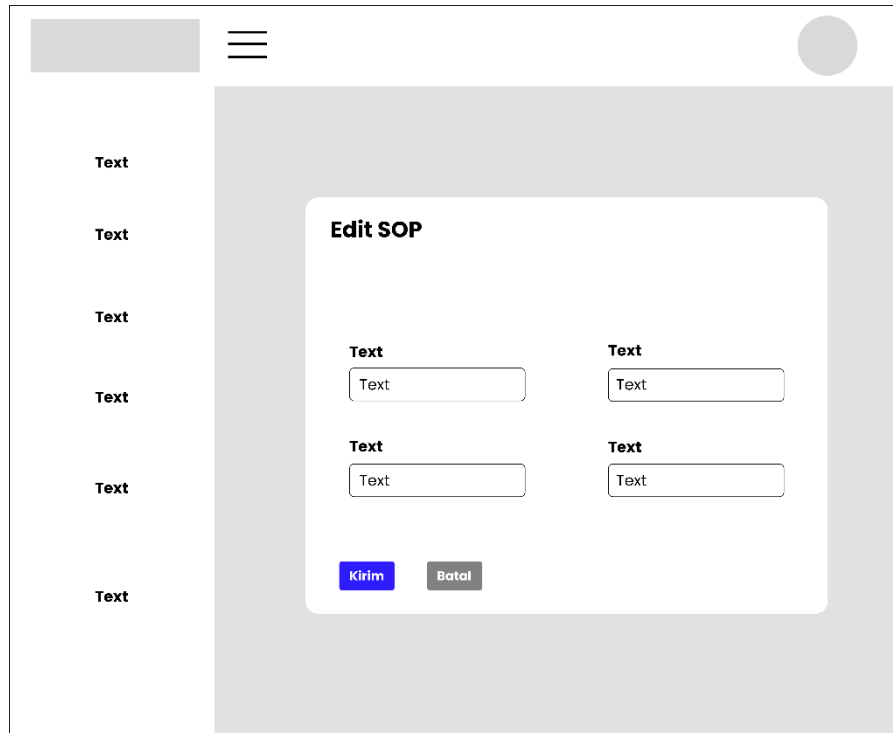
Gambar 69. Activity Diagram Tambah SOP Laboratorium.



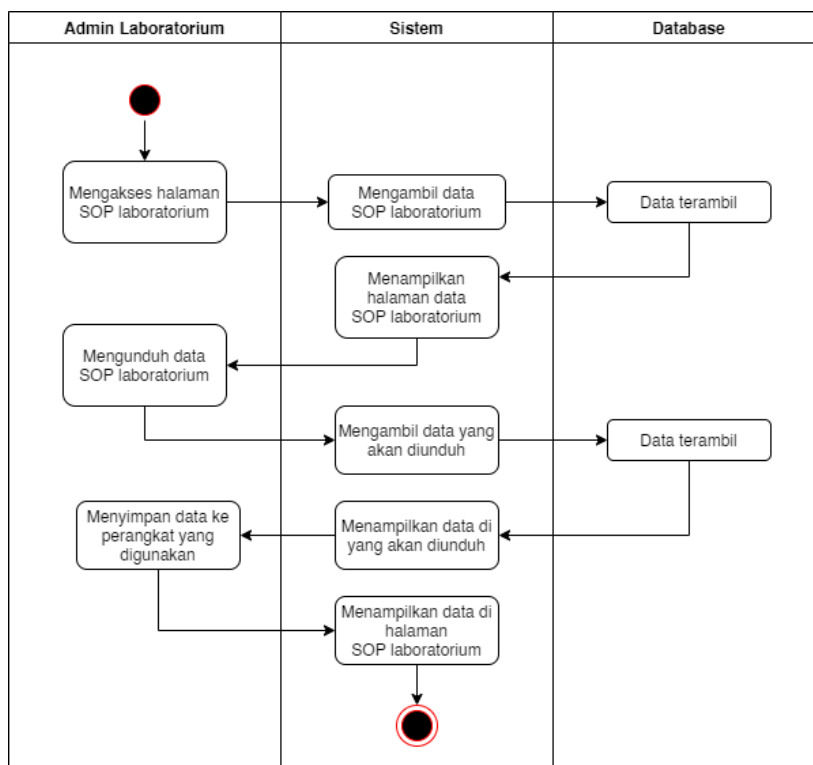
Gambar 70. Desain Antarmuka Tambah SOP Laboratorium.



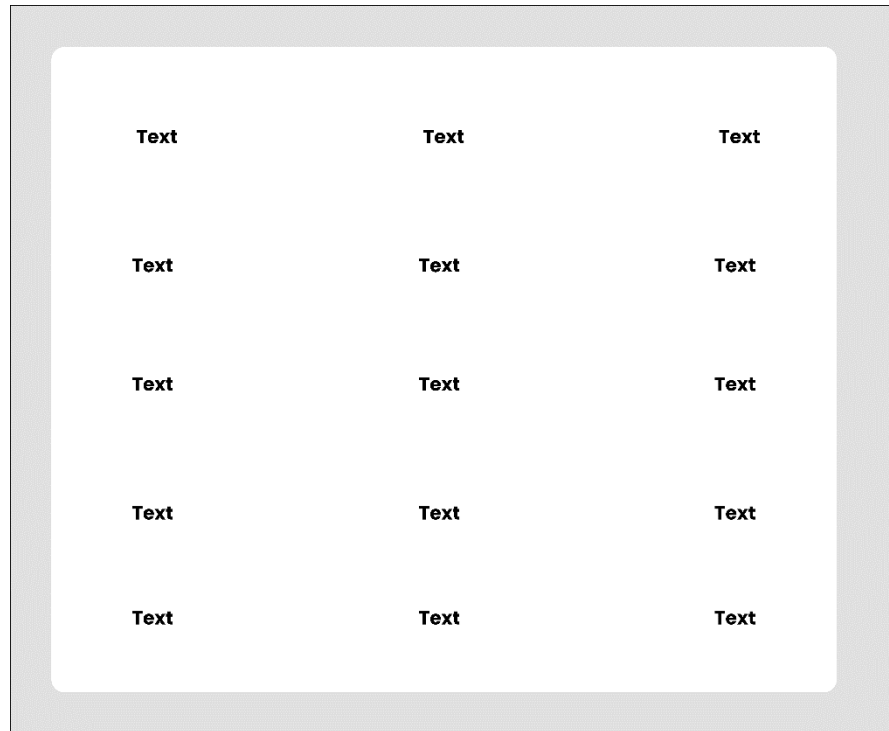
Gambar 71. Activity Diagram Edit SOP Laboratorium.



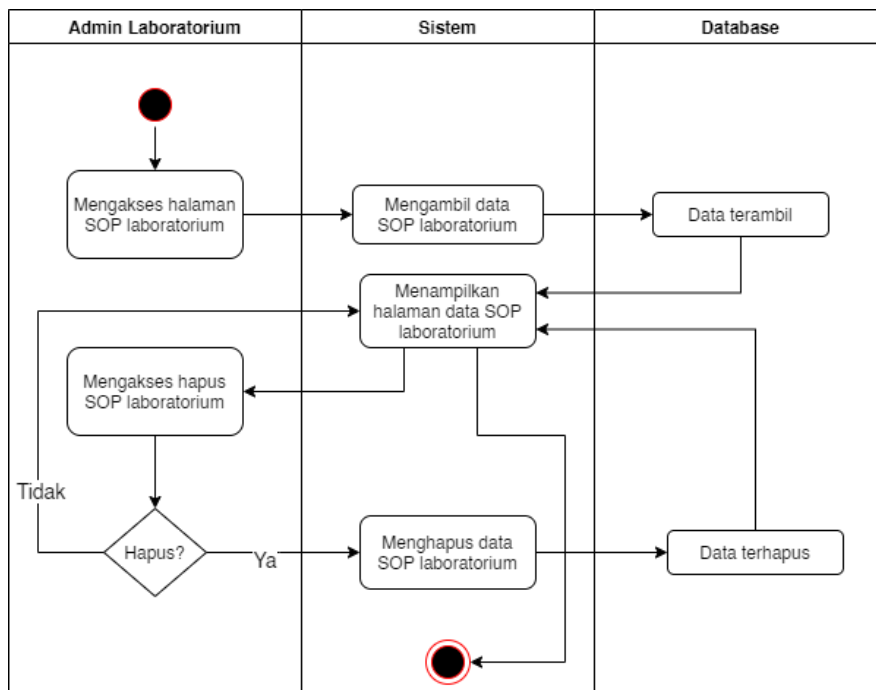
Gambar 72. Desain Antarmuka Edit SOP Laboratorium.



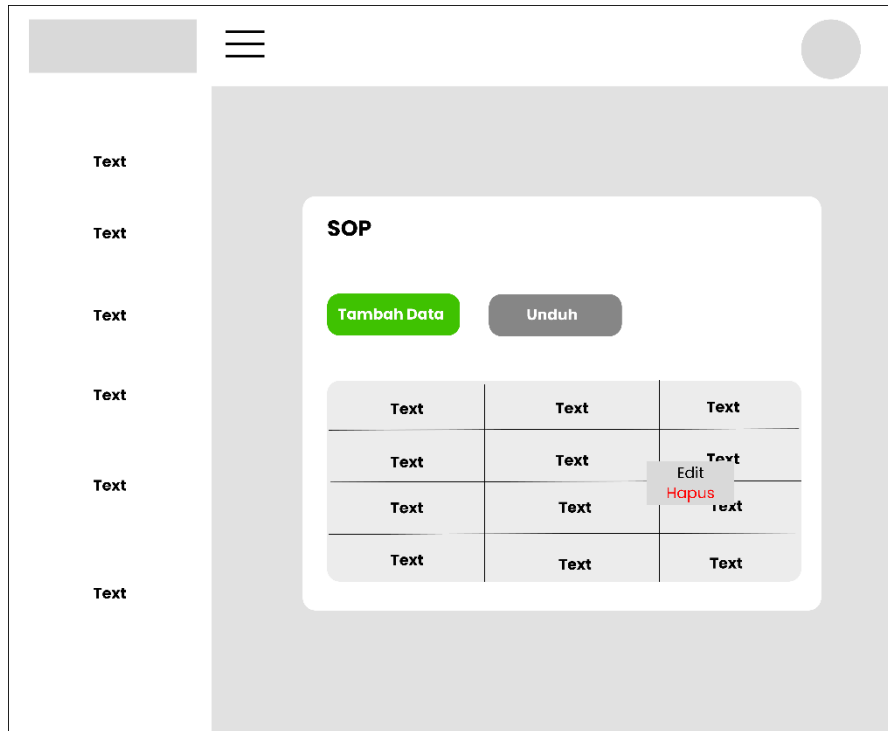
Gambar 73. Activity Diagram Unduh SOP Laboratorium.



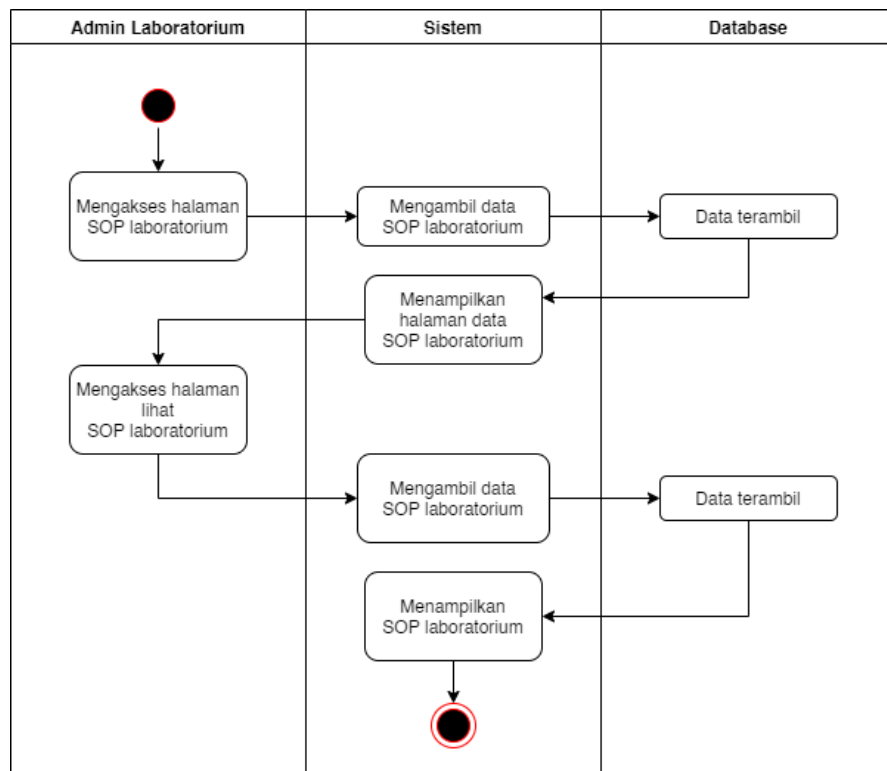
Gambar 74. Desain Antarmuka Unduh Data SOP Laboratorium.



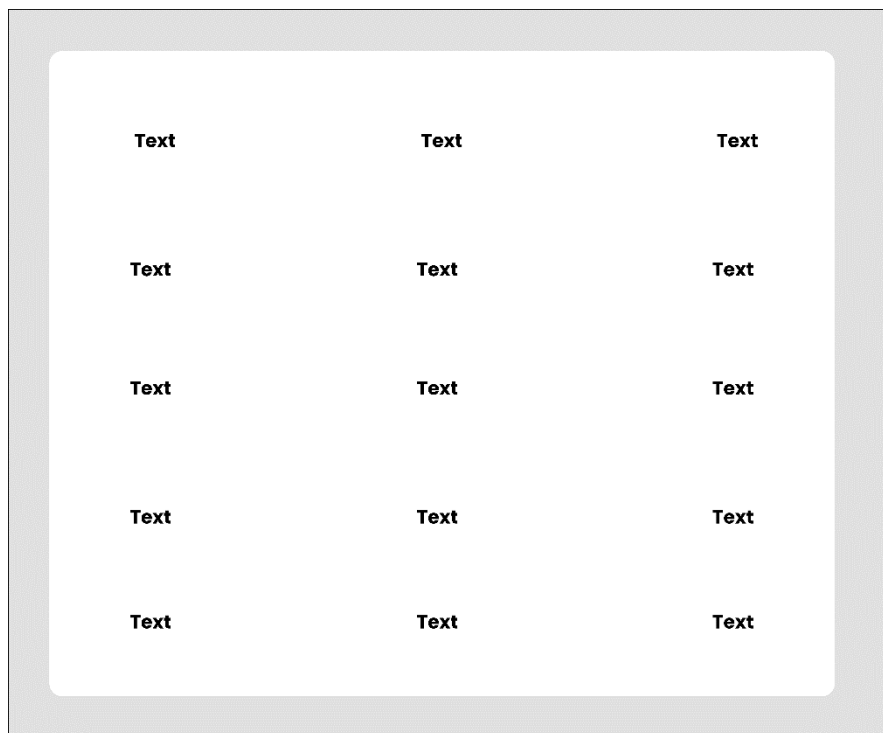
Gambar 75. Activity Diagram Hapus SOP Laboratorium.



Gambar 76. Desain Antarmuka Hapus SOP Laboratorium.



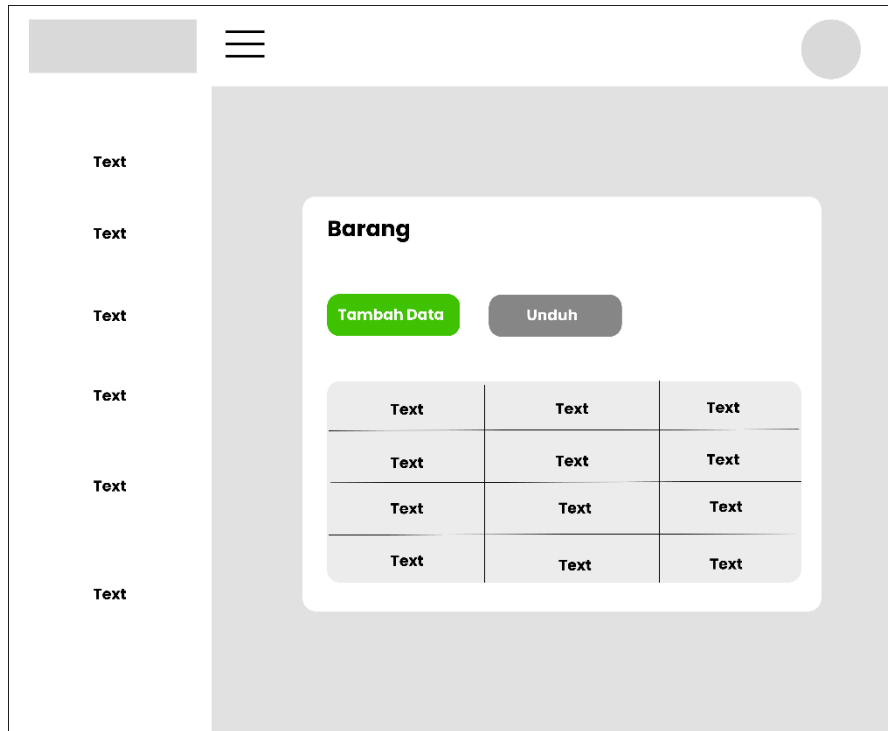
Gambar 77. Activity Diagram Lihat SOP Laboratorium.



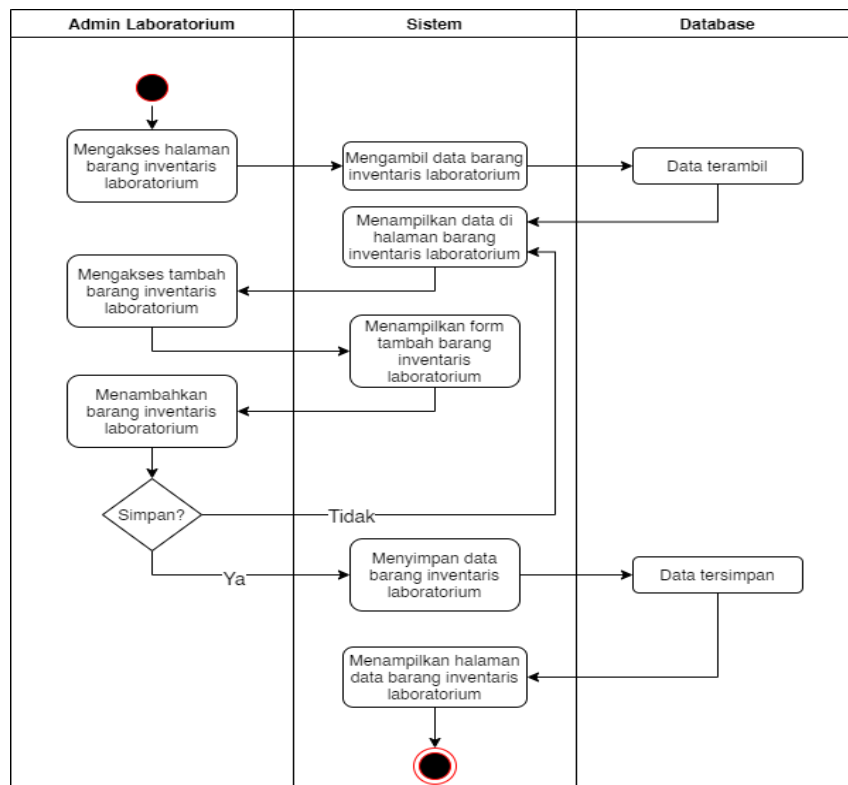
Gambar 78. Desain Antarmuka Lihat SOP Laboratorium.

12. Inventaris Laboratorium

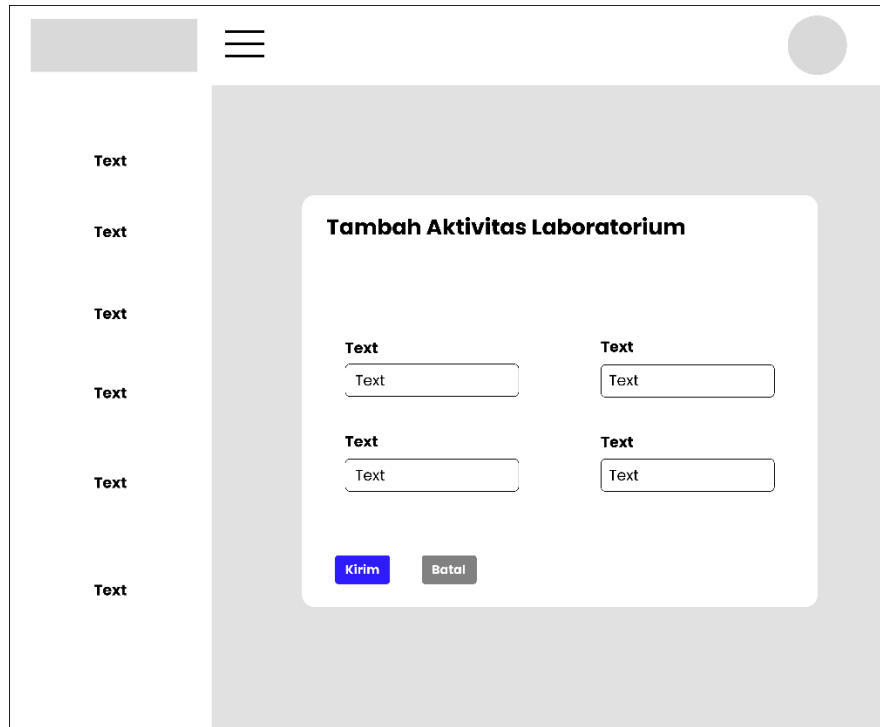
Pada halaman ini, admin laboratorium dapat dapat menambah, mengubah, melihat, mengunduh dan menghapus data inventaris barang yang ada di laboratorium. Halaman ini terhubung dengan tabel barang dalam *database* sistem informasi Chemistry Program Data Center yang terdapat pada lampiran 3. Desain antarmuka dan activity diagram halaman pendataan inventaris barang laboratorium dapat dilihat pada Gambar 79, Gambar 80, Gambar 81, Gambar 82, Gambar 83, Gambar 84, Gambar 85, Gambar 86, Gambar 87, Gambar 88 dan Gambar 89.



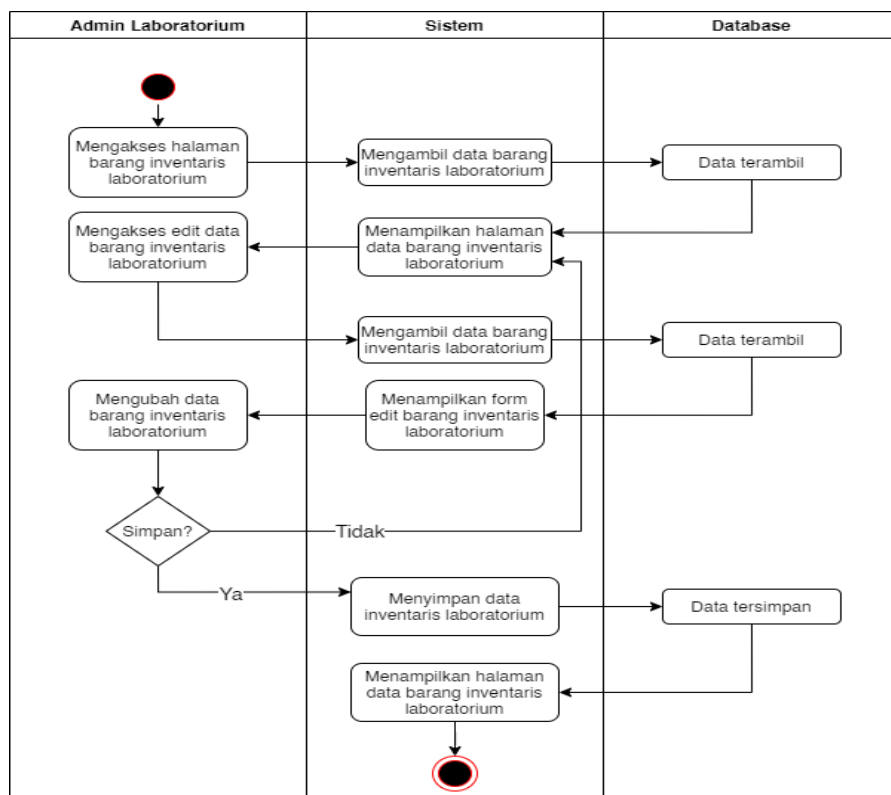
Gambar 79. Desain Antarmuka Inventaris Laboratorium.



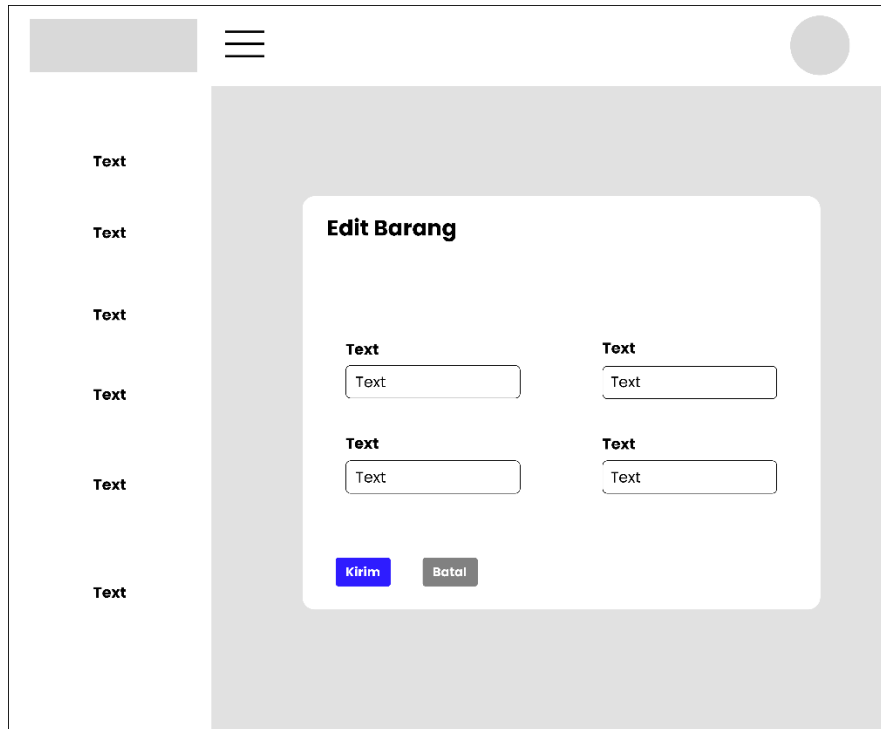
Gambar 80. Activity Diagram Tambah Data Inventaris Laboratorium.



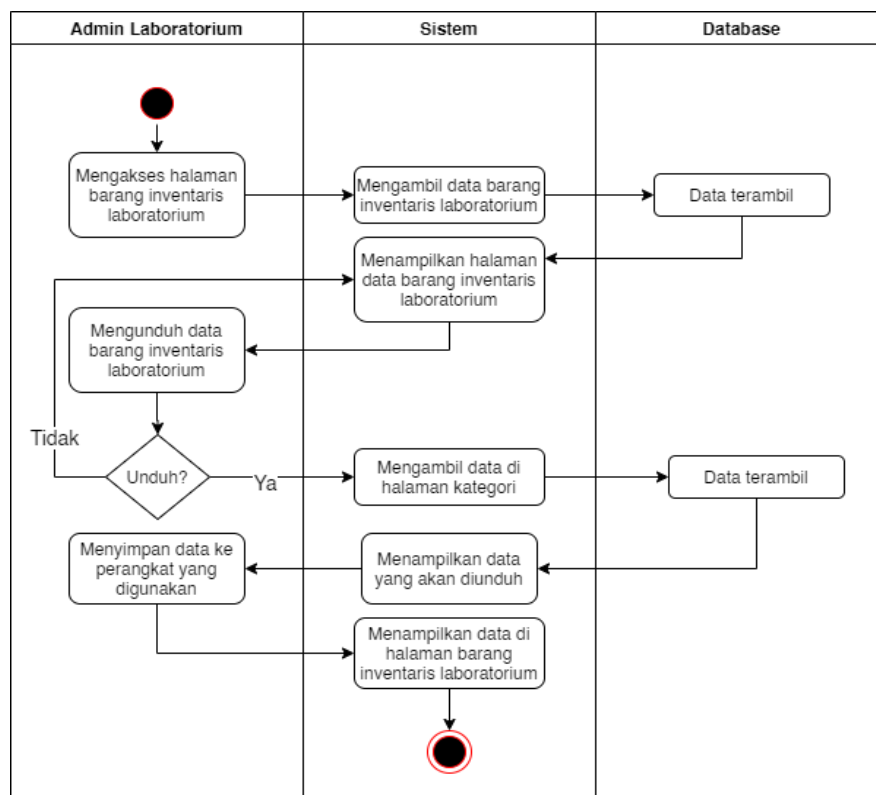
Gambar 81. Desain Antarmuka Tambah Data Inventaris Laboratorium.



Gambar 82. Activity Diagram Edit Inventaris Laboratorium.



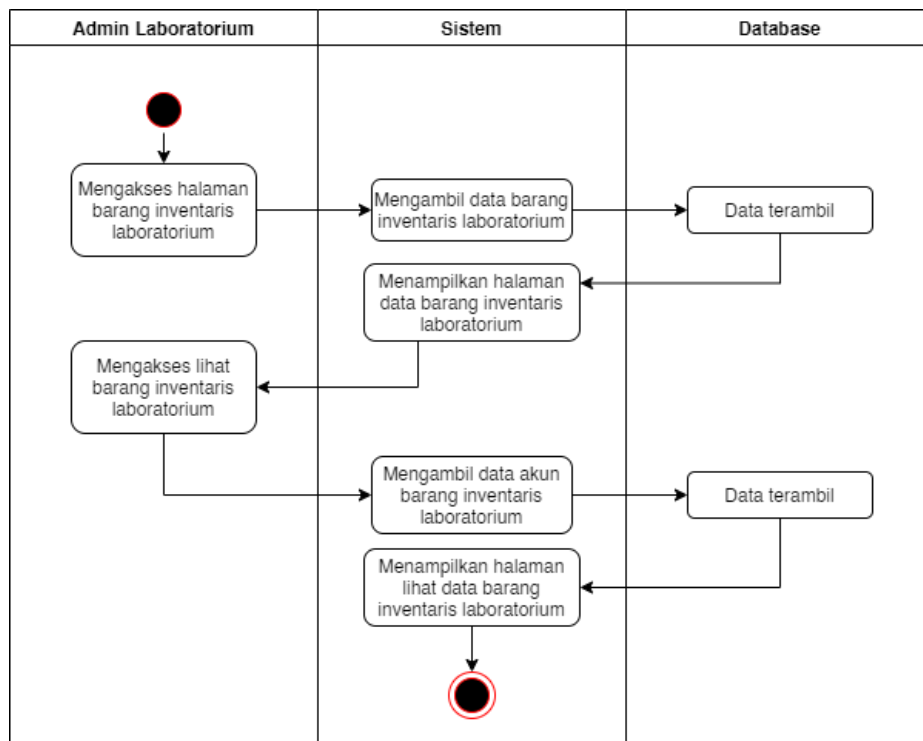
Gambar 84. Desain Antarmuka Edit Data Inventaris Laboratorium.



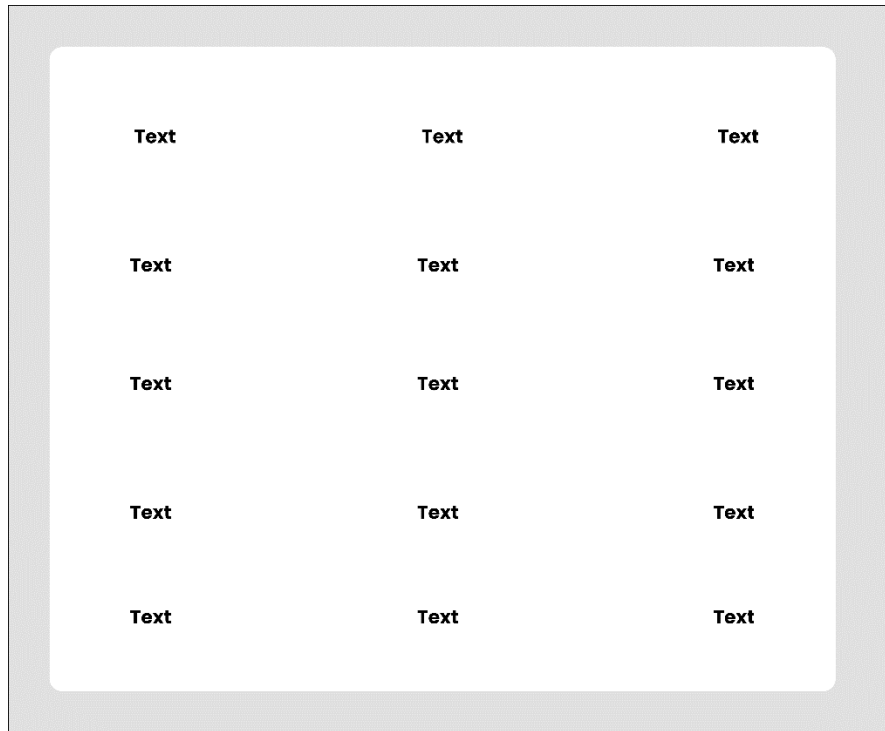
Gambar 83. Activity Diagram Unduh Data Inventaris Laboratorium.

Text	Text	Text
Text	Text	Text
Text	Text	Text
Text	Text	Text
Text	Text	Text

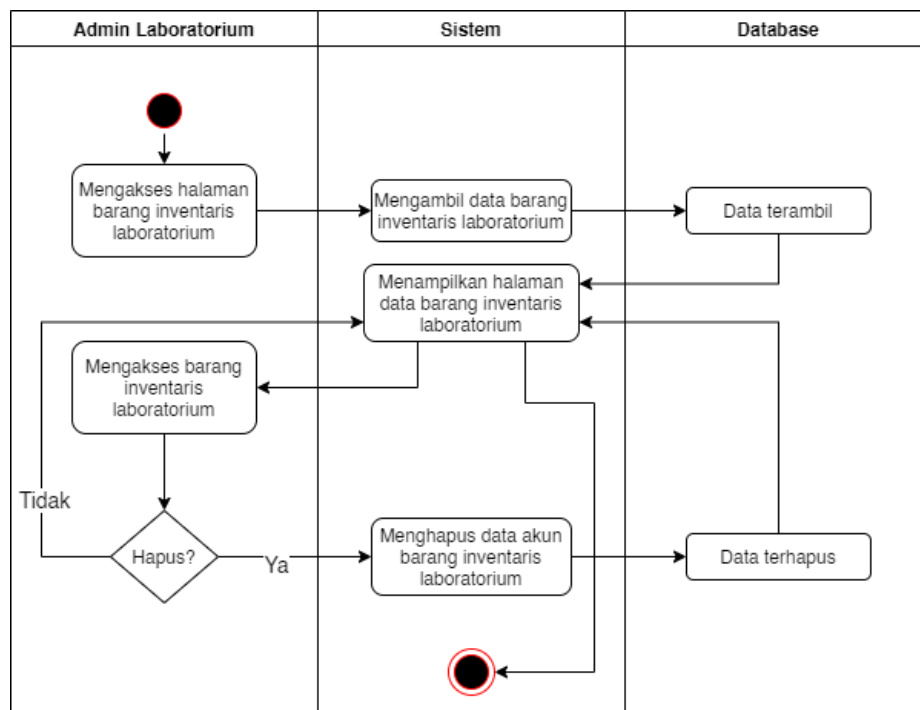
Gambar 85. Desain Antarmuka Unduh Data Inventaris Laboratorium.



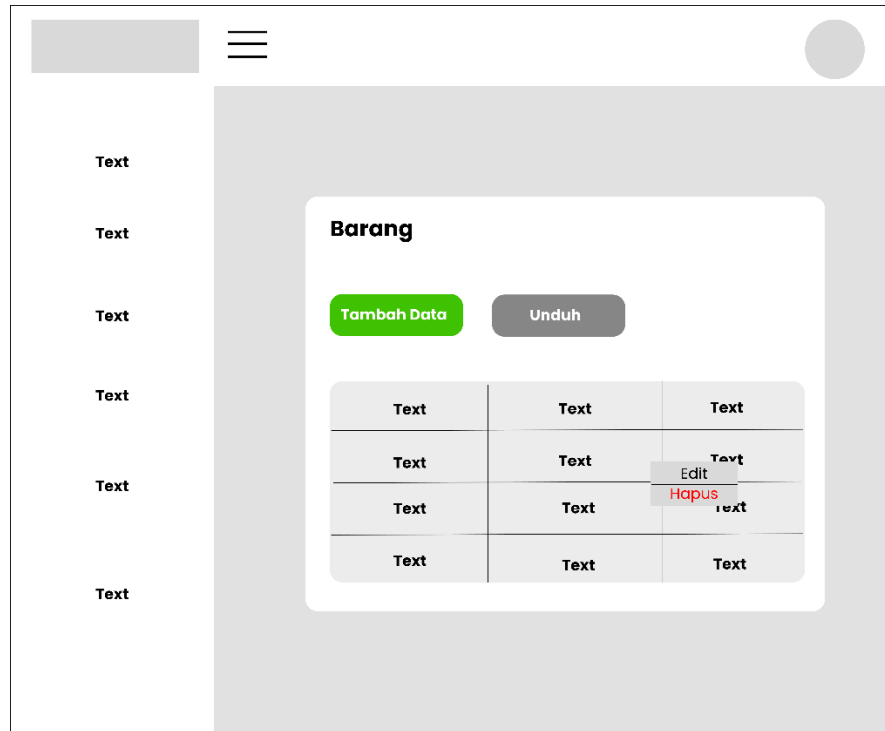
Gambar 86. Activity Diagram Lihat Inventaris Laboratorium.



Gambar 87. Desain Antarmuka Lihat Data Inventaris Laboratorium.



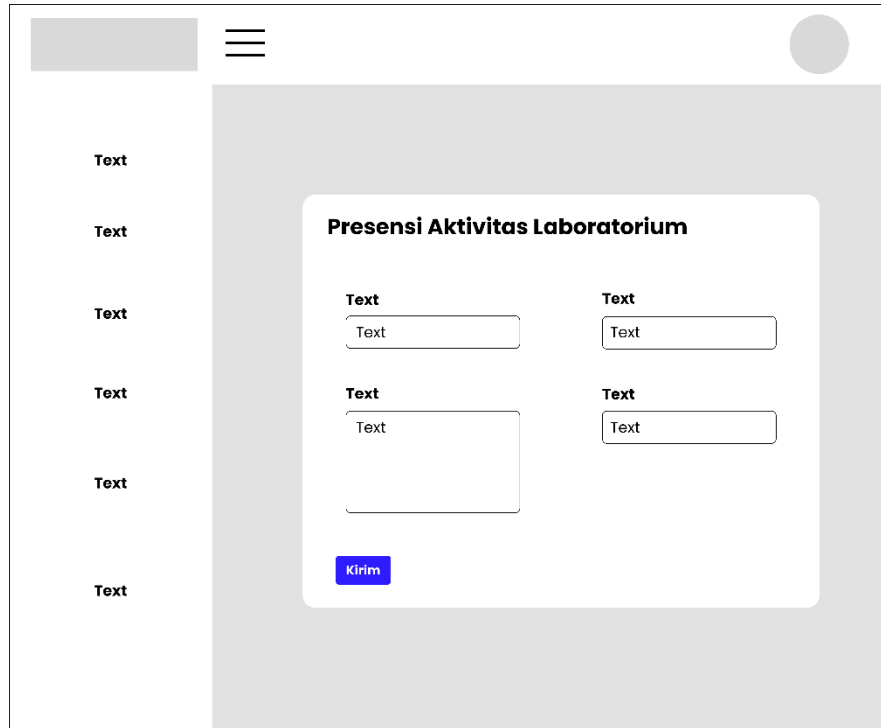
Gambar 88. Activity Diagram Hapus Inventaris Laboratorium.



Gambar 89. Desain Antarmuka Hapus Data Inventaris Laboratorium.

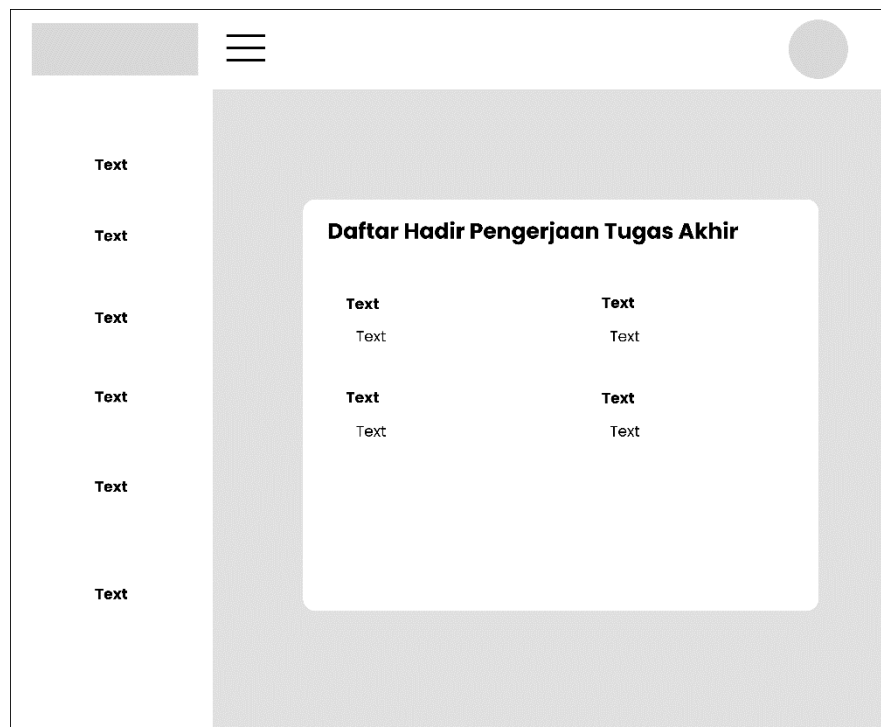
13. Fitur Tambahan Pada Presensi Laboratorium Tugas Akhir Mahasiswa S2

Pada halaman ini, mahasiswa S2 dapat melakukan presensi pada saat menggunakan fasilitas laboratorium yang ada di jurusan kimia. Halaman ini terhubung dengan tabel *activity lab* dalam *database* sistem informasi Chemistry Program Data Center yang terdapat pada lampiran 3. Desain antarmuka dan *activity diagram* dapat dilihat pada gambar sebagai berikut.



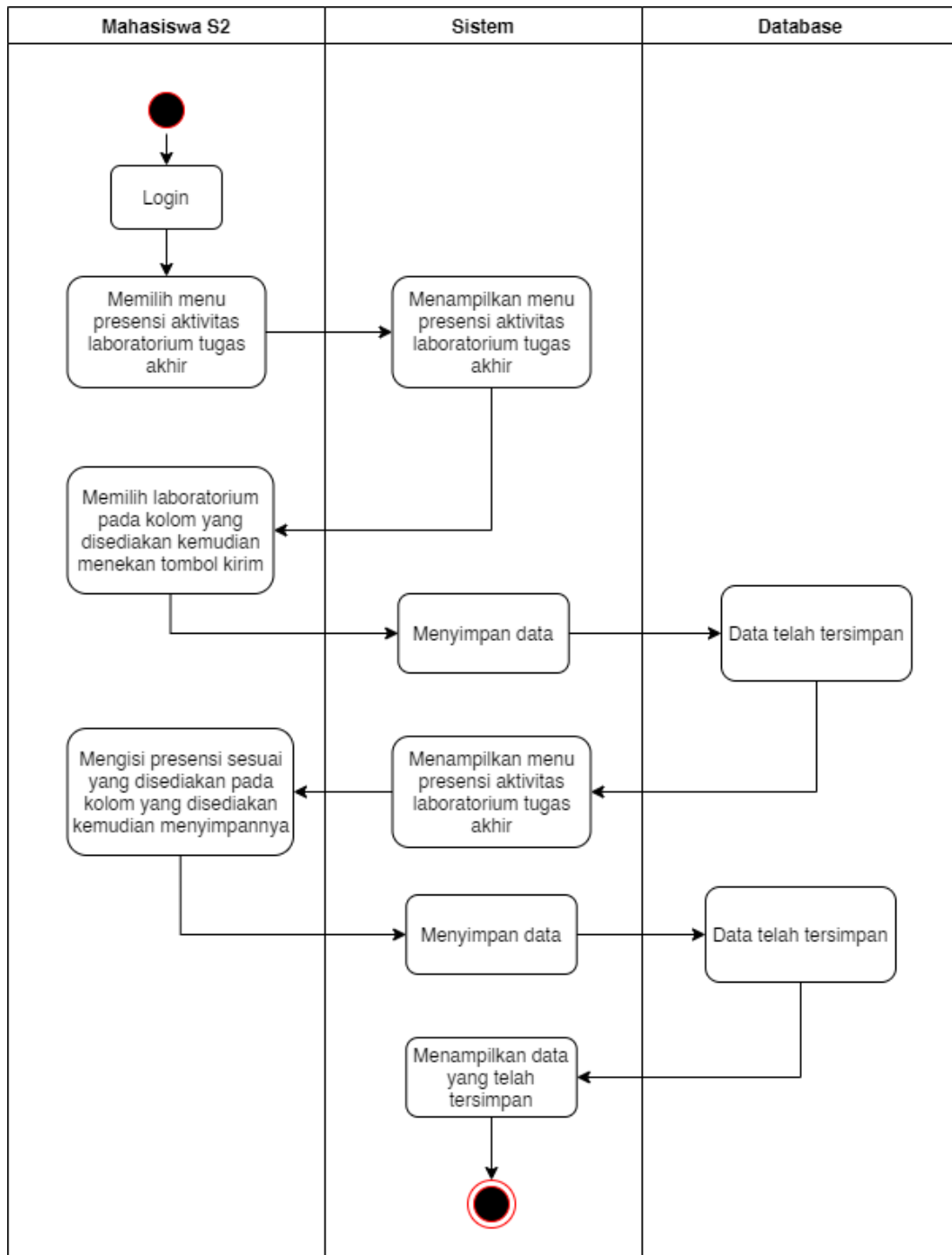
The image shows a web interface for a laboratory activity attendance form. It features a top navigation bar with a hamburger menu icon and a profile icon. A left sidebar contains six 'Text' labels. The main content area has a light gray background and contains a white form titled 'Presensi Aktivitas Laboratorium'. The form includes four text input fields arranged in two columns, a 'Kirim' button at the bottom left, and a 'Text' label at the bottom right.

Gambar 90. Desain Antarmuka Presensi Aktivitas Laboratorium Mahasiswa S2.



The image shows a web interface for a final task attendance list. It features a top navigation bar with a hamburger menu icon and a profile icon. A left sidebar contains six 'Text' labels. The main content area has a light gray background and contains a white form titled 'Daftar Hadir Pengerjaan Tugas Akhir'. The form includes four text input fields arranged in two columns, with a 'Text' label at the bottom right.

Gambar 91. Desain Antarmuka Mahasiswa Sudah Melakukan Presensi.



Gambar 92. Activity Diagram Presensi Aktivitas Laboratorium Mahasiswa S2.

3.7 Penulisan Kode Program

Pada perancangan dan implementasi sistem informasi Chemistry Program Data Center, penulisan program menggunakan bahasa pemrograman PHP dengan *framework* laravel sebagai sistem MVC (*Model View and Controller*) dan MySQL sebagai basis data dari sistem ini. Untuk *text editor* menggunakan Visual Studio Code.

3.8 Pengujian Sistem

Pengujian Alpha atau sering disebut pengujian fungsionalitas pada sistem dilakukan dengan *black box testing*. Pengujian ini dilakukan terhadap setiap fungsi yang ada dan komponen antarmuka yang ada pada sistem. Pengujian dilakukan secara langsung oleh ketua jurusan hingga mahasiswa di Jurusan Kimia FMIPA Universitas Lampung. Pengujian ini berguna untuk membuktikan semua fungsi yang ada pada sistem berjalan dengan baik. Pada pengujian ini diyakini bahwa masukan dan respon yang diterima sama, sehingga terjadi kecocokan antara sistem dan *user*.

3.8.1 Skenario *Black Box Testing*

Skenario pengujian menggunakan *black box testing* dapat dilihat pada Tabel 8.

Tabel 8. Skenario *Black Box Testing*.

No	Kelas Uji	Kasus Uji	Hasil Yang Diharapkan
1	Mengakses halaman kategori laboratorium pada <i>role</i> super admin	Mengakses <i>dropdown</i> inventaris kemudian klik menu tab kategori Mengakses tambah data kategori dengan klik <i>button</i> tambah data Mengakses edit data klik titik 3 pada kolom aksi dan klik edit kategori	Sistem menampilkan halaman kategori beserta data yang ada Sistem menampilkan halaman tambah data kategori Sistem menampilkan halaman edit kategori beserta data yang akan diedit

Lanjutan **Tabel 8** Skenario *Black Box Testing*.

		Mengakses cetak atau unduh data kategori dengan klik tombolnya	Sistem dapat menampilkan data yang akan dicetak atau diunduh
		Mengakses hapus data kategori dengan klik titik 3 pada kolom aksi	Sistem dapat menghapus data kategori
2	Mengakses halaman model laboratorium pada <i>role</i> super admin	Mengakses <i>dropdown</i> inventaris kemudian klik menu tab model	Sistem menampilkan halaman model beserta data yang ada
		Mengakses tambah data model dengan klik <i>button</i> tambah data	Sistem menampilkan halaman tambah data model
		Mengakses edit data model dengan klik titik 3 pada kolom aksi dan klik edit	Sistem menampilkan halaman edit model beserta data yang akan diedit
3	Mengakses halaman akun mahasiswa pada <i>role</i> super admin	Mengakses <i>dropdown</i> akun kemudian klik menu tab mahasiswa	Sistem dapat menampilkan halaman akun mahasiswa beserta data yang ada
		Mengakses edit akun mahasiswa dengan klik titik 3 pada kolom aksi dan pilih edit	Sistem dapat menampilkan halaman edit akun mahasiswa beserta data yang akan diedit
		Mengakses hapus akun mahasiswa dengan klik titik 3 pada kolom aksi dan klik hapus	Sistem dapat menghapus data akun mahasiswa

Lanjutan **Tabel 8** Skenario *Black Box Testing*.

4	Mengakses halaman akun dosen pada <i>role</i> super admin	Mengakses <i>dropdown</i> akun kemudian klik menu tab dosen Mengakses edit akun dosen dengan klik titik 3 pada kolom aksi dan klik edit Mengakses hapus akun dosen dengan klik titik 3 pada kolom aksi dan klik hapus	Sistem dapat menampilkan halaman akun dosen beserta data yang ada Sistem dapat menampilkan halaman edit akun dosen beserta data yang akan diedit Sistem dapat menampilkan halaman edit akun dosen beserta data yang akan diedit
5	Mengakses halaman akun admin pada <i>role</i> super admin	Mengakses <i>dropdown</i> akun kemudian klik menu tab admin Mengakses edit akun admin dengan klik titik 3 pada kolom aksi dan klik edit Mengakses hapus akun admin dengan klik titik 3 pada kolom aksi dan klik hapus	Sistem dapat menampilkan halaman akun admin beserta data yang ada Sistem dapat menampilkan halaman edit akun dosen beserta data yang akan diedit Sistem menghapus akun admin

Lanjutan **Tabel 8** Skenario *Black Box Testing*.

6	Mengakses halaman data NPM mahasiswa pada <i>role</i> super admin	Mengakses halaman data akun mahasiswa dengan cara klik tab data NPM	Sistem dapat menampilkan halaman akun mahasiswa beserta data yang ada
		Mengakses tambah NPM mahasiswa dengan cara klik button tambah NPM mahasiswa	Sistem dapat mengakses halaman tambah data NPM mahasiswa
		Mengakses <i>import</i> NPM mahasiswa dengan cara klik <i>button import</i>	Sistem dapat menampilkan <i>floating window</i> untuk memilih file excel yang akan diunggah
		Mengakses edit NPM mahasiswa dengan klik titik 3 dan pilih edit NPM mahasiswa	Sistem dapat menampilkan halaman edit NPM mahasiswa beserta data yang ada
		Mengakses hapus NPM mahasiswa dengan cara klik <i>button</i> NPM mahasiswa	Sistem dapat menghapus akun mahasiswa
7	Mengakses halaman manajemen akun kepala laboratorium pada <i>role</i> super admin	Mengakses halaman kepala laboratorium dengan cara klik tab kepala laboratorium	Sistem dapat menampilkan halaman kepala laboratorium beserta datanya
		Mengakses unduh atau cetak data kepala laboratorium dengan cara klik <i>button</i> unduh atau cetak	Sistem dapat mngunduh atau mencetak data akun kepala laboratorium

Lanjutan **Tabel 8** Skenario *Black Box Testing*.

8	Mengakses halaman manajemen akun admin laboratorium pada <i>role</i> super admin	Mengakses halaman manajemen akun admin dengan cara klik tab admin jurusan	Sistem dapat menampilkan halaman admin jurusan beserta datanya
		Mengakses unduh atau cetak data admin jurusan dengan cara klik <i>button</i> unduh atau cetak	Sistem dapat mngunduh atau mencetak data akun admin jurusan
9	Mengakses aktivitas laboratorium pada <i>role</i> admin laboratorium	Mengakses halaman aktivitas laboratorium dengan cara klik tab laboratorium	Sistem dapat menampilkan halaman aktivitas laboratorium beserta data yang ada
		Mengakses tambah data aktivitas laboratorium dengan cara klik <i>button</i> tambah data	Sistem dapat menampilkan halaman tambah data aktivitas laboratorium dan mengirimkannya ke <i>database</i>
		Mengakses lihat data aktivitas laboratorium dengan cara klik titik 3 pada kolom aksi kemudian klik <i>view</i>	Sistem dapat menampilkan halaman lihat data aktivitas laboratorium
		Mengubah data aktivitas laboratorium dengan cara klik titik 3 pada kolom aksi kemudian klik edit	Sistem dapat menampilkan halaman edit aktivitas laboratorium beserta data yang ada
		Menghapus data aktivitas laboratorium dengan cara klik titik 3 pada kolom aksi kemudian klik hapus	Sistem dapat menghapus data aktivitas laboratorium

Lanjutan **Tabel 8** Skenario *Black Box Testing*.

10	Mengakses halaman SOP laboratorium <i>role admin</i> laboratorium	Mengakses halaman SOP laboratorium dengan cara klik tab SOP	Sistem dapat menampilkan halaman SOP laboratorium beserta data yang ada
		Mengakses tambah SOP laboratorium dengan cara klik <i>button</i> tambah data	Sistem dapat menampilkan halaman tambah data SOP laboratorium dan mengirimkannya ke <i>database</i>
		Mengakses lihat data SOP dengan cara klik titik 3 pada kolom aksi kemudian klik <i>view</i>	Sistem dapat menampilkan halaman lihat data SOP laboratorium
		Mengubah data SOP dengan cara klik titik 3 pada kolom aksi kemudian klik edit	Sistem dapat menampilkan halaman edit SOP laboratorium beserta data yang ada
		Mengakses unduh atau cetak SOP Laboratorium dengan cara klik <i>button</i> unduh atau cetak	Sistem dapat mngunduh atau mencetak data SOP laboratorium
		Menghapus data SOP dengan cara klik titik 3 pada kolom aksi kemudian klik hapus	Sistem dapat menghapus data SOP laboratorium
11	Mengakses halaman data inventaris laboratorium <i>role admin</i> laboratorium	Mengakses halaman inventaris laboratorium dengan cara klik tab inventaris	Sistem dapat menampilkan halaman inventaris laboratorium beserta data yang ada
		Mengakses tambah inventaris laboratorium dengan cara klik <i>button</i> tambah data	Sistem dapat menampilkan halaman tambah data inventaris laboratorium dan mengirimkannya ke <i>database</i>
		Mengakses lihat data inventaris dengan cara klik	

Lanjutan **Tabel 8** Skenario *Black Box Testing*.

		titik 3 pada kolom aksi kemudian klik view	Sistem dapat menampilkan halaman lihat data inventaris laboratorium
		Mengubah data inventaris dengan cara klik titik 3 pada kolom aksi kemudian klik edit	Sistem dapat menampilkan halaman edit inventaris laboratorium beserta data yang ada
		Mengakses unduh atau cetak inventaris laboratorium dengan cara klik button unduh atau cetak	Sistem dapat mngunduh atau mencetak data inventaris laboratorium
		Menghapus data inventaris dengan cara klik titik 3 pada kolom aksi kemudian klik hapus	Sistem dapat menghapus data inventaris laboratorium
12	Mengakses halaman presensi labratorium tugas akhir mahasiswa S2	Mengakses halaman presensi laboratorium tugas akhir mahasiswa S2	Sistem dapat engakses halaman presensi labratorium tugas akhir mahasiswa S2

V. PENUTUP

5.1 Simpulan

Kesimpulan yang didapatkan dari penelitian ini adalah telah berhasil dikembangkan modul manajemen laboratorium pada sistem informasi chemistry program data center. Modul ini dirancang untuk memudahkan pengelolaan berbagai aspek laboratorium, termasuk inventaris peralatan, hingga penjadwalan penggunaan fasilitas laboratorium. Dengan adanya modul ini, efisiensi operasional laboratorium dapat meningkat, dan pengelolaan data menjadi lebih terstruktur.

Selain itu, pengembangan modul pengelolaan akun juga telah selesai. Modul ini memungkinkan administrasi yang lebih efektif terhadap akun pengguna, termasuk pembuatan, pengeditan, dan penghapusan akun, serta pengaturan hak akses. Kedua modul ini dibangun menggunakan *framework* Laravel 8.

5.2 Saran

Meskipun penelitian ini telah mencapai tujuan yang diharapkan, namun terdapat saran untuk pengembangan selanjuta antara lain:

1. Integrasi dengan sistem informasi akademik Universitas Lampung akan memudahkan super admin untuk tidak lagi perlu memasukkan data NPM secara manual.

2. Melakukan kustomisasi terhadap antarmuka pengguna untuk memudahkan penggantian gambar maupun tema pada situs web jika ada jurusan lain di Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung, yang ingin menerapkan hal yang sama seperti yang dilakukan oleh Jurusan Kimia.

DAFTAR PUSTAKA

- Agustian, H. Y., Finne, L. T., Jørgensen, J. T., Pedersen, M. I., Christiansen, F. V., Gammelgaard, B., & Nielsen, J. A. (2022). Learning outcomes of university chemistry teaching in laboratories: A systematic review of empirical literature
- Arifin, N. Y., Borman, R. I., Ahmad, I., Tyas, S. S., Sulistiani, H., Hardiansyah, A., & Suri, G. P. (2021). Analisa Perancangan Sistem Informasi. (Nama Penerbit).
- Egambaram, O., Hilton, K., Leigh, J., Richardson, R., Sarju, J., Slater, A., & Turner, B. (2022). The future of laboratory chemistry learning, and teaching must be accessible. *Journal of Chemical Education*, 99(12), 3814-3821.
- Farmer, J., Tihanyi, D., & Evans, G. J. 2020. Learning Beyond the Laboratory: A Web Application Framework for Development of Interactive Postlaboratory Exercises Kimia Moozeh. *Journal of Chemical Education*, 97(5), 1481-1486.
- Gallardo-Williams, L. R. (2020). We Should Keep Developing Digital Laboratory Resources in the Postpandemic Era. *Journal of Chemical Education*, 97(9), 2671-2673.
- Lipursari, A. (2013). Konsep Dasar Informasi. Peran Sistem Informasi Manajemen (Sim), 3(2), 1–9. Semarang: Jurnal STIE Semarang.
- Masripah, S., & Ramayanti, L. (2020). Penerapan Pengujian Alpha Dan Beta Pada Aplikasi. *Jurnal Swabumi*, 8(1), 100–105.
- Nasution, Yusuf Ramadhan. (2018). Penerapan Aplikasi Online Angket Persepsi. 3(2), 20–35. Medan: Universitas Islam Negeri Sumatera Utara Medan.
- Pradipta, A., Yusman, M., Shofiana, D. A., & Aristoteles, A. (2021). Perancangan Dan Implementasi Sistem Manajemen Dalam Pengelolaan Data Akademik Berbasis Web di SMA Negeri 1 Liwa. *Jurnal Pepadun*, 2(1), 31–39.
- Prasetya, K. D., Suharjito, & Pratama, D. (2021). Effectiveness Analysis of Distributed Scrum Model Compared to Waterfall approach in Third-

- Party Application Development. *Procedia Computer Science*, 179, 103–111.
- Pressman, R. S. (2020). *Software Engineering: A Practitioner's Approach* (9th ed). McGraw-Hill Education.
- Siking, A., Koniyo, M. H., & Yassin, R. M. T. (2023). Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Pelayanan Pengujian Material Berbasis Web Pada Dinas Pekerjaan Umum dan Penataan Ruang Provinsi Gorontalo. *Journal of System and Information Technology*, 3(2), Juli.
- Stair, R. M., & Reynolds, G. W. (2015). *Principles of Information Systems* (12th ed.). Cengage Learning.
- Suarez, M., Anne, J., Saylor-Miller, K., Mounter, D., & Stanfield, R. (2019). *Design Systems Handbook*. O'Reilly Media.
- Thesinga, T., Feldmann, C., & Burchardt, M. (2021). Agile versus Waterfall Project Management: Decision Model for Selecting the Appropriate Approach to a Project. *Procedia Computer Science*, 181, 746–756.
- Valacich, J. S., & George, J. F. (2017). *Modern systems analysis and design* (edisi ke-8). Pearson.