

**PENGEMBANGAN APLIKASI *DASHBOARD* DATA
PORTOFOLIO DIGITAL MAHASISWA DAN ALUMNI
PROGRAM STUDI TEKNOLOGI INFORMASI**

TUGAS AKHIR

Diajukan oleh:

HADY IKHWAN

NIM. 200705038

**Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi**



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY
BANDA ACEH
2024 M/1446 H**

**PENGEMBANGAN APLIKASI *DASHBOARD* DATA
PORTOFOLIO DIGITAL MAHASISWA DAN ALUMNI
PROGRAM STUDI TEKNOLOGI INFORMASI**

TUGAS AKHIR

Diajukan Kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana (S1)
dalam Ilmu/Prodi Teknologi Informasi

Oleh:
HADY IKHWAN
NIM. 200705038
Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

Disetujui Untuk di Munaqasyahkan Oleh:

Pembimbing I,

Pembimbing II,


Hendri Ahmadian, S.Si., M.I.M
NIP. 198301042014031002


Mulkan Fadhli, M.T
NIP. 198811282020121006

Mengetahui,
Ketua Program Studi Teknologi Informasi



Malahayati, M.T
NIP.198301272015032003

LEMBAR PENGESAHAN

PENGEMBANGAN APLIKASI *DASHBOARD* DATA PORTOFOLIO DIGITAL MAHASISWA DAN ALUMNI PROGRAM STUDI TEKNOLOGI INFORMASI

TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasah Tugas Akhir
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh Dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S-1)
Pada Prodi Teknologi Informasi

Pada Hari/Tanggal : Rabu, 21 Agustus 2024
15 Safar 1446 H

Panitia Ujian Munaqasah Tugas Akhir:

Ketua,



Hendri Ahmadian, S.Si., M.I.M
NIP. 198301042014031002

Sekretaris,



Mulkan Fadhli, M.T
NIP. 198811282020121006

Penguji I,



Khairan AR, M.Kom
NIP. 198607042014031001

Penguji II,

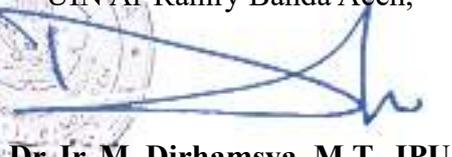


Raihan Islamadina, S.T, M.T
NIP. 198901312020122011

Mengetahui:

Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh,




Dr. Ir. M. Dirhamsya, M.T., IPU
NIP. 196210021988111001

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang Bertanda tanga dibawah ini :

Nama : Hady Ikhwan
NIM : 200705038
Program Studi : Teknologi Informasi
Fakultas : Sains dan Teknologi
Judul Tugas akhir : Pengembangan aplikasi *Dashboard* portofolio digital mahasiswa dan alumni program studi teknologi informasi

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya :

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah tugas akhir orang lain;
3. Tidak menggunakan tugas akhir orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik tugas akhir;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri tugas akhir ini dan mampu bertanggung jawab atas tugas akhir ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas tugas akhir saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang dibuktikan bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 10 Oktober 2024
Yang Menyatakan



(Hady Ikhwan)

ABSTRAK

Nama : Hady Ikhwan
NIM : 200705038
Program Studi : Teknologi Informasi
Judul Tugas akhir : Pengembangan Aplikasi Dashboard Data Portofolio Digital Mahasiswa Dan Alumni Program Studi Teknologi Informasi
Tanggal Sidang : 21 Agustus 2024
Jumlah Halaman : 127 Halaman
Pembimbing I : Hendri Ahmadian, S.Si., M.I.M
Pembimbing II : Mulkan Fadhli, M.T

Perkembangan teknologi digital semakin mendorong kebutuhan akan penyimpanan dan presentasi portofolio mahasiswa dalam format digital. Namun, Program Studi Teknologi Informasi Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh saat ini belum memiliki platform yang terintegrasi untuk mengumpulkan dan menampilkan data mahasiswa dan alumni secara sistematis. Penelitian ini bertujuan untuk mengembangkan sebuah aplikasi dashboard portofolio digital yang dirancang khusus untuk mahasiswa dan alumni Program Studi Teknologi Informasi. *Dashboard* ini diharapkan dapat memfasilitasi mahasiswa dan alumni dalam menyimpan, mengorganisasi, dan menampilkan pencapaian mereka secara terstruktur. Selain itu, aplikasi ini akan dilengkapi dengan fitur verifikasi dan pengawasan oleh pihak program studi, memastikan data yang disajikan sesuai dengan standar akademik yang ditetapkan. Penelitian ini menggunakan metode *Rapid Application Development (RAD)* untuk mempercepat proses pengembangan aplikasi. Hasil dari penelitian ini diharapkan dapat meningkatkan aksesibilitas dan kualitas dalam pengelolaan portofolio digital, serta mendukung pemantauan perkembangan mahasiswa oleh pihak akademik.

Kata Kunci : Portofolio Digital, Mahasiswa dan Alumni, Rapid Application Development (RAD).

ABSTRACT

Name : Hady Ikhwan
NIM : 200705038
Department : Information Technology
Title : Pengembangan Aplikasi Dashboard Data Portofolio Digital Mahasiswa Dan Alumni Program Studi Teknologi Informasi
Date : Agustus 21, 2024
Number of Pages : 127 Pages
Supervisor I : Hendri Ahmadian, S.Si., M.I.M
Supervisor II : Mulkan Fadhli, M.T

The advancement of digital technology increasingly drives the need for digital storage and presentation of student portfolios. However, the Information Technology Program at Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh currently lacks an integrated platform for systematically collecting and displaying student and alumni data. This research aims to develop a digital portfolio dashboard application specifically designed for students and alumni of the Information Technology Program. The dashboard is expected to facilitate students and alumni in storing, organizing, and presenting their achievements in a structured manner. Additionally, the application will include features for verification and monitoring by the program, ensuring that the presented data aligns with established academic standards. This research employs the Rapid Application Development (RAD) method to expedite the application development process. The results of this research are expected to enhance accessibility and quality in digital portfolio management, as well as support academic monitoring of student progress.

Keywords: Digital Portfolio, Students and Alumni, Rapid Application Development (RAD).

KATA PENGANTAR

Bismillahirrahmanirrahim

Segala puji bagi Allah, Tuhan dari semesta alam. Shalawat dan salam semoga selalu tercurah kepada junjungan kita Nabi Muhammad Shallallahu ‘Alaihi Wa sallaam, keluarga dan sahabatnya. Alhamdulillah dengan rahmat Allah yang Maha Rahman dan yang Maha Rahim, sehingga penulis dapat menyelesaikan tugas akhir dengan judul “Pengembangan Aplikasi *Dashboard* Data Portofolio digital Mahasiswa Dan Alumni Program Studi Teknologi Informasi” ini. Tugas akhir ini merupakan salah satu syarat untuk menyelesaikan program studi Strata satu Teknologi Informasi pada Fakultas Sains dan Teknologi di Universitas Islam Negeri Ar-Raniry Banda Aceh.

Ucapan terima kasih penulis sampaikan kepada berbagai pihak yang menjadi sebab dari mereka penulis belajar, mendapatkan ilmu, mendapatkan dukungan, serta mendapatkan hal yang bermanfaat lainnya sehingga penulis sampai pada titik menyelesaikan tugas akhir ini. Terutama dalam konteks ini penulis sampaikan kepada :

1. Bapak Hedri Ahmadian, S.Si., M.I.M dan Bapak Mulkan Fadhli, M.T selaku pembimbing yang selalu bersedia meluangkan waktu dan fikirannya untuk membimbing penulis demi kesempurnaan tugas akhir ini. Ketua dan Sekretaris Program Studi Teknologi Informasi, Malahayati, M.T. Dan Khairan Ar, M.Kom, serta Bapak dan Ibu dosen Program Studi Teknologi Informasi yang telah memberikan ilmu pengetahuan dalam bidang Teknologi Inofrmasi kepada penulis sehingga penulis mampu menyelesaikan tugas akhir ini.
2. Penasehat Akademik, Dr. Ir. M. Dirhamsyah, M.T. yang telah membimbing dan memberikan saran selama masa perkuliahan.
3. Staf Prodi Ibu Cut Ida Rahmadiana S,Si. yang telah membantu membantu penulis dalam hal pengurusan administrasi dan surat-surat untuk keperluan penyelesaian tugas akhir.
4. Teman-teman mahasiswa Prodi Teknologi Informasi dan teman teman penulis di luar perkuliahan yang selalu memberikan dukungan dan

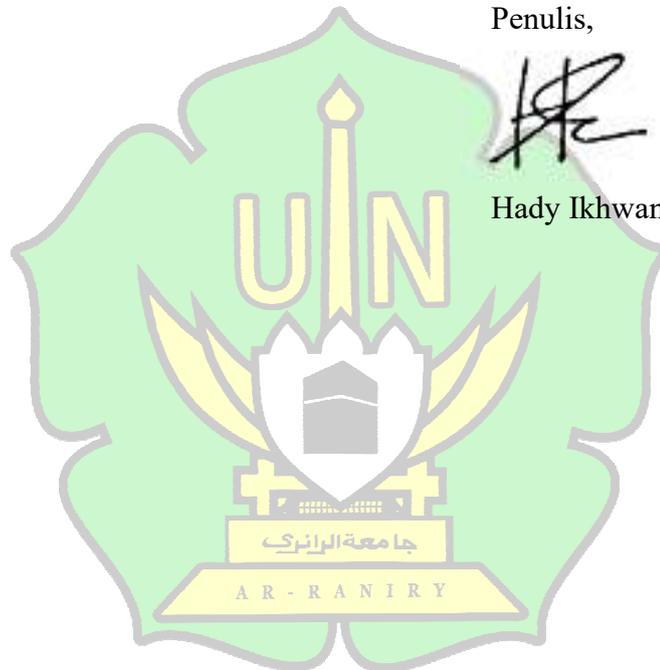
perspektif pemikiran yang membantu peneliti dalam membuat tugas akhir ini.

Akhir kata, penulis menyadari bahwa tugas akhir ini masih jauh dari kesempurnaan, oleh karena itu kritik dan saran yang bersifat membangun sangat penulis harapkan demi mendapatkan hasil yang lebih baik. Semoga perjalanan mempelajari dan berkarya (penyusunan tugas akhir) pada salah satu ilmu milik-Nya ini dapat menghantarkan penulis agar dapat mengenal-Nya dan kekasih-Nya lebih banyak serta mendapatkan ridho dan cinta-Nya yang Maha Rahman dan Rahim.

Banda Aceh, 10 Oktober 2024
Penulis,



Hady Ikhwan



DAFTAR ISI

LEMBAR PERSETUJUAN TUGAS AKHIR	i
LEMBAR PENGESAHAN.....	ii
LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR	iii
ABSTRAK.....	iv
<i>ABSTRACK</i>	v
KATA PENGANTAR	vi
DAFTAR ISI	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian.....	2
1.4. Batasan Masalah.....	3
1.5. Manfaat Penelitian	3
BAB II LANDASAN TEORI.....	4
2.1. <i>Self Branding</i>	4
2.2. Galeri Portofolio Mahasiswa	4
2.3. <i>Tracer study</i>	5
2.4. <i>Head Hunter</i>	5
2.5. <i>Rapid Application Development (RAD)</i>	6
2.6. <i>Unified Modeling Language (UML)</i>	7
2.7. Figma	7
2.8. <i>Website</i>	7
2.9. Bahasa Dart	8
2.10. <i>Framework Flutter</i>	8
2.11. Firebase	9
2.12. Visual Studio Code.....	10
2.13. Penelitian Terdahulu.....	10
BAB III METODE PENELITIAN	13
3.1. Perancangan syarat – syarat	14
3.2. Fase Perancangan	14
3.3. Konstruksi	32

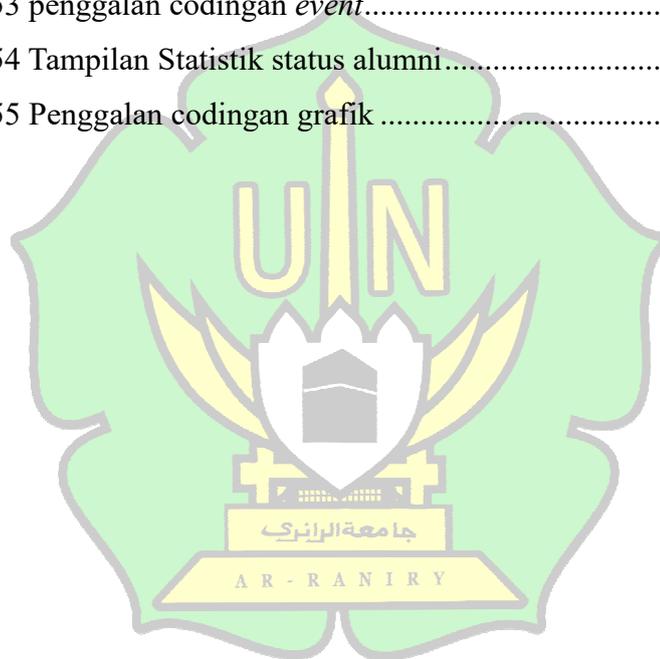
3.4.	Pelaksanaan	32
3.5.	Tempat penelitian	45
BAB IV HASIL DAN PEMBAHASAN		466
4.1	Perancangan	466
4.3.1	Use Case Diagram.....	466
4.3.2	Database Schema	48
4.2	Flowchart User	49
4.3	Konstruksi	766
4.3.1	Fitur Authentication	766
4.3.2	Fitur Search.....	800
4.3.3	Fitur Dashboard admin	811
4.3.4	Fitur Edit Portofolio.....	889
4.3.5	Fitur Portofolio.....	900
4.3.6	Fitur Download CV.....	922
4.3.7	Fitur Pengajuan Tracer study	922
4.3.8	Fitur kuesioner	944
4.3.9	Fitur Event	95
4.3.10	Fitur statistik	97
4.3.11	Fitur Alumni.....	98
4.4	System Testing	99
BAB V KESIMPULAN DAN SARAN		11212
5.1	Kesimpulan	11212
5.2	Saran.....	11212
DAFTAR PUSTAKA.....		11314
LAMPIRAN.....		11515

DAFTAR GAMBAR

Gambar 2. 1 fase RAD	7
Gambar 3. 1 Alur Penelitian	13
Gambar 3. 2 Desain <i>Sign In Screen</i>	24
Gambar 3. 3 Desain <i>Logout</i>	24
Gambar 3. 4 Pencarian Nama.....	25
Gambar 3. 5 Filter Berdasarkan <i>Skill</i>	25
Gambar 3. 6 Filter Berdasarkan <i>Software Skill</i>	26
Gambar 3. 7 <i>Dashboard Tracer study</i>	26
Gambar 3. 8 <i>Dashboard Alumni</i>	27
Gambar 3. 9 <i>Dashboard pertanyaan</i>	27
Gambar 3. 10 <i>Dashboard Grafik Pertanyaan</i>	28
Gambar 3. 11 <i>Dashboard Event</i>	28
Gambar 3. 12 <i>Sign In Screen</i>	29
Gambar 3. 13 <i>Logout</i>	29
Gambar 3. 14 Edit Portofolio	30
Gambar 3. 15 Portofolio.....	30
Gambar 3. 16 <i>CV</i>	31
Gambar 3. 17 Pengajuan <i>Tracer study</i>	32
Gambar 4. 1 <i>Use Case Admin</i>	46
Gambar 4. 2 <i>Use Case Mahasiswa</i>	47
Gambar 4. 3 <i>Use Case Alumni</i>	47
Gambar 4. 4 <i>Database Schema Mahasiswa dan Alumni</i>	48
Gambar 4. 5 <i>Database Schema Admin</i>	49
Gambar 4. 6 <i>Flowchart Login Admin</i>	50
Gambar 4. 7 <i>Flowchart Logout Admin</i>	51
Gambar 4. 8 <i>Flowchart cari nama dan filter</i>	52
Gambar 4. 9 <i>Flowchart Dashboard alumni admni</i>	54
Gambar 4. 10 <i>Flowchart Dashboard pertanyaan</i>	56

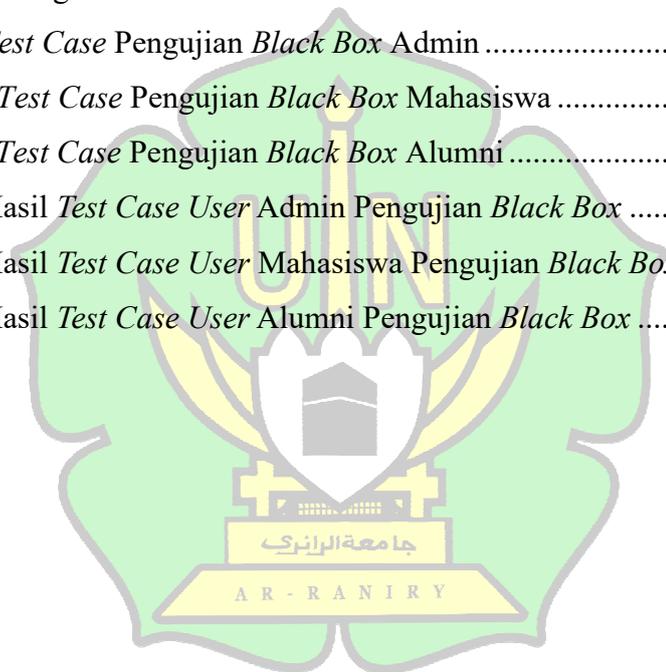
Gambar 4. 11 <i>Flowchart</i> Grafik pertanyaan.....	57
Gambar 4. 12 <i>Flowchart</i> Dashboard Event	59
Gambar 4. 13 <i>Flowchart</i> Dashboard pengajuan	61
Gambar 4. 14 <i>Flowchart</i> SignUp Mahasiswa dan Alumni.....	63
Gambar 4. 15 <i>Flowchart</i> Login Alumni dan mahasiswa.....	64
Gambar 4. 16 <i>Flowchart</i> Logout mahasiswa dan alumni.....	65
Gambar 4. 17 <i>Flowchart</i> Edit portofolio.....	66
Gambar 4. 18 <i>Flowchart</i> Portofolio	68
Gambar 4. 19 Flow download CV	69
Gambar 4. 20 <i>Flowchart</i> pengajuan tracer study.....	71
Gambar 4. 21 <i>Flowchart</i> Kuesioner	72
Gambar 4. 22 <i>Flowchart</i> Event	73
Gambar 4. 23 <i>Flowchart</i> Statistik	74
Gambar 4. 24 <i>Flowchart</i> Alumni	76
Gambar 4. 25 Halaman <i>SignUp</i>	77
Gambar 4. 26 penggalan Codingan <i>SignUp</i>	78
Gambar 4. 27 Halaman <i>Login</i>	78
Gambar 4. 28 Penggalan codingan fitur <i>login</i>	79
Gambar 4. 29 <i>Logout</i>	79
Gambar 4. 30 Codingan <i>Logout</i>	80
Gambar 4. 31 Tampilan <i>Search</i>	80
Gambar 4. 32 Penggalan codingan <i>Search</i>	81
Gambar 4. 33 Tampilan <i>Dashboard</i> alumni	82
Gambar 4. 34 Penggalan codingan <i>Dashboard</i> alumni.....	83
Gambar 4. 35 Tampilan <i>Dashboard</i> pertanyaan.....	83
Gambar 4. 36 Penggalan codingan <i>Dashboard</i> pertanyaan	84
Gambar 4. 37 Tampilan <i>Dashboard</i> grafik pertanyaan	84
Gambar 4. 38 Penggalan codingan <i>Dashboard</i> grafik pertanyaan.....	85
Gambar 4. 39 Tampilan <i>Dashboard</i> Event	86
Gambar 4. 40 Peggalan codingan <i>Event</i>	87
Gambar 4. 41 Tampilan <i>Dashboard</i> pengajuan.....	87
Gambar 4. 42 Penggalan codingan <i>Dashboard</i> pengajuan	88

Gambar 4. 43 Tampilan edit portofolio	89
Gambar 4. 44 Penggalan codingan edit portofolio.....	90
Gambar 4. 45 Tampilan portofolio	90
Gambar 4. 46 Penggalan codingan portofolio.....	91
Gambar 4. 47 Tampilan <i>CV</i>	92
Gambar 4. 48 Tampilan penajuan <i>tracer study</i>	93
Gambar 4. 49 Penggalan codingan pengajuan	94
Gambar 4. 50 Tampilan kuesioner.....	94
Gambar 4. 51 Penggalan codingan kuesioner	95
Gambar 4. 52 tampilan <i>Event</i>	96
Gambar 4. 53 penggalan codingan <i>event</i>	97
Gambar 4. 54 Tampilan Statistik status alumni.....	97
Gambar 4. 55 Penggalan codingan grafik	98



DAFTAR TABEL

Tabel 2. 1 Penelitian terdahulu	11
Tabel 3. 1 Fitur <i>User Admin</i>	15
Tabel 3. 2 Fitur <i>User Mahasiswa</i>	16
Tabel 3. 3 Fitur Alumni	16
Tabel 3. 4 <i>User Story Admin</i>	17
Tabel 3. 5 <i>User Story Mahasiswa</i>	19
Tabel 3. 6 <i>User Story Alumni</i>	21
Tabel 3. 7 Perangkat Lunak.....	22
Tabel 3. 8 Perangkat Keras.....	23
Tabel 3. 9 <i>Test Case Pengujian Black Box Admin</i>	33
Tabel 3. 10 <i>Test Case Pengujian Black Box Mahasiswa</i>	35
Tabel 3. 11 <i>Test Case Pengujian Black Box Alumni</i>	40
Tabel 4. 1 Hasil <i>Test Case User Admin Pengujian Black Box</i>	99
Tabel 4. 2 Hasil <i>Test Case User Mahasiswa Pengujian Black Box</i>	101
Tabel 4. 3 Hasil <i>Test Case User Alumni Pengujian Black Box</i>	106



BAB I

PENDAHULUAN

1.1. Latar Belakang

Di tengah pesatnya perkembangan teknologi digital, dunia pendidikan tinggi semakin mengakui pentingnya penyimpanan dan presentasi portofolio mahasiswa dalam format digital. Tren global menunjukkan peningkatan dalam upaya institusi pendidikan untuk mengarsipkan karya akademik, proyek, dan pencapaian mahasiswa secara digital. Ini menjadi langkah penting dalam mempersiapkan lulusan yang siap bersaing di dunia kerja yang semakin terhubung secara digital.

Namun, Program Studi Teknologi Informasi Universitas UIN Ar-Raniry menghadapi kendala dalam hal ini. Saat ini, belum ada platform yang terintegrasi dan dirancang khusus untuk mengumpulkan dan menampilkan data mengenai mahasiswa dan alumni secara sistematis. Ketiadaan platform yang memadai menyulitkan mahasiswa dalam menunjukkan kompetensi dan pencapaian mereka dengan cara yang terstruktur, sementara pihak akademik kesulitan dalam mengakses data yang dibutuhkan untuk menilai perkembangan mahasiswa selama studi.

Platform profesional seperti LinkedIn memang menawarkan ruang untuk memamerkan profil dan portofolio, namun platform tersebut tidak sepenuhnya sesuai dengan kebutuhan akademik yang spesifik. LinkedIn berfokus pada pasar global dengan cakupan yang luas, sedangkan Program Studi Teknologi Informasi membutuhkan platform yang lebih terarah, yang tidak hanya menampilkan data tetapi juga melalui proses verifikasi dan pengawasan oleh pihak program studi. Hal ini memastikan bahwa semua informasi yang ditampilkan sesuai dengan kurikulum dan standar akademik yang telah ditetapkan.

Penelitian ini mengusulkan pengembangan sebuah *website Dashboard* data portofolio digital yang dirancang khusus untuk mahasiswa dan alumni Program Studi Teknologi Informasi. *Website* ini akan menjadi wadah bagi mahasiswa dan

alumni untuk menyimpan dan menampilkan informasi mengenai pencapaian dan keterampilan mereka, serta dilengkapi dengan fitur untuk verifikasi dan pengawasan oleh pihak program studi. Dengan demikian, data yang disajikan dalam platform ini akan terstruktur dan sesuai dengan standar akademik. Selain itu, *website* ini akan menyediakan alat bagi dosen dan pihak akademik untuk memantau perkembangan dan kualitas mahasiswa sepanjang masa studi mereka.

Melalui pengembangan *website Dashboard* ini, diharapkan mahasiswa dan alumni Program Studi Teknologi Informasi dapat lebih mudah mendokumentasikan dan menunjukkan kompetensi mereka di dunia profesional dengan data yang terverifikasi. Pada saat yang sama, pihak akademik akan mendapatkan alat yang lebih baik untuk mengelola dan menilai portofolio mahasiswa secara menyeluruh. Dalam jangka panjang, platform ini diharapkan dapat memperkuat hubungan antara pendidikan dan industri, serta meningkatkan nilai tambah lulusan di pasar kerja yang semakin kompetitif.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dijabarkan diatas, maka dapat dirumuskan masalahnya sebagai berikut :

1. Fitur apa saja yang diperlukan dalam *website Dashboard* untuk memungkinkan mahasiswa dan alumni menyimpan, mengorganisasi, dan menampilkan data mereka?
2. Bagaimana melakukan pengujian menggunakan *Black Box testing* pada pengembangan aplikasi *Dashboard* data portofolio mahasiswa dan alumni program studi teknologi informasi?

1.3. Tujuan Penelitian

Berdasarkan latar belakang dan rumusan masalah yang telah penulis uraikan, maka tujuan dari penelitian ini adalah sebagai berikut:

1. Menentukan fitur-fitur yang diperlukan dalam *website Dashboard* untuk memungkinkan mahasiswa dan alumni Program Studi Teknologi Informasi menyimpan, mengorganisasi, dan menampilkan data mereka.

2. Melakukan pengujian *Dashboard* data portofolio mahasiswa dan alumni menggunakan *Black Box testing*.

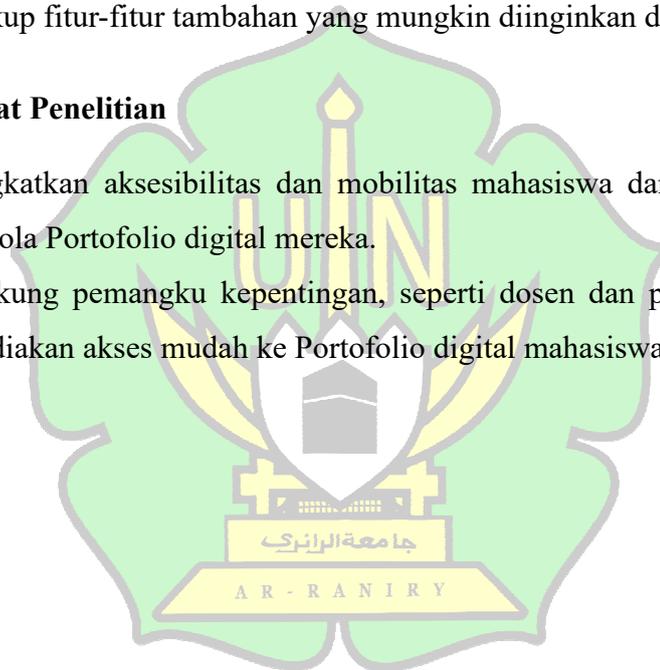
1.4. Batasan Masalah

Berdasarkan latar belakang, rumusan masalah dan tujuan penelitian yang telah penulis uraikan, maka manfaat dari penelitian ini yaitu:

1. Bahasa pemrograman yang digunakan untuk membangun *website* ini adalah Dart dengan *Framework Flutter*.
2. Pengembangan sistem informasi ini akan dibatasi oleh lingkup fungsionalitas yang telah ditentukan dalam rumusan masalah dan tidak akan mencakup fitur-fitur tambahan yang mungkin diinginkan di masa depan.

1.5. Manfaat Penelitian

1. Meningkatkan aksesibilitas dan mobilitas mahasiswa dan alumni dalam mengelola Portofolio digital mereka.
2. Mendukung pemangku kepentingan, seperti dosen dan perekrut, dengan menyediakan akses mudah ke Portofolio digital mahasiswa dan alumni.



BAB II

LANDASAN TEORI

2.1. *Self Branding*

Self-Branding adalah proses strategis di mana individu membangun dan mempromosikan citra diri mereka untuk mencapai tujuan pribadi dan profesional. Konsep ini mengadaptasi prinsip-prinsip branding dari pemasaran produk dan perusahaan untuk diterapkan pada individu. Dalam era digital saat ini, *Self-Branding* menjadi kunci penting bagi mahasiswa dan alumni yang memasuki pasar kerja yang kompetitif. Platform digital seperti media sosial dan *website* portofolio memberikan kesempatan bagi individu untuk menonjol dengan menekankan keahlian, pencapaian, dan nilai-nilai unik mereka.

Pengembangan *website Dashboard* portofolio mahasiswa dan alumni mendukung *Self-Branding* dengan menyediakan sarana untuk mendokumentasikan dan menampilkan pencapaian serta keterampilan secara terstruktur dan terverifikasi. *Website* ini tidak hanya berfungsi sebagai alat dokumentasi tetapi juga sebagai platform untuk membangun brand pribadi secara profesional. Dengan fitur-fitur yang memfasilitasi pencatatan dan verifikasi, mahasiswa dan alumni dapat mempresentasikan brand mereka dengan lebih efektif, meningkatkan daya tarik mereka di mata calon pemberi kerja dan pihak-pihak lain yang relevan.

2.2. Galeri Portofolio Mahasiswa

Galeri portofolio mahasiswa merupakan platform yang berfungsi sebagai wadah untuk menyimpan dan menampilkan berbagai pencapaian dan keterampilan yang telah diperoleh selama masa studi. Dalam konteks pendidikan tinggi, galeri ini memegang peranan penting dalam mendokumentasikan prestasi akademik dan keterampilan praktis mahasiswa secara terstruktur.

Galeri portofolio mahasiswa juga memberikan manfaat tambahan dengan menyediakan data yang terpusat dan terorganisir mengenai perkembangan akademik dan profesional mahasiswa. Hal ini tidak hanya membantu mahasiswa dalam membangun dan mengelola citra profesional mereka, tetapi juga

mempermudah pihak akademik dalam melakukan pemantauan dan evaluasi terhadap kemajuan dan pencapaian mahasiswa.

2.3. *Tracer study*

Tracer study adalah metode untuk mengikuti jejak karier lulusan setelah mereka lulus dari perguruan tinggi. Studi ini mengumpulkan informasi tentang pekerjaan, pengalaman kerja, dan seberapa relevan pendidikan yang diterima dengan dunia kerja. Tujuannya adalah untuk memberi tahu institusi pendidikan seberapa baik kurikulum mereka mempersiapkan lulusan untuk bekerja dan apakah materi yang diajarkan masih sesuai dengan kebutuhan industri.

Website Dashboard portofolio digital bisa sangat membantu dalam *tracer study*. Dengan menyediakan tempat bagi mahasiswa dan alumni untuk menyimpan informasi tentang pencapaian dan keterampilan mereka, *website* ini memudahkan pengumpulan data untuk studi tersebut. Data yang terstruktur dari *Dashboard* ini akan membantu perguruan tinggi melacak perkembangan karier lulusan dan memperbaiki program studi jika diperlukan, sambil memberi lulusan cara yang efektif untuk menunjukkan kompetensi mereka kepada calon pemberi kerja.

2.4. *Head Hunter*

Head hunter adalah orang yang membantu perusahaan menemukan dan merekrut kandidat untuk posisi penting atau spesifik. Mereka bekerja dengan perusahaan untuk memahami jenis keterampilan dan pengalaman yang dibutuhkan untuk sebuah posisi, lalu mencari kandidat yang cocok melalui jaringan profesional dan sumber lainnya. Peran mereka sangat penting untuk memastikan perusahaan mendapatkan orang yang tepat untuk pekerjaan tersebut, terutama untuk posisi yang sulit ditemukan atau sangat khusus.

Website Dashboard portofolio digital diharapkan bisa membantu bagi head hunter. Dengan platform ini, mereka bisa mengakses informasi terstruktur tentang pencapaian dan keterampilan mahasiswa dan alumni. Ini memudahkan mereka untuk menemukan kandidat yang cocok dengan kriteria yang mereka cari. Jadi, head hunter dapat lebih cepat dan mudah menemukan orang yang tepat untuk posisi

yang mereka tawarkan, karena informasi yang ada di *website* ini sudah terverifikasi dan teratur.

2.5. *Rapid Application Development (RAD)*

Pentingnya model RAD terletak pada kemampuannya untuk mengatasi perbedaan pemahaman antara pengembang perangkat lunak dan pelanggan. Sebagian besar pelanggan mungkin kurang memahami aspek teknis yang kompleks dalam pengembangan perangkat lunak. Model RAD hadir sebagai alat yang efektif dalam menjembatani kesenjangan ini. Dengan menggunakan RAD, pengembang perangkat lunak dapat berkomunikasi lebih efisien dengan pelanggan dan menjelaskan secara rinci spesifikasi teknis yang diperlukan (Pricillia, 2021). Hal ini memungkinkan pelanggan untuk lebih jelas dalam menyampaikan kebutuhan mereka kepada pengembang perangkat lunak.

Dengan demikian, RAD bukan hanya sekadar sebuah model pengembangan yang cepat dan adaptif, tetapi juga merupakan alat komunikasi yang efektif dalam menerjemahkan kebutuhan pelanggan ke dalam solusi perangkat lunak yang akurat dan memuaskan. Menurut Martin (Kendall & Kendall, 2003:239), fase dalam RAD dibagi menjadi empat, yaitu :

1. Fase Perencanaan Syarat-syarat

Pada fase ini pengguna memutuskan fungsi apa saja yang harus difiturkan oleh aplikasi tersebut.

2. Fase Desain Pengguna

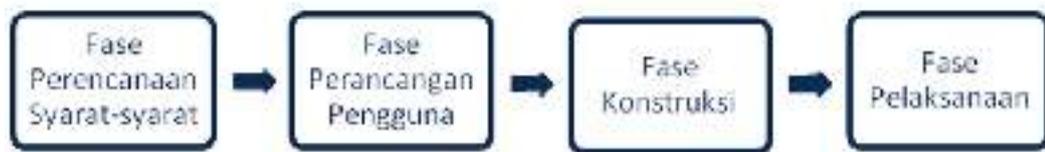
Pada fase ini pengguna diminta membahas aspek-aspek desain Non -teknis dari sistem dengan bimbingan penganalisis. Karena tingginya sifat interaktif, fase ini sering digabungkan dengan fase konstruksi pada workshop desain RAD.

3. Fase Konstruksi

Pada fase ini setiap desain yang diciptakan dalam fase sebelumnya selanjutnya ditingkatkan untuk dilakukan pengkodean sistem. Kemudian setelah tahap ini selesai dilakukan uji kemampuan untuk mendapatkan komentar, dan revisi dari pengguna tingkat tinggi.

4. Fase Pelaksanaan

Tahap terakhir adalah dimana *website* baru di uji coba dan pengenalan terhadap *website* tersebut (Wijaya, 2019).



Gambar 2. 1 fase RAD

2.6. *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah alat yang digunakan untuk menggambarkan dan mendokumentasikan hasil analisis serta desain sistem secara visual. UML juga merupakan kumpulan konvensi pemodelan yang dirancang untuk merepresentasikan sistem perangkat lunak yang berfokus pada objek. Sebagai bahasa pemodelan untuk sistem atau perangkat lunak yang berbasis paradigma berorientasi objek, UML meliputi konsep-konsep dasar seperti klasifikasi struktural, perilaku dinamis, dan manajemen model. Konsep-konsep ini penting saat membuat diagram, yang dikategorikan dalam berbagai *view*. UML mencakup berbagai jenis diagram, termasuk diagram *Use Case*, diagram kelas, diagram *statechart*, diagram aktivitas, diagram urutan, diagram kolaborasi, diagram komponen, dan diagram *deployment* (Ronal et al., 2022).

2.7. *Figma*

Figma adalah aplikasi desain berbasis web yang kompatibel dengan sistem operasi Windows, Linux, dan Mac. Selain untuk merancang antarmuka (UI) dan prototipe, Figma juga menyertakan alat untuk memfasilitasi kolaborasi tim dan memberikan komentar (Vallendito, 2020).

2.8. *Website*

Menurut Hamzah Hartono pada skripsi (Pangestu, 2022) *Website* adalah sekumpulan halaman web dan berkas terkait, termasuk gambar, video, dan

dokumen digital lainnya, yang tersimpan di *server* web. Biasanya, konten tersebut dapat diakses melalui internet. Dalam pengertian yang lebih luas, *website* adalah kumpulan dokumen dan data yang terkelola dengan berbagai fungsi, termasuk kemampuan untuk menampilkan informasi dan menyediakan layanan tertentu.

Sementara itu, menurut Rajak & Muharto pada skripsi (Pangestu, 2022), *Website* adalah sebuah sistem yang mencakup berbagai halaman yang saling terhubung dan dapat diakses melalui berbagai perangkat, termasuk komputer, ponsel pintar, dan berbagai sistem operasi. *Website* juga dapat dilihat sebagai alamat atau *uniform resource locator (URL)* yang menyimpan informasi dan data, yang dapat diakses oleh pengguna dari berbagai lokasi.

Kesimpulannya, *website* adalah sebuah entitas digital yang terdiri dari serangkaian halaman yang terhubung dan dapat diakses melalui berbagai perangkat serta sistem operasi, menawarkan akses ke beragam informasi dan data.

2.9. Bahasa Dart

Dart adalah bahasa pemrograman open-source yang terstruktur dan dirancang untuk memfasilitasi pembuatan aplikasi web kompleks berbasis *browser*. Dengan Dart, pengguna dapat menjalankan aplikasi langsung di *browser* yang mendukung Dart atau mengonversi kode Dart menjadi JavaScript untuk kompatibilitas yang lebih luas. (Sofi & Dharmawan, n.d.). Dart menawarkan fitur utama seperti sintaksis yang mirip dengan bahasa pemrograman lain, berbasis kelas, dan opsi tipe data, mempermudah penggunaannya. Meskipun Dart adalah bahasa yang berjalan dalam satu thread, ia memiliki model konkurensi yang disebut "*isolates*" untuk mendukung eksekusi paralel.

Selain opsi menjalankan kode Dart di *browser* atau mengonversinya menjadi JavaScript, pengguna juga dapat menjalankan kode Dart secara langsung melalui baris perintah. Mesin virtual Dart memungkinkan pengembangan aplikasi di sisi *klien* dan *server* dengan bahasa yang sama. Dengan sintaksis yang mirip dengan bahasa seperti Java, C#, dan JavaScript, Dart dirancang agar mudah dipahami oleh para pengembang.

2.10. Framework Flutter

Flutter adalah kit pengembangan perangkat lunak (SDK) yang dikembangkan oleh Google untuk membangun aplikasi seluler. SDK ini, yang bersifat open-source, memfasilitasi pembuatan aplikasi yang dapat dijalankan di platform Android dan iOS, serta mendukung sistem operasi Google Fuchsia. Flutter ditulis dalam bahasa C, C++, dan Dart, dan menggunakan Skia Graphics Engine sebagai dasar untuk rendering grafis.

Salah satu fitur unggulan Flutter adalah kemampuannya untuk menyediakan berbagai widget yang dapat disesuaikan sepenuhnya, memungkinkan pengembang menciptakan antarmuka pengguna yang sangat asli. Flutter juga menawarkan akses ke perpustakaan Desain Material dari Google, yang dikenal dengan desainnya yang elegan, serta widget Cupertino yang dirancang untuk menciptakan antarmuka yang sesuai dengan tampilan iOS. Dengan Flutter, pengembang dapat merancang dan membangun antarmuka dengan cepat, memastikan bahwa tampilan aplikasi tetap konsisten di seluruh emulator, simulator, dan perangkat fisik pada platform iOS dan Android. Ini membuat proses pengembangan aplikasi yang responsif dan menarik menjadi lebih efisien dan efektif.

2.11. Firebase

Firebase adalah perusahaan yang berfokus pada layanan *cloud* dan *backend* yang terletak di San Francisco, California. Didirikan oleh Andrew Lee dan James Tamplin pada tahun 2011, Firebase pertama kali diluncurkan pada tahun 2012 dengan menawarkan *database cloud real-time* secara gratis. Produk unggulan Firebase adalah *database* yang memiliki API untuk memudahkan pengembang dalam menyimpan dan menyinkronkan data di berbagai *klien*.

Firebase adalah platform *backend* yang menyediakan solusi untuk layanan *database* dan sinkronisasi data secara *real-time*. Ini memungkinkan pengembang untuk membuat API yang dapat diintegrasikan dengan berbagai *klien* serta menyimpan data secara *cloud*. Firebase juga menawarkan fitur keamanan untuk melindungi *database* menggunakan *server* dan aturan khusus. Selain itu, platform ini mendukung *hosting file* statis dengan kemampuan distribusi melalui CDN dan SNL (Sonita & Fardianitama, 2018).

2.12. Visual Studio Code

Visual Studio Code (VS Code) adalah editor teks yang sangat dikenal, dikembangkan oleh Microsoft. Salah satu keunggulan utama dari VS Code adalah kemampuannya sebagai editor yang efisien dan ringan, yang kompatibel dengan berbagai sistem operasi seperti Linux, Mac, dan Windows. Dengan dukungan lintas platform, VS Code mempermudah pengembang dari berbagai latar belakang untuk menggunakan alat ini. Meskipun VS Code mendukung bahasa pemrograman seperti JavaScript, Typescript, dan Node.js secara langsung, keunggulannya terletak pada kemampuannya untuk memperluas dukungannya ke berbagai bahasa pemrograman lainnya melalui plugin yang tersedia di *marketplace*-nya. Ini memungkinkan pengguna untuk menyesuaikan pengalaman pengkodean mereka dengan lebih fleksibel sesuai dengan kebutuhan dan preferensi mereka.

VS Code menawarkan berbagai fitur yang memudahkan pengembangan perangkat lunak, seperti Intellisense yang memberikan saran kode otomatis, integrasi Git untuk manajemen versi, dan kemampuan debugging yang kuat. Selain itu, pengguna dapat memperluas fungsionalitas VS Code dengan menambahkan ekstensi sesuai kebutuhan proyek mereka. Microsoft juga secara rutin merilis pembaruan bulanan, menjaga agar editor ini selalu terbaru dengan fitur terbaru, perbaikan, dan peningkatan performa, menjadikannya lebih unggul dibandingkan *teks editor* lain yang mungkin jarang mendapatkan pembaruan.

VS Code adalah editor kode sumber yang bersifat *open source*, memungkinkan pengembang global untuk memeriksa dan berkontribusi terhadap pengembangannya melalui GitHub. Keterbukaan ini memberikan kesempatan bagi komunitas untuk ikut serta dalam peningkatan dan perbaikan, menjadikan VS Code sebagai pilihan utama di kalangan pengembang. Partisipasi komunitas memastikan alat ini tetap *up-to-date* dan sesuai dengan kebutuhan pengguna, menjaga relevansi dan kualitasnya dalam industri perangkat lunak. (Costa Pinarria et al., n.d.).

2.13. Penelitian Terdahulu

Dibutuhkannya referensi atau penelitian terkait bagi peneliti yang ingin menulis guna untuk menghindari duplikasi dan pelanggaran hak cipta. Penelitian terdahulu berfungsi untuk bahan tolak ukur dan acuan. Tentunya untuk melihat

perbedaan dengan penelitian ini, maka dalam landasan teori ini penulis mencantumkan hasil-hasil penelitian terdahulu sebagai berikut:

Tabel 2. 1 Penelitian terdahulu

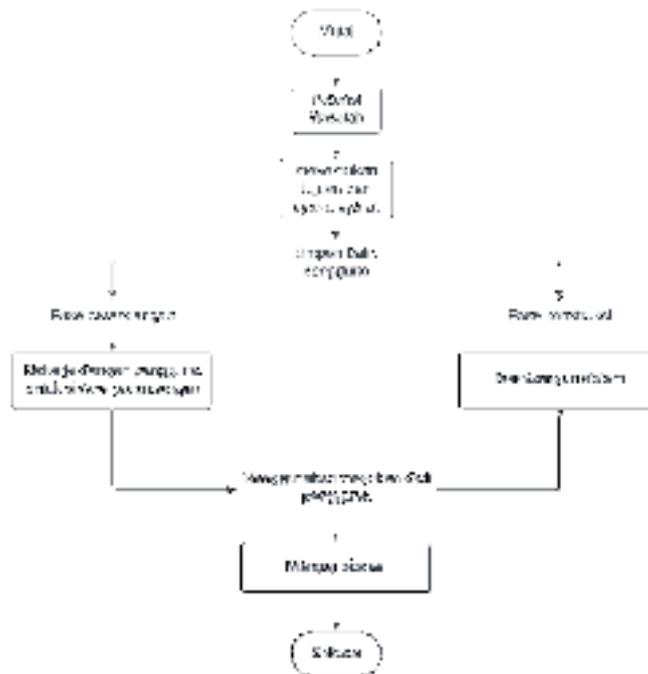
No	Nama Peneliti	Judul	Tahun	Deskripsi
1.	(Fernando, 2018).	Rancang Bangun Aplikasi Portofolio digital Hasil Karya Mahasiswa Unsera Menggunakan Metode Scrum	2018	Mengumpulkan informasi produk atau hasil karya mahasiswa UNSERA baik itu kategori mahasiswa aktif, alumni, Unit Kegiatan Mahasiswa (UKM) dan Himpunan Mahasiswa Jurusan. Diharapkan dengan adanya aplikasi Portofolio digital hasil karya mahasiswa Unsera dapat bermanfaat bagi beberapa pihak baik pihak kampus, mahasiswa dan masyarakat umum. Model pengembangan <i>website</i> digunakan Scrum dan menggunakan bahasa pemrograman PHP untuk pembangunan aplikasi <i>website</i> dan MySQL untuk pengelolaan <i>database</i> pada sisi <i>server</i> , serta dalam perancangannya menggunakan UML dan template <i>website</i> HTML.

No	Nama Peneliti	Judul	Tahun	Deskripsi
2.	(Putra, 2018).	“Rancang Bangun Sistem Informasi Portofolio digital Dosen”	2018	Portofolio yang dimaksudkan pada penelitian ini adalah catatan kegiatan yang diikuti maupun produk yang telah dikerjakan oleh dosen yang kemudian sistem mengolahnya menjadi sebuah portofolio elektronik. Pada pengembangan sistemnya digunakan MySQL sebagai <i>database</i> manajemen sistemnya, dan <i>Framework</i> Laravel untuk mengembangkan antarmuka dan sistem transaksinya.
3.	(Saputri et al., 2022).	Pembangunan Sistem Informasi Galeri Portofolio Mahasiswa Arsitektur Universitas Atma Jaya Yogyakarta Berbasis <i>Website</i>	2022	Membangun sistem informasi yang dapat membantu dan memudahkan mahasiswa memiliki portofolio dalam satu media terpusat yang dapat menangani berbagai tipe media agar memudahkan mahasiswa dalam mengategorisasikan hasil karyanya dan departemen arsitektur dapat menunjukkan kualitas pendidikannya kepada masyarakat umum melalui kumpulan portofolio mahasiswa yang dipamerkan dalam bentuk galeri.

BAB III

METODE PENELITIAN

Metode yang digunakan pada rancang bangun *website Dashboard* portofolio mahasiswa dan alumni adalah RAD (*Rapid Application Development*). RAD memungkinkan untuk mengelola risiko dengan lebih baik. Melalui iterasi dan pengujian yang sering, sehingga dapat mengidentifikasi dan memperbaiki masalah lebih awal dalam siklus pengembangan. Ini berarti bahwa jika ada fitur yang tidak berfungsi sesuai harapan atau ada kebutuhan yang berubah, pengembang dapat dengan cepat menyesuaikan dan memperbaikinya tanpa harus menunggu hingga tahap akhir proyek. Fleksibilitas ini memberikan keuntungan besar dalam proyek yang mungkin memiliki kebutuhan pengguna yang belum sepenuhnya jelas di awal, seperti pengembangan aplikasi *Dashboard* yang mungkin memerlukan penyesuaian berdasarkan umpan balik pengguna. Tahapan penelitian dapat dilihat pada gambar 3.1



Gambar 3. 1 Alur Penelitian

3.1. Perancangan syarat – syarat

Perencanaan Syarat-Syarat Pada tahap ini dilakukan pengidentifikasian tujuan-tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. Secara terperinci yang penulis lakukan pada tahap tujuan dan syarat-syarat informasi adalah:

- a) Menganalisa sistem yang berjalan
- b) Mengidentifikasi permasalahan yang ada
- c) Memberikan solusi permasalahan yang dihadapi

Hasil yang penulis dapatkan dari tahap Tujuan dan Syarat- syarat Informasi adalah:

- a) Mendapatkan pemahaman tentang bagaimana mahasiswa saat ini menyusun dan mengelola portofolio mereka, baik melalui alat digital yang sudah ada atau proses manual jika ada.
- b) Mengetahui bahwa sistem yang lebih efisien dan terotomatisasi dibutuhkan untuk mengatasi permasalahan ini.
- c) Menyimpulkan bahwa solusi yang diperlukan adalah perancangan dan pengembangan *website* menggunakan *Framework* Flutter yang dapat membantu mahasiswa mencatat, mengorganisir, dan menyimpan portofolio mereka secara *online*.

3.2. Fase Perancangan

Pada tahap ini, dilakukan perancangan proses yang merujuk pada pemodelan dan penjelasan rinci mengenai berbagai proses yang akan terjadi di dalam sistem. Ini mencakup langkah-langkah, alur kerja, interaksi antara komponen, serta cara data akan diproses dan mengalir melalui sistem secara efisien.

1. Kebutuhan Fungsional

Kebutuhan fungsional menggambarkan fungsi dan fitur yang harus ada pada *website* yang akan dikembangkan. Tabel 3.1 menjelaskan fitur-fitur yang akan disediakan dalam *Dashboard* data portofolio mahasiswa dan alumni, yang dibagi menjadi tiga kategori pengguna: admin, mahasiswa, dan alumni. Setiap kategori

memiliki akses yang berbeda; fitur-fitur untuk admin tidak dapat diakses oleh mahasiswa dan sebaliknya. Tabel berikut menyajikan fitur-fitur yang tersedia untuk pengguna admin.

Tabel 3. 1 Ftur *User Admin*

No	Fitur	Deskripsi
1.	<i>Authentication</i>	Fitur ini akan menangani kebutuhan proses otentikasi admin seperti <i>login</i> dan <i>Logout</i> .
2.	Pencarian portofolio mahasiswa	Fitur ini akan menampilkan pencarian seluruh mahasiswa. Tersedia juga fitur filter untuk melihat portofolio mahasiswa berdasarkan <i>softskill</i> dan <i>software skill</i>
3.	<i>Dashboard Tracer study</i>	Fitur ini memperlihatkan beragam informasi terkait alumni seperti : jumlah alumni, jumlah pertanyaan, jumlah <i>Event</i> dan status alumni.
4.	<i>Dashboard alumni</i>	Fitur ini memperlihatkan informasi terkait alumni, baik itu nama, nim, Alamat, dll. Juga terdapat fitur untuk melihat portofolio keseluruhan dari alumni.
5.	<i>Dashboard pertanyaan</i>	Fitur ini untuk menambah pertanyaan – pertanyaan yang ingin ditanyakan kepada para alumni pada saat mereka mengisi kuesiner <i>tracer study</i> .
6.	<i>Dashboard grafik</i>	Fitur ini akan menampilkan seluruh grafik dari setiap point pertanyaan dalam kuesioner dari para alumni.
7.	<i>Dashboard Event</i>	Fitur ini untuk menambah informasi terkait <i>Event</i> atau kegiatan yang diperuntukan untuk para alumni

No	Fitur	Deskripsi
8.	<i>Dashboard</i> pengajuan <i>Tracer study</i>	Fitur ini untuk checking berkas dan <i>approvalment</i> untuk para alumni Ketika mereka mengajukan <i>tracer study</i> .

Sumber: Data diolah (2023)

Berikut merupakan fitur-fitur *User* mahasiswa yang dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Fitur *User* Mahasiswa

No	Fitur	Deskripsi
1.	<i>Authentication</i>	Fitur ini akan menangani kebutuhan proses otentikasi admin seperti <i>login</i> dan <i>Logout</i> .
2.	Edit portofolio	Fitur ini terdiri dari <i>7 step</i> yang mana seluruhnya dibutuhkan untuk seluruh informasi terkait denan portofolio mahasiswa.
3.	Portofolio	Fitur ini untuk melihat <i>Result</i> dari hasil seluruh portofolio mahasiswa yang telah diisi pada <i>step</i> edit portofolio
	<i>Download CV</i>	Fitur ini untuk <i>Generate</i> otomatis <i>CV</i> berbentuk <i>file</i> PDF yang mana data tersebut didapatkan dari seluruh informasi yang ada didalam portofolio mahasiswa
	<i>Tracer study</i>	Fitur ini diperuntukan bagi para mahasiswa aktif yang telah berubah status menjadi alumni, dengan mengupload berkas yang mendukung bahwa mereka telah lulus.

Sumber: Data diolah (2023)

Berikut merupakan fitur-fitur *User* alumni yang dapat dilihat pada Tabel 3.3

Tabel 3. 3 Fitur Alumni

No	Fitur	Deskripsi
----	-------	-----------

1.	<i>Authentication</i>	Fitur ini akan menangani kebutuhan proses otentikasi admin seperti <i>login</i> dan <i>Logout</i> .
2.	Edit portofolio	Fitur ini terdiri dari 7 <i>step</i> yang mana seluruhnya dibutuhkan untuk seluruh informasi terkait dengan portofolio alumni.
3.	Portofolio	Fitur ini untuk melihat <i>Result</i> dari hasil seluruh portofolio alumni yang telah diisi pada <i>step</i> edit portofolio
4.	<i>Download CV</i>	Fitur ini untuk <i>Generate</i> otomatis <i>CV</i> berbentuk <i>file</i> PDF yang mana data tersebut didapatkan dari seluruh informasi yang ada didalam portofolio alumni
5.	Kuesioner	Fitur ini untuk para alumni mengisi berbagai <i>form</i> pertanyaan untuk kebutuhan pihak akademik terkait <i>tracer study</i> .
6.	<i>Event</i>	Fitur ini berisikan informasi terkait tentang <i>Event</i> atau acara dan kegiatan yang diperuntukan untuk alumni
	alumni	Fitur ini berisikan grafik informasi terkait tentang Gambaran umum tentang status seluruh alumni yang ada di prdi TI
	<i>Profile</i>	Fitur ini berisikan menu edit yang terkait dengan ganti <i>email</i> , ganti <i>Password</i> , dll.

Berikut *User stories* dari kebutuhan fitur *User* admin yang telah di kategorikan dapat dilihat pada Tabel 3.5.

Tabel 3. 4 *User Story* Admin

No	Fitur	Judul <i>User Story</i>	<i>User Story</i>
1.	<i>Authentication</i>	<i>Sign In</i>	Sebagai admin yang terdaftar, saya ingin <i>login</i> ke aplikasi web dengan akun yang terdaftar, sehingga saya bisa akses fitur admin.
		<i>Logout</i>	Sebagai admin yang sudah <i>login</i> , saya ingin <i>Logout</i> dari aplikasi web.

No	Fitur	Judul <i>User Story</i>	<i>User Story</i>
2.	Pencarian portofolio mahasiswa	Cari nama	Sebagai admin saya ingin mencari portofolio mahasiswa berdasarkan nama
		Filter <i>softskill</i>	Sebagai admin saya ingin filter pencarian mahasiswa dengan menerapkan kriteria <i>softskill</i> tertentu
		Filter <i>software skill</i>	Sebagai admin saya ingin filter pencarian mahasiswa dengan menerapkan kriteria <i>software skill</i> tertentu
3.	<i>Dashboard Tracer study</i>	Jumlah alumni	Sebagai admin saya ingin melihat total jumlah alumni yang sudah terdaftar didalam sistem.
		Jumlah pertanyaan	Sebagai admin saya ingin melihat total jumlah pertanyaan kuesioner
		Jumlah <i>Event</i>	Sebagai admin saya ingin melihat total jumlah <i>Event</i> .
		Grafik status alumni	Sebagai admin saya ingin melihat grafik status alumni
4.	<i>Dashboard Alumni</i>	Data alumni	Sebagai admin saya melihat seluruh data dari alumni.
		Portofolio alumni	Sebagai admin saya bisa mengakses seluruh portofolio alumni secara lengkap
		hapus	Sebagai admin saya bisa menghapus accout alumni yang terdaftar didalam system.
5.	<i>Dashboard pertanyaan</i>	Tambah pertanyaan	Sebagai admin saya dapat menambah pertanyaan kuesioner untuk para alumni untuk kebutuhan <i>tracer study</i> .
		Hapus pertanyaan	Sebagai admin saya dapat menghapus pertanyaan kuesioner untuk para alumni.

No	Fitur	Judul <i>User Story</i>	<i>User Story</i>
6.	Dashboar Grafik	Eksplor seluruh jawaban kuesioner	Sebagai admin saya ingin melihat seluruh jawaban dari kuesioner yang telah di input pada <i>step</i> sebelumnya.
7.	<i>Dashboard Event</i>	Tambah <i>Event</i>	Sebagai admin saya dapat menambah <i>Event</i> untuk para alumni.
		Hapus <i>Event</i>	Sebagai admin saya dapat menghapus <i>Event</i> untuk para alumni.
8.	Pengajuan <i>tracer study</i>	Data pengajuan	Sebagai admin saya dapat melihat seluruh pengajuan untuk <i>tracer study</i> dari para alumni.
		Lihat <i>file</i>	Sebagai admin saya dapat melihat <i>file</i> atau berkas yang diajukan oleh alumni sebagai prasyarat agar bisa merubah status dari mahasiswa menjadi alumni.
		approve	Sebagai admin saya dapat <i>approve</i> pengajuan <i>tracer study</i> dengan prasyarat jikalau berkas yang di ajukan telah di validasi.
		hapus	Sebagai admin saya dapat hapus pengajuan <i>tracer study</i> dengan prasyarat jikalau berkas yang di ajukan tidak tervalidasi.

Sumber: Data diolah (2023)

Berikut *User stories* dari kebutuhan fitur *User* mahasiswa yang telah di kategorikan dapat dilihat pada Tabel 3.5.

Tabel 3. 5 *User Story* Mahasiswa

No	Fitur	Judul <i>User Story</i>	<i>User Story</i>
1.	<i>Authenticatio n</i>	<i>Sign In</i>	Sebagai mahasiswa saya dapat membuat <i>account</i> baru.
		Log In	Sebagai mahasiswa saya dapat masuk kedalam system dengan

			<i>account</i> yang telah dibuat pada <i>step Sign In</i> sebelumnya.
2.	Edit portofolio	<i>Profile info</i>	Sebagai mahasiswa saya dapat mengisi terkait informais pribadi secara umum.
		<i>Achievement</i>	Sebagai mahasiswa saya dapat menambahkan <i>achievement</i> diri saya
		<i>Experience</i>	Sebagai mahasiswa saya dapat menambahkan <i>experience</i> diri saya
		<i>Personal skill</i>	Sebagai mahasiswa saya dapat menambahkan <i>personal skill</i> diri saya
		<i>Softskill</i>	Sebagai mahasiswa saya dapat menambahkan <i>Softskill</i>
		<i>Software skill</i>	Sebagai mahasiswa saya dapat menambahkan <i>Software skill</i>
		Nilai peminatan	Sebagai mahasiswa saya dapat mengisi seluruh nilai mata kuliah yang tersedia untuk melihat <i>statistic</i> dari peminatan saya.
3.	Portofolio	<i>Show portofolio</i>	Sebagai mahasiswa saya dapat melihat dari keseluruhan dari hasil edit portofolio sebelumnya.
4.	<i>Download CV</i>	<i>Download CV</i>	Sebagai mahasiswa saya dapat <i>mendownload CV</i> dari hasil seluruh data yang telah saya isi pada berbagai <i>form</i> untuk portofolio.
5.	<i>Tracer study</i>	telah lulus	Sebagai mahasiswa saya dapat mengajukan diri untuk merubah status menajdi seorang alumni jika saya telah dinyatakan lulus.

Sumber: Data diolah (2023)

Berikut *User stories* dari kebutuhan fitur *User* alumni yang telah di kategorikan dapat dilihat pada Tabel 3.6

Tabel 3. 6 *User Story* Alumni

No	Fitur	Judul <i>User Story</i>	<i>User Story</i>
1.	<i>Authenticatio n</i>	<i>Sign In</i>	Sebagai alumni saya dapat membuat <i>account</i> baru.
		Log In	Sebagai alumni saya dapat masuk kedalam system dengan <i>account</i> yang telah dibuat pada <i>step Sign In</i> sebelumnya.
2.	Edit portofolio	<i>Profile info</i>	Sebagai alumni saya dapat mengisi terkait informais pribadi secara umum.
		<i>Achievement</i>	Sebagai alumni saya dapat menambahkan <i>achievement</i> diri saya
		<i>Experience</i>	Sebagai alumni saya dapat menambahkan <i>experience</i> diri saya
		<i>Personal skill</i>	Sebagai alumni saya dapat menambahkan <i>personal skill</i> diri saya
		<i>Softskill</i>	Sebagai alumni saya dapat menambahkan <i>Softskill</i>
		<i>Software skill</i>	Sebagai alumni saya dapat menambahkan <i>Software skill</i>
		Nilai peminatan	Sebagai alumni saya dapat mengisi seluruh nilai mata kuliah yang tersedia untuk melihat <i>statistic</i> dari peminatan saya.
3.	Portofolio	<i>Show portofolio</i>	Sebagai alumni saya dapat melihat dari keseluruhan dari hasil edit portofolio sebelumnya.
4.	<i>Download CV</i>	<i>Download CV</i>	Sebagai alumni saya dapat <i>mendownload CV</i> dari hasil seluruh data yang telah saya isi pada berbagai <i>form</i> untuk portofolio.

5.	Kuesioner	Mengisi kuesioner <i>tracer study</i>	Sebagai alumni saya dapat mengisi kuesioner yang berisi pertanyaan – pertanyaan terkait alumni.
6.	<i>Event</i>	Melihat <i>Event</i>	Sebagai alumni saya dapat melihat <i>Event</i> atau kegiatan dari pihak prodi
7.	Alumni	Melihat grafik status para alumni	Sebagai alumni saya dapat melihat grafik kumulatif dari status para alumni
8.	<i>Profile</i>	Merubah informasi account	Sebagai alumni saya dapat merubah <i>email</i> yang awalnya menggunakan <i>email</i> “@student.ar-raniry.ac.id” menjadi <i>email</i> biasa.

2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah perangkat lunak dan perangkat keras yang digunakan. Perangkat lunak yang digunakan dalam membangun *Website* ini dapat dilihat pada Tabel 3.7.

Tabel 3. 7 Perangkat Lunak

No	Nama	Versi
1	Visual Studio Code	1.84
2	Flutter	3.13.9
3	Dart	3.1.5
4	Figma	-
5	Firebase	10.17.0

Sumber: Data diolah (2023)

Ada pun perangkat keras yang digunakan dalam membangun *Website* dapat dilihat pada Tabel 3.8.

Tabel 3. 8 Perangkat Keras

No	Nama	Versi
1	Laptop	Lenovo, ideapad slim 3
2	Windows	11
3	Ram 8 GB	3.13.9

Sumber: Data diolah (2023)

3. Design UI *website*

Desain yang telah mencapai tingkat detail yang tinggi, memastikan bahwa elemen-elemen seperti warna, ukuran, jarak, dan bentuk telah ditentukan dengan akurasi yang sangat tepat (Prasetyo, 2022). Berikut desain dari *website Dashboard* data portofolio mahasiswa dan alumni.

A. Admin

Berikut merupakan desain UI dari admin

a. *Authentication Screen*

Authentication di bagi menjadi dua yaitu *Sign In* dan *Logout*. *Sign In* adalah halaman untuk masuk ke dalam menu admin. Pada halaman ini tersedia *form* yang harus diisi oleh admin seperti *email*, dan *Password*. Berikut merupakan desain dari *Sign In Screen*.



Gambar 3. 2 Desain *Sign In Screen*

Logout adalah fitur yang hanya dapat diakses oleh admin. Fitur ini berfungsi untuk mengganti status *Authentication User* dari yang sebelumnya *login* menjadi mahasiswa (tidak *login*). Berikut merupakan desain dari fitur *Logout*.



Gambar 3. 3 Desain *Logout*

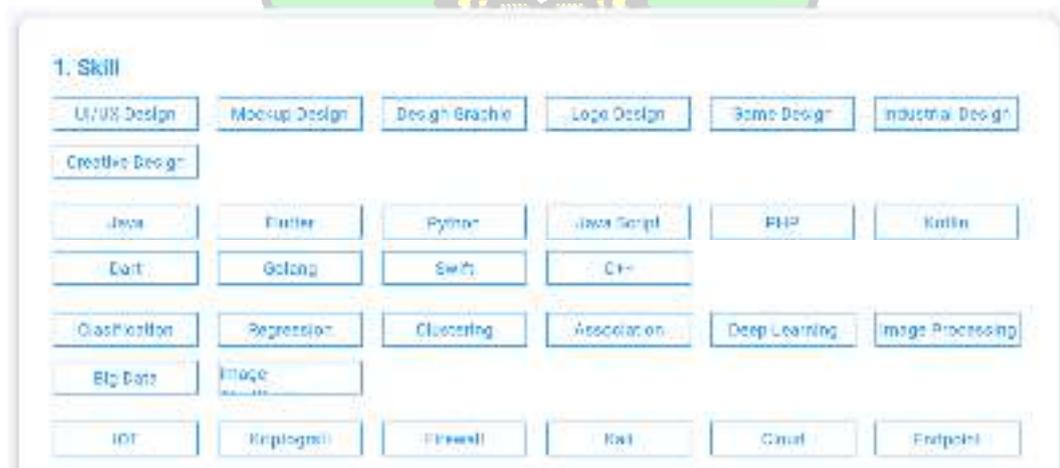
b. Pencarian portofolio mahasiswa

Pencarian portofolio mahasiswa memasukkan kata kunci berupa nama mahasiswa untuk mencari portofolio mahasiswa berdasarkan nama. Kemudian tersedia juga fitur filter untuk mencari portofolio mahasiswa yang lebih spesifik.



Gambar 3. 4 Pencarian Nama

Fitur untuk filter pencarian portofolio mahasiswa terdapat 2 kriteria tertentu, yaitu berdasarkan *softskill* mahasiswa dan *software skill* mahasiswa. Memudahkan untuk mencari portofolio mahasiswa berdasarkan spesifikasi tertentu.



Gambar 3. 5 Filter Berdasarkan Skill

2. Software Skill

Design Visual



Vidio Editor



Machine Learning



Programming



Gambar 3. 6 Filter Berdasarkan *Software Skill*

c. *Dashboard tracer study*

Dashboard tracer study menampilkan informasi umum terkait tentang informasi terkait alumni. Berisikan informasi jumlah alumni yang sudah terdaftar didalam sistem, jumlah pertanyaan kuesioner, jumlah *Event*/kegiatan yang sudah dibuat dan status dari keseluruhan dari alumni.



Gambar 3. 7 *Dashboard Tracer Study*

d. Dashboard Pertanyaan

Pada *Dashboard* ini berisikan informasi terkait *Profile* dari alumni yang sudah terdaftar didalam system. Disini terdapat fitur untuk melihat akses portofolio para alumni secara keseluruhannya. Terdapat juga fitur untuk hapus data alumni.

No	Nama	NIM	Alamat	Universitas	No Telp	Tgl
1	Amir	12121212				
2	123	123				
3	11111	1111				
4	121	121				

Gambar 3. 8 Dashboard Alumni

e. *Dashboard Pertanyaan*

Dashboard pertanyaan berikan tempat admin untuk menambah dan menghapus pertanyaan kuesioner kepada seluruh alumni. Pertanyaan-pertanyaan tersebut berguna untuk mentracking seluruh alumni.

No	Partisipan	Jawaban A	Jawaban B	Jawaban C	Jawaban D
1	Pengalaman Anda dalam menggunakan aplikasi ini? (jika ada, tolong jelaskan pengalaman Anda)	Sangat Baik	Cukup	Belum pernah dan tidak pernah	Belum pernah dan tidak pernah
2	Apakah Anda merasa puas dengan layanan ini?	Sangat Baik	Cukup	Belum pernah dan tidak pernah	Belum pernah dan tidak pernah
3	Apakah Anda merasa puas dengan layanan ini? (jika ada, tolong jelaskan pengalaman Anda)	Sangat Baik	Cukup	Belum pernah dan tidak pernah	Belum pernah dan tidak pernah
4	Apakah Anda merasa puas dengan layanan ini? (jika ada, tolong jelaskan pengalaman Anda)	Sangat Baik	Cukup	Belum pernah dan tidak pernah	Belum pernah dan tidak pernah

Gambar 3. 9 Dashboard pertanyaan

f. *Dashboard Grafik*

Dashboard grafik ini berisikan informasi terkait hasil dari jawaban dari para alumni. Grafik akan menampilkan jawaban kuesioner per setiap pertanyaannya.



Gambar 3. 10 Dashboard Grafik Pertanyaan

g. *Dahsboard Event*

Dashboard Event terdapat fitur untuk menambah *Event* baru untuk para alumni, dan juga fitur hapus jikalau *Event* tersebut telah terlaksana.

The screenshot shows a dashboard with a purple sidebar and a main content area titled 'Data Event'. It features a table with the following data:

No	Aksi	Status	Lokasi	Tanggal
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4

Gambar 3. 11 Dashboard Event

B. Mahasiswa

Berikut ini merupakan tampilan keseluruhan dari *User* Mahasiswa

a. *Authentication*

Authentication di bagi menjadi dua yaitu *Sign In* dan *Logout*. *Sign In* adalah halaman untuk masuk ke dalam menu admin . Pada halaman

ini tersedia *form* yang harus diisi oleh admin seperti *email*, dan *Password*. Berikut merupakan desain dari *Sign In Screen*.



Gambar 3. 12 Sign In Screen

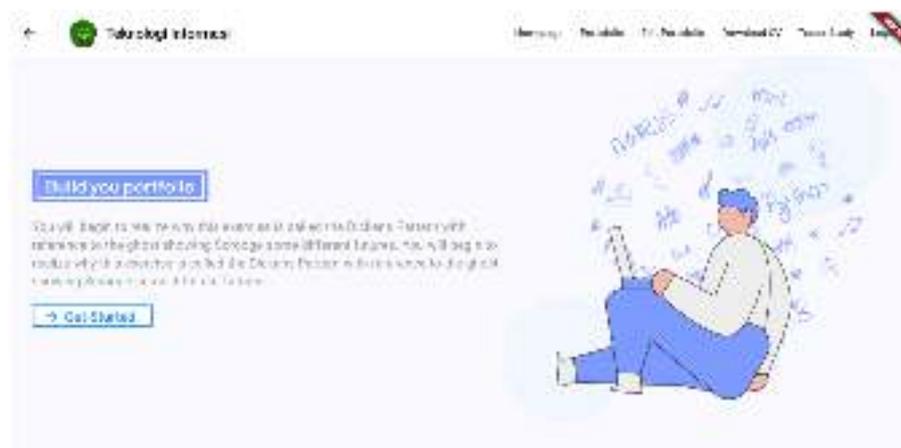
Logout adalah fitur yang hanya dapat diakses oleh admin. Fitur ini berfungsi untuk mengganti status *Authentication User* dari yang sebelumnya *login* menjadi mahasiswa (tidak *login*). Berikut merupakan desain dari fitur *Logout*.



Gambar 3. 13 Logout

b. Edit Portofolio

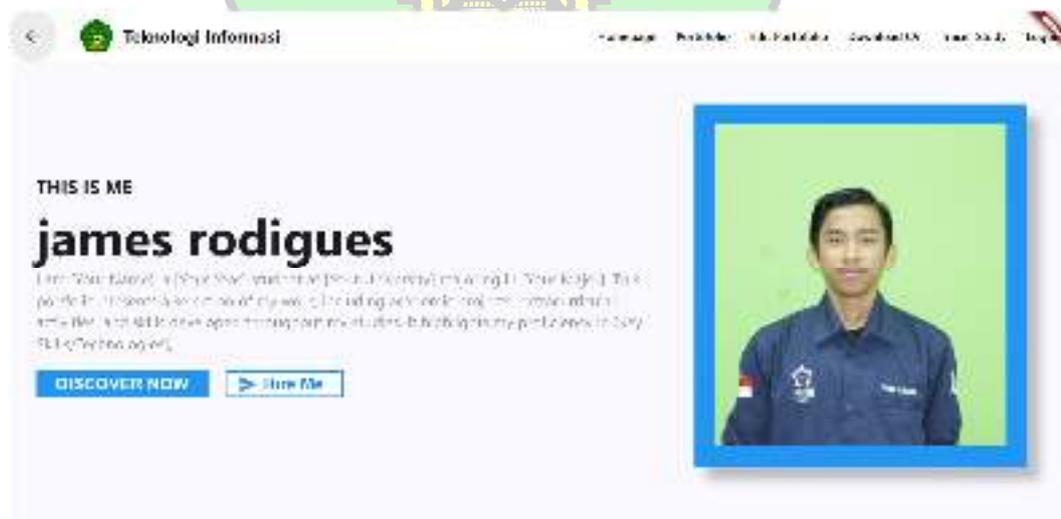
Edit portofolio mahasiswa berisikan berbagai *form* untuk mengisian data pribadi yang berkaitan dengan kebutuhan portofolio mahasiswa. Terdapat 7 *step* pengisian portofolio mahasiswa terdiri dari : *Profile info, achievement, experience, personal skills, softskill, software skill* dan nilai mata kuliah peminatan.



Gambar 3. 14 Edit Portofolio

c. Portofolio

Portofolio mahasiswa berisikan informasi keseluruhan data yang telah diisi pada *step* sebelumnya yaitu edit portofolio.



Gambar 3. 15 Portofolio

d. *Download CV*

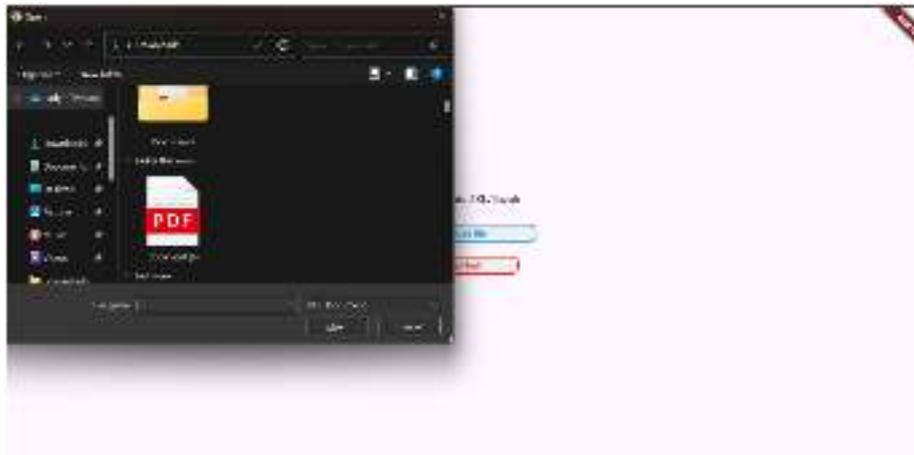
Download CV merupakan fitur untuk mengunduh *CV* otomatis yang di *Generate* oleh *website* ini. Data yang tersedia didalam *CV* berasal dari data-data yang ada pada portofolio.



Gambar 3. 16 *CV*

e. *Tracer study*

Trace study merupakan fitur untuk bagi para mahasiswa yang telah lulus untuk merubah statusnya menjadi seorang alumni dengan mengupload data yang menyatakan bahwa mereka telah lulus.



Gambar 3. 17 Pengujian *Tracer study*

3.3. Konstruksi

Pada tahap Konstruksi, peneliti melanjutkan dengan proses pembuatan program berdasarkan rancangan-rancangan yang telah ditetapkan sebelumnya. Pembuatan program ini dilakukan dengan menggunakan bahasa pemrograman Dart, yang diintegrasikan dengan *Framework Flutter*. Langkah ini memungkinkan peneliti untuk mengimplementasikan setiap elemen desain dalam bentuk kode yang berfungsi sesuai dengan kebutuhan *Website*.

3.4. Pelaksanaan

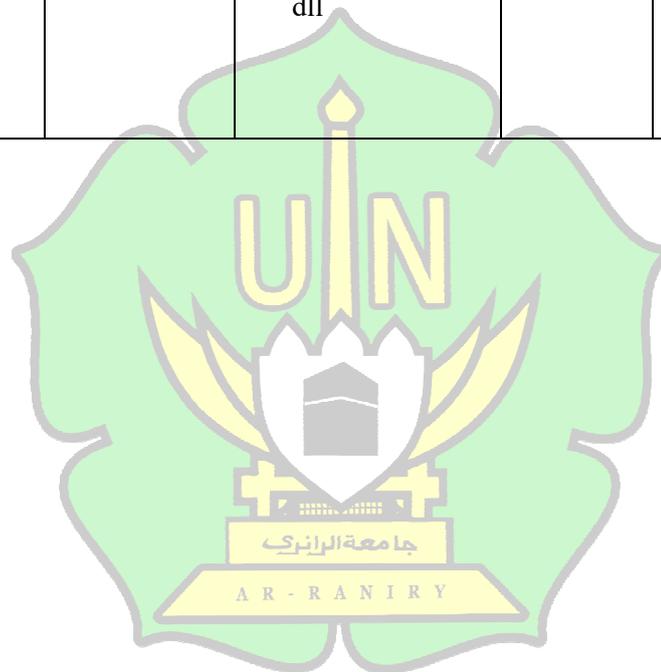
Pada tahapan ini, dilakukan pengujian menyeluruh terhadap sistem informasi Portofolio digital yang telah dibangun. Dalam rangka menguji sistem, penulis mengadopsi metode pengujian *Black Box*, yang berfokus pada pemeriksaan persyaratan fungsional perangkat lunak, sebagaimana dijelaskan oleh Pressman (2002: 551). Sebagai langkah penutup dari fase pelaksanaan ini, penulis juga mengundang tanggapan dan masukan dari pengguna terkait pengalaman mereka dalam menggunakan *website* ini. Hal ini dilakukan untuk memperoleh umpan balik yang berguna sebagai bahan evaluasi dan perbaikan selanjutnya.

Tabel 3. 9 Test Case Pengujian Black Box Admin

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
1.	Authentication	login	Melakukan Login pada website	<ol style="list-style-type: none"> Masuk ke menu login Masukkan Username Password khusus admin yang disediakan Tekan tombol login untuk masuk 	email = admin@gmail.com Password = admin123	User berhasil masuk ke halaman homepage admin	
		Logout	Melakukan Logout pada website	<ol style="list-style-type: none"> User admin menekan menu Logout. 		User keluar dari halaman admin dan mengganti status Authentication User dari yang sebelumnya login menjadi guest (tidak login).	
2.	Dashboard Tracer study	Lihat Jumlah alumni keseluruhan		<ol style="list-style-type: none"> Mengklik menu tambah barang. User admin memilih kategori barang yang ingin dihapus. 		1. User admin dapat melihat data jumlah alumni secara keseluruhan	
		Lihat Jumlah pertanyaan keseluruhan	Lihat jumlah pertanyaan yang sudah dibuat pada menu Dashboard pertanyaan	<ol style="list-style-type: none"> User admin klik menu dashboard pertanyaan Lihat jumlah total pertanyaan yang sudah dibuat 		1. User admin dapat melihat data jumlah pertanyaan secara keseluruhan	
		Lihat Jumlah Event keseluruhan	Lihat jumlah pertanyaan yang sudah dibuat pada menu Dashboard Event	<ol style="list-style-type: none"> User admin klik menu dashboard Event Lihat jumlah total Event yang sudah dibuat 		1. User admin dapat melihat data jumlah Event secara keseluruhan	

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
		Lihat status alumni				1. <i>User</i> admin dapat melihat data jumlah status alumni secara keseluruhan	
4.	<i>Dashboard</i> alumni	Portofolio alumni	Melihat portofolio alumni secara keseluruhan	<ol style="list-style-type: none"> 1. Masuk kedalam <i>Dashboard</i> alumni 2. Mengklik icon pencarian pada menu bagian dari data alumni 		1. <i>User</i> admin dapat melihat secara langsung data portofolio alumni secara menyeluruh	
		Hapus alumni	Menghapus seluruh data dari alumni yang dipilih	<ol style="list-style-type: none"> 1. Masuk kedalam <i>Dashboard</i> alumni 2. Mengklik icon hapus pada menu bagian dari data alumni 		1. <i>User</i> admin dapat menghapus secara langsung data portofolio alumni secara menyeluruh	
6.	<i>Dashboard</i> pertanyaan	Tambah pertanyaan	Menambah pertanyaan untuk kuesioner para alumni	<ol style="list-style-type: none"> 1. Masuk kedalam <i>Dashboard</i> pertanyaan 2. Klik menu tambah untuk menambah pertanyaan 3. Masukkan pertanyaan beserta jawaban pilihannya 4. Klik <i>button</i> tambah 		1. Setelah mengisi pertanyaan beserta dengan jawabannya, maka pertanyaan didalam kuesioner juga bertambah	
7.	<i>Dashboard</i> grafik	Grafik pertanyaan	Melihat grafik hasil jawaban alumni pada setiap pertanyaan	<ol style="list-style-type: none"> 1. Masuk kedalam <i>Dashboard</i> grafik 2. Lihat seluruh jawaban pertanyaan para alumni 		1. Data yang ditampilkan adalah data yang valid, baik dari jumlah pertanyaan serta juga jawaban dari alumni.	

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
8.	Dashboard Event	Tambah Event	Tambah informasi Event untuk informasi kegiatan yang diperuntukan untuk para alumni	<ol style="list-style-type: none"> Masuk kedalam menu Dashboard Event lalu mengklik button tambah Event. Masukkan form untuk menambah Event seperti, deskripsi, tanggal, Lokasi dll 		1. Setelah mengisi Event beserta dengan seluruh informasi yang diperlukan, maka Event juga bertambah.	



Tabel 3. 10 Test Case Pengujian Black Box Mahasiswa

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
1.	Authentication	login	Melakukan Login pada website	<ol style="list-style-type: none"> Masuk ke menu login Masukkan Username Password telah terdaftar Tekan tombol login untuk masuk 	email = admin@gmail.com Password = admin123	User berhasil masuk ke halaman homepage mahasiswa	

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
		<i>Logout</i>	Melakukan <i>Logout</i> pada <i>website</i>	1. <i>User</i> mahasiswa menekan menu <i>Logout</i> .		<i>User</i> keluar dari halaman mahasiswa dan mengganti status <i>Authentication User</i> dari yang sebelumnya <i>login</i> menjadi <i>guest</i> (tidak <i>login</i>).	
		<i>Sign Up</i>	Melakukan pendaftaran akun	<ol style="list-style-type: none"> 1. Mengklik <i>button SignUp</i> pada <i>login page</i> 2. Memasukkan <i>form</i> pendaftaran akun, terdiri dari : <i>NIM, Username, email, dan Password</i> 		<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa dapat melakukan pendaftaran <i>account</i> mahasiswa 2. <i>Email</i> yang didaftar wajib menggunakan kata kunci “@student.ar-raniry.ac.id” 	
2.	Edit portofolio	<i>Info Profile</i>	Mengisi informasi umum terkait <i>Profile</i>	<ol style="list-style-type: none"> 1. Mengklik bagian edit portofolio pada bagian menu <i>header</i> 2. Masuk kedalam <i>form</i> info <i>Profile</i> yang berada pada <i>step 1</i> 3. klik <i>button save</i> untuk menyimpan 		1. <i>User</i> mahasiswa mengisi seluruh <i>form</i> yang berisi informasi pribadi mahasiswa dan Ketika ditekan <i>button “save”</i> maka data akan	

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
						tersimpan di database	
		<i>achievement</i>	Mengisi <i>form achievement</i> yang berisikan tahun, nama kompetisi, ranking, deskripsi	<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 2 2. Isi <i>form achievement</i> 3. Jika <i>achievement</i> lebih dari 1, klik <i>button add</i> disebalah kanan form 4. klik <i>button save</i> untuk menyimpan 		<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa mengisi seluruh <i>form</i> yang berisi <i>aachievement</i> mahasiswa dan Ketika ditekan <i>button "save"</i> maka data akan tersimpan di <i>database</i> 	
		<i>experience</i>	Mengisi <i>form experience</i> yang berisikan tahun, nama institusi, posisi, deskripsi	<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 3 2. Isi <i>form experience</i> 3. Jika <i>experience</i> lebih dari 1, klik <i>button add</i> disebalah kanan form 4. klik <i>button save</i> untuk menyimpan 		<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa mengisi seluruh <i>form</i> yang berisi <i>experience</i> mahasiswa dan Ketika ditekan <i>button "save"</i> maka data akan tersimpan di <i>database</i> 	
		<i>Personal skill</i>	Menambahkan <i>personal skill</i>	<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 4 2. Klik <i>button add</i> jika ingin menambahkan <i>personal skill</i> 3. Jika ingin menghapus klik <i>button hapus</i> 4. klik <i>button save</i> untuk menyimpan 		<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa menambah <i>personal skill</i> dan dapat mengisi <i>personal skill</i> 2. <i>User</i> mahasiswa menghapus <i>personal skill</i> dan dapat menghapus 	

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
						3. Ketika ditekan <i>button save</i> maka data tersimpan didalam <i>database</i>	
		<i>softskill</i>	Menambahka n keahlian pribadi	<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 5 2. Pilih <i>softskill</i> sesuai dengan keahlian mahasiswa pada pilihan yang tersedia 3. klik <i>button save</i> untuk menyimpan 		<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa dapat memilih <i>softskill</i> yang mereka kuasai pada pilihan yang tersedia dan jika membatalkan pilihan klik Kembali pilihan tersebut 2. Ketika ditekan <i>button save</i> maka data tersimpan didalam <i>database</i> 	
		<i>Software skill</i>	Menambahka n keahlian pada <i>software</i> tertentu	<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 6 2. Pilih <i>software skill</i> sesuai dengan keahlian mahasiswa pada pilihan yang tersedia 3. klik <i>button save</i> untuk menyimpan 		<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa dapat memilih <i>software skill</i> yang mereka kuasai pada pilihan yang tersedia dan jika membatalkan pilihan klik Kembali pilihan tersebut 2. Ketika ditekan <i>button save</i> maka data tersimpan didalam <i>database</i> 	

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
		Mata kuliah peminatan	Mengisi nilai mata kuliah yang terkait dengan 3 peminatan	<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 7 2. Masukkan nilai mata kuliah yang terdaftar didalam <i>form</i> mata kuliah sesuai dengan nilai yang tertera pada khs 3. klik <i>button save</i> untuk menyimpan 		<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa dapat mengisi nilai didalam <i>form</i> nilai 2. Ketika ditekan <i>button save</i> maka data tersimpan didalam <i>database</i> 	
3.	portofolio	Portofolio mahasiswa	Melihat portofolio mahasiswa secara keseluruhan sesuai dengan data yang sudah diisi pada <i>step</i> sebelumnya	<ol style="list-style-type: none"> 3. Klik menu portofolio pada <i>header</i> 		<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa dapat melihat seluruh data yang telah diisi pada “edit portofolio” sebelumnya, data telah tersaji semua pada menu portofolio ini 	
6.	Download CV	Generate CV	Generate CV otomatis dan data didalamnya berisikan data dari portofolio	<ol style="list-style-type: none"> 1. Klik menu <i>download CV</i> pada <i>header</i> 		<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa dapat mendownload CV bertipe <i>file</i> PDF 	
7.	Tracer study	Pengajuan tracer study	Pengajuan tracer study berisikan mengirimkan salah satu berkas yang menyatakan bahwa mahasiswa telah lulus	<ol style="list-style-type: none"> 1. Pilih menu <i>tracer study</i> pada <i>header</i> 2. Pilih <i>file</i> pernyataan kelulusan 3. Klik selesai untuk mengirim 		<ol style="list-style-type: none"> 1. <i>File</i> yang dikirim oleh mahasiswa akan tersampaikan pada <i>Dashboard</i> pengajuan pada <i>Dashboard</i> admin 2. Jika admin <i>approve file</i> tersebut maka 	

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
						mahasiswa yang mengajukan akan berubah status menjadi seorang alumni	

Tabel 3. 11 Test Case Pengujian Black Box Alumni

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
1.	Authentication	login	Melakukan Login pada website	<ol style="list-style-type: none"> Masuk ke menu login Masukkan Username Password telah terdaftar Tekan tombol login untuk masuk 	email = admin@gmail.com Password = admin123	User berhasil masuk ke halaman homepage alumni	
		Logout	Melakukan Logout pada website	<ol style="list-style-type: none"> User alumni menekan menu Logout. 		User keluar dari halaman mahasiswa dan mengganti status Authentication User dari yang sebelumnya login menjadi guest (tidak login).	

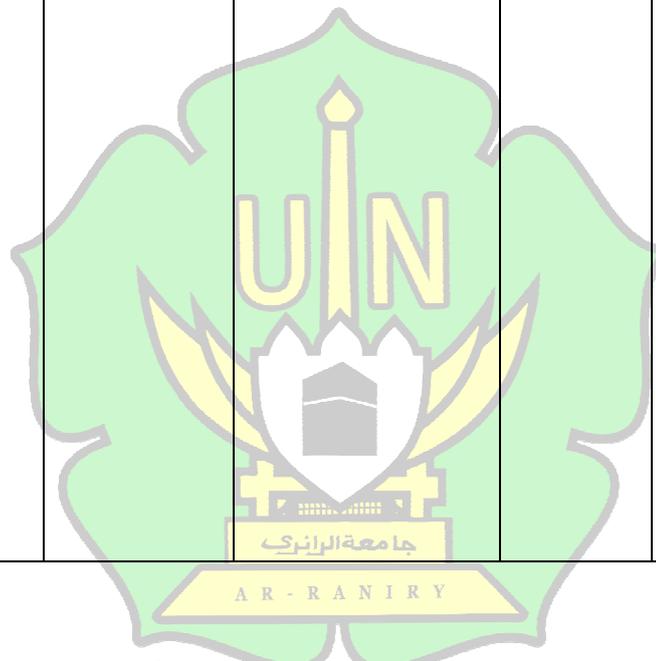
No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
		Sign Up	Melakukan pendaftaran akun	<ol style="list-style-type: none"> 1. Mengklik <i>button SignUp</i> pada <i>login page</i> 2. Memasukkan <i>form</i> pendaftaran akun, terdiri dari : NIM, <i>Username</i>, <i>email</i>, dan <i>Password</i> 		<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa dapat melakukan pendaftaran <i>account</i> mahasiswa 2. <i>Email</i> yang didaftar wajib menggunakan kata kunci “@student.ar-raniry.ac.id” 	
2.	Edit portofolio	Info Profile	Mengisi informasi umum terkait Profile	<ol style="list-style-type: none"> 1. Mengklik bagian edit portofolio pada bagian menu <i>header</i> 2. Masuk kedalam <i>form</i> info Profile yang berada pada <i>step</i> 1 3. klik <i>button save</i> untuk menyimpan 		<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa mengisi seluruh <i>form</i> yang berisi informasi pribadi mahasiswa dan Ketika ditekan <i>button</i> “<i>save</i>” maka data akan tersimpan di <i>database</i> 	
		achievement	Mengisi <i>form</i> achievement yang berisikan tahun, nama kompetisi, ranking, deskripsi	<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 2 2. Isi <i>form</i> achievement 3. Jika <i>achievement</i> lebih dari 1, klik <i>button add</i> disebalah kanan form 4. klik <i>button save</i> untuk menyimpan 		<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa mengisi seluruh <i>form</i> yang berisi <i>achievement</i> mahasiswa dan Ketika ditekan <i>button</i> “<i>save</i>” maka data akan tersimpan di <i>database</i> 	
		experience	Mengisi <i>form</i> experience yang berisikan tahun, nama institusi,	<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 3 2. Isi <i>form</i> experience 3. Jika <i>experience</i> lebih dari 1, klik 		<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa mengisi seluruh <i>form</i> yang berisi <i>experience</i> mahasiswa 	

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
			posisi, deskripsi	<p><i>button add</i> disebalah kanan form</p> <p>4. klik <i>button save</i> untuk menyimpan</p>		dan Ketika ditekan <i>button "save"</i> maka data akan tersimpan di <i>database</i>	
		<i>Personal skill</i>	Menambahk an <i>personal skill</i>	<p>1. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 4</p> <p>2. Klik <i>button add</i> jika ingin menambahkan <i>personal skill</i></p> <p>3. Jika ingin menghapus klik <i>button hapus</i></p> <p>4. klik <i>button save</i> untuk menyimpan</p>		<p>1. <i>User</i> mahasiswa menambah <i>personal skill</i> dan dapat mengisi <i>personal skill</i></p> <p>2. <i>User</i> mahasiswa menghapus <i>personal skill</i> dan dapat menghapus</p> <p>3. Ketika ditekan <i>button save</i> maka data tersimpan didalam <i>database</i></p>	
		<i>softskill</i>	Menambahk an keahlian pribadi	<p>1. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 5</p> <p>2. Pilih <i>softskill</i> sesuai dengan keahlian mahasiswa pada pilihan yang tersedia</p> <p>3. klik <i>button save</i> untuk menyimpan</p>		<p>1. <i>User</i> mahasiswa dapat memilih <i>softskill</i> yang mereka kuasai pada pilihan yang tersedia dan jika membatalkan pilihan klik Kembali pilihan tersebut</p> <p>2. Ketika ditekan <i>button save</i> maka data tersimpan didalam <i>database</i></p>	

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
		<i>Software skill</i>	Menambahkan keahlian pada <i>software</i> tertentu	<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 6 2. Pilih <i>software skill</i> sesuai dengan keahlian mahasiswa pada pilihan yang tersedia 3. klik <i>button save</i> untuk menyimpan 		<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa dapat memilih <i>software skill</i> yang mereka kuasai pada pilihan yang tersedia dan jika membatalkan pilihan klik Kembali pilihan tersebut 2. Ketika ditekan <i>button save</i> maka data tersimpan didalam <i>database</i> 	
		Mata kuliah peminatan	Mengisi nilai mata kuliah yang terkait dengan 3 peminatan	<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 7 2. Masukkan nilai mata kuliah yang terdaftar didalam <i>form</i> mata kuliah sesuai dengan nilai yang tertera pada <i>khs</i> 3. klik <i>button save</i> untuk menyimpan 		<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa dapat mengisi nilai didalam <i>form</i> nilai 2. Ketika ditekan <i>button save</i> maka data tersimpan didalam <i>database</i> 	
3.	portofolio	Portofolio mahasiswa	Melihat portofolio mahasiswa secara keseluruhan sesuai dengan data yang sudah diisi pada <i>step</i> sebelumnya	<ol style="list-style-type: none"> 1. Klik menu portofolio pada <i>header</i> 		<ol style="list-style-type: none"> 1. <i>User</i> mahasiswa dapat melihat seluruh data yang telah diisi pada “edit portofolio” sebelumnya, data telah tersaji semua pada menu portofolio ini 	

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
6.	Download CV	Generate CV	Generate CV otomatis dan data didalamnya berisikan data dari portofolio	1. Klik menu <i>download CV</i> pada <i>header</i>		1. User mahasiswa dapat mendownload CV bertipe file PDF	
7.	Kuesioner	Menjawab kuesioner	Isi <i>form</i> kuesioner dengan pertanyaan yang telah diisi oleh admin	1. Pilih menu kuesioner pada <i>header</i> 2. Pilih jawaban dari <i>form</i> pertanyaan yang tersedia 3. Klik <i>save</i> untuk mengirim		1. Seluruh jawaban dari User alumni akan terekam dan akan terkirim kedalam <i>database</i>	
8.	Event	Melihat Event/kegiatan	Event yang tersedia berasal dari Event yang telah diisi oleh admin	1. Pilih menu Event pada <i>header</i>		1. Event yang tersedia pada menu Event, seharusnya berisikan data yang diinput oleh admin ada <i>Dashboard Event</i>	
9.	Statistik	Melihat <i>statistic</i> status alumni	<i>Statistic</i> yang tersedia berasal dari kalkulasi status pekerjaan alumni yang telah mengisi status alumni	1. Pilih menu Event pada <i>header</i>		1. Statistik yang tersedia seharusnya berisikan data dari kalkulasi status alumni yang sudah terdaftar pada <i>website</i> ini secara keseluruhan	
10.	Alumni	Ganti <i>email</i> , <i>Password</i> dan status alumni	Mengganti <i>email</i> yang sudah terdaftar sebelumnya menggunakan akun kampus, <i>Password</i>	1. Klik <i>button ganti email</i> 2. Klik <i>button ganti Password</i> 3. Pilih status alumni 4. Klik <i>save</i> untuk mengirim		1. Ketika <i>button ganti email</i> ditekan maka muncul popup untuk mengisi <i>email</i> yang baru, kemudian klik ganti <i>email</i> maka	

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
			dan status alumni			<p>akan terkirim konfirmasi otomatis melalui pesan <i>email</i> dari <i>email</i> tersebut</p> <p>2. Ketika ditekan <i>button</i> ganti <i>Password</i> maka akan muncul popup untuk mengisi <i>email</i> yang terdaftar didaman akun alumni saat itu, dan kemudian klik <i>button</i> ganti <i>Password</i> maka secara otomatis akan terkirim ke Alamat <i>email</i> tersebut sebuah pesan konfirmasi untuk mengganti <i>Password</i></p>	



1. Tempat penelitian

Penelitian ini dilakukan di Prodi Teknologi Informasi yang berada di Fakultas Sain dan Teknologi, Universitas Islam Negeri Ar-Raniry Banda Aceh.

BAB IV

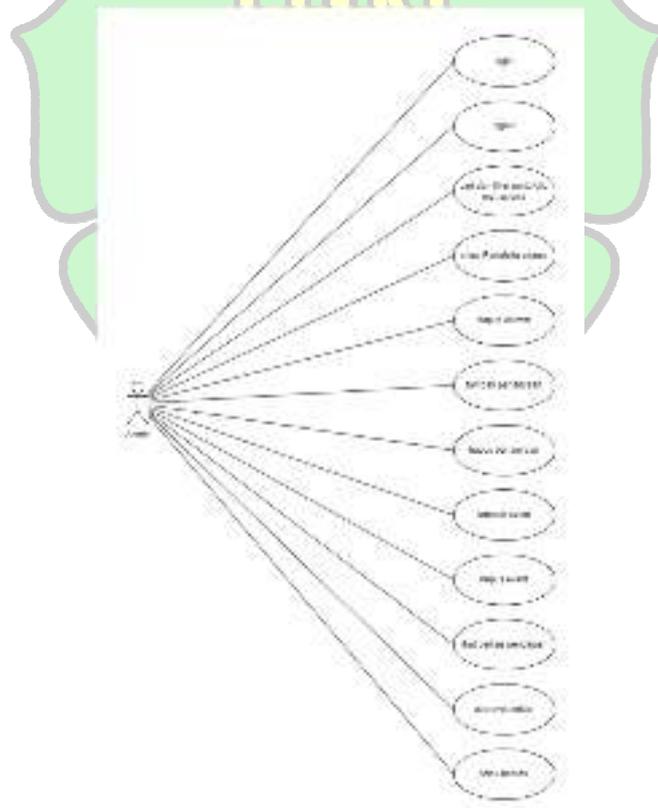
HASIL DAN PEMBAHASAN

4.1 Perancangan

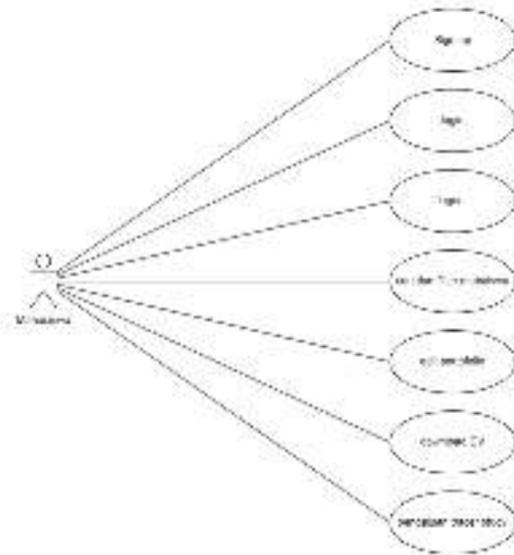
Bagian ini akan mengulas tentang desain dan pemodelan dalam pengembangan *website*. Fokus utama dari desain ini adalah memastikan bahwa *website* yang dibuat memenuhi kebutuhan yang telah diuraikan dalam bentuk *User Story* pada bab sebelumnya.

4.3.1 Use Case Diagram

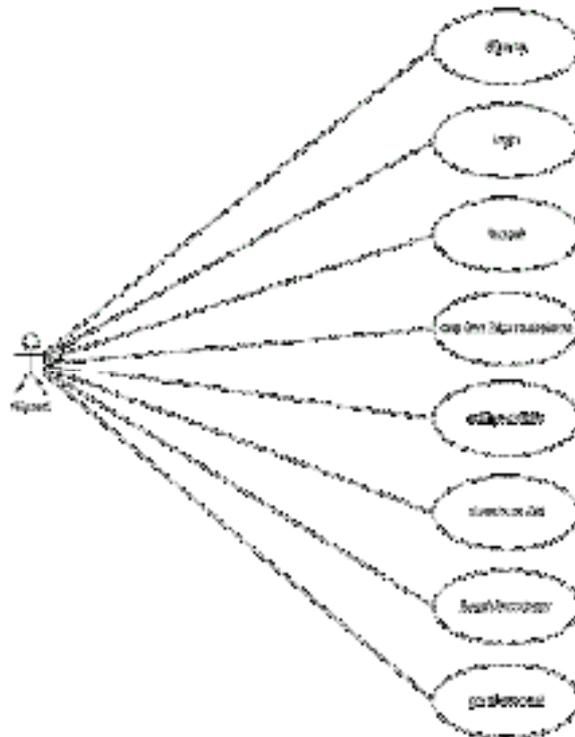
Diagram *Use Case* memperlihatkan bagaimana aktor berinteraksi dengan sistem dan menjelaskan *Use Case* melalui langkah-langkah yang sederhana. Diagram ini menawarkan pandangan umum tentang hubungan antara aktor, *Use Case*, dan system (Al et al., 2020). Berikut merupakan *Use Case Diagram*:



Gambar 4. 1 Use Case Admin



Gambar 4. 2 Use Case Mahasiswa



Gambar 4. 3 Use Case Alumni

4.3.2 Database Schema

Skema *database* adalah struktur yang menentukan bagaimana data diatur dalam sebuah *database* relasional. Ini meliputi elemen seperti nama tabel, kolom, tipe data, dan keterkaitan antar entitas yang relevan (ibm, 2024). Berikut merupakan *database Schema* dari *website* ini :



Gambar 4. 4 Database Schema Mahasiswa dan Alumni



Gambar 4.5 Database Schema Admin

4.2 Flowchart User

Activity Diagram berfungsi untuk memvisualisasikan alur proses dan urutan kegiatan yang terjadi ketika pengguna berinteraksi dengan situs *website*. Di bawah ini, akan dapat dilihat diagram aktivitas yang menggambarkan fitur-fitur yang sedang dikembangkan atau dirancang dalam pembuatan *website*.

1. Fitur Admin

Ada beberapa fitur yang hanya dapat diakses oleh *User* dengan tipe admin. Berikut merupakan *Activity Diagram* dari setiap fitur tersebut.

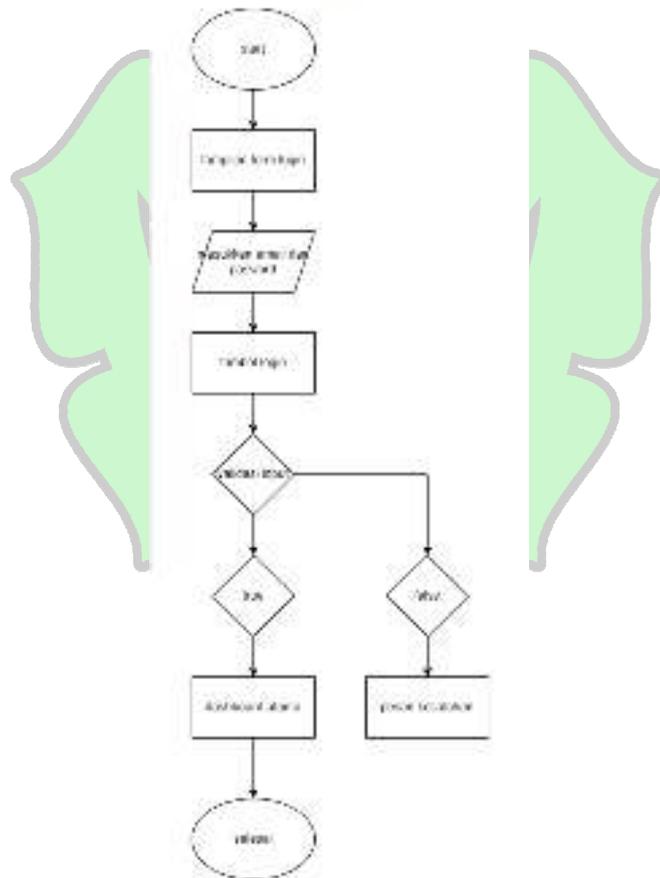
A. Authentication

Fitur *Authentication* dibagi menjadi 2, yaitu *Sign In* dan *Logout*. Berikut merupakan *activity diagramnya*

a. Log In

Gambar dibawah menggambarkan rangkaian proses *Sign In* yang dilakukan *User* di *Dashboard* data portofolio mahasiswa dan alumni. Adapun rinciannya ialah sebagai berikut:

- *User* mengklik *button Login* pada menu *guest*
- *User* memasukan *email* dan *Password*, kemudian menekan *button login*
- kemudian aplikasi melakukan pengecekan apakah *email* dan *Password* telah benar atau salah sesuai dengan *email* dan *Password* admin *firebase Authentication*, jika salah maka akan muncul *pop up* notifikasi bahwa *email* yang diinput salah, dan jika benar maka aplikasi akan menampilkan halaman *homepage* admin.

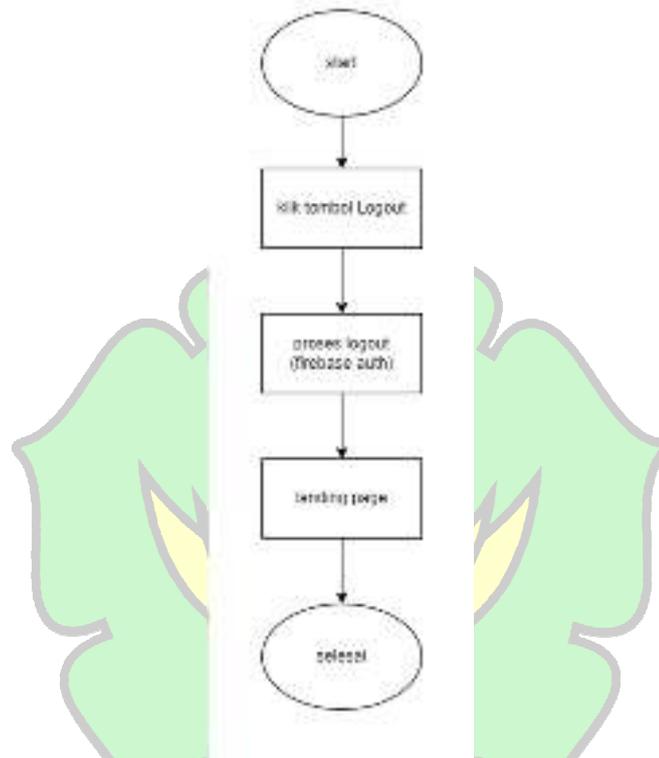


Gambar 4. 6 Flowchart *Login* Admin

b. *Logout*

Gambar dibawah menggambarkan rangkaian proses *Logout* yang dilakukan *User* di *Dashboard* data portofolio mahasiswa dan alumni. Adapun rinciannya ialah sebagai berikut:

- *User* mengklik *button Logout* pada menu
- Kemudian aplikasi akan melakukan *Logout* pada *database* dan menampilkan halaman landing page



Gambar 4. 7 Flowchart Logout Admin

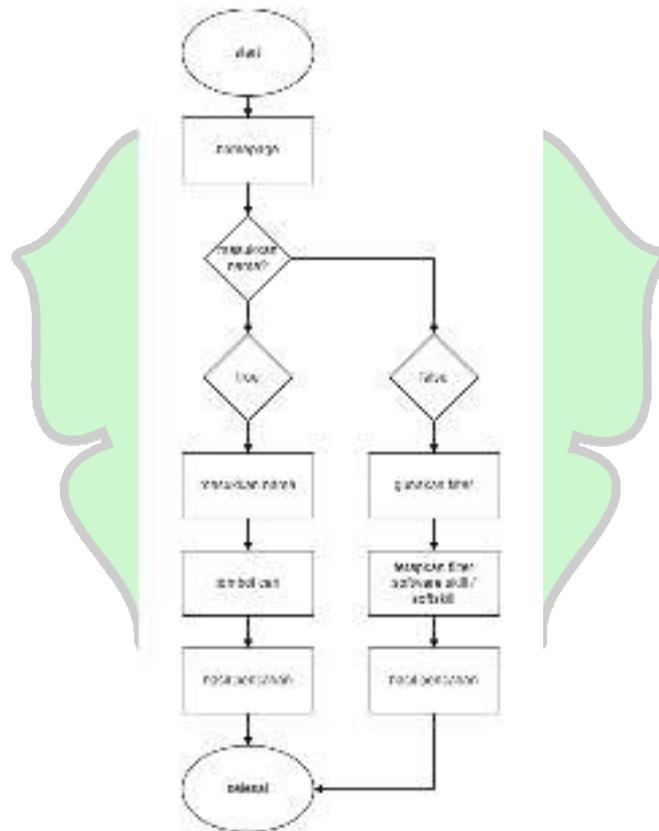
B. Cari nama dan filter portofolio mahasiswa

Fitur cari nama dan filter portofolio mahasiswa berikut ini adalah rincian dari *Flowchart* fitur tersebut

- Proses dimulai ketika admin mengakses fitur pencarian dan filter mahasiswa dalam antarmuka admin.
- *Homepage* admin yang berisi fitur pencarian dan filter mahasiswa ditampilkan. Di sini, admin dapat melihat dua opsi utama: pencarian berdasarkan nama dan filter berdasarkan *softskill* serta *software skill*.
- Admin memutuskan apakah ingin mencari mahasiswa berdasarkan nama, ini adalah langkah Keputusan. Jika iya, admin memilih untuk mencari berdasarkan nama, lanjutkan ke langkah berikutnya yaitu

menampilkan hasil pencarian berdasarkan nama. Jika tidak, admin lanjut ke langkah filter.

- Pada Langkah filter, admin memilih kriteria *softskill* dan/atau *software skill* yang relevan.
- Admin menerapkan filter yang telah dipilih untuk menyaring hasil pencarian dan Hasil akhir yang sesuai dengan filter ditampilkan kepada admin.
- Proses pencarian dan filter selesai, dan admin melihat hasil yang diinginkan.

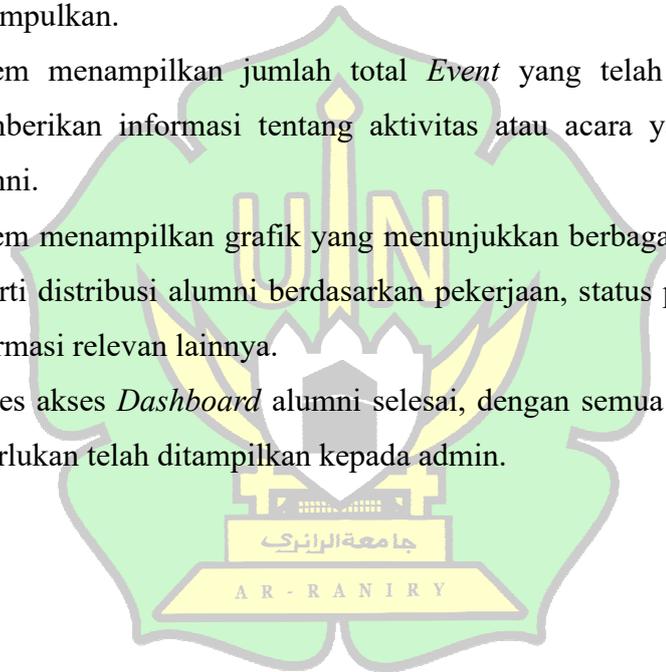


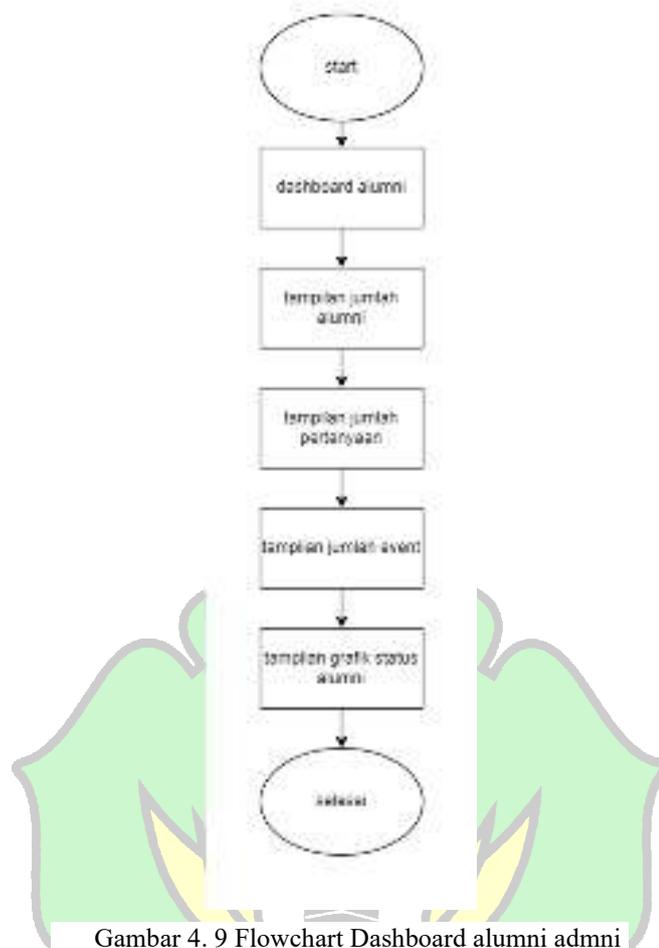
Gambar 4. 8 Flowchart cari nama dan filter

C. Dashboard alumni

Fitur *Dashboard* alumni berikut ini adalah rincian dari *Flowchart User* admin untuk fitur *Dashboard* alumni sebagai berikut :

- Proses dimulai ketika admin memutuskan untuk mengakses fitur *Dashboard* alumni.
- Sistem menampilkan antarmuka khusus untuk *Dashboard* alumni, yang menyajikan berbagai informasi penting tentang alumni.
- Sistem menampilkan jumlah total alumni yang terdaftar dalam sistem. Ini memberikan gambaran umum tentang populasi alumni.
- Sistem menampilkan jumlah total pertanyaan yang ada dalam kuesioner. Ini membantu admin untuk memahami sejauh mana data kuesioner dikumpulkan.
- Sistem menampilkan jumlah total *Event* yang telah dilakukan. Ini memberikan informasi tentang aktivitas atau acara yang melibatkan alumni.
- Sistem menampilkan grafik yang menunjukkan berbagai status alumni, seperti distribusi alumni berdasarkan pekerjaan, status pendidikan, dan informasi relevan lainnya.
- Proses akses *Dashboard* alumni selesai, dengan semua informasi yang diperlukan telah ditampilkan kepada admin.





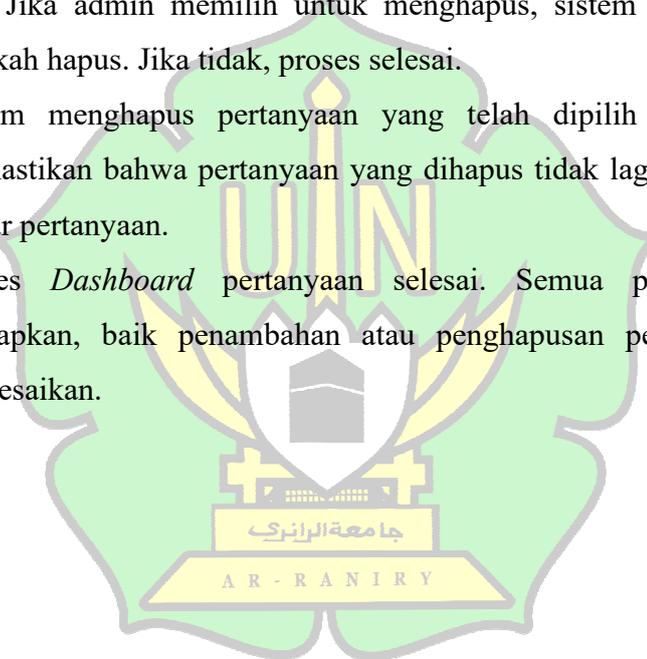
Gambar 4. 9 Flowchart Dashboard alumni admini

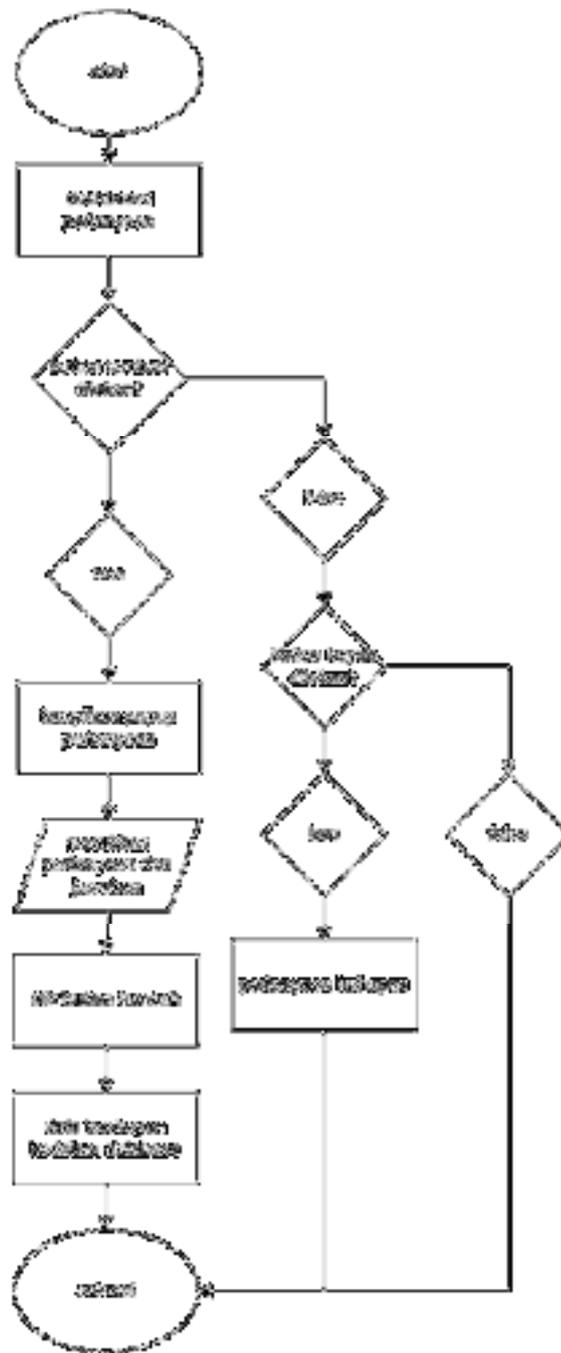
D. *Dashboard* pertanyaan

Fitur *Dashboard* pertanyaan berikut ini adalah rincian dari *Flowchart User* admin untuk fitur *Dashboard* pertanyaan sebagai berikut :

- Proses dimulai ketika admin mengakses fitur *Dashboard* Pertanyaan di antarmuka admin.
- Antarmuka *Dashboard* pertanyaan ditampilkan, menyediakan opsi untuk menambah atau menghapus pertanyaan serta menampilkan daftar pertanyaan yang ada.
- Admin memutuskan apakah akan menambah pertanyaan baru atau tidak. Jika admin memilih untuk menambah, sistem melanjutkan ke popup form. Jika tidak, sistem melanjutkan ke opsi hapus.

- Sistem menampilkan popup yang berisi *form* untuk admin mengisi pertanyaan baru dan jawabannya.
- Admin mengisi *form* dengan pertanyaan dan jawaban baru yang diinginkan. Ini merupakan input data dari admin.
- Admin menekan tombol tambah setelah mengisi form. Ini menginisiasi penyimpanan data ke dalam system.
- Sistem menyimpan data pertanyaan dan jawaban yang baru ditambahkan ke dalam *database*. Pertanyaan baru kini tersedia untuk ditampilkan di daftar pertanyaan.
- Admin memutuskan apakah akan menghapus pertanyaan yang sudah ada. Jika admin memilih untuk menghapus, sistem melanjutkan ke langkah hapus. Jika tidak, proses selesai.
- Sistem menghapus pertanyaan yang telah dipilih dari *database*, memastikan bahwa pertanyaan yang dihapus tidak lagi tersedia dalam daftar pertanyaan.
- Proses *Dashboard* pertanyaan selesai. Semua perubahan yang diterapkan, baik penambahan atau penghapusan pertanyaan, telah diselesaikan.



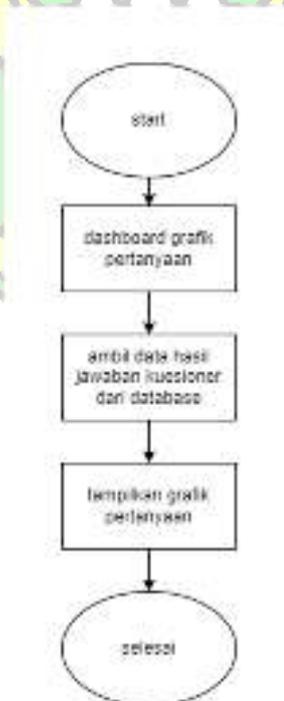


Gambar 4. 10 Flowchart Dashboard pertanyaan

E. Grafik pertanyaan

Fitur grafik pertanyaan berikut ini adalah rincian dari *Flowchart User* admin untuk fitur grafik pertanyaan sebagai berikut:

- Proses dimulai ketika admin mengakses fitur *Dashboard* Grafik Kuesioner di antarmuka admin.
- Sistem menampilkan antarmuka *Dashboard* grafik kuesioner. *Dashboard* ini dirancang untuk memungkinkan admin melihat berbagai grafik yang menunjukkan hasil jawaban kuesioner dari alumni.
- Sistem menghubungi *database* untuk mengambil data jawaban kuesioner dari alumni. Data ini mencakup jawaban untuk setiap pertanyaan dalam kuesioner yang telah diisi oleh alumni.
- Grafik yang telah dihasilkan ditampilkan pada antarmuka *Dashboard*. Admin dapat melihat grafik-grafik ini untuk menganalisis hasil kuesioner, memahami tren, dan membuat keputusan berbasis data.
- Proses *Dashboard* Grafik Kuesioner selesai. Semua grafik telah ditampilkan dan admin dapat mengevaluasi data. Proses berakhir setelah grafik ditampilkan di *Dashboard*.



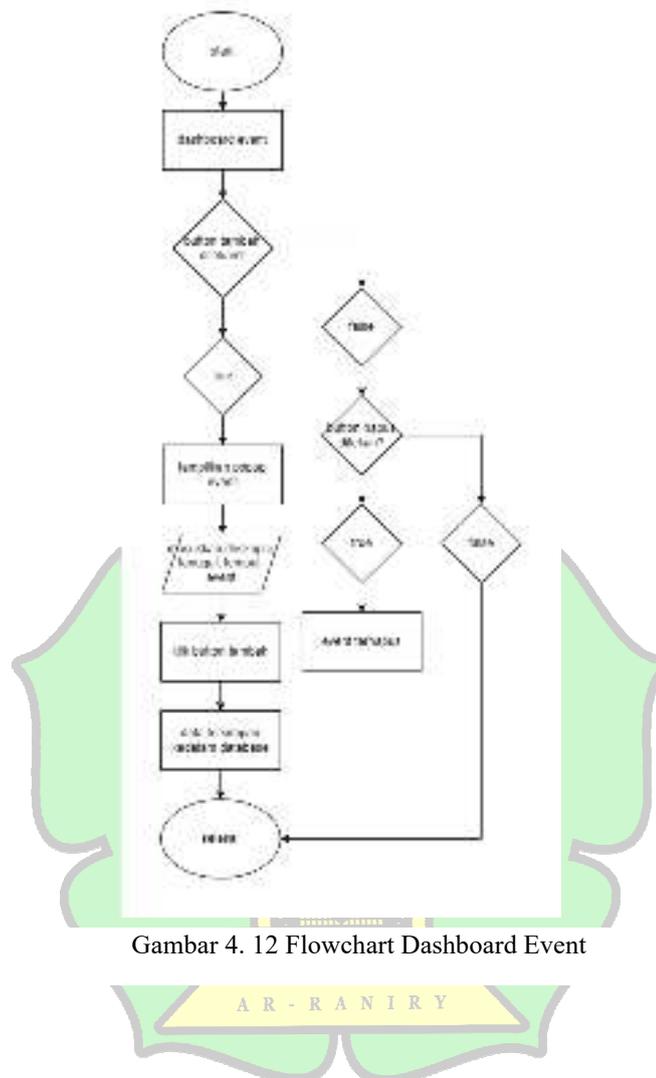
Gambar 4. 11 Flowchart Grafik pertanyaan

F. Dashboard Event

Fitur *Dashboard Event* berikut ini adalah rincian dari *Flowchart User* admin untuk fitur *Dashboard Event* sebagai berikut:

- Proses dimulai ketika admin mengakses fitur *Dashboard Event* di antarmuka admin.
- Antarmuka *Dashboard Event* ditampilkan, menyediakan opsi untuk menambah atau menghapus *Event* serta menampilkan daftar *Event* yang ada.
- Admin memutuskan apakah akan menambah *Event* baru atau tidak. Jika admin memilih untuk menambah, sistem melanjutkan ke popup form. Jika tidak, sistem melanjutkan ke opsi hapus.
- Sistem menampilkan popup yang berisi *form* untuk admin mengisi *form Event* yang berisikan judul, deskripsi, tanggal dan tempat.
- Admin menekan tombol tambah setelah mengisi form. Ini menginisiasi penyimpanan data ke dalam system.
- Sistem menyimpan data *Event* yang baru ditambahkan ke dalam *database*. *Event* baru kini tersedia untuk ditampilkan di daftar *Event*.
- Admin memutuskan apakah akan menghapus *Event* yang sudah ada. Jika admin memilih untuk menghapus, sistem melanjutkan ke langkah hapus. Jika tidak, proses selesai.
- Sistem menghapus *Event* yang telah dipilih dari *database*, memastikan bahwa *Event* yang dihapus tidak lagi tersedia dalam daftar *Event*.

- Proses *Dashboard Event* selesai. Semua perubahan yang diterapkan, baik penambahan atau penghapusan *Event*, telah diselesaikan.



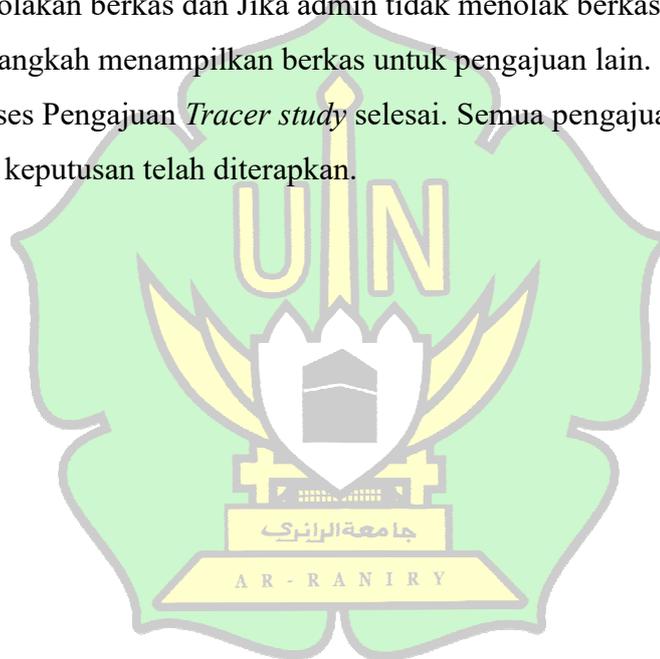
Gambar 4. 12 Flowchart Dashboard Event

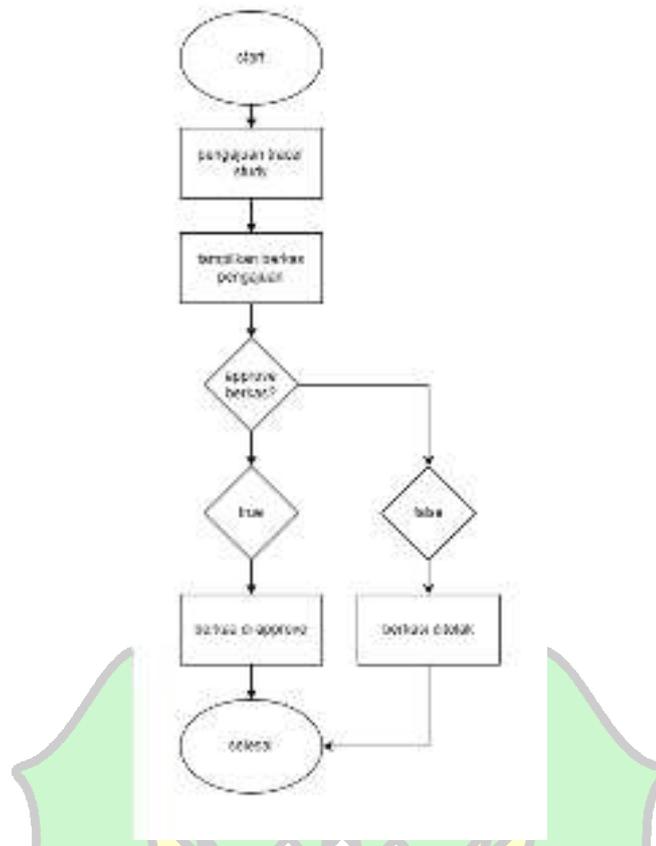
G. Dashboard pengajuan

Fitur *Dashboard* pengajuan berikut ini adalah rincian dari *Flowchart User* admin untuk fitur *Dashboard* pengajuan sebagai berikut:

- Proses dimulai ketika admin mengakses fitur Pengajuan *Tracer study* di antarmuka admin.
- Sistem menampilkan antarmuka yang berisi daftar pengajuan berkas kelulusan dari mahasiswa serta opsi untuk menampilkan, menyetujui, atau menolak berkas.

- Sistem menampilkan berkas kelulusan yang dikirimkan oleh mahasiswa untuk peninjauan oleh admin.
- Admin memutuskan apakah akan menyetujui berkas yang ditampilkan.
- Jika admin memilih untuk menyetujui, lanjutkan ke langkah berikutnya dan Jika admin memilih untuk menolak berkas, lanjutkan ke langkah penolakan berkas.
- Sistem memperbarui status mahasiswa menjadi alumni dan menyimpan perubahan tersebut dalam *database*.
- Admin memutuskan apakah akan menolak berkas yang ditampilkan.
- Jika admin memilih untuk menolak berkas, lanjutkan ke langkah penolakan berkas dan Jika admin tidak menolak berkas, proses kembali ke langkah menampilkan berkas untuk pengajuan lain.
- Proses Pengajuan *Tracer study* selesai. Semua pengajuan telah diproses dan keputusan telah diterapkan.





Gambar 4. 13 Flowchart Dashboard pengajuan

2. Fitur Mahasiswa dan alumni

Ada beberapa fitur *Dashboard* data portofolio mahasiswa dan alumni yang hanya dapat diakses oleh *User* dengan tipe mahasiswa dan alumni diantaranya adalah *Authentication*, edit portofolio, portofolio, *download CV*, pengajuan *tracer study*, kuesioner, *Event*, statistik dan alumni. Berikut merupakan *Activity Diagram* dari setiap fitur tersebut.

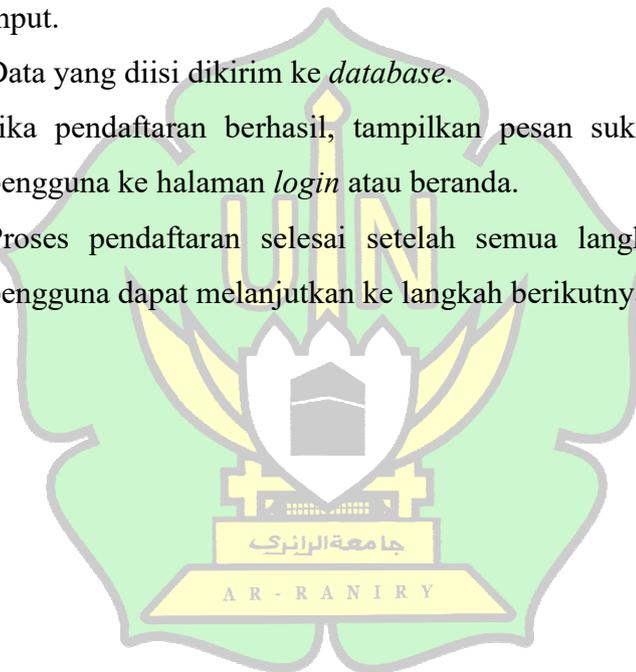
A. Authentication

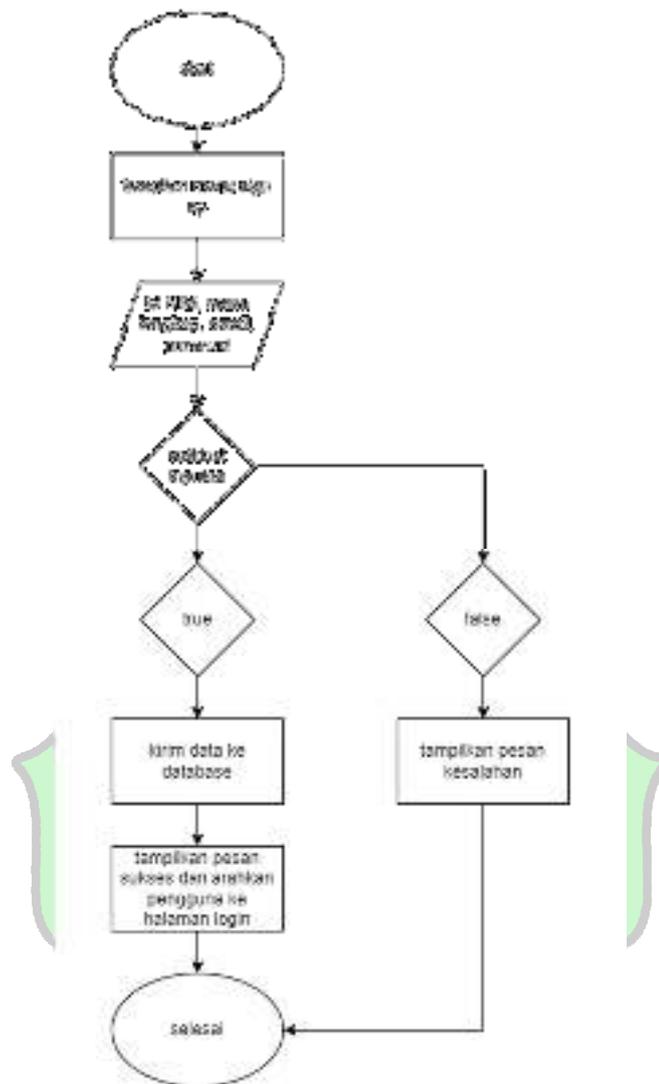
Fitur *Authentication* dibagi menjadi 3, yaitu *sign up*, *Sign In* dan *Logout*. Berikut merupakan *Flowchart User* fitur *Authentication*

a. Sign Up

Gambar dibawah menggambarkan rangkaian proses *Sign In* yang dilakukan *User* di *Dashboard* data portofolio mahasiswa dan alumni. Adapun rinciannya ialah sebagai berikut:

- Proses dimulai ketika pengguna mengakses halaman pendaftaran.
- Sistem menampilkan formulir pendaftaran yang meminta informasi pengguna, termasuk NIM, nama lengkap/*Username*, *email*, dan *Password*.
- Pengguna mengisi formulir dengan informasi yang diperlukan untuk mendaftar.
- Sistem memvalidasi input dari formulir untuk memastikan bahwa semua data diisi dengan benar dan dalam format yang sesuai.
- Jika input valid, lanjutkan ke langkah berikutnya. Jika ada kesalahan, tampilkan pesan kesalahan dan minta pengguna untuk memperbaiki input.
- Data yang diisi dikirim ke *database*.
- Jika pendaftaran berhasil, tampilkan pesan sukses dan arahkan pengguna ke halaman *login* atau beranda.
- Proses pendaftaran selesai setelah semua langkah selesai, dan pengguna dapat melanjutkan ke langkah berikutnya.





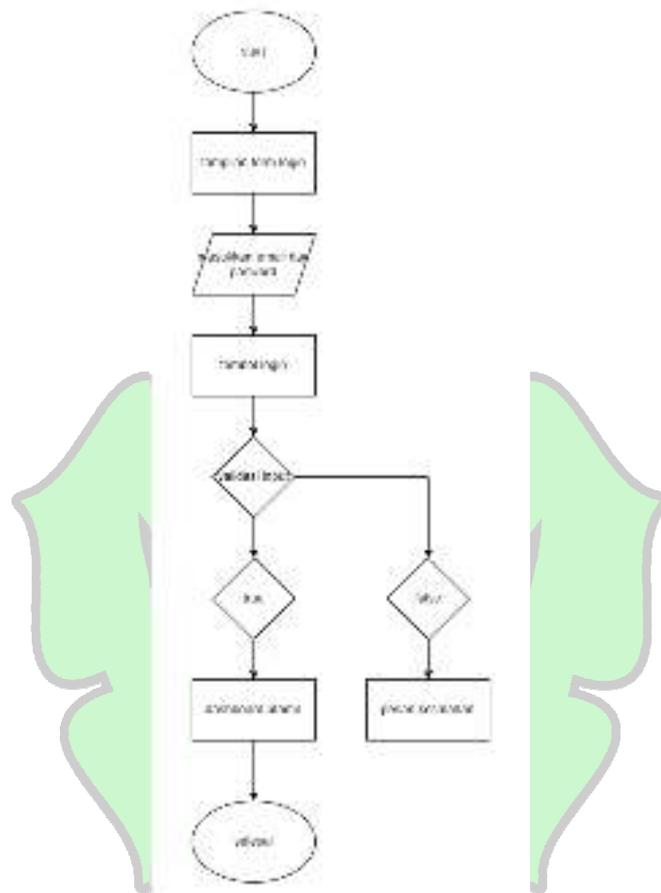
Gambar 4. 14 Flowchart SignUp Mahasiswa dan Alumni

b. Log In

Gambar dibawah menggambarkan rangkaian proses *Sign In* yang dilakukan *User* di *Dashboard* data portofolio mahasiswa dan alumni. Adapun rinciannya ialah sebagai berikut:

- *User* mengklik *button Login* pada menu *guest*
- *User* memasukan *email* dan *Password*, kemudian menekan *button login*

- kemudian aplikasi melakukan pengecekan apakah *email* dan *Password* telah benar atau salah sesuai dengan *email* dan *Password* admin *firebase Authentication*, jika salah maka akan muncul *pop up* notifikasi bahwa *email* yang diinput salah, dan jika benar maka aplikasi akan menampilkan halaman *homepage* admin.



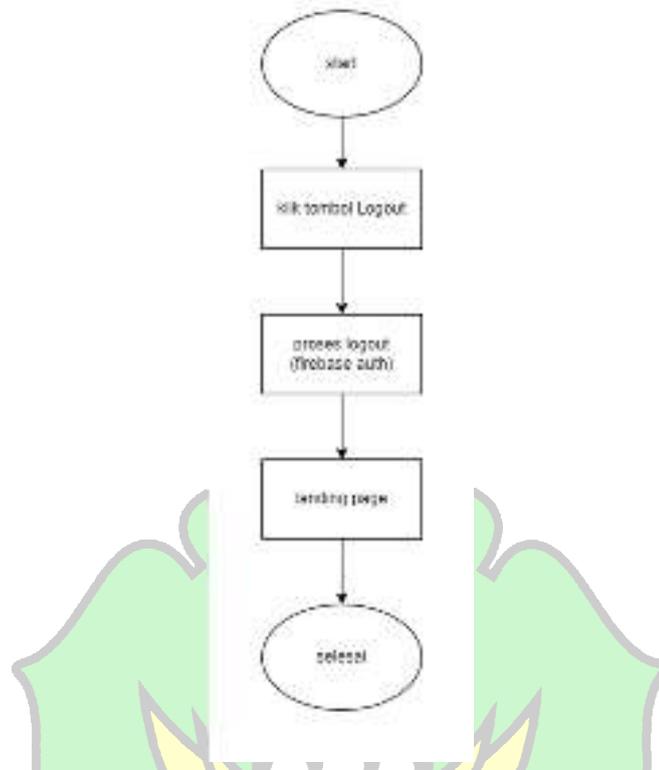
Gambar 4. 15 Flowchart *Login* Alumni dan mahasiswa

b. *Logout*

Gambar dibawah menggambarkan rangkaian proses *Logout* yang dilakukan *User* di *Dashboard* data portofolio mahasiswa dan alumni. Adapun rinciannya ialah sebagai berikut:

- *User* mengklik *button Logout* pada menu

- Kemudian aplikasi akan melakukan *Logout* pada *database* dan menampilkan halaman landing page



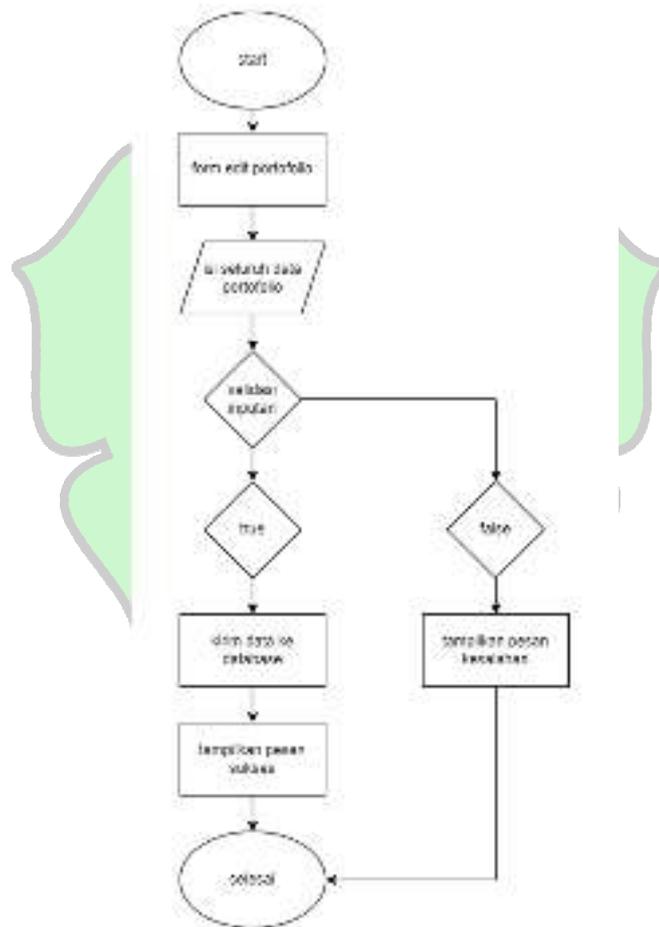
Gambar 4. 16 Flowchart Logout mahasiswa dan alumni

B. Edit portofolio

Fitur edit portofolio berisikan berbagai *form* informasi yang dibutuhkan untuk sebagai informasi pribadi mahasiswa dan alumni yang di lampirkan didalam portofolio. Berikut ini adalah *Flowchart* untuk edit portofolio

- Proses dimulai ketika mahasiswa dan alumni mengakses fitur Edit Portofolio di antarmuka pengguna mereka.
- Sistem menampilkan formulir yang berisi berbagai informasi yang harus diisi oleh mahasiswa dan alumni untuk portofolio mereka.
- Mahasiswa dan alumni mengisi formulir dengan data portofolio yang diperlukan, seperti info *Profile*, *achievement*, *experience*, *softskill*, dan informasi relevan lainnya.

- Sistem memvalidasi input dari formulir untuk memastikan bahwa semua data diisi dengan benar dan dalam format yang sesuai.
- Jika semua input valid, lanjutkan ke langkah berikutnya dan Jika ada kesalahan, tampilkan pesan kesalahan dan minta mahasiswa untuk memperbaiki input.
- Data formulir yang telah diisi dikirim ke *server* untuk diproses lebih lanjut.
- Proses pengeditan portofolio selesai setelah semua langkah selesai dan data berhasil disimpan atau diperbaiki.

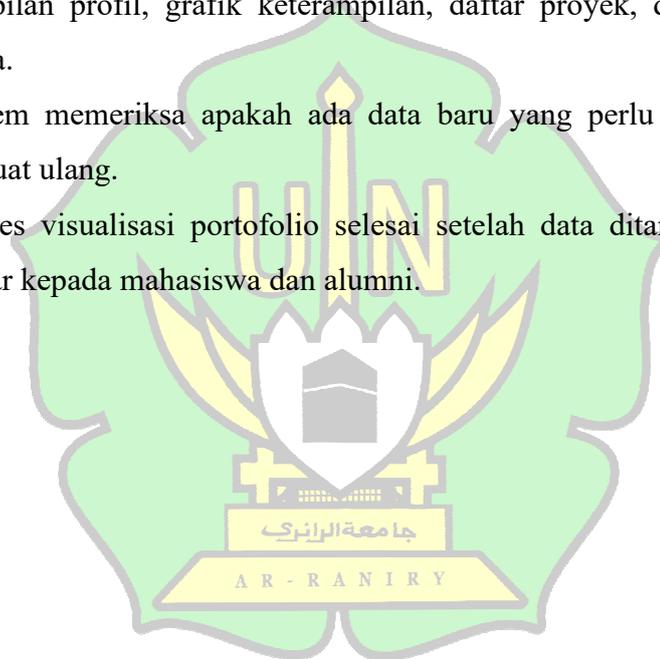


Gambar 4. 17 Flowchart Edit portofolio

C. Portofolio

Fitur Portofolio yang merupakan hasil visualisasi dari data yang diisi pada tahap Edit Portofolio sebelumnya. Berikut ini adalah *Flowchart* untuk portofolio

- Proses dimulai ketika mahasiswa dan alumni mengakses fitur Portofolio untuk melihat data portofolio mereka.
- Sistem mengambil data portofolio yang telah disimpan dari *server* untuk ditampilkan.
- Sistem memproses data yang diambil untuk memastikan format dan informasi siap untuk ditampilkan. Ini termasuk memformat data, mengorganisasi informasi, dan melakukan pengolahan yang diperlukan.
- Sistem menampilkan portofolio dalam format visual yang sesuai, seperti tampilan profil, grafik keterampilan, daftar proyek, dan pengalaman kerja.
- Sistem memeriksa apakah ada data baru yang perlu diperbarui atau dimuat ulang.
- Proses visualisasi portofolio selesai setelah data ditampilkan dengan benar kepada mahasiswa dan alumni.





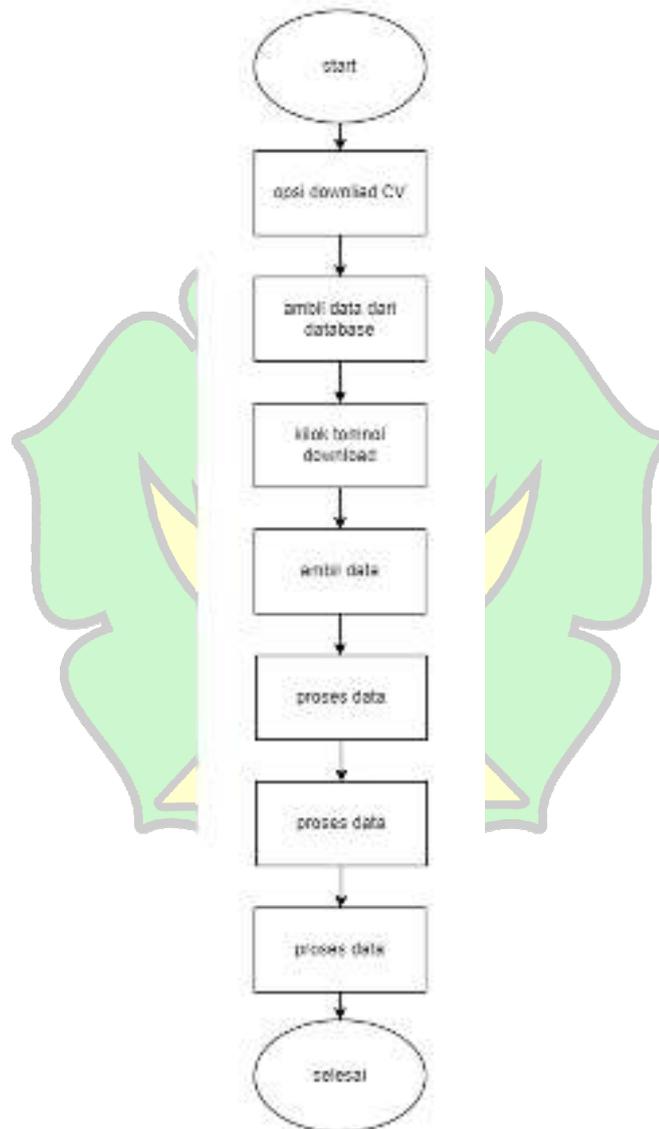
Gambar 4. 18 Flowchart Portofolio

D. *Download CV*

Fitur Portofolio yang merupakan hasil visualisasi dari data yang diisi pada tahap Edit Portofolio sebelumnya. Berikut ini adalah *Flowchart* untuk portofolio

- Proses dimulai ketika mahasiswa atau alumni mengakses fitur *Download CV*.
- Sistem menampilkan opsi atau tombol untuk mengunduh *CV* pada antarmuka pengguna.
- Pengguna mengklik tombol atau opsi untuk memulai unduhan *CV* mereka.
- Sistem mengambil data portofolio terbaru yang telah diisi sebelumnya dari *server*.
- Sistem memproses data yang diambil untuk menyusun format *CV* yang sesuai, termasuk format dokumen seperti PDF.

- Sistem menyediakan *file CV* untuk diunduh oleh pengguna, memungkinkan mereka untuk menyimpan salinan *CV* di perangkat mereka.
- Proses unduhan *CV* selesai setelah *file CV* berhasil diunduh oleh pengguna.

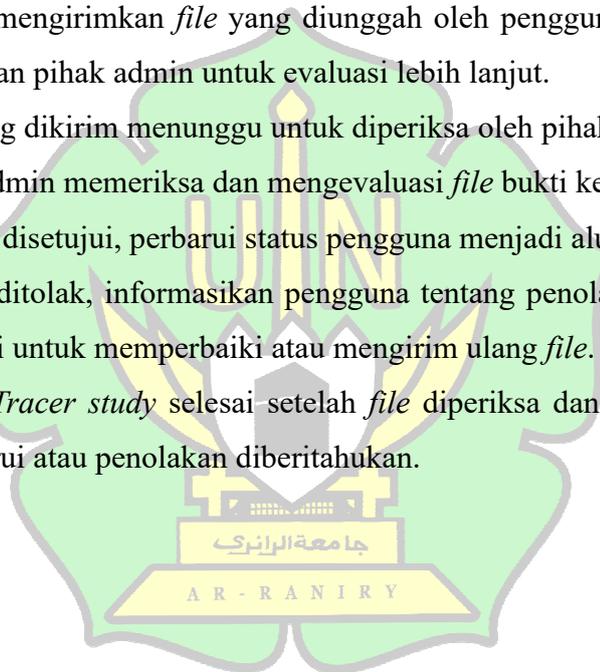


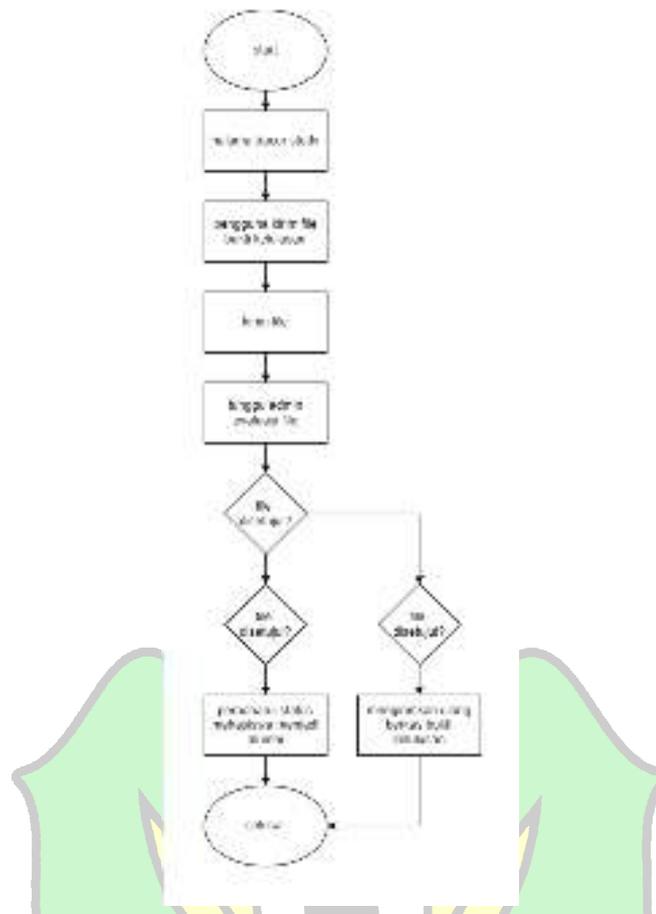
Gambar 4. 19 Flow download *CV*

E. Pengajian *tracer study*

Fitur pengajuan taracer study hanya diperuntukan untuk seorang mahasiswa aktif yang sudah lulus diperkuliahan, pengajuan ini untuk merubah status mereka menjadi seorang alumni dengan mengirimkan bukti kelulusan mereka, berikut ini adalah *Flowchart* dari pengajuan *tracer study* :

- Proses dimulai ketika mahasiswa atau alumni mengakses fitur *Tracer study* di antarmuka pengguna mereka.
- Sistem menampilkan halaman dengan formulir untuk mengirimkan *file* bukti kelulusan.
- Pengguna mengunggah dan mengirimkan *file* yang diperlukan untuk verifikasi kelulusan.
- Sistem mengirimkan *file* yang diunggah oleh pengguna ke *Dashboard* pengajuan pihak admin untuk evaluasi lebih lanjut.
- *File* yang dikirim menunggu untuk diperiksa oleh pihak admin.
- Pihak admin memeriksa dan mengevaluasi *file* bukti kelulusan.
- Jika *file* disetujui, perbarui status pengguna menjadi alumni, dan
- ika *file* ditolak, informasikan pengguna tentang penolakan dan berikan instruksi untuk memperbaiki atau mengirim ulang *file*.
- Proses *Tracer study* selesai setelah *file* diperiksa dan status pengguna diperbarui atau penolakan diberitahukan.





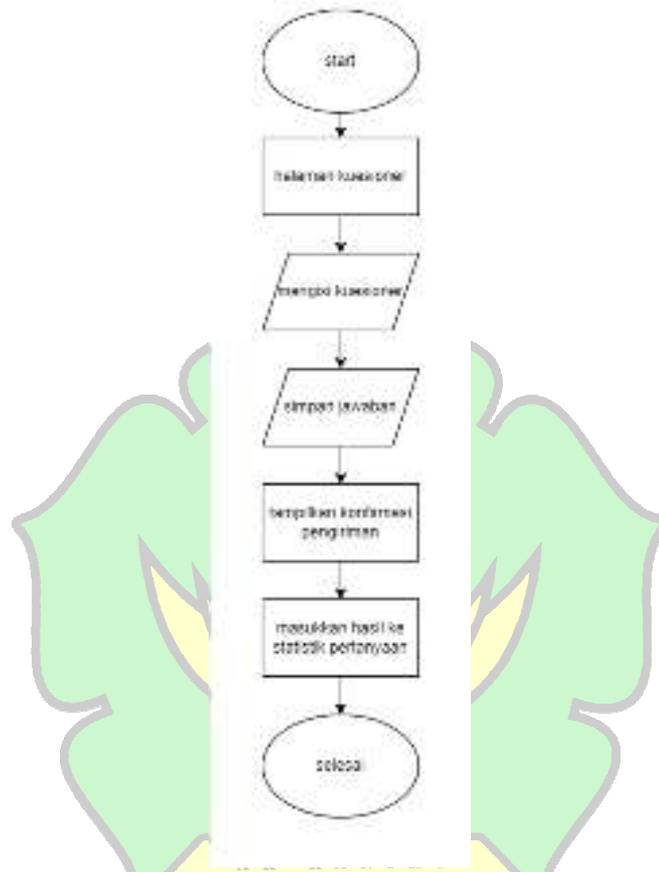
Gambar 4. 20 Flowchart pengajuan tracer study

F. Kuesioner

Fitur kuesioner diperuntukan untuk alumni, mereka mengisi kuesioner yang berisikan pertanyaan yang telah di tambahkan oleh admin pada *Dashboard* pertanyaan. Berikut ini adalah *Flowchart* dari kuesioner :

- Proses dimulai ketika alumni mengakses fitur Kuesioner.
- Menampilkan halaman kuesioner kepada alumni dengan daftar pertanyaan pilihan berganda.
- Alumni mengisi kuesioner dengan memilih jawaban untuk setiap pertanyaan (Input).
- Setelah mengisi kuesioner, alumni mengirimkan jawaban mereka (Output).
- Sistem menyimpan jawaban yang telah dikirimkan ke *database*.

- Menampilkan pesan konfirmasi bahwa jawaban telah berhasil dikirim dan disimpan.
- Hasil jawaban yang disimpan akan diperbarui dalam statistik pertanyaan pada *Dashboard* admin.



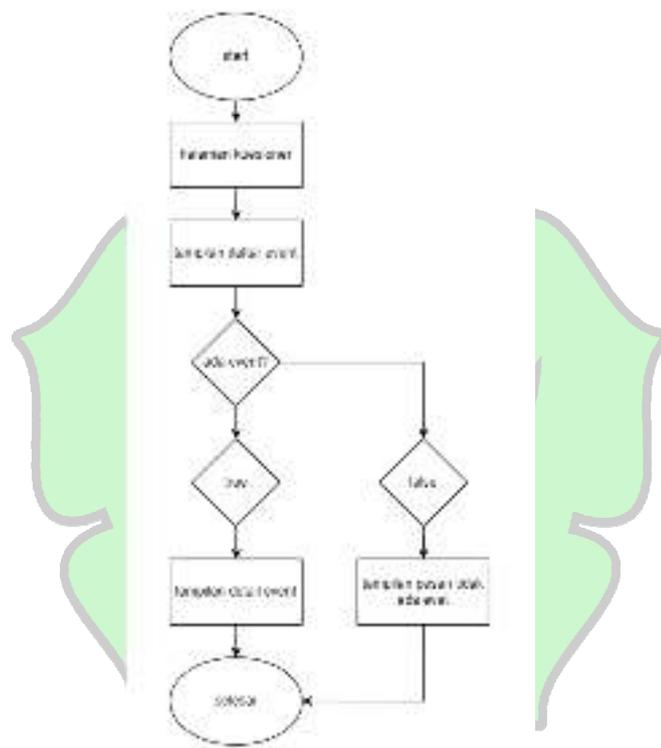
Gambar 4. 21 Flowchart Kuesioner

G. *Event*

Fitur *Event* berisikan informasi terkait *Event* atau kegiatan atau informasi yang bersifat khusus dan di peruntukan untuk para alumni, berikut ini adalah *Flowchart* dari *Event* :

- Proses dimulai ketika pengguna (alumni) mengakses fitur *Event* di *website*.
- Sistem menampilkan daftar semua *Event* yang diatur oleh admin dan diperuntukkan bagi alumni.
- Sistem memeriksa apakah ada *Event* yang perlu ditampilkan.

- Jika ada *Event*, sistem melanjutkan untuk menampilkan detail *Event*, Jika tidak ada *Event*, sistem menampilkan pesan bahwa tidak ada *Event* saat ini.
- Jika ada *Event*, sistem menampilkan informasi lengkap tentang setiap *Event*, termasuk deskripsi, tanggal, waktu, dan lokasi.
- Proses tampilan *Event* selesai setelah informasi tentang *Event* ditampilkan atau pesan ketidakadaan *Event* diberikan.

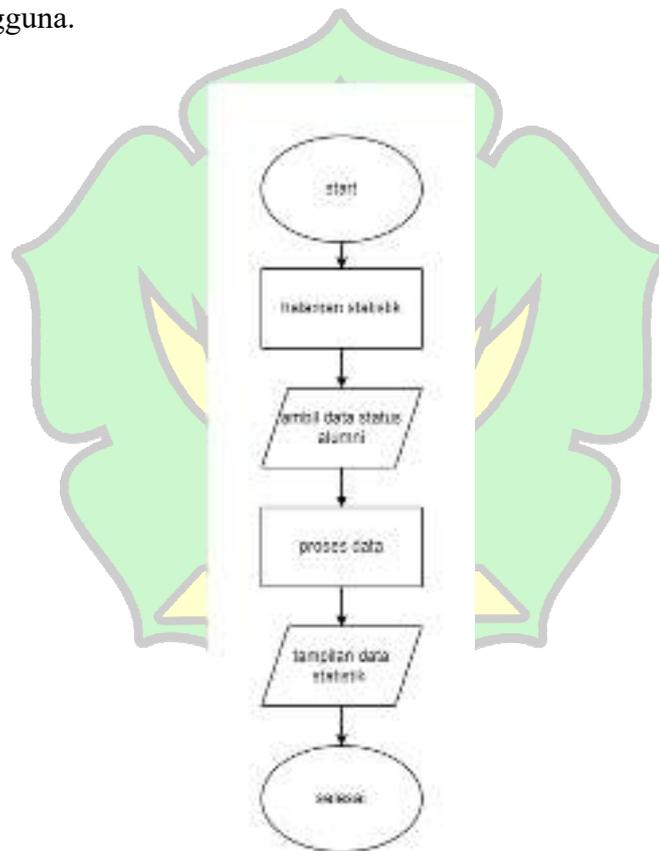


Gambar 4. 22 Flowchart Event

H. Statistik

Fitur statistik berisikan informasi terkait *statistic* dari status pekerjaan seluruh alumni, berikut ini adalah *Flowchart* dari *statistic* :

- Proses dimulai ketika pengguna mengakses fitur Statistik dalam website.
- Sistem mengambil data terkini mengenai status pekerjaan dari seluruh alumni. Ini adalah langkah input di mana data diambil dari *database*.
- Data yang diambil diproses untuk menghitung statistik, seperti chart alumni yang bekerja, mencari pekerjaan, atau tidak bekerja.
- Setelah data diproses, sistem menampilkan informasi statistik dalam bentuk grafik atau tabel yang mudah dibaca. Ini adalah langkah output di mana hasil statistik ditampilkan kepada pengguna.
- Proses tampilan statistik selesai setelah informasi ditampilkan kepada pengguna.



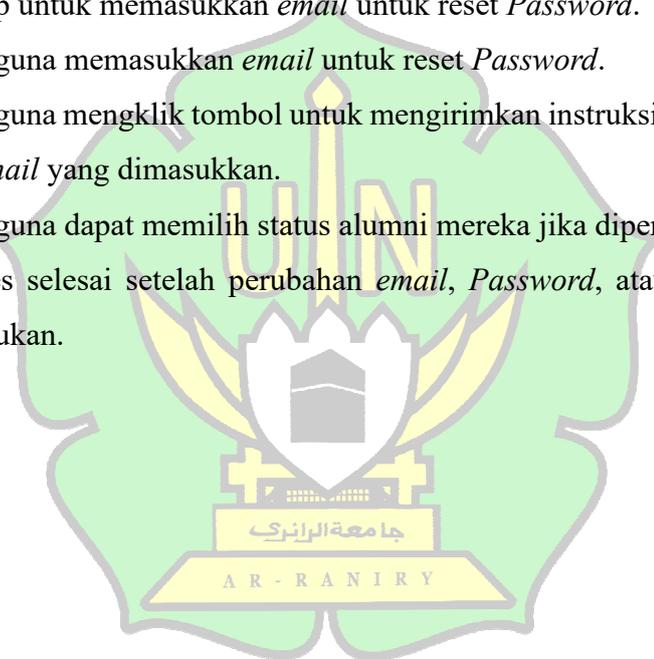
Gambar 4. 23 Flowchart Statistik

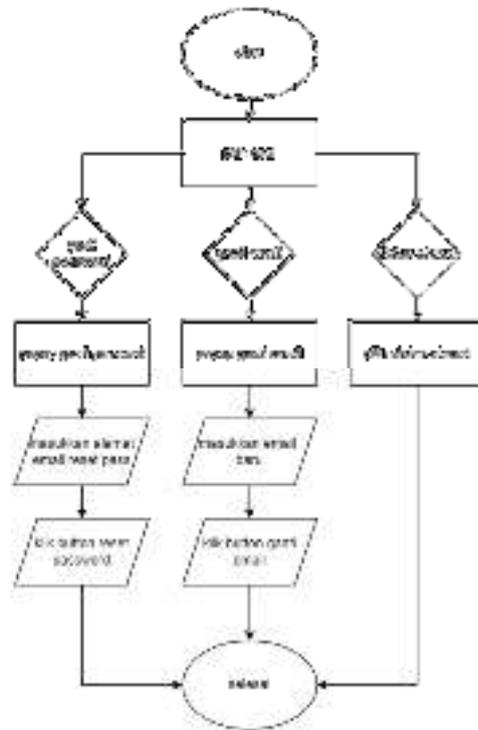
I. Alumni

Fitur alumni berisikan fitur ganti *email*, *Password* dan status alumni. Fitur ini tersedia dikarenakan jika seorang mahasiswa telah menjadi alumni, maka

account google kampus akan dihapus, maka disediakan fitur untuk anti *email* untuk para alumni. Berikut ini adalah *Flowchart* nya :

- Proses dimulai ketika pengguna mengakses fitur Alumni.
- Pengguna memilih opsi yang ingin dilakukan dari fitur Alumni: Ganti *Email*, Ganti *Password*, atau Pilih Status Alumni.
- Jika opsi yang dipilih adalah Ganti *Email*, sistem menampilkan popup untuk memasukkan *email* baru.
- Pengguna memasukkan *email* baru mereka.
- Pengguna mengklik tombol untuk menyimpan *email* baru.
- Jika opsi yang dipilih adalah Ganti *Password*, sistem menampilkan popup untuk memasukkan *email* untuk reset *Password*.
- Pengguna memasukkan *email* untuk reset *Password*.
- Pengguna mengklik tombol untuk mengirimkan instruksi reset *Password* ke *email* yang dimasukkan.
- Pengguna dapat memilih status alumni mereka jika diperlukan.
- Proses selesai setelah perubahan *email*, *Password*, atau status alumni dilakukan.





Gambar 4. 24 Flowchart Alumni

4.3 Konstruksi

Pada tahap Konstruksi, peneliti fokus pada pembuatan perangkat lunak dengan mengikuti desain yang telah ditetapkan sebelumnya. Dalam proses ini, bahasa pemrograman Dart digunakan bersama *Framework* Flutter untuk mengembangkan *website*. Tahap ini melibatkan implementasi elemen-elemen desain ke dalam kode, memastikan bahwa setiap komponen berfungsi sesuai dengan persyaratan situs web.

4.3.1 Fitur Authentication

Adapun hasil dari tampilan aplikasi *web* untuk fitur *Authentication*, diantaranya adalah halaman *Sign In Screen*, dan *Logout*, Penjelasan lebih lanjut dikemas pada poin-poin berikut.

1. Halaman Sign Up

halaman dari hasil *User Story Sign Up* merupakan bagian dari fitur *Authentication*. Halaman ini berfungsi untuk proses *Sign Up* untuk mahasiswa dan alumni. Pada halaman ini, *User* mahasiswa dan alumni diminta untuk mengisi data berisikan nim, nama lengkap, *email* dan *Password*.



Gambar 4. 25 Halaman SignUp

Pada gambar dibawah berfungsi Dalam proses pendaftaran pengguna, validasi domain *email* memastikan alamat *email* yang digunakan sesuai dengan kriteria yang ditetapkan. Fungsi `checkString` mengekstrak domain dari alamat *email* dengan memotong substring mulai dari karakter '@' hingga akhir. Misalnya, untuk *email* `User@student.ar-raniry.ac.id`, fungsi ini akan menghasilkan `@student.ar-raniry.ac.id`. Hasil ini kemudian dibandingkan dengan `emailUIN`, yang merupakan domain *email* yang diizinkan (misalnya, `@student.ar-raniry.ac.id`). Jika hasilnya cocok, pendaftaran diteruskan; jika tidak, aplikasi menampilkan pesan kesalahan, memastikan hanya *email* yang valid yang dapat digunakan untuk mendaftar.

```

if (!checkString(a, == email)) {
  try {
    //add new account
    user = (await FirebaseAuth.instance
      .createUserWithEmailAndPassword(email: a, password: b))
      .user!;

    //add username
    await DatabaseService()
      .user!.firebaseAuth.instance.currentUser!.uid)
      .addUserData(name, nis, email);

    if (!x || !y) {
      await FirebaseAuth.instance.signOut();
      print("Sign up berhasil");
      return true;
    }
  } on FirebaseAuthException catch (e) {
    setError = e.message.toString();
  }
  return false;
} else {
  ScaffoldMessenger.of(context)
    .showSnackBar(SnackBar(content: Text("Sorry, Not ITM Account")));
}
}

```

Gambar 4. 26 penggalan Codingan Signup

2. Halaman *Log In*

Tampilan gambar dibawah adalah halaman dari hasil *User Story Logiin* merupakan bagian dari fitur *Authentication*. Halaman ini berfungsi untuk proses *login User* admin, mahasiswa dan alumni. Pada halaman ini, *User* diminta untuk mengisi dua data, yaitu *email* dan *Password*, sebagai kredensial untuk keperluan otentikasi saat melakukan *login* ke aplikasi.



Gambar 4. 27 Halaman *Login*

Gambar dibawah merupakan *function* untuk menangani fitur *login*. Pada *function* terdapat 2 parameter string yaitu *email* dan *Password*. Parameter pada *function* ini berfungsi sebagai data yang akan dikirimkan ke *firebase*. Ketika pengiriman data ke *firebase* berhasil serta *email* dan *Passwordnya* sesuai dengan

data pada *firebase*, maka *function* akan mengembalikan nilai *true*. Nama jika terjadi kesalahan pada saat pengiriman data, maka akan dikembalikan pesan *error* yang dikirim oleh *firebase*.

```
//login
Future loginWithEmailAndPassword(String email, String password) async {
  try{
    User user =
      (await FirebaseAuth.signInWithEmailAndPassword(
        email: email, password: password)).user!;

    if (user != null){
      return true;
    }
  } on FirebaseAuthException catch (e) {
    return e.message;
  }
}
```

Gambar 4. 28 Penggalan codingan fitur *login*

3. Fitur *Logout*

Tampilan pada Gambar dibawah adalah halaman dari hasil *User Story Logout* merupakan bagian dari fitur *Authentication*. Halaman ini berfungsi untuk proses *Logout User* admin, mahasiswa dan alumni. Pada halaman ini, *User* admin akan mengkilik *button* “*Logout*” yang telah disediakan diatas kanan layar untuk mengganti status *Authentication User* dari yang sebelumnya *login* menjadi *guest* (tidak *login*).



Gambar 4. 29 Logout

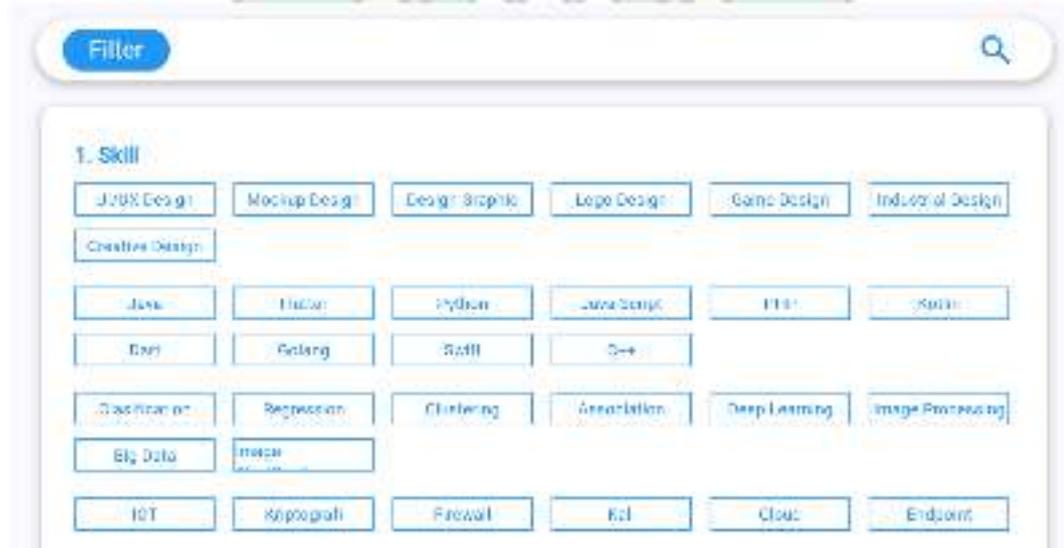
Gambar 4.17 merupakan *function* untuk menangani fitur *Logout*. Pada *function* dengan nama *signOut* ini, akan dilakukan *sign out (Logout)*. Ketika *Logout* yang dilakukan gagal maka akan dikembalikan pesan *error* yang dikirim oleh *firebase*.

```
//signout
Future signOut() async {
  try{
    await firebaseAuth.signOut();
  } on FirebaseAuthException catch (e) {
    return e.message;
  }
}
```

Gambar 4. 30 Codingan Logout

4.3.2 *Fitur Search*

Fitur *Search* berfungsi untuk pencarian nama untuk portofolio mahasiswa dan alumni. Didalam fitur tersebut juga tersematkan filter tambahan, yang Dimana *User* dapat mencari mahasiswa dengan kriteria keterampilan tertentu.



Gambar 4. 31 Tampilan Search

Pada gambar (), Bagian Filtering Nama Pengguna dari fungsi *SearchNamaUser* bekerja dengan memeriksa setiap nama pengguna untuk melihat apakah nama tersebut mengandung substring yang sesuai dengan nama yang dicari.

Proses ini dilakukan dengan membandingkan setiap karakter nama yang dicari dengan karakter nama pengguna secara case-insensitive. Jika nama pengguna cocok, pengguna tersebut ditambahkan ke hasil pencarian. Metode ini membantu dalam menemukan dan menyaring pengguna berdasarkan nama dengan akurasi dan relevansi.

Bagian Menyortir Hasil Pencarian Berdasarkan Keterampilan mengurutkan daftar hasil pencarian `hasilSearch` berdasarkan jumlah keterampilan pengguna. Menggunakan metode bubble sort, elemen dalam daftar dibandingkan dan ditukar jika keterampilan pengguna pada indeks saat ini kurang dari pada indeks berikutnya. Dengan cara ini, daftar `hasilSearch` diurutkan dari pengguna dengan keterampilan terbanyak ke pengguna dengan keterampilan paling sedikit.

```
searchNamaUser() {}
hasilSearch.clear();

//filter nama
for (int i = 0; i < seluruhDataPengguna.length; i++) {
    boolean benar = false;
    for (int j = 0; j < namaYangDiSearch.length; j++) {
        if (namaYangDiSearch[j].toLowerCase() ==
            seluruhDataPengguna[i].nama.toLowerCase().charAt(j)) {
            benar = true;
        } else {
            benar = false;
            break;
        }
    }
    if (benar == true) {
        hasilSearch.add(seluruhDataPengguna[i]);
    }
}

for (int i = 0; i < hasilSearch.length - 1; i++) {
    database temp = new database();
    if (hitungTotalSkill(hasilSearch, i) <
        hitungTotalSkill(hasilSearch, i + 1)) {
        temp = hasilSearch[i];
        hasilSearch[i] = hasilSearch[i + 1];
        hasilSearch[i + 1] = temp;
    }
}
```

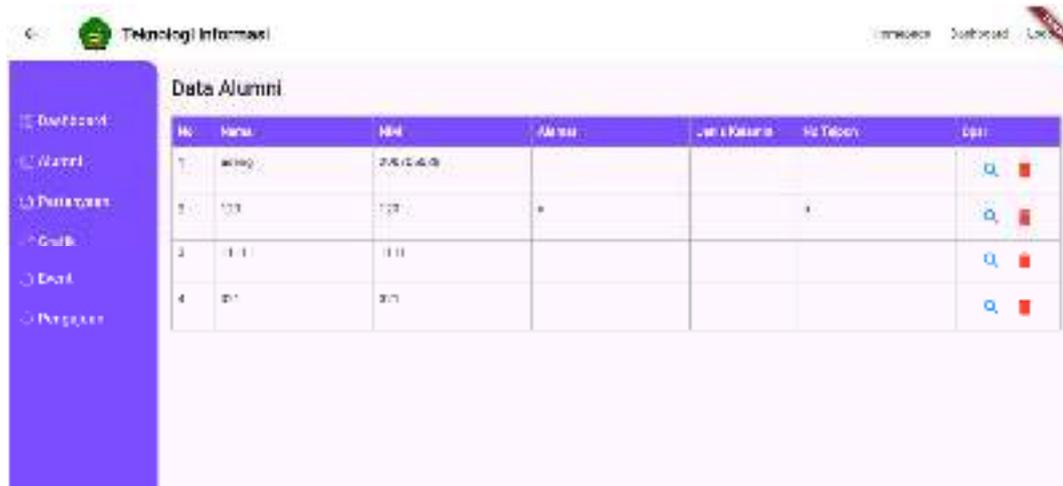
Gambar 4. 32 Penggalan codingan search

4.3.3 Fitur Dashboard admin

Adapun hasil dari tampilan *website* untuk fitur *Dashboard admin*, diantaranya adalah fitur *Dashboard alumni*, *Dashboard pertanyaan*, *Dashboard grafik*, *Dashboard Event* dan pengajuan *tracerstudy*. Penjelasan lebih lanjut dikemas pada poin-poin berikut.

1. *Dashboard* alumni

Tampilan pada Gambar 4.24 adalah hasil *User Story Dashboard* alumni merupakan tempat Dimana *User* admin dapat melihat data dari alumni yang sudah terdaftar dan dapat melihat portofolio mereka secara keseluruhan.



No	Nama	NIM	Alamat	Jenis Kelamin	No Telepon	Aksi
1	ahmad	2016040				 
2	123	123				 
3	111	111				 
4	211	211				 

Gambar 4. 33 Tampilan dashboard alumni

Berikut ini merupakan codigan untuk mengambil data dari firebase. Metode ``getDataAlumni()`` bertugas mengambil data alumni dari koleksi "*User*" di *Firestore*. Pertama, metode ini menghapus semua data lama di list ``SeluruhAlumni`` untuk memastikan list kosong sebelum data baru diambil. Kemudian, metode ini melakukan query pada koleksi "*User*" dan memproses setiap dokumen yang memiliki field ``Alumni`` bernilai *true*. Untuk setiap dokumen yang valid, metode ini membuat objek baru dari kelas ``alumni``, menetapkan berbagai atribut seperti nomor urut, NIM, nama, alamat, jenis kelamin, nomor telepon, dan ID dokumen dari field yang relevan dalam dokumen *Firestore*. Setelah data diproses, objek

`alumni` yang baru ditambahkan ke dalam list `SeluruhAlumni`, sehingga list tersebut diperbarui dengan informasi terbaru mengenai semua alumni.

```

getDatoAlumni() async {
  SeluruhAlumni.clear();
  await FirebaseFirestore.instance.collection("alumni").get().then(
    (value) {
      value.data.forEach((element) {
        if (element["alumni"]) {
          SeluruhAlumni.add(element);
          SeluruhAlumni[SeluruhAlumni.length - 1].no =
            SeluruhAlumni.length.toString();
          SeluruhAlumni[SeluruhAlumni.length - 1].nama = element["nama"];
          SeluruhAlumni[SeluruhAlumni.length - 1].asalat = element["asalat"];
          SeluruhAlumni[SeluruhAlumni.length - 1].jenisKelamin =
            element["jenisKelamin"];
          SeluruhAlumni[SeluruhAlumni.length - 1].telepon =
            element["handphone"];
          SeluruhAlumni[SeluruhAlumni.length - 1].id = element.id;
        }
      });
    }
  );
}

```

Gambar 4. 34 Penggalan codingan dashboard alumni

2. *Dashboard* pertanyaan

Tampilan pada Gambar 4.24 adalah hasil *User Story Dashboard* pertanyaan merupakan tempat Dimana *User* admin dapat menambahkan dan menghapus pertanyaan yang akan diajukan kepada para alumni pada kuesioner *tracer study*.



Gambar 4. 35 Tampilan dashboard pertanyaan

Berikut ini merupakan codigan fitur tambah pertanyaan. Bagian kode ini mengelola proses menambahkan pertanyaan baru ke dalam sistem. Saat pengguna

mengetuk tombol "Tambah", data dari field input diperbarui dalam objek `alumniData` dan dikirim ke Firebase menggunakan metode `tambahPertanyaan` dari `Databaseservice`. Setelah data ditambahkan, field input dibersihkan, tampilan diperbarui, dan dialog input ditutup.

```

DashboardPertanyaan:
  onTap: () {
    // Menanyakan user pertanyaan dengan nilai yang diinputkan pengguna
    alumniData[alumniData.length - 1].no = alumniData.length;
    alumniData[alumniData.length - 1].pertanyaan = pertanyaan.text;
    alumniData[alumniData.length - 1].jawabA = jawabanA.text;
    alumniData[alumniData.length - 1].jawabB = jawabanB.text;
    alumniData[alumniData.length - 1].jawabC = jawabanC.text;
    alumniData[alumniData.length - 1].jawabD = jawabanD.text;

    // Menanyakan user soal dan jawaban yang akan dimasukkan ke pertanyaan ke
    // Firebase Database realtime
    userNo = FirebaseAuth.instance.currentUser!.uid;
    // Menambahkan pertanyaan ke Firebase realtime
    // Menambahkan jawaban ke Firebase realtime
  }

```

Gambar 4. 36 Penggalan codingan dashboard pertanyaan

3. Dashboard grafik pertanyaan

Hasil *User Story Dashboard* grafik pertanyaan merupakan tempat Dimana *User* admin dapat melihat hasil dari jawaban *User* alumni dari pertanyaan kuesioner yang diajukan sebelumnya.



Gambar 4. 37 Tampilan dashboard grafik pertanyaan

Berikut ini merupakan codingan fitur tambah pertanyaan. Metode `getData()` bertugas mengambil dan memproses data dari *Firestore* dalam dua langkah utama: mengambil data pertanyaan dan jawaban pengguna. Pertama, metode ini mengosongkan list `seluruhPertanyaan` untuk memastikan bahwa data yang lama tidak mengganggu data yang baru. Kemudian, metode ini mengambil semua dokumen dari koleksi "pertanyaan" di dalam dokumen "0" pada koleksi "Admin" dan memproses setiap dokumen. Setiap dokumen yang diambil ditambahkan sebagai objek baru dari kelas `pertanyaan` ke dalam list `seluruhPertanyaan`, dengan menetapkan atribut seperti pertanyaan, jawaban A hingga D, dan ID dokumen yang dikonversi menjadi integer. Setelah data pertanyaan diproses, metode ini memanggil `susunData()` untuk mengorganisir data yang telah diambil.

```

ADMIN, BANYAK {
  addDataPertanyaan() {
    //ambil data pertanyaan
    await firebase.firestore().collection('Admin')
      .doc('0')
      .collection('pertanyaan')
      .get()
      .then(
        () => {
          seluruhPertanyaan = [];
          seluruhPertanyaan.add(pertanyaan);
          seluruhPertanyaan[seluruhPertanyaan.length - 1].jawabanA =
            jawabanA;
          seluruhPertanyaan[seluruhPertanyaan.length - 1].jawabanB =
            jawabanB;
          seluruhPertanyaan[seluruhPertanyaan.length - 1].jawabanC =
            jawabanC;
          seluruhPertanyaan[seluruhPertanyaan.length - 1].jawabanD =
            jawabanD;
          seluruhPertanyaan[seluruhPertanyaan.length - 1].doc =
            doc;
        }
      );
    susunData();
  }
}

const firebaseAdmin = firebase.firestore().collection('Admin').get().then(
  () => {
    value.doc().collection('pertanyaan')
      .get().then(
        () => {
          if (document.getElementById('id')) {
            jawabanBaru.add({
              pertanyaan: pertanyaan,
              jawabanA: jawabanA,
              jawabanB: jawabanB,
              jawabanC: jawabanC,
              jawabanD: jawabanD,
              doc: doc
            });
          }
        }
      );
  }
);

```

Gambar 4. 38 Penggalan codingan dashboard grafik pertanyaan

4. Dashboard Event

Hasil *User Story Dashboard Event* merupakan tempat Dimana *User* admin dapat menambah *Event* baru untuk para alumni. *Event* berisikan informasi terkait kegiatan yang melibatkan para alumni.

No	Judul	Tempat	Tanggal	Deskripsi/Post
1
2
3
4

Gambar 4. 39 Tampilan dashboard event

Berikut ini merupakan codingan fitur tambah *Event*. `GestureDetector` pada kode ini menangani penambahan data *Event* ketika pengguna mengetuk tombol "Tambah". Prosesnya dimulai dengan mengupdate elemen terakhir di list `alumniData` dengan nilai-nilai dari `TextEditingController` seperti judul, tempat, tanggal, dan deskripsi *Event*. Setelah itu, data ini dikirim ke *Firestore* melalui metode `tambahEvent()` dari kelas `Databaseservice`. Selanjutnya, *form* direset dengan `clearEditor()`, tampilan UI diperbarui menggunakan `setState()`, dan dialog saat ini ditutup dengan `Navigator.pop(context)`. Keseluruhan proses memastikan data *Event* baru ditambahkan, UI diperbarui, dan *form* dibersihkan setelah pengguna mengklik tombol.

Berikut ini merupakan codingan fitur *approve* dan tolak pengajuan *tracer study*. terdapat dua *IconButton* yang masing-masing memiliki fungsi untuk menerima atau menolak pengajuan. *IconButton* pertama dengan ikon *Icons.check_box* dan warna biru digunakan untuk menerima pengajuan; ketika tombol ini ditekan, ia memanggil metode *terimaPengajuan* dari *Databaseservice*, yang memperbarui status pengajuan di *Firestore* sesuai dengan ID pengajuan yang terkait. Setelah berhasil, *setState()* dipanggil untuk memperbarui tampilan aplikasi dan mencetak pesan "penerimaan berhasil" di konsol.

IconButton kedua dengan ikon *Icons.delete* dan warna merah digunakan untuk menolak pengajuan; tombol ini memanggil metode *tolakPengajuan* dari *Databaseservice*, yang juga memperbarui status pengajuan di *Firestore*. Setelah pengajuan ditolak, tampilan diperbarui dengan *setState()* dan pesan "penolakan berhasil" dicetak di konsol. Keduanya memastikan bahwa interaksi pengguna dengan data pengajuan dikelola secara dinamis dan responsif dalam aplikasi.



```
IconButton(
  icon: Icon(Icons.check_box,
    color: Colors.blue), // Icon
  onPressed: () async {
    await DatabaseService(
      userID: "") // DatabaseService
      .terimaPengajuan(
        Seluruhpengajuan[i]
          .id);
    setState(() {});
    print("penerimaan berhasil");
  },
), // IconButton
IconButton(
  icon: Icon(Icons.delete,
    color: Colors.red), // Icon
  onPressed: () async {
    await DatabaseService(
      userID: "") // DatabaseService
      .tolakPengajuan(
        Seluruhpengajuan[i]
          .id);
    setState(() {});
    print("penolakan berhasil");
  },
), // IconButton
```

Gambar 4. 42 Penggalan codingan dashboard pengajuan

4.3.4 Fitur Edit Portofolio

Fitur edit portofolio berisikan berbagai *form* yang berisi seluruh informasi yang dibutuhkan dalam terbentuknya sebuah portofolio. Didalam fitur ini terdiri dari 7 *step* pengisian informasi. Fitur ini dapat di akses dan digunakan oleh *User* mahasiswa dan alumni.



Gambar 4. 43 Tampilan edit portofolio

Berikut ini merupakan codingan untuk mengirim data dan menyimpan data salah satu dari edit portofolio yaitu *Profile* info. Fungsi `saveData()` bertugas untuk menyimpan atau memperbarui informasi pengguna di *database Firestore*. Fungsi ini dimulai dengan mendapatkan ID pengguna yang sedang *login* melalui `FirebaseAuth.instance.currentUser!.uid`, lalu membuat instance `Databaseservice` dengan ID tersebut untuk mengakses metode penyimpanan data. Selanjutnya, `saveData()` memanggil metode `addBiodataUser()` dari instance `Databaseservice`, yang menerima berbagai parameter data yang diambil dari kontroler *form* seperti nama, alamat, nomor telepon, *email*, serta link ke profil media sosial dan URL gambar profil pengguna. Fungsi ini menggunakan `await` untuk memastikan bahwa proses penyimpanan atau pembaruan data selesai dengan benar sebelum melanjutkan ke langkah berikutnya. Dengan cara ini, `saveData()` memastikan bahwa informasi pengguna di *Firestore* selalu akurat dan terbaru, mendukung pengelolaan data yang efisien dalam *website*.

```

saveData() async {
  await DatabaseService(userId: FirebaseAuth.instance.currentUser!.uid)
    .postDataUser(
      nameController.text,
      addressController.text,
      phoneController.text,
      emailController.text,
      instagramController.text,
      twitterController.text,
      facebookController.text,
      linkedinController.text,
      githubController.text,
      descriptionController.text,
      uriImage!,);
}

```

Gambar 4. 44 Penggalan codingan edit portofolio

4.3.5 Fitur Portofolio

Fitur portofolio berisikan visualisasi dari seluruh *form* edit portofolio yang sebelumnya diisi oleh mahasiswa dan alumni. Didalamnya menampilkan seluruh berbagai informasi seperti : informasi pribadi, *achievement*, *experience*, *personal skill*, *softskil*, *software skill*, dan *statistic*.



Gambar 4. 45 Tampilan portofolio

Berikut ini merupakan codingan untuk mengambil seluruh data yang di input oleh *User* pada tahap sebelumnya yaitu “edit portofolo” dari firebase. Fungsi ``getDataUser()`` bertujuan untuk mengambil dan memproses data pengguna dari Firebase *Firestore*. Dimulai dengan menghapus data lama melalui pemanggilan ``deleteData()``, fungsi ini kemudian mengakses koleksi ``User`` untuk mendapatkan

informasi dasar seperti nama pengguna, URL gambar profil, deskripsi, serta link ke profil media sosial dan informasi kontak. Setelah itu, data tentang prestasi, pengalaman kerja, keterampilan pribadi, keterampilan teknis, dan perangkat lunak yang dikuasai pengguna diambil dari berbagai sub-koleksi, dengan setiap *item* yang relevan ditambahkan ke daftar lokal yang sesuai.

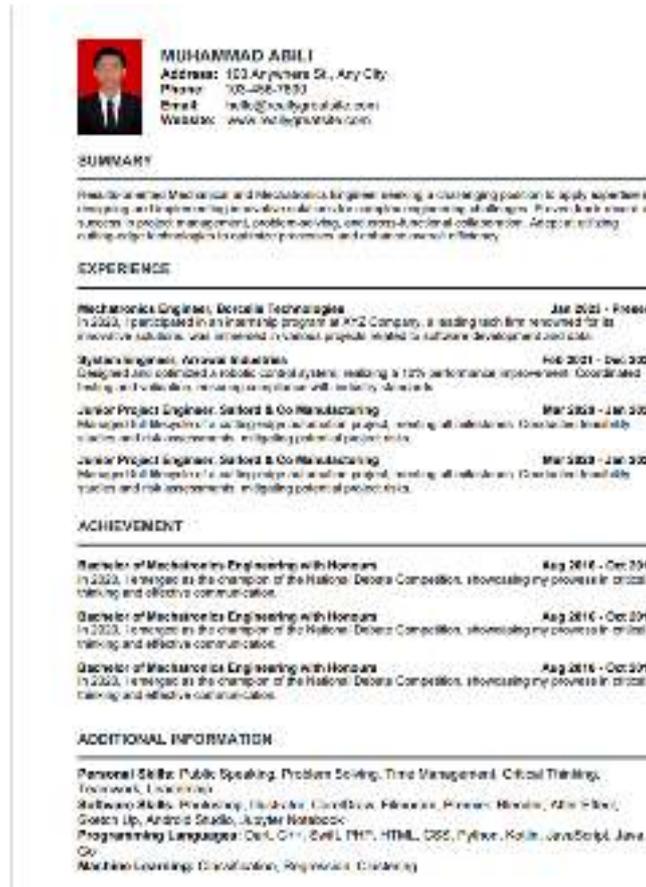
Fungsi ini juga mengambil nilai mata kuliah per semester dari sub-koleksi `nilaiMK`, yang kemudian diolah dan dikategorikan. Akhirnya, data yang telah diambil diproses lebih lanjut untuk pengolahan dan konversi menggunakan beberapa metode tambahan. Dengan langkah-langkah ini, fungsi `getDataUser()` memastikan bahwa aplikasi memiliki data pengguna yang lengkap dan terkini, memungkinkan tampilan dan manajemen informasi pengguna yang akurat.

```
269  getDataUser() async {}
270  deleteData();
271
272  //software
273  setDataPathSoftwareDesignVisual();
274  setDataPathSoftwareMachineLearning();
275  setDataPathSoftwareProgramming();
276  setDataPathSoftwareSecurity();
277  setDataPathSoftwareVideoEditor();
278
279  await FirebaseFirestore.instance
280    .collection("User")
281    .doc(firebaseAuth.instance.currentUser!.uid)
282    .get()
283    .then(
284      (value) {
285        nameUser = value['username'];
286        urlImage = value['linkFotoProfile'];
287        deskripsi = value['deskripsi'];
288        instagram = value['linkInstagram'];
289        github = value['linkGithub'];
290        twitter = value['linkTwitter'];
291        facebook = value['linkFacebook'];
292        linkedin = value['linkLinkedin'];
293        email = value['Email'];
294        whatsapp = value['noHandphone'];
295      },
296    );
```

Gambar 4. 46 Pengalan codingan portfolio

4.3.6 Fitur Download CV

Fitur fitur *download CV* diperuntukan untuk mahasiswa dan alumni. *User* dapat mendownload *CV* yang didalamnya berisi data dan informasi pribadi yang diambil dari portofolio mereka seperti : *experience*, *achievement* dan additional informasi terkait *softskil* mereka. *Download CV* dapat didownload dalam bentuk tipe *file PDF*. Berikut merupakan contoh dari *Generate CV*



Gambar 4. 47 Tampilan *CV*

4.3.7 Fitur Pengajuan Tracer study

Fitur pengajuan *tracer study* diperuntukan untuk mahasiswa dengan kriteria tertentu. Kriteria yang dimaksud adalah mahasiswa yang telah dinyatakan lulus oleh pihak universitas dan mereka ingin merubah status mereka menjadi seorang alumni bukan lagi seorang mahasiswa biasa. Pengajuan tersebut mahasiswa diminta untuk mengupload berkas yang menandakan mereka sudah lulus, seperti contohnya SKL dan ijazah.


```

31 uploadDocbar() async {
32   try {
33     //upload ke firebase storage
34     ref = FirebaseStorage.instance
35       .ref()
36       .child('perguruan/paperstate')
37       .child('$id' + FilePicker1.Files.first.name);
38
39     final metadata = SettableMetadata(contentType: 'application/pdf');
40
41     uploadTask = ref.putData(FilePicker1.Files.first.bytes(), metadata);
42
43     await uploadTask.whenComplete(() => null);
44   } on FirebaseException catch (e) {
45     print(e);
46   }
47 }

```

Gambar 4. 49 Penggalan codingan pengajuan

4.3.8 Fitur kuesioner

Fitur kuesioner ini hanya dapat diakses oleh alumni dan Fitur kuesiner berisikan berbagai pertanyaan yang di berikan oleh admin untuk keperluan *tracer study* para alumni. Alumni mengisi seluruh pertanyaan dan kemudian mengirimkan jawaban mereka. Jawaban mereka disimpan dan di ditampilkan didalam *Dashboard* grafik pada *User* admin.



Gambar 4. 50 Tampilan kuesioner

Berikut ini merupakan codingan untuk mengambil seluruh data pertanyaan beserta jawaban dari menu “*Dashboard* pertanyaan” yang di input oleh admin. Fungsi `loadDataPertanyaan()` bertugas untuk mengambil data pertanyaan dari koleksi *Firestore* yang terletak di sub-koleksi `pertanyaan` dalam dokumen dengan ID "0" pada koleksi "Admin". Fungsi ini mengisi daftar `_questions` dengan objek

tersedia di berikan oleh admin pada *Dashboard Event*. Berisikan judul kegiatan, deskripsi kegiatan, tempat kegiatan dan jadwal kegiatan.



Gambar 4. 52 tampilan event

Berikut ini merupakan codingan untuk mengambil seluruh data *Event* beserta dari menu "*Dashboard Event*" yang di input oleh admin. Fungsi `getData()` berfungsi untuk mengambil data *Event* dari koleksi *Firestore* yang terletak di sub-koleksi `'Event'` dalam dokumen dengan ID "0" pada koleksi "Admin". Fungsi ini mengambil semua dokumen dalam koleksi `'Event'` dan kemudian mengisi daftar `'seluruhEvent'` dengan objek `'EventItem'` yang baru, yang masing-masing diisi dengan informasi *Event* dari dokumen, seperti deskripsi, judul, tanggal, dan tempat. Setelah data diambil dan diproses, daftar `'seluruhEvent'` akan memuat semua *Event* yang tersedia, sehingga dapat digunakan untuk ditampilkan di antarmuka pengguna

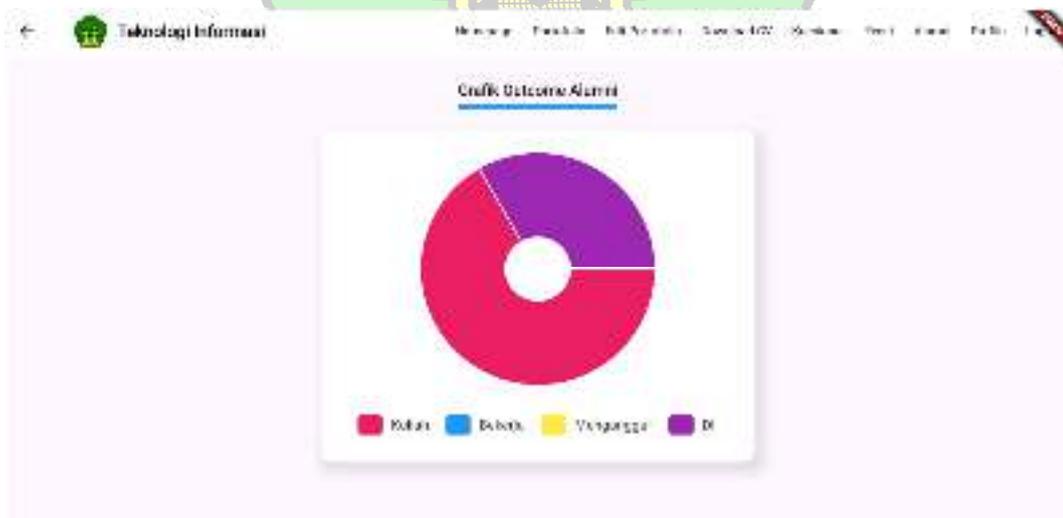
aplikasi. Fungsi ini memastikan bahwa informasi *Event* terbaru dan relevan tersedia dan siap untuk ditampilkan kepada pengguna.

```
getEvent(): async {
  await FirebaseFirestore.instance
    .collection("Admin")
    .doc("9")
    .collection("event")
    .get()
    .then(
      (value) {
        return docs.forEach(
          (element) {
            seluruhEvent.add(eventData());
            seluruhEvent[seluruhEvent.length - 1].deskripsi =
              element["Deskripsi"];
            seluruhEvent[seluruhEvent.length - 1].judul = element["Judul"];
            seluruhEvent[seluruhEvent.length - 1].tanggal = element["Tanggal"];
            seluruhEvent[seluruhEvent.length - 1].tempat = element["Tempat"];
          }
        );
      }
    );
}
```

Gambar 4. 53 penggalan codingan eventr

4.3.10 Fitur statistik

Fitur statistik ini hanya dapat diakses oleh alumni dan Fitur *statistic* berisikan informasi terkait hasil grafik dari keseluruhan alumni tentang status para alumni. Fitur ini menampilkan grafik dengan kriteria didalamnya : kuliah, bekerja, tidak bekerja, dll.



Gambar 4. 54 Tampilan Statistik status alumni

Berikut ini merupakan codingan untuk mengambil seluruh data status alumni yang mana data tersebut didapat dari inputan *User* alumni. Fungsi `getDataGrafik()` bertujuan untuk mengumpulkan dan menghitung data status pekerjaan dari pengguna yang terdaftar sebagai alumni dalam koleksi `'User'` di *Firestore*. Fungsi ini mengambil semua dokumen dari koleksi `'User'`, kemudian memeriksa setiap dokumen untuk menentukan apakah pengguna adalah alumni (berdasarkan nilai `'Alumni'` yang harus `'true'`). Setelah itu, berdasarkan status pekerjaan yang tertera dalam setiap dokumen (`'statusPekerjaan'`), fungsi ini menghitung jumlah alumni yang berada dalam kategori berbeda: "kuliah", "bekerja", "tidak bekerja", dan "dll". Hasil perhitungan ini dapat digunakan untuk menampilkan statistik atau grafik tentang distribusi status pekerjaan alumni di aplikasi, membantu dalam analisis atau pelaporan terkait keterampilan dan penempatan alumni.



```
getDataGrafik() async {
  const firebaseFirestore = firebase.firestore.collection("User").get().then(
    (value) => {
      value.docs.forEach(
        (element) => {
          if (element["Alumni"]) {
            switch (element["statusPekerjaan"].toString()) {
              case "kuliah":
                kuliah++;
                break;
              case "bekerja":
                bekerja++;
                break;
              case "tidak bekerja":
                Menganggur++;
                break;
              case "dll":
                dll++;
                break;
            }
          }
        }
      );
    }
  );
}
```

Gambar 4. 55 Penggalan codingan grafik

4.3.11 Fitur Alumni

Fitur alumni ini hanya dapat diakses oleh alumni dan fitur alumni berisikan *form* yang didalamnya *User* dapat mengganti *email*, *Password* dan status alumni. Fitur ini tersedia dikarenakan, jika seorang mahasiswa telah lulus maka *account*

mereka sebagai mahasiswa akan dihapus. Fitur ini memudahkan agar para alumni dapat mengganti *account* mahasiswa mereka dengan *email* biasa.

4.4 System Testing

Setelah kode program selesai pada tahap konstruksi, langkah selanjutnya adalah melakukan pengujian sistem. Proses ini mencakup pengujian berbagai skenario dengan menggunakan berbagai kasus uji. Sesuai dengan penjelasan di bab sebelumnya, pengujian dilakukan secara menyeluruh dari awal hingga akhir (end-to-end) untuk memastikan bahwa seluruh sistem yang telah terintegrasi berfungsi dengan baik.

Tabel 4. 1 Hasil *Test Case User Admin* Pengujian *Black Box*

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
1.	Authentication	login	Melakukan Login pada website	<ol style="list-style-type: none"> 4. Masuk ke menu login 5. Masukkan Username Password khusus admin yang disediakan 6. Tekan tombol login untuk masuk. 	<i>email</i> = admin@gmail.com <i>Password</i> = admin123	User berhasil masuk ke halaman homepage admin	Berhasil
		Logout	Melakukan Logout pada website	<ol style="list-style-type: none"> 2. User admin menekan menu Logout. 		User keluar dari halaman admin dan mengganti status Authentication User dari yang sebelumnya login menjadi guest (tidak login).	Berhasil
2.	Dashboard Tracer study	Lihat Jumlah alumni keseluruhan		<ol style="list-style-type: none"> 3. Mengklik menu tambah barang. 4. User admin memilih kategori barang yang ingin dihapus. 		2. User admin dapat melihat data jumlah alumni secara keseluruhan	Berhasil

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
		Lihat Jumlah pertanyaan keseluruhan	Lihat jumlah pertanyaan yang sudah dibuat pada menu <i>Dashboard</i> pertanyaan	3. <i>User</i> admin klik menu dashboard pertanyaan 4. Lihat jumlah total pertanyaan yang sudah dibuat		2. <i>User</i> admin dapat melihat data jumlah pertanyaan secara keseluruhan	Berhasil
		Lihat Jumlah <i>Event</i> keseluruhan	Lihat jumlah pertanyaan yang sudah dibuat pada menu <i>Dashboard Event</i>	3. <i>User</i> admin klik menu dashboard <i>Event</i> 4. Lihat jumlah total <i>Event</i> yang sudah dibuat		2. <i>User</i> admin dapat melihat data jumlah <i>Event</i> secara keseluruhan	Berhasil
		Lihat status alumni				2. <i>User</i> admin dapat melihat data jumlah status alumni secara keseluruhan	Berhasil
4.	<i>Dashboard</i> alumni	Portofolio alumni	Melihat portofolio alumni secara keseluruhan	4. Masuk kedalam <i>Dashboard</i> alumni 5. Mengklik icon pencarian pada menu bagian dari data alumni		2. <i>User</i> admin dapat melihat secara langsung data portofolio alumni secara menyeluruh	Berhasil
		Hapus alumni	Menghapus seluruh data dari alumni yang dipilih	3. Masuk kedalam <i>Dashboard</i> alumni 4. Mengklik icon hapus pada menu bagian dari data alumni		2. <i>User</i> admin dapat menghapus secara langsung data portofolio alumni secara menyeluruh	Berhasil
6.	<i>Dashboard</i> pertanyaan	Tambah pertanyaan	Menambah pertanyaan untuk kuesioner para alumni	5. Masuk kedalam <i>Dashboard</i> pertanyaan 6. Klik menu tambah untuk menambah pertanyaan 7. Masukkan pertanyaan		2. Setelah mengisi pertanyaan beserta dengan jawabannya, maka pertanyaan didalam	Berhasil

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
				beserta jawaban pilihannya 8. Klik <i>button</i> tambah		kuesioner juga bertambah	
7.	<i>Dashboard</i> grafik	Grafik pertanyaan	Melihat grafik hasil jawaban alumni pada setiap pertanyaan	3. Masuk kedalam <i>Dashboard</i> grafik 4. Lihat seluruh jawaban pertanyaan para alumni		2. Data yang ditampilkan adalah data yang valid, baik dari jumlah pertanyaan serta juga jawaban dari alumni.	Berhasil
8.	<i>Dashboard</i> Event	Tambah <i>Event</i>	Tambah informasi <i>Event</i> untuk informasi kegiatan yang diperuntukan untuk para alumni	4. Masuk kedalam menu <i>Dashboard</i> <i>Event</i> 5. lalu mengklik <i>button</i> tambah <i>Event</i> . 6. Masukkan <i>form</i> untuk menambah <i>Event</i> seperti, deskripsi, tanggal, Lokasi dll		2. Setelah mengisi <i>Event</i> beserta dengan seluruh informasi yang diperlukan, maka <i>Event</i> juga bertambah.	Berhasil

Tabel 4. 2 Hasil *Test Case* User Mahasiswa Pengujian *Black Box*

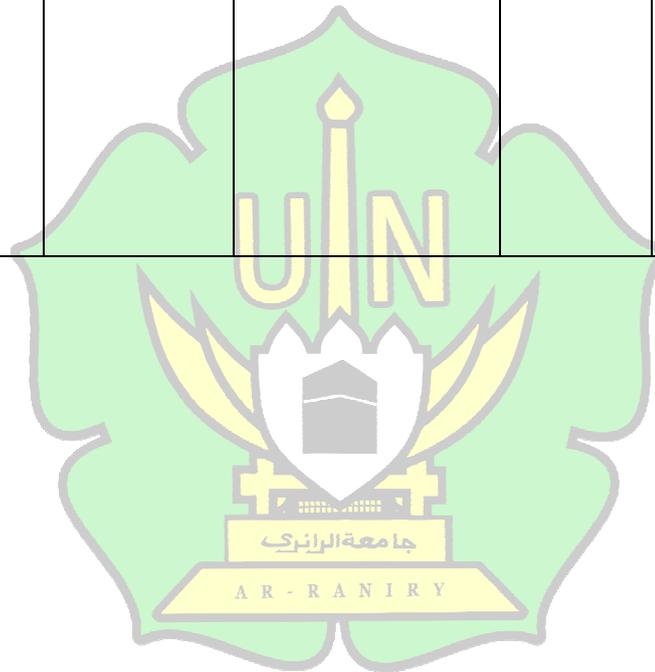
No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
1.	Authentication	login	Melakukan Login pada website	<ol style="list-style-type: none"> 4. Masuk ke menu login 5. Masukkan Username Password telah terdaftar 6. Tekan tombol login untuk masuk 	<i>email =</i> admin@gmail.com <i>Password =</i> admin123	User berhasil masuk ke halaman homepage mahasiswa	Berhasil
		Logout	Melakukan Logout pada website	<ol style="list-style-type: none"> 2. User mahasiswa menekan menu Logout. 		User keluar dari halaman mahasiswa dan mengganti status Authentication User dari yang sebelumnya login menjadi guest (tidak login).	Berhasil
		Sign Up	Melakukan pendaftaran akun	<ol style="list-style-type: none"> 3. Mengklik <i>button SignUp</i> pada login page 4. Memasukkan form pendaftaran akun, terdiri dari : NIM, Username, email, dan Password 		<ol style="list-style-type: none"> 3. User mahasiswa dapat melakukan pendaftaran account mahasiswa 4. Email yang didaftar wajib menggunakan kata kunci "@student.ar-raniry.ac.id" 	Berhasil

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
2.	Edit portofolio	Info Profile	Mengisi informasi umum terkait Profile	<ol style="list-style-type: none"> 4. Mengklik bagian edit portofolio pada bagian menu header 5. Masuk kedalam form info Profile yang berada pada step 1 6. klik button save untuk menyimpan 		2. User mahasiswa mengisi seluruh form yang berisi informasi pribadi mahasiswa dan Ketika ditekan button "save" maka data akan tersimpan di database	Berhasil
		achievement	Mengisi form achievement yang berisikan tahun, nama kompetisi, ranking, deskripsi	<ol style="list-style-type: none"> 5. User mahasiswa masuk kedalam step yang ke 2 6. Isi form achievement 7. Jika achievement lebih dari 1, klik button add disebalah kanan form 8. klik button save untuk menyimpan 		2. User mahasiswa mengisi seluruh form yang berisi achievement mahasiswa dan Ketika ditekan button "save" maka data akan tersimpan di database	Berhasil
		experience	Mengisi form experience yang berisikan tahun, nama institusi, posisi, deskripsi	<ol style="list-style-type: none"> 5. User mahasiswa masuk kedalam step yang ke 3 6. Isi form experience 7. Jika experience lebih dari 1, klik button add disebalah kanan form 8. klik button save untuk menyimpan 		2. User mahasiswa mengisi seluruh form yang berisi experience mahasiswa dan Ketika ditekan button "save" maka data akan tersimpan di database	Berhasil
		Personal skill	Menambahkan personal skill	<ol style="list-style-type: none"> 5. User mahasiswa masuk kedalam step yang ke 4 6. Klik button add jika ingin menambahkan personal skill 		4. User mahasiswa menambah personal skill dan dapat mengisi personal skill	Berhasil

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
				7. Jika ingin menghapus klik <i>button</i> hapus 8. klik <i>button save</i> untuk menyimpan		5. <i>User</i> mahasiswa menghapus <i>personal skill</i> dan dapat menghapus 6. Ketika ditekan <i>button save</i> maka data tersimpan didalam <i>database</i>	
		<i>softskill</i>	Menambahkan keahlian pribadi	4. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 5 5. Pilih <i>softskill</i> sesuai dengan keahlian mahasiswa pada pilihan yang tersedia 6. klik <i>button save</i> untuk menyimpan		3. <i>User</i> mahasiswa dapat memilih <i>softskill</i> yang mereka kuasai pada pilihan yang tersedia dan jika membatalkan pilihan klik Kembali pilihan tersebut 4. Ketika ditekan <i>button save</i> maka data tersimpan didalam <i>database</i>	Berhasil
		<i>Software skill</i>	Menambahkan keahlian pada <i>software</i> tertentu	4. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 6 5. Pilih <i>software skill</i> sesuai dengan keahlian mahasiswa pada pilihan yang tersedia 6. klik <i>button save</i> untuk menyimpan		3. <i>User</i> mahasiswa dapat memilih <i>software skill</i> yang mereka kuasai pada pilihan yang tersedia dan jika membatalkan pilihan klik Kembali pilihan tersebut 4. Ketika ditekan <i>button save</i> maka data tersimpan	Berhasil

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
						didalam database	
		Mata kuliah peminatan	Mengisi nilai mata kuliah yang terkait dengan 3 peminatan	4. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 7 5. Masukkan nilai mata kuliah yang terdaftar didalam <i>form</i> mata kuliah sesuai dengan nilai yang tertera pada <i>khs</i> 6. klik <i>button save</i> untuk menyimpan		3. <i>User</i> mahasiswa dapat mengisi nilai didalam <i>form</i> nilai 4. Ketika ditekan <i>button save</i> maka data tersimpan didalam database	Berhasil
3.	portofolio	Portofolio mahasiswa	Melihat portofolio mahasiswa secara keseluruhan sesuai dengan data yang sudah diisi pada <i>step</i> sebelumnya	6. Klik menu portofolio pada <i>header</i>		2. <i>User</i> mahasiswa dapat melihat seluruh data yang telah diisi pada “edit portofolio” sebelumnya, data telah tersaji semua pada menu portofolio ini	Berhasil
6.	Download CV	Generate CV	Generate CV otomatis dan data didalamnya berisikan data dari portofolio	2. Klik menu <i>download CV</i> pada <i>header</i>		2. <i>User</i> mahasiswa dapat mendownload CV bertipe <i>file</i> PDF	Berhasil

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
7.	Tracer study	Pengajuan tracer study	Pengajuan tracer study berisikan mengirim salah satu berkas yang menyatakan bahwa mahasiswa telah lulus	<ol style="list-style-type: none"> 4. Pilih menu tracer study pada header 5. Pilih file pernyataan kelulusan 6. Klik selesai untuk mengirim 		<ol style="list-style-type: none"> 3. File yang dikirim oleh mahasiswa akan tersampaikan pada Dashboard pengajuan pada Dashboard admin 4. Jika admin approve file tersebut maka mahasiswa yang mengajukan akan berubah status menjadi seorang alumni 	Berhasil



Tabel 4. 3 Hasil *Test Case User* Alumni Pengujian *Black Box*

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
1.	Authenticat ion	login	Melakukan Login pada website	<ol style="list-style-type: none"> 4. Masuk ke menu login 5. Masukkan Username Password telah terdaftar 6. Tekan tombol login untuk masuk 	email = admin@gmail.com Password = admin123	User berhasil masuk ke halaman homepage alumni	Berhasil
		Logout	Melakukan Logout pada website	<ol style="list-style-type: none"> 2. User alumni menekan menu Logout. 		User keluar dari halaman mahasiswa dan mengganti status Authentication User dari yang sebelumnya login menjadi guest (tidak login).	Berhasil
		Sign Up	Melakukan pendaftaran akun	<ol style="list-style-type: none"> 3. Mengklik button SignUp pada login page 4. Memasukkan form pendaftaran akun, terdiri dari : NIM, Username, email, dan Password 		<ol style="list-style-type: none"> 3. User mahasiswa dapat melakukan pendaftaran account mahasiswa 4. Email yang didaftar wajib menggunakan kata kunci "@student.ar-raniry.ac.id" 	Berhasil
2.	Edit portofolio	Info Profile	Mengisi informasi umum terkait Profile	<ol style="list-style-type: none"> 4. Mengklik bagian edit portofolio pada bagian menu header 5. Masuk kedalam form info Profile yang berada pada step 1 		<ol style="list-style-type: none"> 2. User mahasiswa mengisi seluruh form yang berisi informasi pribadi mahasiswa dan Ketika ditekan button 	Berhasil

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
				6. klik <i>button save</i> untuk menyimpan		“ <i>save</i> ” maka data akan tersimpan di <i>database</i>	
		<i>achievement</i>	Mengisi <i>form achievement</i> yang berisikan tahun, nama kompetisi, ranking, deskripsi	5. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 2 6. Isi <i>form achievement</i> 7. Jika <i>achievement</i> lebih dari 1, klik <i>button add</i> disebalah kanan form 8. klik <i>button save</i> untuk menyimpan		2. <i>User</i> mahasiswa mengisi seluruh <i>form</i> yang berisi <i>achievement</i> mahasiswa dan Ketika ditekan <i>button “save”</i> maka data akan tersimpan di <i>database</i>	Berhasil
		<i>experience</i>	Mengisi <i>form experience</i> yang berisikan tahun, nama institusi, posisi, deskripsi	5. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 3 6. Isi <i>form experience</i> 7. Jika <i>experience</i> lebih dari 1, klik <i>button add</i> disebalah kanan form 8. klik <i>button save</i> untuk menyimpan		2. <i>User</i> mahasiswa mengisi seluruh <i>form</i> yang berisi <i>experience</i> mahasiswa dan Ketika ditekan <i>button “save”</i> maka data akan tersimpan di <i>database</i>	Berhasil
		<i>Personal skill</i>	Menambahkan <i>personal skill</i>	5. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 4 6. Klik <i>button add</i> jika ingin menambahkan <i>personal skill</i> 7. Jika ingin menghapus klik <i>button hapus</i> 8. klik <i>button save</i> untuk menyimpan		4. <i>User</i> mahasiswa menambah <i>personal skill</i> dan dapat mengisi <i>personal skill</i> 5. <i>User</i> mahasiswa menghapus <i>personal skill</i> dan dapat menghapus	Berhasil

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
						6. Ketika ditekan <i>button save</i> maka data tersimpan didalam <i>database</i>	
		<i>softskill</i>	Menambahka n keahlian pribadi	4. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 5 5. Pilih <i>softskill</i> sesuai dengan keahlian mahasiswa pada pilihan yang tersedia 6. klik <i>button save</i> untuk menyimpan		3. <i>User</i> mahasiswa dapat memilih <i>softskill</i> yang mereka kuasai pada pilihan yang tersedia dan jika membatalkan pilihan klik Kembali pilihan tersebut 4. Ketika ditekan <i>button save</i> maka data tersimpan didalam <i>database</i>	Berhasil
		<i>Software skill</i>	Menambahka n keahlian pada <i>software</i> tertentu	4. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 6 5. Pilih <i>software skill</i> sesuai dengan keahlian mahasiswa pada pilihan yang tersedia 6. klik <i>button save</i> untuk menyimpan		3. <i>User</i> mahasiswa dapat memilih <i>software skill</i> yang mereka kuasai pada pilihan yang tersedia dan jika membatalkan pilihan klik Kembali pilihan tersebut 4. Ketika ditekan <i>button save</i> maka data tersimpan didalam <i>database</i>	Berhasil

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
		Mata kuliah peminatan	Mengisi nilai mata kuliah yang terkait dengan 3 peminatan	<ol style="list-style-type: none"> 4. <i>User</i> mahasiswa masuk kedalam <i>step</i> yang ke 7 5. Masukkan nilai mata kuliah yang terdaftar didalam <i>form</i> mata kuliah sesuai dengan nilai yang tertera pada khs 6. klik <i>button save</i> untuk menyimpan 		<ol style="list-style-type: none"> 3. <i>User</i> mahasiswa dapat mengisi nilai didalam <i>form</i> nilai 4. Ketika ditekan <i>button save</i> maka data tersimpan didalam <i>database</i> 	Berhasil
3.	portofolio	Portofolio mahasiswa	Melihat portofolio mahasiswa secara keseluruhan sesuai dengan data yang sudah diisi pada <i>step</i> sebelumnya	<ol style="list-style-type: none"> 2. Klik menu portofolio pada <i>header</i> 		<ol style="list-style-type: none"> 2. <i>User</i> mahasiswa dapat melihat seluruh data yang telah diisi pada “edit portofolio” sebelumnya, data telah tersaji semua pada menu portofolio ini 	Berhasil
6.	Download CV	Generate CV	Generate CV otomatis dan data didalamnya berisikan data dari portofolio	<ol style="list-style-type: none"> 2. Klik menu <i>download CV</i> pada <i>header</i> 		<ol style="list-style-type: none"> 2. <i>User</i> mahasiswa dapat mendownload CV bertipe <i>file</i> PDF 	Berhasil
7.	Kuesioner	Menjawab kuesioner	Isi <i>form</i> kuesioner dengan pertanyaan yang telah diisi oleh admin	<ol style="list-style-type: none"> 4. Pilih menu kuesioner pada <i>header</i> 5. Pilih jawaban dari <i>form</i> pertanyaan yang tersedia 6. Klik <i>save</i> untuk mengirim 		<ol style="list-style-type: none"> 2. Seluruh jawaban dari <i>User</i> alumni akan terekam dan akan terkirim kedalam <i>database</i> 	Berhasil

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
8.	Event	Melihat <i>Event</i> /kegiatan	<i>Event</i> yang tersedia berasal dari <i>Event</i> yang telah diisi oleh admin	2. Pilih menu <i>Event</i> pada <i>header</i>		2. <i>Event</i> yang tersedia pada menu <i>Event</i> , seharusnya berisikan data yang diinput oleh admin ada <i>Dashboard Event</i>	Berhasil
9.	Statistik	Melihat <i>statistic</i> status alumni	<i>Statistic</i> yang tersedia berasal dari kalkulasi status pekerjaan alumni yang telah mengisi status alumni	2. Pilih menu <i>Event</i> pada <i>header</i>		2. Statistik yang tersedia seharusnya berisikan data dari kalkulasi status alumni yang sudah terdaftar pada <i>website</i> ini secara keseluruhan	Berhasil
10.	Alumni	Ganti <i>email</i> , <i>Password</i> dan status alumni	Mengganti <i>email</i> yang sudah terdaftar sebelumnya menggunakan akun kampus, <i>Password</i> dan status alumni	5. Klik <i>button ganti email</i> 6. Klik <i>button ganti Password</i> 7. Pilih status alumni 8. Klik <i>save</i> untuk mengirim		4. Ketika <i>button ganti email</i> ditekan maka muncul popup untuk mengisi <i>email</i> yang baru, kemudian klik ganti <i>email</i> maka akan terkirim konfirmasi otomatis melalui pesan <i>email</i> dari <i>email</i> tersebut 5. Ketika ditekan <i>button ganti Password</i> maka akan muncul popup untuk mengisi <i>email</i> yang terdaftar didaman akun alumni saat itu, dan	Berhasil

No	Feature	User Story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/ Fail
						kemudian klik button ganti Password maka secara otomatis akan terkirim ke Alamat email tersebut sebuah pesan konfirmasi untuk mengganti Password	



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari penjelesan pada bab sebelumnya, maka dapat disimpulkan beberapa hal berikut :

1. *Website Dashboard* data portofolio mahasiswa dan alumni menghasilkan beberapa fitur yaitu *Authentication*, *Dashboard admin*, *edit portofolio*, *portofolio*, *download CV*, *pengajuan tracer study*, *kuesioner*, *Event*, *statistic* dan alumni.
2. Fitur-fitur yang dibangun berjalan dengan baik dari hasil pengujian *Black Box* yang diuji.

5.2 Saran

Setelah penelitian ini, peneliti memiliki beberapa saran yang dapat diterapkan pada penelitian berikutnya. Adapun saran tersebut adalah sebagai berikut :

1. Pengembangan *Dashboard* data portofolio mahasiswa dan alumni pada penelitian ini menggunakan metode *Rapid Application Development*, pada penelitian berikutnya dapat dilakukan dengan menerapkan metode pengembangan perangkat lunak lainnya.
2. Pengembangan *Dashboard* data portofolio mahasiswa dan alumni pada penelitian ini menggunakan *Framework flutter*, pada penelitian berikutnya dapat dilakukan dengan menggunakan *Framework* lainnya.

DAFTAR PUSTAKA

153-Article Text-461-1-10-20210506. (n.d.).

Fernando, D. (2018). RANCANG BANGUN APLIKASI PORTOFOLIO DIGITAL HASIL KARYA MAHASISWA UNSERA MENGGUNAKAN METODE SCRUM. *Jurnal Sistem Informasi*, 1.

Hidayat, T., & Muttaqin, M. (2018). Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda *Online* menggunakan *Black Box Testing* dengan Metode Equivalence Partitioning dan Boundary Value Analysis. In *Jurnal Teknik Informatika UNIS JUTIS* (Vol. 6, Issue 1). www.ccssenet.org/cis

Mudharabah, P. (n.d.). RANCANG BANGUN SISTEM ELECTRONIC CUSTOMER RELATIONSHIP MANAGEMENT (E-CRM) PADA.

Putra, A. R. (2018). RANCANG BANGUN SISTEM INFORMASI PORTOFOLIO DIGITAL DOSEN (Studi Kasus : Jurusan Teknik Informatika Universitas Negeri Surabaya). In *Jurnal Manajemen Informatika* (Vol. 8). <https://www.smackcoders.com>.

Rashidi, J., Candra Brata, K., & Hendra Brata, A. (2022). *Pembangunan Aplikasi Program Membership Wahana Go-Kart menggunakan Flutter dengan Pustaka Deepface (Studi Kasus CV Indi Karya Teknik)* (Vol. 6, Issue 7). <http://j-ptiik.ub.ac.id>

Saputri, S. L., Adithama, S. P., & Santoso, A. J. (2022). Pembangunan Sistem Informasi Galeri Portofolio Mahasiswa Arsitektur Universitas Atma Jaya Yogyakarta Berbasis *Website*. In *Jurnal Informatika Atma Jogja* (Vol. 3, Issue 2).

Sofi, N., & Dharmawan, R. (n.d.). PERANCANGAN APLIKASI BENGKEL CSM BERBASIS ANDROID MENGGUNAKAN *FRAMEWORK* FLUTTER (BAHASA DART). *JTS*, 1(2).

tesis-husain-asyari-wijaya-15721251010. (n.d.).

Ulum Fajar, H., Putra Kharisma, A., & Bhawiyuga, A. (2022). *Pengembangan Aplikasi Layanan Kesehatan berbasis Web untuk Skrining Pendengaran menggunakan Arsitektur Clean (Studi Kasus: Fakultas Kedokteran Universitas Brawijaya)* (Vol. 6, Issue 4). <http://j-ptiik.ub.ac.id>

UNIKOM_Waldi Aditia Sirait_BAB II. (n.d.).

Ronal, Yunita, & yuliana. (2022). *Desain Unified Modeling Language (UML) Dalam Perancangan Aplikasi Hauling Trip Di Industri Tambang Batubara. Jurnal Teknik Informatika Dan Sistem Informasi, 9(4).* <http://jurnal.mdp.ac.id>

Vallendito, B. (2020). *PEMODELAN USER INTERFACE DAN USER EXPERIENCE MENGGUNAKAN DESIGN THINKING.*

Pangestu, E. I. (2022). *PERANCANGAN WEBSITE PEMERINTAH DESA SEBAGAI MEDIA PENYEBARAN INFORMASI BAGI MASYARAKAT BERBASIS WEB.*



LAMPIRAN

<https://github.com/hadyikhwan/tugas-akhir-hady-ikhwan.git>

