



KERJA PRAKTIK - IF184801

**Pengembangan Ulang Sisi Frontend Aplikasi
“Eventeer” Berbasis Website**

PT Telekomunikasi Indonesia Tbk

Telkom Corporate University

Digital Amoeba

Jl. Gegerkalong Hilir, Sukarasa, Kec. Sukasari, Kota Bandung, Jawa Barat 40152

Periode: 16 Februari 2023 – 30 Juni 2023

Oleh:

Monica Narda Davita

5025201009

Pembimbing Departemen

Dr. Baskoro Adi P., S.Kom., M.Kom.

Pembimbing Lapangan

Mahathir Aldiafis Mochammad

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2023

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

DAFTAR ISI.....	iv
DAFTAR GAMBAR	ix
DAFTAR KODE SUMBER.....	xiii
LEMBAR PENGESAHAN.....	xviii
KATA PENGANTAR.....	xxi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan.....	2
1.3. Manfaat	3
1.4. Rumusan Masalah.....	3
1.5. Lokasi dan Waktu Kerja Praktik	3
1.6. Metodologi Kerja Praktik	4
1.6.1. Pencarian Kebutuhan	4
1.6.2. Perencanaan.....	4
1.6.3. Desain.....	4
1.6.4. Implementasi Sistem	5
1.6.5. <i>Release and Feedback</i>	5
1.6.6. <i>Track and Monitor</i>	5
1.7. Sistematika Laporan.....	6
1.7.1. Bab I Pendahuluan.....	6

1.7.2.	Bab II Profil Perusahaan.....	6
1.7.3.	Bab III Tinjauan Pustaka	6
1.7.4.	Bab IV Analisis dan Perancangan Infrastruktur Sistem	6
1.7.5.	Bab V Implementasi Sistem	6
1.7.6.	Bab VI Hasil Pengerjaan.....	6
1.7.7.	Bab VII Kesimpulan dan Saran	6
BAB II PROFIL PERUSAHAAN.....		8
2.1.	Profil PT Telekomunikasi Indonesia Tbk & Digital Amoeba	8
2.2.	Lokasi	9
BAB III TINJAUAN PUSTAKA.....		11
3.1.	Pemrograman Web	11
3.2.	HTML.....	11
3.3.	Javascript.....	12
3.4.	ReactJS	12
3.5.	NextJS.....	13
3.6.	Bootstrap	13
3.7.	Agile.....	14
3.8.	Scrum.....	14
BAB IV ANALISIS DAN PERANCANGAN INFRASTRUKTUR SISTEM		16
4.1.	Analisis Sistem.....	16
4.1.1.	Definisi Umum Aplikasi.....	16

4.1.2	Analisis Kebutuhan.....	17
4.2.	Perancangan Infrastruktur Sistem.....	18
4.2.1.	Entity Relationship Diagram (ERD) Eventeer ...	18
4.2.2.	Activity Diagram Login.....	19
4.2.3.	Activity Diagram Continue With Google.....	20
4.2.4.	Activity Diagram Homepage.....	21
4.2.5.	Activity Diagram Edit Profile.....	22
4.2.6.	Activity Diagram Chat	23
4.2.7.	Activity Diagram Notification	24
4.2.2.	Entitiy Relationship Diagram (ERD) Landing Page Digital Amoeba	26
BAB V IMPLEMENTASI SISTEM.....		28
5.1.	Pengembangan Ulang Eventeer	28
5.1.1.	Landing Page	28
5.1.2.	Login	39
5.1.3.	Homepage	47
5.1.4.	Edit Profile	52
5.1.5.	Chat	66
5.1.6.	Notification (Tab Activity)	68
5.2.	Pengembangan Landing Page Digital Amoeba	88
5.2.1.	Introduction (Hero Section).....	88
5.2.2.	History	91
BAB VI HASIL Pengerjaan		94

6.1.	Pengembangan Ulang Eventeer	94
6.1.1.	Landing Page	94
6.1.2.	Login	99
6.1.3.	Homepage	102
6.1.4.	Edit Profile	104
6.1.5.	List Chat.....	108
6.1.6.	Notification.....	110
6.7.	Landing Page Digital Amoeba.....	114
6.7.1.	Introduction	114
6.7.2.	History	115
BAB VII KESIMPULAN DAN SARAN.....		119
7.1.	Kesimpulan.....	119
7.2.	Saran.....	120
DAFTAR PUSTAKA.....		122
BIODATA PENULIS.....		124

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 1 Struktur Organisasi Telkom Corporate University	8
Gambar 2 ERD Eventeer Revamped.....	18
Gambar 3 Activity Diagram: Login	19
Gambar 4 Activity Diagram: Continue with Google	20
Gambar 5 Activity Diagram: Homepage	21
Gambar 6 Activity Diagram: Profile Page.....	22
Gambar 7 Activity Diagram: Chat	23
Gambar 8 Activity Diagram: Notification Tab Activity (Joined Community).....	24
Gambar 9 Activity Diagram: Notification Tab Activity (Managed Community)	24
Gambar 10 Activity Diagram: Notification Tab Updates	25
Gambar 11 ERD Landing Page Digital Amoeba	26
Gambar 12 Tampilan: Landing Page Hero Section.....	94
Gambar 13 Tampilan: Landing Page Advantages Section	95
Gambar 14 Tampilan: Landing Page Features Section	95
Gambar 15 Tampilan: Landing Page Achievement Section	96
Gambar 16 Tampilan: Landing Page Download Section...	96
Gambar 17 Tampilan: Landing Page Testimonial Section	97
Gambar 18 Tampilan: Landing Page Get AccountSection	97
Gambar 19 Tampilan: Landing Page Footer Section	98
Gambar 20 Tampilan: Login Default View	99
Gambar 21 Tampilan: Login with Google.....	99
Gambar 22 Tampilan: Modal Login Sukses	100
Gambar 23 Tampilan: Modal Email Telah Terdaftar	100
Gambar 24 Tampilan: Modal Percobaan Login Maksimum	101
Gambar 25 Tampilan: Modal Email Belum Terdaftar	101

Gambar 26 Tampilan: Homepage Default View.....	102
Gambar 27 Tampilan: Modal Hola Homepage.....	103
Gambar 28 Tampilan: Modal Profile	103
Gambar 29 Tampilan: Edit Profile Personal Profile	104
Gambar 30 Tampilan: Edit Profile Profession	104
Gambar 31 Tampilan: Edit Profile Interest	105
Gambar 32 Tampilan: Edit Profile Portfolio.....	105
Gambar 33 Tampilan: Modal Add Portfolio	106
Gambar 34 Tampilan: Edit Profile Social Links.....	106
Gambar 35 Tampilan: Modal Save Profile Confirmation .	107
Gambar 36 Tampilan: Toast Edit Profile Sukses	107
Gambar 37 Tampilan: List Files Media	108
Gambar 38 Tampilan: List Files Document.....	108
Gambar 39 Tampilan: Modal Show Photo	109
Gambar 40 Tampilan: Notification Page Tab Activity Joined Community	110
Gambar 41 Tampilan: Notification Page Tab Activity Managed Community	111
Gambar 42 Tampilan: Notification Sidebar Tab Activity Joined Community.....	111
Gambar 43 Tampilan: Notification Sidebar Tab Activity Managed Community	112
Gambar 44 Tampilan: Modal Member Request	112
Gambar 45 Tampilan: Modal Member Request Respond	113
Gambar 46 Tampilan: Landing Page Digital Amoeba Hero Section	114
Gambar 47 Tampilan: Landing Page Digital Amoeba History 2018.....	115
Gambar 48 Tampilan: Landing Page Digital Amoeba History 2019.....	115

Gambar 49 Tampilan: Landing Page Digital Amoeba History 2020	116
Gambar 50 Tampilan: Landing Page Digital Amoeba History 2021	116
Gambar 51 Tampilan: Landing Page Digital Amoeba History 2022	117

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 1 Landing page: index.jsx	28
Kode Sumber 2 Landing Page: Navbar.jsx	29
Kode Sumber 3 Landing Page: Navbar.scss	29
Kode Sumber 4 Landing Page: Introduction.jsx	30
Kode Sumber 5 Landing Page: Introduction.scss	30
Kode Sumber 6 Landing Page: Advantages.scss	31
Kode Sumber 7 Landing Page: Advantages.jsx	31
Kode Sumber 8 Landing Page: Features.scss	32
Kode Sumber 9 Landing Page: Features.jsx	32
Kode Sumber 10 Landing Page: Achievement.jsx	33
Kode Sumber 11 Landing Page: Achievement.scss	33
Kode Sumber 12 Landing Page: Download.jsx	34
Kode Sumber 13 Landing Page: Download.scss	34
Kode Sumber 14 Landing Page: Testimonial.scss	35
Kode Sumber 15 Landing Page: Testimonial.jsx	35
Kode Sumber 17 Landing Page: GetAccount.jsx	36
Kode Sumber 16 Landing Page: GetAccount.scss 2/2	36
Kode Sumber 18 Landing Page: GetAccount.scss 1/2	36
Kode Sumber 19 Landing Page: Footer.scss	37
Kode Sumber 20 Landing Page: Footer.jsx	37
Kode Sumber 21 Landing Page: Constant Languages	38
Kode Sumber 22 Login: index.jsx 1/5	39
Kode Sumber 23 Login: index.jsx 3/5	40
Kode Sumber 24 Login: index.jsx 2/5	40
Kode Sumber 25 Login: index.jsx 5/5	41
Kode Sumber 26 Login: index.jsx 4/5	41
Kode Sumber 27 Login: ModalEmailsNotRegistered.jsx ..	42
Kode Sumber 28 Login: ModalEmailsAlreadyRegistered.jsx ..	42

Kode Sumber 29 Login: ModalLoginSuccess.jsx	43
Kode Sumber 30 Login: ModalMaxLoginAttempt.jsx	44
Kode Sumber 31 Component: ModalConfirmation.jsx	45
Kode Sumber 32 Login: NextAtuh.js.....	46
Kode Sumber 33 Homepage: index.jsx 1/3.....	47
Kode Sumber 34 Homepage: index.jsx 2/3.....	48
Kode Sumber 35 Homepage: index.jsx 3/3.....	49
Kode Sumber 36 Homepage: ModalHolaHomepage.jsx...	50
Kode Sumber 37 Homepage: ModalProfile.jsx 2/2	51
Kode Sumber 38 Homepage: ModalProfile.jsx 1/2	51
Kode Sumber 39 Edit Profile: index.jsx 2/11	52
Kode Sumber 40 Edit Profile: index.jsx 1/11	52
Kode Sumber 41 Edit Profile: index.jsx 4/11	53
Kode Sumber 42 Edit Profile: index.jsx 3/11	53
Kode Sumber 43 Edit Profile: index.jsx 6/11	54
Kode Sumber 44 Edit Profile: index.jsx 5/11	54
Kode Sumber 45 Edit Profile: index.jsx 7/11	55
Kode Sumber 46 Edit Profile: index.jsx 8/11	56
Kode Sumber 47 Edit Profile: index.jsx 9/11	56
Kode Sumber 48 Edit Profile: index.jsx 10/11	57
Kode Sumber 49 Edit Profile: index.jsx 11/11	58
Kode Sumber 50 Edit Profile: PersonalProfile.jsx	59
Kode Sumber 51 Edit Profile: Profession.jsx	60
Kode Sumber 52 Edit Profile: Interest.jsx	61
Kode Sumber 53 Edit Profile: Portfolio.jsx	62
Kode Sumber 54 Edit Profile: SocialLinks.jsx	63
Kode Sumber 55 Edit Profile: ModalAddPortfolio.jsx	64
Kode Sumber 56 Edit Profile: ModalSaveEditProfile.jsc ...	65
Kode Sumber 57 Edit Profile: ModalEditProfileSuccess.jsx	65
Kode Sumber 58 Chat: ListFiles.jsx.....	66

Kode Sumber 59 Chat: ModalShowPhoto.jsx	67
Kode Sumber 60 Notification: index.jsx 1/10	68
Kode Sumber 61 Notification: index.jsx 2/10	69
Kode Sumber 62 Notification: index.jsx 3/10	70
Kode Sumber 63 Notification: index.jsx 4/10	71
Kode Sumber 64 Notification: index.jsx 5/10	72
Kode Sumber 65 Notification: index.jsx 6/10	73
Kode Sumber 66 Notification: index.jsx 7/10	74
Kode Sumber 67 Notification: index.jsx 8/10	75
Kode Sumber 68 Notification: index.jsx 9/10	76
Kode Sumber 69 Notification: index.jsx 10/10	77
Kode Sumber 70 Notification: NotificationSidebar.jsx 1/8 .	77
Kode Sumber 71 Notification: NotificationSidebar.jsx 2/8 .	78
Kode Sumber 72 Notification: NotificationSidebar.jsx 3/8 .	79
Kode Sumber 73 Notification: NotificationSidebar.jsx 4/8 .	80
Kode Sumber 74 Notification: NotificationSidebar.jsx 5/8 .	81
Kode Sumber 75 Notification: NotificationSidebar.jsx 6/8 .	82
Kode Sumber 76 Notification: NotificationSidebar.jsx 7/8 .	83
Kode Sumber 77 Notification: NotificationSidebar.jsx 8/8 .	84
Kode Sumber 78 Edit Profile: ManagedCommunity.jsx.....	84
Kode Sumber 79 Edit Profile: JoinedCommunity.jsx.....	85
Kode Sumber 80 Edit Profile: ModalMemberRequestNotif.jsx.....	86
Kode Sumber 81 Edit Profile: Helper dateNotificationFormatter	87
Kode Sumber 82 Landing Page Digital Amoeba: index.jsx	88
Kode Sumber 83 Landing Page DigitalAmoeba: Introduction.jsx	89
Kode Sumber 84 Landing Page Digital Amoeba: Introduction.scss	90

Kode Sumber 85 Landing Page Digital Amoeba: History.jsx91

Kode Sumber 86 Landing Page Digital Amoeba:
History.scss.....92

[Halaman ini sengaja dikosongkan]

**LEMBAR PENGESAHAN
KERJA PRAKTIK**

**Pengembangan Ulang Sisi Frontend Aplikasi “Eventeer”
Berbasis Website**

Oleh:

Monica Narda Davita

5025201009

Disetujui oleh Pembimbing Kerja Praktik:

1. Dr. Baskoro Adi P., S.Kom.,
M.Kom.
NIP. 198702182014041001

(Pembimbing Departemen)

2. Mahathir Aldiafis
Mochammad



(Pembimbing Lapangan)

[Halaman ini sengaja dikosongkan]

Pengembangan Ulang Sisi Frontend Aplikasi “Eventeer” Berbasis Website

Nama Mahasiswa : Monica Narda Davita
NRP :5025201009
Departemen : Teknik Informatika FTEIC-ITS
Pembimbing Departemen : Dr. Baskoro Adi P., S.Kom.,
M.Kom.
Pembimbing Lapangan : Mahathir Aldiafis Mochammad

ABSTRAK

Eventeer merupakan platform inovasi karya Digital Amoeba PT. Telekomunikasi Indonesia Tbk yang memwadhahi berbagai pihak dalam mencari komunitas yang diinginkan, mengatur komunitas yang dibuat, dan menjadi media pembelajaran serta penyebaran informasi khususnya dalam bidang teknologi. Proses perombakan aplikasi Eventeer memakan waktu selama kurang lebih empat (4) bulan dan dikerjakan menggunakan metode Agile. Dari sisi pengembang Frontend, pengerjaan dimulai dari fitur Landing Page, Registrasi dan Login menggunakan email, Telkom Single Sign On (SSO), atau Google SSO, dilanjutkan dengan Homepage, Community Page, Profile, Chat, Notifikasi, Detail Community, Detail Event, Notifikasi, dan diakhiri dengan pengerjaan Final Project. Secara umum, pengembangan ulang aplikasi Eventeer berbasis website dari sisi Frontend telah diselesaikan dengan baik dan memenuhi 90% Minimum Viable Product (MVP) yang ditentukan walaupun masih diperlukan beberapa penyempurnaan.

Kata Kunci : NextJS, Agile, Bootstrap, Pemrograman web, Scrum

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur saya haturkan kepada Tuhan Yesus Kristus yang telah memberi penulis kesempatan dan kekuatan sehingga penulis dapat menyelesaikan kegiatan magang dan laporan kerja praktik dengan tepat waktu. Penyusunan laporan kerja praktik ini dibuat berdasarkan hasil magang penulis di PT. Telekomunikasi Indonesia Tbk (Digital Amoeba). Laporan ini dibuat untuk memenuhi persyaratan kelulusan Program Magang & Studi Independen Bersertifikat Merdeka Belajar Kampus Merdeka (MSIB MBKM) Periode 4 dan juga mata kuliah Magang serta Kerja Praktik Program Pendidikan Strata Satu (S1) Program Studi Teknik Informatika di Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Pengalaman baru yang penulis dapatkan semasa melakukan magang semakin menambah wawasan dan gambaran mengenai cara kerja seorang pengembang Frontend Web di PT. Telekomunikasi Indonesia (Digital Amoeba).

Selama penulis membuat laporan kerja praktik ini, penulis banyak diberi motivasi serta doa dari orangtua, kakak, dan keluarga yang senantiasa selalu ada untuk memberikan dukungan baik dalam rupa moral maupun materi.

Dalam kesempatan ini, penulis berterima kasih kepada pihak-pihak yang telah membantu proses penulisan laporan kerja praktik magang ini. Secara khusus penulis sampaikan terima kasih kepada :

1. Bapak Prof. Dr. Ir. Adi Soeprijanto, M.T. selaku Wakil Rektor I Bidang Akademik dan Kemahasiswaan ITS yang telah memberi persetujuan dan dukungan bagi penulis untuk mengikuti kegiatan magang.

2. Ibu Dr. Eng. Chastine Faticah, S.Kom, M.Kom. selaku Ketua Program Studi Teknik Informatika Institut Teknologi Sepuluh Nopember (ITS) Surabaya yang telah membantu kelancaran kerja magang dengan mengeluarkan surat kerja magang dan rekomendasi mahasiswa.

3. Bapak Ary Mazharuddin Shiddiqi, S.Kom., M.Comp.Sc., Ph.D. selaku Sekretaris I bidang Kemahasiswaan yang telah membimbing dan mengarahkan penulis dalam hal konversi Satuan Kredit Semester (SKS) untuk program magang penulis.

4. Bapak Abdul Munif, S.Kom., M.Sc. selaku Koordinator Tim Konsultasi Kemahasiswaan Teknik Informatika ITS yang selalu sigap dalam mengurus administrasi penulis dalam mendaftar program MSIB MBKM Periode 4.

5. Dr. Baskoro Adi P., S.Kom.,M.Kom. selaku dosen monitoring dan evaluasi serta pembimbing penulis selama melakukan kegiatan magang dan kerja praktik sehingga penulis dapat melakukan konversi SKS.

6. Bapak Mahathir Aldiafis M. Nasution selaku Pembimbing Lapangan sekaligus mentor penulis dalam mendalami dunia Frontend Web untuk platform Eventeer di Digital Amoeba.

7. Rivaldi Agustinus Nugroho Siringoringo selaku rekan penulis di Frontend Web Developer Eventeer yang telah membantu penulis dalam belajar banyak hal baru dan yang selalu berkenan menjawab pertanyaan teknis penulis.

8. Risa Augusta, Farih Sartika, Rayhan Ravianda, Laida Lavenia, dan Reihan Manzis selaku tim Frontend Developer Digital Amoeba yang secara langsung maupun tidak langsung memberi penulis ilmu-ilmu baru.

Akhir kata penulis mengucapkan terima kasih banyak pada semua pihak yang telah membantu penulis dalam proses penyelesaian laporan kerja praktik ini.

Penulis berharap dan berterima kasih atas segala saran dan masukan dari pembaca, menerima saran serta masukan tersebut dengan hati terbuka. Akhir kata, besar harapan penulis agar laporan kerja praktik ini dapat bermanfaat bagi kemajuan semua pihak.

Surabaya, 10 Juli 2023

Monica Narda Davita

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. Latar Belakang

Digital Amoeba merupakan inkubator yang dinaungi oleh PT. Telkom Indonesia dalam mengembangkan ide-ide para karyawannya yang dapat menciptakan inovasi dan talenta-talenta digital. Dalam pencapaian misinya tersebut, Digital Amoeba menerapkan tiga (3) prinsip berikut: Build People, Build Business, dan Build System. Terdapat lebih dari 10.000 talenta, 3000 ide, dan 500 inkubasi yang didukung oleh Digital Amoeba (bdk. Web Digital Amoeba). Salah satu produk yang dikembangkan oleh Digital Amoeba ialah Eventeer, platform *all-in-one* untuk penyelenggaraan berbagai komunitas khususnya komunitas berbasis teknologi.

Kemunculan aplikasi Eventeer tidak terlepas dari fakta bahwa semakin pesatnya penggunaan teknologi yang ada di Indonesia yang secara tidak langsung menyebabkan munculnya pegiat teknologi (Mildawati, 2016). Eventeer hadir untuk mewadahi kegiatan berbagai komunitas mulai dari mencari komunitas, berkomunikasi, mengadakan dan mengatur acara, hingga mendapatkan sertifikat sesuai dengan program yang diikuti. Namun, perkembangan zaman juga memunculkan kebutuhan baru bagi para pengguna sehingga platform Eventeer memerlukan perombakan untuk menjawab kebutuhan-kebutuhan baru tersebut.

Pengembangan platform Eventeer tidak terlepas dari peran para tim teknis yang terdiri dari Frontend Web Developer, Backend, System Analyst yang merangkap menjadi juga Quality Assurance, dan Development Operations. Frontend Web Developer merupakan seorang yang bertugas dalam mengembangkan antarmuka website sehingga pengguna website dapat berinteraksi dengan sistem (Mufti Prasetyo et al., 2022). Oleh karena itu, peran Frontend Web Developer bagi pengembangan aplikasi Eventeer berbasis website sangat krusial, terutama dalam pengembangan ulang Eventeer menjadi aplikasi dengan antarmuka dan beberapa fitur yang baru.

1.2. Tujuan

Adapun tujuan dari pengembangan ulang aplikasi Eventeer berbasis website dari sisi Frontend Web Developer adalah sebagai berikut:

1. Mengembangkan media yang praktis bagi para pencari maupun pencipta berbagai komunitas khususnya komunitas berbasis teknologi.
2. Menyediakan berbagai fitur yang dapat mempermudah aktivitas komunitas dalam berdiskusi dan menyusun program bagi anggota komunitas.
3. Menyebarkan informasi komunitas kepada para anggota maupun pihak eksternal yang tepat sasaran.

1.3. Manfaat

Manfaat yang diperoleh dengan adanya website Eventeer antara lain adalah mempermudah pembuat komunitas untuk menyebarluaskan informasi dan mencari peserta yang tepat sasaran. Selain itu, Eventeer mempermudah pelaksanaan acara dari suatu komunitas dalam platform yang praktis dan memiliki fitur yang cukup lengkap. Dengan demikian, para pembuat komunitas tidak perlu repot-repot menggunakan banyak platform untuk mengelola acaranya dan para pencari komunitas tidak terlalu susah untuk mencari komunitas maupun acara yang mereka minati.

1.4. Rumusan Masalah

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana para peminat teknologi maupun peminat bidang lainnya dapat mengadakan suatu acara dengan praktis?
2. Bagaimana kami menjawab kebutuhan para pencari dan penggiat komunitas tersebut?
3. Bagaimana suatu komunitas dapat menggaet peserta untuk acara mereka secara tepat sasaran?

1.5. Lokasi dan Waktu Kerja Praktik

Kegiatan magang dan kerja praktik dilaksanakan secara luring di kantor Digital Amoeba (Jl. Gegerkalong Hilir, Bandung).

Adapun kerja praktik dimulai pada tanggal 16 Februari 2023 hingga 30 Juni 2023.

1.6. Metodologi Kerja Praktik

Metodologi dalam pembuatan laporan kerja praktik meliputi :

1.6.1. Pencarian Kebutuhan

Langkah pertama yang dilakukan sebelum mengembangkan platform Eventeer versi kedua (pengembangan ulang) ialah pencarian kebutuhan yang dilakukan oleh tim *sales* dan tim produk.

1.6.2. Perencanaan

Selanjutnya ialah tahap perencanaan yang dilakukan oleh Product Owner (PO) dalam menyusun timeline pengerjaan. Pada sisi tim pengembang (AMATech), tahap perencanaan diisi dengan penentuan kerangka kerja dan *design system* apa yang akan digunakan dalam melakukan pengembangan. Untuk sisi Frontend, akan digunakan *tech-stack* ReactJS, NextJS versi 13, dan Bootstrap versi 5.

1.6.3. Desain

Langkah ketiga ialah tahap desain yang dilakukan oleh tim desain (AMAVisual). Pembuatan desain diawali dengan pembuatan *Low Fidelity Design* dan disempurnakan melalui *High Fidelity Design*. Selain AMAVisual, System Analyst juga melakukan perancangan basis data berupa pembuatan *Activity Diagram* dan *Entity Relationship Diagram* (ERD) untuk digunakan saat proses pengembangan oleh AMATech.

1.6.4. Implementasi Sistem

Setelah desain sudah selesai dibuat, maka sistem sudah siap dikembangkan oleh AMATech yang dikerjakan secara paralel. Frontend melakukan *slicing* dan integrasi API, Backend melakukan pembuatan API sesuai kebutuhan Frontend, Quality Assurance membuat dan menguji *test-case* sesuai dengan fitur yang dikembangkan, dan DevOps melakukan konfigurasi *deployment* website serta sistem untuk melihat anomali pada website.

1.6.5. Release and Feedback

Setelah suatu fitur telah dikembangkan maka akan ada langkah selanjutnya yang disebut dengan Release and Feedback. Pada tahapan ini, dilakukan *Sprint Review and Retrospective* untuk menilai dan memberi masukan terhadap pekerjaan yang sudah dilakukan selama satu sprint tersebut.

1.6.6. Track and Monitor

Pada tahapan ini, AMATech melakukan revisi dan penyempurnaan fitur yang telah dilakukan sebagai upaya preventif terjadinya *error* lain. Tahapan ini adalah tahap terakhir pada proses pengembangan Eventeer yang menggunakan metode Agile. Selanjutnya proses akan kembali ke tahap pertama yaitu tahap pencarian kebutuhan.

1.7. Sistematika Laporan

1.7.1. Bab I Pendahuluan

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

1.7.2. Bab II Profil Perusahaan

Bab ini berisi gambaran umum Digital Amoeba mulai dari profil hingga lokasi perusahaan.

1.7.3. Bab III Tinjauan Pustaka

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

1.7.4. Bab IV Analisis dan Perancangan Infrastruktur Sistem

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

1.7.5. Bab V Implementasi Sistem

Bab ini berisi kode sumber yang dikerjakan penulis dalam mengembangkan platform Eventeer dan Landing Page Digital Amoeba.

1.7.6. Bab VI Hasil Pengerjaan

Bab ini menampilkan hasil akhir pengerjaan pengembangan ulang platform Eventeer berbasis website dan Landing Page Digital Amoeba

1.7.7. Bab VII Kesimpulan dan Saran

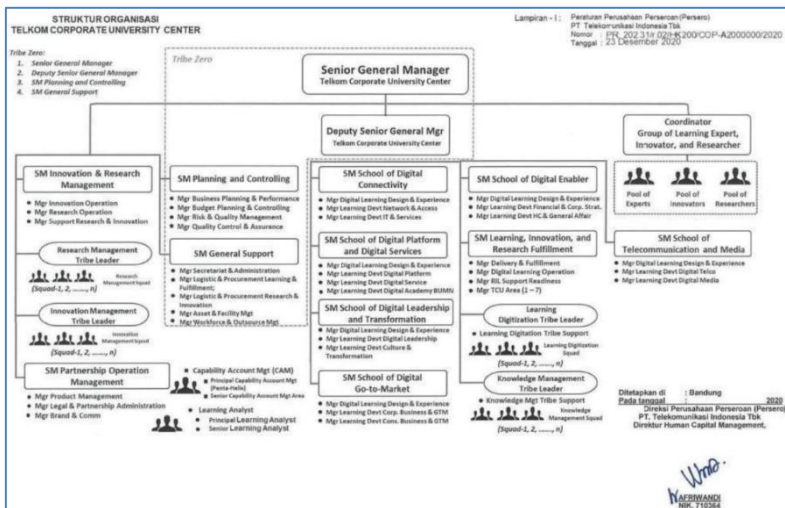
Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

[Halaman ini sengaja dikosongkan]

BAB II PROFIL PERUSAHAAN

2.1. Profil PT Telekomunikasi Indonesia Tbk & Digital Amoeba

PT Telekomunikasi Indonesia Tbk merupakan Badan Usaha Milik Negara (BUMN) yang bergerak di bidang jasa layanan teknologi informasi dan komunikasi (TIK) dan jaringan telekomunikasi. Digital Amoeba merupakan inkubator yang dinaungi oleh PT Telekomunikasi Indonesia Tbk. Berikut merupakan struktur organisasi yang menaungi Digital Amoeba



Gambar 1 Struktur Organisasi Telkom Corporate University

Berdasarkan struktur organisasi di atas, Digital Amoeba berada di divisi Tribe Leader Innovation Management Corpu.

2.2. Lokasi

Jl. Gegerkalong Hilir, Sukarasa, Kec. Sukasari,
Kota Bandung, Jawa Barat 40152

[Halaman ini sengaja dikosongkan]

BAB III

TINJAUAN PUSTAKA

3.1. Pemrograman Web

Web atau World Wide Web adalah ruang informasi yang berisi dokumen dan *resource* web lainnya yang dapat diidentifikasi melalui sebuah URL (Uniform Resource Locators, contohnya www.google.com) dan diakses ketika terkoneksi dengan internet. Halaman penyedia dokumen di dalam web dapat disebut sebagai website yang dapat terkoneksi satu dengan lainnya (*hyperlink*).

Pemrograman web adalah proses pembuatan suatu halaman agar bisa diakses oleh semua orang. Dalam pembuatan website, diperlukan sebuah standar pada website agar semua orang dapat membaca informasi dalam keadaan yang berbeda. Standar tersebut ialah HTML (Hypertext Markup Language). Oleh sebab itu, pemrograman web memiliki tugas untuk menciptakan suatu halaman sesuai standar HTML agar semua orang memiliki akses pada informasi di dalam halaman tersebut (Khairul Kawistara and Hidayatullah, 2020).

3.2. HTML

HTML merupakan bahasa standar internasional yang digunakan untuk membuat halaman web. HTML menggambarkan struktur dan isi semantik dari sebuah dokumen. HTML biasanya digabungkan dengan Cascading Style Sheet (CSS) dan Javascript. CSS untuk memperindah tampilan dan Javascript untuk *client side scripting language* (A. Faradilla., 2022).

3.3. Javascript

Javascript adalah sebuah bahasa tingkat tinggi yang dinamis. Javascript memiliki banyak sekali fungsionalitas seperti *web application*, backend, *desktop application*, Internet of Things (IoT), dan lain - lain. Pada buku kerja praktik ini Javascript digunakan untuk *client side scripting language* yang tertanam pada HTML sebuah website. Javascript juga memiliki banyak *library* yang dapat digunakan contohnya NodeJS, AxiosJS, BluebirdJS, VueJS, AngularJS, ReactJS, AnimateJS, dan lain - lain. Pada kerja praktik ini, penulis menggunakan *library* ReactJS dengan kerangka kerja (*framework*) NextJS (Reactor, 2021).

3.4. ReactJS

ReactJS merupakan *open-source library* yang dilengkapi untuk mengembangkan suatu antar muka secara tidak langsung untuk suatu halaman aplikasi. ReactJS memampukan para pengembangan perangkat lunak untuk menciptakan suatu website yang cukup besar yang memerlukan data dan perubahan pada beberapa waktu tanpa harus memuat ulang halaman (Rawat and Mahajan, 2020). Berbeda dengan Javascript Vanilla yang tidak bisa berisi HTML, pada ReactJS pengembang dapat menuliskan sintaks HTML. Dengan demikian, ReactJS merupakan *library* yang cukup krusial dalam pengembangan sisi Frontend aplikasi website secara praktis.

3.5. NextJS

NextJS merupakan suatu *framework* ReactJS yang memiliki berbagai fitur. Apabila suatu aplikasi berbasis website memerlukan *rendering* dari sisi server (*server-side-rendering*), maka penggunaan ReactJS tidak akan memenuhi kebutuhan tersebut—diperlukan kerangka kerja NextJS (Fariz, Lazuardy and Anggraini, 2022).

Pada kerja praktik penulis, penggunaan NextJS berada pada autentikasi pengguna ketika Login dengan Google dan menyimpan *state* pengguna saat masuk. Selain itu, karena sifat NextJS yang merupakan *server-side-rendering*, penulis memanfaatkan fitur Image Optimization milik NextJS.

3.6. Bootstrap

Bootstrap merupakan sistem *open-source* dan gratis bagi pengembangan sisi frontend aplikasi berbasis website. Bootstrap terdiri dari kerangka HTML dan CSS untuk tipografi, bentuk, *route*, dan berbagai bagian antar muka lainnya—serta memungkinkan penggunaan Javascript. Bootstrap merupakan bahasa yang minim sintaks untuk menciptakan *template*, gambar, tombol, portofolio, menu, dan lain sebagainya yang lebih responsif (Durganath and Edreena, 2016).

Pada kerja praktik penulis, Bootstrap dipergunakan untuk melakukan responsivitas layar, pembuatan *navigation bar* (*navbar*), *offcanvas*, tombol, modal, dan sebagainya.

3.7. Agile

Digital Amoeba menerapkan budaya kerja berupa Agile dan dengan menggunakan metode Scrum. Agile merupakan pendekatan ringan yang bertujuan untuk mengatasi metode pengembangan perangkat lunak yang kompleks dan untuk mengurangi biaya selama pengerjaan pada tahap apa pun, hal ini dilakukan dengan mengatur penugasan-penugasan dan koordinasi melalui sekumpulan nilai-nilai dan prinsip tertentu (Al-Saqqa, Sawalha and Abdelnabi, 2020).

3.8. Scrum

Scrum sendiri merupakan salah satu metode yang digunakan dalam menerapkan pendekatan Agile dan merupakan metode yang cukup populer di era modern ini (Masood, Hoda and Blincoe, 2022).

[Halaman ini sengaja dikosongkan]

BAB IV

ANALISIS DAN PERANCANGAN INFRASTRUKTUR SISTEM

4.1. Analisis Sistem

Pada bab ini akan dijelaskan mengenai tahapan dalam membangun infrastruktur aplikasi Eventeer. Hal tersebut dijelaskan ke dalam dua bagian, definisi umum aplikasi dan analisis kebutuhan.

4.1.1. Definisi Umum Aplikasi

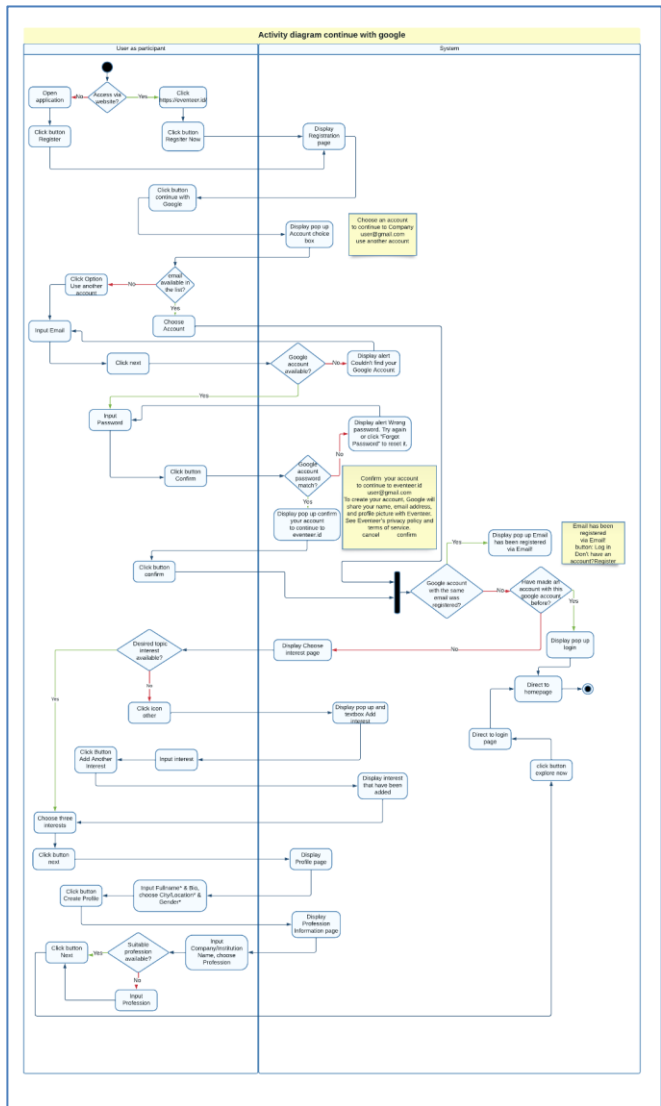
Eventeer merupakan platform *all-in-one* bagi para pencipta maupun pencari komunitas. Alur penggunaan platform Eventeer dimulai dengan pengenalan platform Eventeer melalui Landing Page. Selanjutnya pengguna dapat melakukan registrasi yang dapat dilakukan menggunakan email, Google, maupun Telkom SSO (Single Sign On). Setelah pengguna mengisi berbagai data diri dan minat komunitasnya, pengguna dapat masuk melalui akun yang telah dibuat dan akan diarahkan ke Homepage. Pada Homepage, pengguna dapat melihat sekilas komunitas dan acara sesuai dengan minatnya. Pada Homepage juga disediakan arahan untuk ke fitur lain seperti fitur Profil, Notifikasi, melihat detail komunitas dan event, mengakses media pembelajaran, melakukan interaksi melalui fitur *chat*, dan keluar (*logout*).

4.1.2 Analisis Kebutuhan

Berikut merupakan fitur-fitur yang disediakan Eventeer:

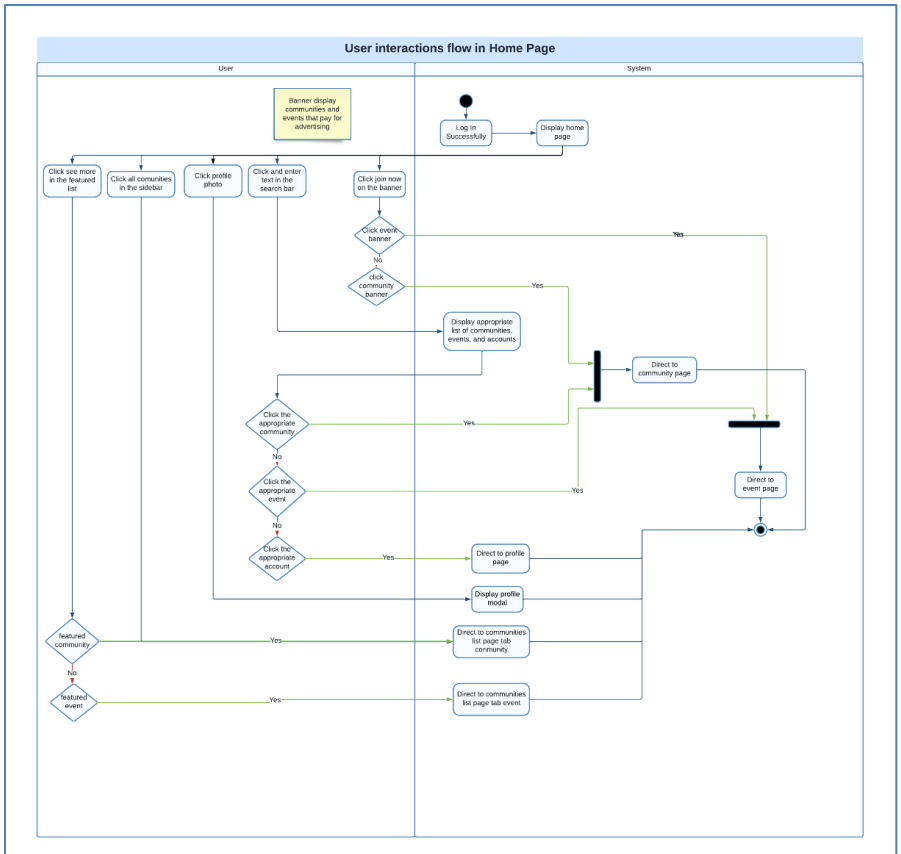
1. Menciptakan dan/atau menemukan komunitas berdasarkan minat
 - a. Pengguna dapat berkolaborasi dengan pengguna lain.
 - b. Pengguna dapat mengunggah konten terkait komunitas.
 - c. Pengguna dapat melakukan diskusi yang terorganisir.
 - d. Pengguna dapat berinteraksi dengan pengguna lain.
2. Menciptakan dan mengatur acara yang interaktif bagi peserta
 - a. Pengguna dapat mengelola acara.
 - b. Pengguna dapat membuat acara.
 - c. Sistem menyediakan tiket webinar atau acara.
3. Menyediakan media pembelajaran
 - a. Pengguna dapat mengunggah konten edukatif.
 - b. Pengguna dapat berdiskusi melalui fitur pembelajaran.

4.2.3. Activity Diagram Continue With Google



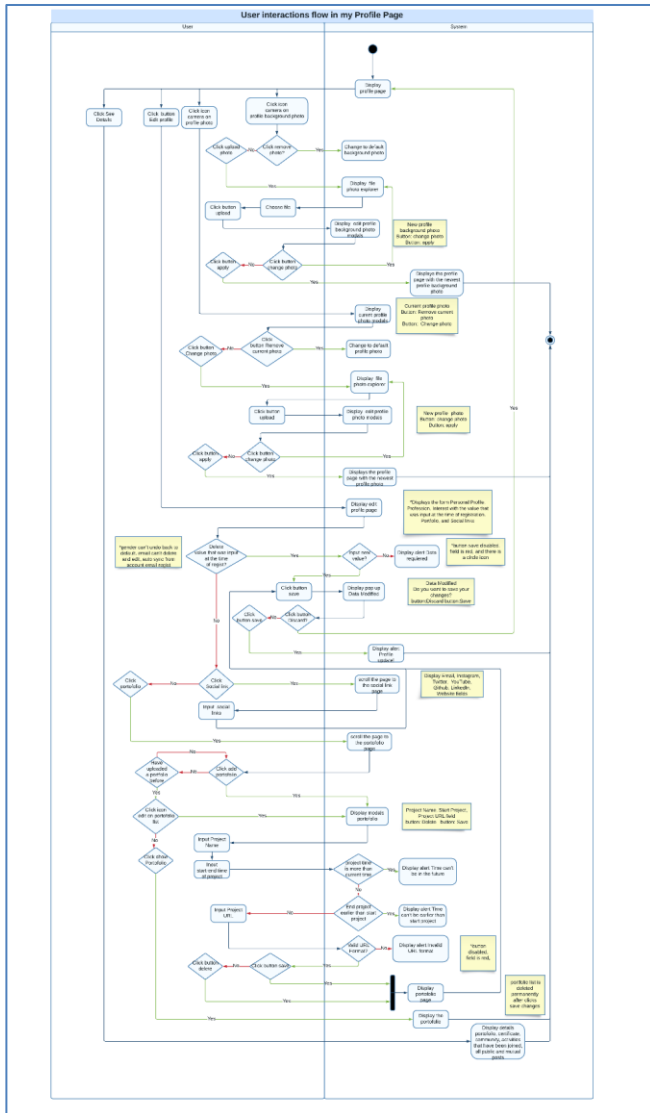
Gambar 4 Activity Diagram: Continue with Google

4.2.4. Activity Diagram Homepage



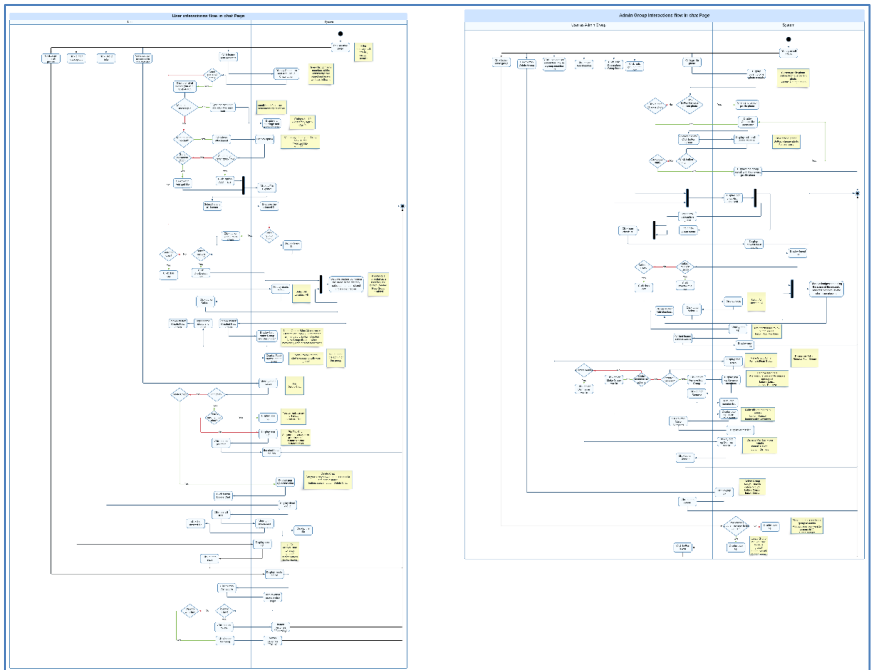
Gambar 5 Activity Diagram: Homepage

4.2.5. Activity Diagram Edit Profile

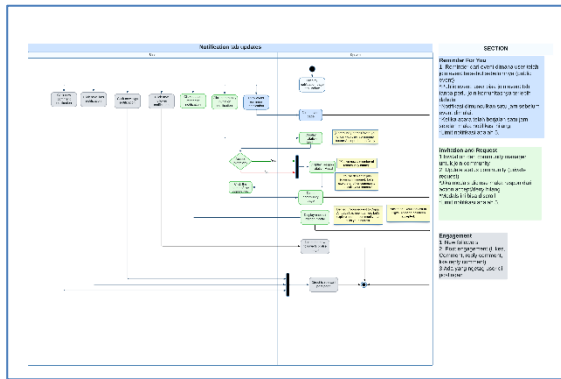


Gambar 6 Activity Diagram: Profile Page

4.2.6. Activity Diagram Chat

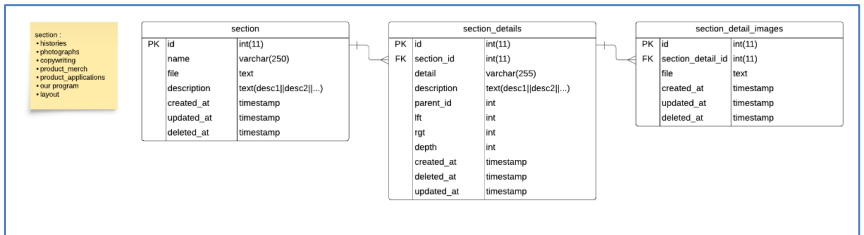


Gambar 7 Activity Diagram: Chat



Gambar 10 Activity Diagram: Notification Tab Updates

4.2.2. Entity Relationship Diagram (ERD) Landing Page Digital Amoeba



Gambar 11 ERD Landing Page Digital Amoeba

[Halaman ini sengaja dikosongkan]

BAB V

IMPLEMENTASI SISTEM

Bab ini membahas tentang implementasi dari sistem yang kami buat. Implementasi ini akan dibagi ke dalam dua bagian, yaitu bagian pengerjaan pengembangan ulang Eventeer dan pengerjaan Landing Page Digital Amoeba sesuai dengan bagian yang dikerjakan oleh penulis.

5.1. Pengembangan Ulang Eventeer

5.1.1. Landing Page

5.1.1.1. Index.jsx



```
import React, {useState} from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import Header from './components/Header';
import Footer from './components/Footer';
import Eventeer from './components/Eventeer';
import LandingPage from './components/LandingPage';
import {BrowserRouter} from 'react-router-dom';

const App = () => {
  return (
    <div>
      <Header />
      <Eventeer />
      <LandingPage />
      <Footer />
    </div>
  );
};

ReactDOM.render(<App />, document.getElementById('root'));
export default App;
```

Kode Sumber 1 Landing page: index.jsx

5.1.1.6. Achievement Component

```

import React from 'react';
import { div, heading, space } from '@components';
import {
  ulItems,
  RichTextAndBottomCenterText,
  RichTextOnly,
  RichTextPhone,
} from 'react-icom/042';
import { container } from 'react-bootstrap';

const Achievement = ({
  totalUsers = null,
  connectFriends = null,
  builtCommunities = null,
  successfulEvents = null,
  achLanguage,
}) => {
  return (
    <Container>
      <div style={16} />
      <div className="x-100 0-100 x-achievement">
        <div className="x-container">
          <div className="x-title-ach-container">
            <heading marginless className="x-ach-title"><achLanguage.title></heading>
            </div>
            <div style={14} />
            <heading marginless className="x-ach-cta">
              <achLanguage.cta>
            </heading>
            <div style={14} />
            <div className="x-ach-test-container">
              <heading marginless className="x-ach-cta-test">
                <achLanguage.description>
              </heading>
              <div>
                <div style={14} />
                <div className="x-achievement-test">
                  <heading className="x-ach-infoggris-mua">
                    <totalUsers>
                  </heading>
                  <heading className="x-ach-infoggris-test"><achLanguage.totalUsers></heading>
                </div>
                <div>
                  <div className="x-achievement-test">
                    <div className="x-ach-infoggris-mua">
                      <connectFriends>
                    </div>
                    <heading>
                      <achLanguage.connectFriends>
                    </heading>
                </div>
                <div>
                  <div className="x-achievement-test">
                    <div className="x-ach-infoggris-mua">
                      <builtCommunities>
                    </div>
                    <heading>
                      <achLanguage.builtCommunities>
                    </heading>
                </div>
                <div>
                  <div className="x-achievement-test">
                    <div className="x-ach-infoggris-mua">
                      <successfulEvents>
                    </div>
                    <heading>
                      <achLanguage.successfulEvents>
                    </heading>
                </div>
              </div>
            </div>
            <div style={16} />
          </div>
        </div>
      </div>
    </Container>
  );
};

export default Achievement;

```

Kode Sumber 10 Landing Page:
Achievement.jsx

```

<div style={16} />
<div className="x-100 0-100 x-achievement">
  <div className="x-container">
    <div className="x-title-ach-container">
      <heading marginless className="x-ach-title"><achLanguage.title></heading>
    </div>
    <div style={14} />
    <heading marginless className="x-ach-cta">
      <achLanguage.cta>
    </heading>
    <div style={14} />
    <div className="x-ach-test-container">
      <heading marginless className="x-ach-cta-test">
        <achLanguage.description>
      </heading>
      <div>
        <div style={14} />
        <div className="x-achievement-test">
          <heading className="x-ach-infoggris-mua">
            <totalUsers>
          </heading>
          <heading className="x-ach-infoggris-test"><achLanguage.totalUsers></heading>
        </div>
        <div>
          <div className="x-achievement-test">
            <div className="x-ach-infoggris-mua">
              <connectFriends>
            </div>
            <heading>
              <achLanguage.connectFriends>
            </heading>
        </div>
        <div>
          <div className="x-achievement-test">
            <div className="x-ach-infoggris-mua">
              <builtCommunities>
            </div>
            <heading>
              <achLanguage.builtCommunities>
            </heading>
        </div>
        <div>
          <div className="x-achievement-test">
            <div className="x-ach-infoggris-mua">
              <successfulEvents>
            </div>
            <heading>
              <achLanguage.successfulEvents>
            </heading>
        </div>
      </div>
    </div>
    <div style={16} />
  </div>
</div>
</Container>
);
};

export default Achievement;


```

Kode Sumber 11
Landing Page:
Achievement.scss

5.1.1.7. Download Component

```
import React from 'react';
import { Div, Heading, Spacer } from '@components';
import Image from 'next/image';
import Link from 'next/link';
import appstore from '../public/assets/images/App Store.svg';
import playstore from '../public/assets/images/Play Store.svg';
import phone from '../public/assets/images/phoneDownload.svg';
import { Container } from 'react-bootstrap';

const Download = ({downloadLanguage}) => {
  return (
    <Div className="x-download-container">
      <Container className="d-flex">
        <Div id="cta" className="x-download-cta">
          <Div className="x-cta-download-container">
            <Heading marginless className="x-download-cta-title">
              (downloadLanguage.cta){ " " }
            </Heading>
            <Spacer size={16} />
            <Heading marginless className="x-download-cta-text">
              (downloadLanguage.description)
            </Heading>
            <Spacer size={32} />
            <Div className="x-buttons">
              <Link href="/" />
              <Image
                src={playstore}
                alt="/" />
              </Link>
              <Link href="/" />
              <Image
                src={appstore}
                alt="/" />
              </Link>
            </Div>
          </Div>
        </Container>
        <Spacer axis="horizontal" size={54} className="d-sm-none d-lg-block" />
        <Div id="img" className="d-sm-none d-lg-block">
          <Image className="x-phones-download" src={phone} alt="/" />
        </Div>
      </Div>
    </Div>
  );
};
```

 **Monica Narda Davita**

Kode Sumber 12 Landing Page: Download.jsx

```
.x-download-container {
  display: flex;
  flex-direction: row;
  height: 180px;
  background-image: url('../public/assets/images/bg.png');
  justify-content: space-between;
  media only screen and (max-width: 600px) {
    height: fit-content;
    background: var(--ab1-500);
  }
}
.x-download-cta {
  padding: 16px 0 0 120px;
  display: flex;
  flex-direction: column;
  place-items: flex-start;
  media only screen and (max-width: 600px) {
    padding: 40px 16px;
    width: 100%;
  }
}
.x-cta-download-container {
  width: 600px;
  display: flex;
  flex-direction: column;
  place-items: flex-start;
  media only screen and (max-width: 992px) {
    padding: 32px 40px 40px 40px;
    width: 80%;
  }
}
.x-cta-download-container {
  width: 600px;
  display: flex;
  flex-direction: column;
  place-items: flex-start;
  media only screen and (max-width: 992px) {
    padding: 40px 16px;
    width: 100%;
  }
}
.x-buttons {
  display: flex;
  flex-direction: row;
  padding-top: 12px;
}
.x-download-cta-title {
  font-family: 'Poppins-Bold', sans-serif;
  font-weight: 700;
  font-size: 32px;
  text-align: start;
  color: #000000;
  line-height: 40px;
  media only screen and (max-width: 992px) {
    font-size: 20px;
    line-height: 26px;
  }
}
.x-download-cta-text {
  font-weight: 400;
  font-size: 16px;
  text-align: start;
  color: #000000;
  line-height: 24px;
}
.x-phones-download {
  width: auto;
  height: auto;
  media only screen and (max-width: 992px) {
    object-fit: cover;
  }
  media only screen and (max-width: 600px) {
    display: none;
  }
}
```


 **Monica Narda Davita**

Kode Sumber 13 Landing Page: Download.scss

5.1.1.9. Get Account Component

```
import React from 'react';
import { Div, Heading, Spacer } from '@components';
import Link from 'next/link';
import { BsArrowRight } from 'react-icons/bs';
import { Container } from 'react-bootstrap';

const GetAccount = ({getAcLanguage}) => {
  return (
    <Container className="x-get-account">
      <Div className="x-get-account-cta">
        <Heading marginInass className="x-get-account-cta-title">
          {getAcLanguage.cta}
        </Heading>
        <Spacer size={12} />
        <Heading className="x-get-account-cta-text">
          {getAcLanguage.desc}
        </Heading>
      </Div>
      <Spacer size={32} />
      <Link href="/register" style={{ textDecoration: 'none' }}>
        <Div className="x-get-acc-btn">
          <Heading centered className="x-get-account-btn">{getAcLanguage.registerNow}</Heading>
          <Spacer axis='horizontal' size={8} />
          <BsArrowRight
            style={{ color: '#FFFFFF', padding: '0 0 2px 6px' }}
            size={25}
          />
        </Div>
      </Link>
      <Spacer size={64} />
    </Container>
  );
};
```

 **Monica Narda Davita**

Kode Sumber 17 Landing Page: GetAccount.jsx

```
.x-get-account {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  @media only screen and (max-width: 992px) {
    padding: 30px 50px;
  }
  @media only screen and (max-width: 600px) {
    align-items: flex-start;
    padding: 30px 50px;
  }
}

.x-get-account-cta {
  display: flex;
  flex-direction: column;
  align-items: center;
}

.x-get-acc-btn {
  width: fit-content;
  display: flex;
  flex-direction: row;
  justify-content: center;
  align-items: center;
  background-color: var(--abl-500);
  border-radius: 8px;
  padding: 10px 20px;
}
```

*Kode Sumber 18
Landing Page:
GetAccount.scss 1/2*

```
.x-get-account-cta-title {
  font-family: 'Poppins-Bold', sans-serif;
  font-weight: 700;
  font-size: 30px;
  text-align: center;
  color: #192434;
  line-height: 40px;
  @media only screen and (max-width: 992px) {
    font-size: 28px;
    line-height: 36px;
  }
}

.x-get-account-cta-text {
  font-weight: 400;
  font-size: 16px;
  text-align: center;
  color: #41455d;
  line-height: 24px;
  @media only screen and (max-width: 992px) {
    font-size: 16px;
    line-height: 24px;
  }
}


.x-get-account-btn {
  font-weight: 600;
  font-size: 18px;
}
```

*Kode Sumber 16
Landing Page:
GetAccount.scss 2/2*

5.1.2.2. ModalEmailIsNotRegistered Component

```
import { Fragment, memo } from 'react';
import { Button } from 'react-bootstrap';
import { ModalConfirmation } from '@components';

const ModalEmailIsNotRegistered = ({ show, onHide, onExited }) => {
  return (
    <ModalConfirmation
      modalClassName='x-modal-email-is-not-registered'
      size='lg'
      show={show}
      onHide={onHide}
      onExited={onExited}
      title='Your Email is Not Registered!'
      description='Make sure to use the email associated with your Eventeez account'
      customFooter={
        <Fragment>
          <Button
            onClick={onHide}
            className='x-try-again-btn btn btn-primary paragraph-p16 line-height-md bs-50 fw-500 text-center text-decoration-none bg-ab1-900 corner-radius-4'
            >
            Try again
          </Button>
        </Fragment>
      }
    />
  );
};
```

 **Monica Neerda Devi** @monicalreadyregistered


Kode Sumber 27 Login: ModalEmailsNotRegistered.jsx

5.1.2.3. ModalEmailIsAlreadyRegistered Component

```
import { Fragment, memo } from 'react';
import { Button } from 'react-bootstrap';
import Link from 'next/link';
import { ModalConfirmation, Spacer } from '@components';
import { useRouter } from 'next/router';
import { signOut } from 'next-auth/react';

const ModalEmailIsAlreadyRegistered = ({ show, onHide, onExited, push }) => {
  const { push } = useRouter();

  return (
    <ModalConfirmation
      modalClassName='x-modal-email-is-not-registered'
      size='lg'
      show={show}
      onHide={onHide}
      onExited={onExited}
      title='Email has been registered via Email!'
      customFooter={
        <Fragment>
          <Button
            onClick={onHide}
            className='x-try-again-btn btn btn-primary paragraph-p16 line-height-md bs-50 fw-500 text-center text-decoration-none bg-ab1-900 corner-radius-4'
            >
            Log In
          </Button>
          <Fragment>
            <Form>
              <Form.Control type='text' value={email} />
              <Form.Control type='password' value={password} />
              <Form.Button type='submit' value='Log In' />
            </Form>
          </Fragment>
          <Form>
            <Form.Control type='text' value={email} />
            <Form.Control type='password' value={password} />
            <Form.Button type='submit' value='Register' />
          </Form>
        </Fragment>
      }
    />
  );
};
```

 **Monica Neerda Devi** @monicalreadyregistered

Kode Sumber 28 Login: ModalEmailsAlreadyRegistered.jsx

5.1.2.4. ModalLoginSuccess Component


```
import { memo } from 'react';
import { useRouter } from 'next/router';
import { ModalConfirmation } from '@components';

import SuccessImg from '@public/assets/gifs/Success 200x200.gif';

const ModalLoginSuccess = ({ show, onHide, onExited }) => {
  const { push } = useRouter();

  const escapeKeyDownHandler = () => {
    push({ pathname: '/' });
  };

  return (
    <ModalConfirmation
      modalClassName='x-modal-success-reset-password'
      show={show}
      onHide={onHide}
      onExited={onExited}
      onEscapeKeyDown={escapeKeyDownHandler}
      modalImg={SuccessImg}
      title='Log In Successfully!'
      description=''
      backdrop={true}
    />
  );
};
```

 **Monica Narda Davita** LoginSuccess);

Kode Sumber 29 Login: ModalLoginSuccess.jsx

5.1.2.5. ModalMaxLoginAttempt Component

```
import { memo, useEffect, useState } from 'react';
import { useRouter } from 'next/router';
import { Button } from 'react-bootstrap';
import { Div, ModalConfirmation } from '@components';

const ModalEmailIsNotRegistered = ({ show, onHide, onExited }) => {
  const [timeLeft, setTimeLeft] = useState(60);


  useEffect(() => {
    if (!show) return;

    if (timeLeft < 0) {
      return;
    }

    setTimeout(() => setTimeLeft((prevTime) => prevTime - 1), 1000);
  }, [timeLeft, show]);
  const { push } = useRouter();

  const handleClickForgotPassword = (e) => {
    e.preventDefault();
    push('/forgot-password');
  };

  const maxLoginDescription = 'Please try again in ${timeLeft} seconds';
  return (
    <ModalConfirmation
      modalClassName='x-modal-email-is-not-registered'
      size='lg'
      show={show}
      onHide={onHide}
      onExited={onExited}
      title='You've Reached The Log In Attempt Limit!'
      description={timeLeft >= 1 ? maxLoginDescription : ''}
      customFooter={
        <div className='pl-00 ml-0'>
          <button
            onClick={handleClickForgotPassword}
            className='x-try-again-btn btn btn-primary paragraph-p16 line-height-md bs-fw-semibold text-center text-decoration-none bg-abl-500 corner-radius-4'
          >
            Forgot Password
          </button>
          <button
            onClick={onHide}
            className='x-try-again-btn btn btn-primary paragraph-p16 line-height-md abl-600 fw-semibold text-center text-decoration-none bg-abl-25 corner-radius-4 mt-3'
          >
            Continue
          </button>
        </div>
      }
    />
  );
};
```

 **Monica Narda Davita** EmailIsNotRegistered;

Kode Sumber 30 Login: ModalMaxLoginAttempt.jsx

5.1.2.7. NextAuth for Login with Google

```
import NextAuth from 'next-auth';
import GoogleProvider from 'next-auth/providers/google';
export const authOptions = {
  providers: [
    GoogleProvider({
      clientId: process.env.NEXT_PUBLIC_GOOGLE_CLIENT_ID,
      clientSecret: process.env.NEXT_PUBLIC_GOOGLE_CLIENT_SECRET,
      authorization: {
        params: {
          prompt: "consent",
          access_type: "offline",
          response_type: "code"
        }
      },
    },
  ],
  callbacks: {
    session: async ({ session, token }) => {
      if (session?.user) {
        session.user.id = token.uid;
      }
      return session;
    },
    jwt: async ({ user, token }) => {
      if (user) {
        token.uid = user.id;
      }
      return token;
    },
  },
  session: {
    strategy: 'jwt',
  },
  secret: process.env.NEXT_PUBLIC_NEXTAUTH_SECRET,
}
export default NextAuth(authOptions);
```



Monica Narda Davita

[@fault](#) NextAuth(authOptions);

Kode Sumber 32 Login: NextAuth.js



5.1.3. Homepage

5.1.3.1. Index.jsx

```
import React, { useEffect, useState } from 'react';
import { useSelector } from 'react-redux';
import { useRouter } from 'next/router';
import { isEmpty } from 'lodash-es';
import * as cookie from 'cookie';

import Image from 'next/image';
import { Container, Row, Col } from 'react-bootstrap';
import {
  Div,
  Heading,
  Layout,
  Button,
  ModalHolaHomepage,
  CommunityCard,
  EventCard,
  Spacer,
} from '@components';

import { getHomepage } from '@services/actions/Api';

import { FiArrowRight } from 'react-icons/fi';
 Monica Narda Davita /public/assets/icons/Magnet.svg';
 Party from '../public/assets/icons/Party Popper.svg';
```

Kode Sumber 33 Homepage: index.jsx 1/3

```
export const getServerSideProps = async (context) => {
  const cookies = cookie.parse(context.req.headers.cookie);

  const userToken = cookies.user_token;

  let featuredCommunities = [];
  let featuredEvents = [];

  await getHomepage(userToken)
    .then((response) => {
      const { code, result } = response;

      if (code === 200) {
        const { featured_community, featured_event } = result;

        featuredCommunities = featured_community;
        featuredEvents = featured_event;
      }
    })
    .catch((error) => console.log(error));

  return {
    props: {
      featuredCommunities: featuredCommunities ?? null,
      featuredEvents: featuredEvents ?? null,
    },
  };
};
```

 **Monica Narda Davita**
@fault Homepage;

Kode Sumber 34 Homepage: index.jsx 2/3

5.1.3.2. ModalHolaHomepage Component

```
import React from 'react'
import { Div, Heading } from '@components'
import Image from 'next/image'
import WavingHand from '../././public/assets/icons/Waving Hand.svg'

const ModalHolaHomepage = ({name}) => {
  return (
    <Div backgroundColor='pr-80' flex absolute center gap='big' radius=(6) className='x-modal-hola'>
      <Div className='x-icon-wave-container' backgroundColor='abl-50' gap='sm' radius=(3) flex justifyCenter alignCenter >
        <Image alt='/' src={WavingHand}/>
      </Div>
      <Div flex column between gap='sm'>
        <Heading marginless fontFamily='poppins' weight='bold' size='h5' color='ne-900' text={`Hello, ${name} ?? 'Hello, Andika Leonardus Suzya' } />
        <Heading marginless type='normal' size='h6' text='Update the new features of the community' />
      </Div>
    </Div>
  )
}

Monica Narda Davita
@fault ModalHolaHomepage
```

Code Sumber 36 Homepage: ModalHolaHomepage.jsx


```

//validation
const isSubmitBtnDisabled =
  formik.errors.personalProfile ||
  formik.errors.profession ||
  formik.errors.interest ||
  formik.errors.socialLinks ||
  formik.values.personalProfile.fullName.length === 0 ||
  formik.values.profession.companyName.length === 0;

// ALL SUGGESTIONS FOR DROPDOWN USAGE
// city suggestion
const suggestionsCities = allCities?.filter((city) =>
  city.city_name
  .toLowerCase()
  .includes(formik.values.personalProfile.city.toLowerCase())
);

// interest suggestion
const suggestionsInterests = allInterests?.filter((interest) =>
  interest.interest_name
  .toLowerCase()
  .includes(formik.values.interest.interest_name.toLowerCase())
);

//profession suggestions
const suggestionsProfessions = allProfessions.filter((profession) =>
  profession.job_title
  .toLowerCase()
  .includes(formik.values.profession.profession.toLowerCase())
);

//month start portfolio suggestions
const suggestionsMonthStart = months?.filter((month) =>
  month.month
  .toLowerCase()
  .includes(formik.values.portfolio.monthStartDate?.toLowerCase())
);

// month end portfolio suggestions
const suggestionsMonthEnd = months?.filter((month) =>
  month.month
  .toLowerCase()
  .includes(formik.values.portfolio.monthEndDate?.toLowerCase())
);

//year start portfolio suggestions
const suggestionsYearStart = years?.filter((year) =>
  year.year
  ?.toLowerCase()
  .includes(formik.values.portfolio.yearStartDate.toLowerCase())
);

//year end portfolio suggestions
const suggestionsYearEnd = years?.filter((year) =>
  year.year
  ?.toLowerCase()
  .includes(formik.values.portfolio.yearEndDate.toLowerCase())
);

```


Monica Narda Davita

Kode Sumber 42 Edit Profile: index.jsx 3/11

```

//validation
const isSubmitBtnDisabled =
  formik.errors.personalProfile ||
  formik.errors.profession ||
  formik.errors.interest ||
  formik.errors.socialLinks ||
  formik.values.personalProfile.fullName.length === 0 ||
  formik.values.profession.companyName.length === 0;

// ALL SUGGESTIONS FOR DROPDOWN USAGE
// city suggestion
const suggestionsCities = allCities?.filter((city) =>
  city.city_name
  .toLowerCase()
  .includes(formik.values.personalProfile.city.toLowerCase())
);

// interest suggestion
const suggestionsInterests = allInterests?.filter((interest) =>
  interest.interest_name
  .toLowerCase()
  .includes(formik.values.interest.interest_name.toLowerCase())
);

//profession suggestions
const suggestionsProfessions = allProfessions.filter((profession) =>
  profession.job_title
  .toLowerCase()
  .includes(formik.values.profession.profession.toLowerCase())
);

//month start portfolio suggestions
const suggestionsMonthStart = months?.filter((month) =>
  month.month
  .toLowerCase()
  .includes(formik.values.portfolio.monthStartDate?.toLowerCase())
);

// month end portfolio suggestions
const suggestionsMonthEnd = months?.filter((month) =>
  month.month
  .toLowerCase()
  .includes(formik.values.portfolio.monthEndDate?.toLowerCase())
);

//year start portfolio suggestions
const suggestionsYearStart = years?.filter((year) =>
  year.year
  ?.toLowerCase()
  .includes(formik.values.portfolio.yearStartDate.toLowerCase())
);

//year end portfolio suggestions
const suggestionsYearEnd = years?.filter((year) =>
  year.year
  ?.toLowerCase()
  .includes(formik.values.portfolio.yearEndDate.toLowerCase())
);

```

Kode Sumber 41 Edit Profile: index.jsx 4/11

```

// City select handler
const _citySelectHandler = (city) => {
  formik.setValues({
    ...formik.values,
    personalProfile: {
      ...formik.values.personalProfile,
      city: city.city_name,
      cityId: city.id_city,
    },
  });
};

// Interest select handler
const _interestSelectHandler = (interest) => {
  formik.setValues({
    ...formik.values,
    interest: {
      ...formik.values.interest,
      interest_name: interest.interest_name,
      interest_id: interest?.id_interest,
    },
  });
  addItem({ interest_name: interest.interest_name });
};

// Profession select handler
const _professionSelectHandler = (profession) => {
  formik.setValues({
    ...formik.values,
    profession: {
      ...formik.values.profession,
      profession: profession.job_title,
      professionId: profession.id_job,
    },
  });
};

// Month start select handler
const _monthStartSelectHandler = (month) => {
  formik.setValues({
    ...formik.values,
    portfolio: {
      ...formik.values.portfolio,
      monthStartDate: month.month,
      monthStartId: month.monthId,
    },
  });
};

// Month End select handler
const _monthEndSelectHandler = (month) => {
  formik.setValues({
    ...formik.values,
    portfolio: {
      ...formik.values.portfolio,
      monthEndDate: month.month,
      monthEndId: month.monthId,
    },
  });
};

// Year Start select handler
const _yearStartSelectHandler = (year) => {
  formik.setValues({
    ...formik.values,
    portfolio: {
      ...formik.values.portfolio,
      yearStartDate: year.year,
      yearStartId: year.yearId,
    },
  });
};

// Year End select handler
const _yearEndSelectHandler = (year) => {
  formik.setValues({
    ...formik.values,
    portfolio: {
      ...formik.values.portfolio,
      yearEndDate: year.year,
      yearEndId: year.yearId,
    },
  });
};

```

Kode Sumber 44 Edit Profile: index.jsx 5/11

```

//Profession reset value handler
const _professionResetValueHandler = () => {
  formik.setValues({
    ...formik.values,
    profession: {
      ...formik.values.profession,
      profession: '',
      professionId: 0,
    },
  });
};

//Interest reset value handler
const _interestResetValueHandler = () => {
  formik.setValues({
    ...formik.values,
    interest: {
      ...formik.values.interest,
      interest_name: '',
      interest_id: 0,
    },
  });
};

//City reset value handler
const _cityResetValueHandler = () => {
  formik.setValues({
    ...formik.values,
    personalProfile: {
      ...formik.values.personalProfile,
      city: '',
      cityId: 0,
    },
  });
};

//Month Start reset value handler
const _monthStartResetValueHandler = () => {
  formik.setValues({
    ...formik.values,
    portfolio: {
      ...formik.values.portfolio,
      monthStartDate: '',
      monthStartId: 0,
    },
  });
};

//Month End reset value handler
const _monthEndResetValueHandler = () => {
  formik.setValues({
    ...formik.values,
    portfolio: {
      ...formik.values.portfolio,
      monthEndDate: '',
      monthEndId: 0,
    },
  });
};

//Year Start reset value handler
const _yearStartResetValueHandler = () => {
  formik.setValues({
    ...formik.values,
    portfolio: {
      ...formik.values.portfolio,
      yearStartDate: '',
      yearStartId: 0,
    },
  });
};

//Year End reset value handler
const _yearEndResetValueHandler = () => {
  formik.setValues({
    ...formik.values,
    portfolio: {
      ...formik.values.portfolio,
      yearEndDate: '',
      yearEndId: 0,
    },
  });
};

```

Kode Sumber 43 Edit Profile: index.jsx 6/11

```
//Function to check the validation of each field
const _handleSubmit = (values) => {
  const errors = {};

  // Personal Profile Validation
  if (values.personalProfile.fullName.length <= 0 {
    errors.personalProfile = {
      ...errors.personalProfile,
      fullName: "Data required",
    };
  }

  if (values.personalProfile.city.length <= 0 && values.personalProfile.city) {
    errors.personalProfile = {
      ...errors.personalProfile,
      city: "Data required",
    };
  }

  if (values.personalProfile.city.length <= 0 && values.personalProfile.city) {
    errors.personalProfile = {
      ...errors.personalProfile,
      city: "Data required",
    };
  }

  // Profession Validation
  if (values.profession?.companyName?.length <= 0 && values?.profession?.companyName) {
    errors.profession = {
      ...errors.profession,
      companyName: "Data required",
    };
  }

  if (values.profession?.profession?.length <= 0 && values?.profession?.profession) {
    errors.profession = {
      ...errors.profession,
      profession: "Data required",
    };
  }

  // Income Validation
  if (!isNaN(values.income?.log?.interests) &&
    !isNaN(values.income?.log?.interests)) {
    errors.income = {
      ...errors.income,
      interests: "Data required"
    }
  }

  //Portfolio Validation
  if (values.portfolio?.yearStarted < new Date().getFullYear()) {
    errors.portfolio = {
      ...errors.portfolio,
      monthStarted: "Year can't be in the future",
    };
  }

  if (values.portfolio?.yearEnded < new Date().getFullYear()) {
    errors.portfolio = {
      ...errors.portfolio,
      monthStarted: "Year can't be in the future",
    };
  }

  if (values.portfolio?.yearEnded < values.portfolio?.yearStarted && values.portfolio?.yearStarted) {
    errors.portfolio = {
      ...errors.portfolio,
      monthStarted: "Year can't be earlier than start project",
    };
  }

  if (values.portfolio?.yearStarted <= values.portfolio?.monthEnded && values.portfolio?.monthStarted && values.portfolio?.monthEnded) {
    errors.portfolio = {
      ...errors.portfolio,
      monthStarted: "Year can't be earlier than start project",
    };
  }

  //Social Media Validation
  if (!values.socialLinks?.instagram &&
    values.socialLinks?.instagram) {
    errors.socialLinks = {
      ...errors.socialLinks,
      instagram: "Invalid Instagram link",
    };
  }

  if (!values.socialLinks?.twitter &&
    values.socialLinks?.twitter) {
    errors.socialLinks = {
      ...errors.socialLinks,
      twitter: "Invalid twitter link",
    };
  }

  if (!values.socialLinks?.github &&
    values.socialLinks?.github) {
    errors.socialLinks = {
      ...errors.socialLinks,
      github: "Invalid github link",
    };
  }

  if (!values.socialLinks?.youtube &&
    values.socialLinks?.youtube) {
    errors.socialLinks = {
      ...errors.socialLinks,
      youtube: "Invalid Youtube link",
    };
  }

  if (!values.socialLinks?.linkedin &&
    values.socialLinks?.linkedin) {
    errors.socialLinks = {
      ...errors.socialLinks,
      linkedin: "Invalid LinkedIn link",
    };
  }

  if (!values.socialLinks?.website &&
    values.socialLinks?.website) {
    errors.socialLinks = {
      ...errors.socialLinks,
      website: "Invalid website link",
    };
  }


  if (!values.portfolio?.projectTitle &&
    values.portfolio?.projectTitle) {
    errors.portfolio = {
      ...errors.portfolio,
      projectTitle: "Invalid title",
    };
  }
};
```

Kode Sumber 45 Edit Profile: index.jsx 7/11

```

// Making scroll effect to each component
EditProfileSections.map((section, id) => {
  return (refs[section.id] = useRef(null));
});
const scrollToSection = (sectionId) => {
  const sectionRef = refs[sectionId].current;
  if (sectionRef) {
    sectionRef.scrollToView({
      behavior: 'smooth',
      block: 'start',
    });
  }
  setActiveSection(sectionId);
}

```



Monica Narda Davita

Kode Sumber 46 Edit Profile: index.jsx 8/11

```

// function to delete interest from frontend
const deleteItem = (id) => {
  const updatedItems = interestList.filter( // Find the interest that want to be deleted based on the id
    (interest) => interest.id_interest !== id
  );
  setInterestList(updatedItems); // update the interest list array
  setDeleteInterest([...deleteInterest, id]); // store the id of the deleted interest
};


// function to add interest
const addItem = (value) => {
  if (interestList.length < 3) { // add condition
    setInterestList([...interestList, value]); // update the interest list array so it will be displayed
    setAddInterest([...addInterest, value.interest_name]); // store the interest id so it can be POST-ed to the API
  }
};

// function to edit portfolio that has already added
const editExistingPortfolio = (item) => {
  if (Number.isInteger(item.id_portfolio)) { // check if the id is integer or not (if it's integer then it is from the backend)
    var updateExistingPortfolio = [...editPortfolio]; // make temporary array to store the initial value
    const itemIndex = editPortfolio.findIndex( // find the portfolio's index that want to be edited
      (portfolio) => portfolio.id_portfolio === item.id_portfolio
    );
    if (itemIndex !== -1) { // if the portfolio exists
      updateExistingPortfolio[itemIndex] = item; // change the data into the temp. array
      setEditPortfolio(updateExistingPortfolio); // store the temp array into the initial array
      setPostEditPortfolio([...postEditPortfolio, item]); // store the edited portfolio into the special array (so we can POST it later)
    }
  } else { // if the id is string (it means the portfolio is not yet in the backend/server, only in the frontend)
    var updatePortfolio = [...addPortfolio]; // create temp. array that consists of all portfolios
    console.log(item.project_name);
    const itemIndex = addPortfolio.findIndex(
      (portfolio) => portfolio.project_name === item.id_portfolio
    );
    if (itemIndex !== -1) {
      updatePortfolio[itemIndex] = item;
      setAddPortfolio(updatePortfolio); // store the edited data back into the initial array
    }
  }
};

// function that handles storing the added portfolio
const handleAddPortfolio = (value) => {
  setAddPortfolio([...addPortfolio, value]);
};

// function that handles delete portfolio
const deletePortfolio = (item) => {
  console.log(item);
  if (typeof item === 'string') { // check if the item is string, then we know that it's still in the frontend
    const updatedItems = addPortfolio.filter( // filter all the portfolio except for the portfolio that want to be deleted (based on the name)
      (portfolio) => portfolio.project_name !== item
    );
    setAddPortfolio(updatedItems); // store the updated array into the initial array
    setDeleteItemPortfolio([...deleteItemPortfolio, item]); // store the deleted portfolio item into the deleted array for we can POST it later
  } else {
    const updatedItems = editPortfolio.filter( // delete the portfolio based on the id (because it's already stored in the server)
      (portfolio) => portfolio.id_portfolio !== item
    );
    setEditPortfolio(updatedItems);
    setDeleteItemPortfolio([...deleteItemPortfolio, item]);
  }
};

```



Monica Narda Davita

Kode Sumber 47 Edit Profile: index.jsx 9/11

5.1.4.3. Profession Component

```
import React from 'react'
import { Div, Heading, Label, InputSelect, InputText } from '@components'
import { Form } from 'react-bootstrap'
import classNames from 'classnames';
import SearchIcon from '@public/assets/icons/icon-search.svg';
import { Input } from '@components/form';

const Profession = React.forwardRef((props, ref) => {

  const {values, errors, changeCompanyName, professionChange, professionSelect, professionSuggestions, professionReset} = props
  return (
    <Div ref={ref} flex column centered gap='md' className='p-big x-personal-profile-container' radius={4}>
      <Label text='PROFESSION' weight='bold' />
      <hr className="w-100 bg-ne-75"></hr>
      <Form>
        <Div className='w-100' flex column gap='md'>
          <Input
            value={values?.companyName}
            onChange={changeCompanyName}
            className='fw-semibold label-l16 pr-900'
            label='Company/Institution Name'
            id='company'
            placeholder='Example: Telkom Indonesia'
            isError={errors?.companyName}
            errorHint={errors?.companyName}
          />
          <InputSelect
            value={values?.profession}
            onChange={professionChange}
            className='fw-semibold label-l16 pr-900'
            labelText='Profession'
            inputId='profession'
            placeholder='Example: Software Engineer'
            isError={errors?.profession}
            errorLabelTitle={errors?.profession}
            showErrorIcon={true}
            onResetValue={professionReset}
            suggestions={professionSuggestions}
            suggestionOnClick={professionSelect}
            idFieldName='id_job'
            suggestionTitleFieldName='job_title'
            extraIcon={SearchIcon}
          />
        </Div>
      </Form>
    </Div>
  );
});
```


 **Monica Narda Davita**
@fault Profession

Kode Sumber 51 Edit Profile: Profession.jsx

5.1.4.8. ModalSaveEditProfile Component

```
import { Fragment, memo, useEffect, useState } from 'react';
import { Div, ModalConfirmation, Button, Heading, Paragraph } from '@components';
import { useRouter } from 'next/router';

const ModalSaveEditProfile = ({
  show,
  onHide,
  onExited,
  onSubmit,
  isError,
  onNormal,
}) => {
  const [windowInnerWidth, setWindowInnerWidth] = useState(0);
  const push = useRouter();
  useEffect(() => {
    setWindowInnerWidth(window.innerWidth);
  }, [show]);
  return (
    <ModalConfirmation
      header={tname}
      modalClassName='x-modal-email-confirmation'
      size='lg'
      show={show}
      onHide={onHide}
      onExited={onHide}
      title={<Heading text='Data Modified' size='h5' weight='semibold' fontFamily='poppins' />}
      description={<Paragraph text='Do you want to save your changes?' size='p18' color='ne-600' weight='regular' />}
      customFooter={
        <Fragment>
          <Div
            width='fit-content'
            flex
            center
            gap='x-big'
            className='x-btn-wrapper'
          >
            <Button border='none' size='l18' className='py-sm px-xx-lg' onClick={onNormal} backgroundColor='v2-pr-25' color='v2-pr-500' text='Discard' />
            <Button disabled={isError} size='l18' className='py-sm px-xx-lg' onClick={onSubmit} backgroundColor='v2-pr-500' text='Save' />
          </Div>
        </Fragment>
      }
    />
  );
};
```


 **Monica Narda Davita**
@fault ModalSaveEditProfile

Kode Sumber 56 Edit Profile: ModalSaveEditProfile.jsc

5.1.4.9. ModalEditProfileSuccess Component

```
import React from 'react';
import { BsCheckCircle } from 'react-icons/bs';
import { Div, Heading } from '@components';
import { HiOutlineX } from 'react-icons/hi';

const ModalEditProfileSuccess = ({onHide, message = 'Your changes are saved successfully', title = 'Profile update!'}) => {
  return (
    <Div backgroundColor='pp-00' flex absolute justifyStart alignStart gap='big' radius={6} className='x-modal-toast'>
      <Div backgroundColor='ss-100' className='gap-sm p-big corner-radius-8 border border-s border-ss-50' flex justifyCenter alignCenter >
        <BsCheckCircle size={20} color='var(--ss-500)' />
      </Div>
      <Div flex column between gap='sm'>
        <Heading marginless fontFamily='poppins' weight='bold' size='h5' color='ne-900' text={title} />
        <Heading marginless type='normal' size='h6' text={message}/>
      </Div>
      <HiOutlineX size={20} color='var(--ne-500)' onClick={onHide} className='pointer' />
    </Div>
  );
};
```


 **Monica Narda Davita**
@fault ModalEditProfileSuccess

Kode Sumber 57 Edit Profile: ModalEditProfileSuccess.jsx

5.1.5.2. ModalShowPhoto Component

```
import { Fragment, memo, useEffect, useState } from 'react';
import { Div, ModalConfirmation, Button, Heading, Paragraph, Spacer } from '@components';
import Image from 'next/image';

const ModalShowPhoto = ({selectedImage, handleCloseLightbox, onHide, onExited, show, download}) => {
  return (
    <ModalConfirmation
      hasExitIcon={true}
      modalClassName='x-modal-email-confirmation'
      size='lg'
      show={show}
      onHide={onHide}
      onExited={onExited}
      customFooter={
        <Div flex center column className="lightbox">
          <Div flex center className="lightbox-content">
            <img style={{width: '480px', height:'520px'}} src={selectedImage} alt="Enlarged Image" />
          </Div>
          <Spacer size={24} />
          <Button variant='primary' text='Download' onClick={download} />
        </Div>
      }
    />
  )
}
```



Monica Narda Davita
@fault ModalShowPhoto

Kode Sumber 59 Chat: ModalShowPhoto.jsx

5.1.6. Notification (Tab Activity)

5.1.6.1. Index.jsx

```
import React, { useState, useEffect } from 'react'
import {
  Div,
  Heading,
  Label,
  Paragraph,
  Spacer,
  Button,
  ManagedCommunity,
  Layout,
  JoinedCommunity,
  UpdateComponent,
  ModalInvitationNotif,
  ModalRequestNotif,
} from '@components';
import { Containerz } from "react-bootstrap";
import classNames from "classnames";
import ProfilePict from "../../public/assets/images/Person.svg";
import NotifPostAsset from "../../public/assets/images/NotifPostAsset.svg";
import NotifEmpty from "../../public/assets/images/img-empty-notification.png";
import ModalMemberRequestNotif from "@components/modal/ModalMemberRequestNotif";
import {
  getNotificationActivityJoined,
  getNotificationActivityManaged,
  getNotificationUpdates,
  postReadAllNotification,
  postReadNotification,
  postInvitationConfirmation,
} from "@services/actions/Api";
import { getAuthToken } from "@services/Utils/Helper";
import { Monica Narda Davita } from "next/image";
import Image from "next/image";
```

Kode Sumber 60 Notification: index.jsx 1/10

```

const Notification = () => {
  const tab = ["Updates", "Activity"];
  const tabActivity = ["joined", "managed"];
  const userToken = getUserToken();

  const [tabState, setTabState] = useState(tab[0]);
  const [tabStateActivity, setTabStateActivity] = useState(tabActivity[0]);
  const [showModalInvitation, setShowModalInvitation] = useState(false);
  const [showModalRequestNotif, setShowModalRequestNotif] = useState(false);
  const [showModalMemberRequest, setShowModalMemberRequest] = useState(false);
  const [updatesNotif, setUpdatesNotif] = useState();
  const [joinedCommunityNotif, setJoinedCommunityNotif] = useState();
  const [managedCommunityNotif, setManagedCommunityNotif] = useState();
  const [invitationNotif, setInvitationNotif] = useState(DUMMY_DATA_INVITATION);
  const [memberRequestNotif, setMemberRequestNotif] = useState();
  const [communityId, setCommunityId] = useState();
  const [communityName, setCommunityName] = useState();
  const [dateModal, setDateModal] = useState();
  const [idNotifModal, setIdNotifModal] = useState()

  //useEffect
  useEffect() => {
    getNotificationUpdates(userToken).then((response) => {
      setUpdatesNotif(response.result);
    });

    getNotificationActivityJoined(userToken).then((response) => {
      setJoinedCommunityNotif(response.result);
    });

    getNotificationActivityManaged(userToken).then((response) => {
      setManagedCommunityNotif(response.result);
      let indexWithModal = -1;
      for (let i = 0; i < response.result.reminder.length; i++) {
        if (response.result.reminder[i].additional_data.modal !== null) {
          indexWithModal = i;
          break;
        }
      }
      setMemberRequestNotif(response.result?.reminder[indexWithModal]?.additional_data?.modal)
      if (response.result?.reminder[indexWithModal]?.additional_data?.modal) {
        setCommunityName(response.result?.reminder[indexWithModal].content)
        setCommunityId(response.result?.reminder[indexWithModal].community?.id_community)
        setDateModal(response.result?.reminder[indexWithModal].created_at)
        setIdNotifModal(response.result?.reminder[indexWithModal].id_notif)
      }
    });
  }
}

```



Monica Narda Davita

Kode Sumber 61 Notification: index.jsx 2/10

```
const handleJoinedMarkAll = () => {
  postReadAllNotification(userToken, { tab: "joined" }).then((response) => {
    const { code, result } = response;
    if (code === 200) {
      const updatedData = { ...joinedCommunityNotif };
      Object.keys(updatedData).forEach((notificationType) => {
        updatedData[notificationType] = updatedData[notificationType].map(
          (notification) => {
            return { ...notification, status: "read" };
          }
        );
      });
      setJoinedCommunityNotif(updatedData);
    }
  });
};

const handleManagedMarkAll = () => {
  postReadAllNotification(userToken, { tab: "managed" }).then((response) => {
    const { code, result } = response;
    if (code === 200) {
      const updatedData = { ...managedCommunityNotif };
      Object.keys(updatedData).forEach((notificationType) => {
        updatedData[notificationType] = updatedData[notificationType].map(
          (notification) => {
            return { ...notification, status: "read" };
          }
        );
      });
      setManagedCommunityNotif(updatedData);
    }
  });
};

const handleReadNotification = (id, section) => {
  postReadNotification(userToken, { id_notif: id }).then((response) => {
    const { code, result } = response;
    if (code === 200) {
      if (tabState === "Activity") {
        if (tabStateActivity === "joined") {
          var tempArr = { ...joinedCommunityNotif };
          const itemIndex = joinedCommunityNotif[section].findIndex(
            (notif) => notif.id_notif === id
          );
          if (itemIndex !== -1) {
            tempArr[section][itemIndex].status = "read";
            setJoinedCommunityNotif(tempArr);
          }
        } else if (tabStateActivity === "managed") {
          var tempArr = { ...managedCommunityNotif };
          const itemIndex = managedCommunityNotif[section].findIndex(
            (notif) => notif.id_notif === id
          );
          if (itemIndex !== -1) {
            tempArr[section][itemIndex].status = "read";
            setManagedCommunityNotif(tempArr);
          }
        }
      } else if (tabState === "Updates") {
        var tempArr = { ...updatesNotif };
        const itemIndex = updatesNotif[section].findIndex(
          (notif) => notif.id_notif === id
        );
        if (itemIndex !== -1) {
          tempArr[section][itemIndex].status = "read";
          setUpdatesNotif(tempArr);
        }
      }
    }
  });
};
```



Kode Sumber 62 Notification: index.jsx 3/10

```
const handleMemberAccepted = (id) => {
  console.log("In id user" + id);
  postInvitationConfirmationUserToken, {
    id_community: communityId,
    id_user: id,
    tab: "managed",
    confirmation: "accept",
    id_notif: idNotifModal
  })
  .then((response) => {
    const { code, result } = response;
    if (code === 200) {
      console.log('success')
    }
  })
  const updatedData = memberRequestNotif.map(member => {
    if (member.id_user === id) {
      return {
        ...member,
        status: "accepted"
      };
    }
    return member;
  });
  console.log('update data', updatedData)
  setMemberRequestNotif(updatedData);
}

const handleMemberDenied = (id) => {
  postInvitationConfirmationUserToken, { id_community: communityId, id_user: id, tab: 'managed', confirmation: 'denied', id_notif: idNotifModal })
  .then((response) => {
    const { code, result } = response;
    if (code === 200) {
      console.log('success')
    }
  })
  const updatedData = memberRequestNotif.map(member => {
    if (member.id_user === id) {
      return {
        ...member,
        status: "denied"
      };
    }
    return member;
  });
  console.log('update data', updatedData)
  setMemberRequestNotif(updatedData);
}
```

Kode Sumber 63 Notification: index.jsx 4/10

```
<Layout title="Notification">
  <Div flex className="x-notification-header">
    <Div className="x-header w-100">
      <Heading
        size="h1"
        weight="bold"
        className="pt-xxx-lg pl-xxx-big pb-xxx-big"
        fontFamily="poppins"
        text="Notifications"
      />
    </Div>
  </Div>
  <Spacer size={40} />
  <Container>
    <Div flex width="100%" className="border-bottom border-1 border-ne-400">
      <Div
        onClick={() => setTabState(tab[0])}
        pointer
        className={classNames(
          "border-bottom border-1",
          tabState === "Updates" && "border-v2-pr-500"
        )}
        width="fit-content"
      >
        <Div className="py-x-sm px-sm">
          <Label
            className="fw-semibold label-l16"
            color={tabState === "Updates" ? "v2-pr-500" : "ne-400"}
            text="Updates"
          />
        </Div>
      </Div>
      <Div
        onClick={() => setTabState(tab[1])}
        pointer
        className={classNames(
          "border-bottom border-1",
          tabState === "Activity" && "border-v2-pr-500"
        )}
        width="fit-content"
      >
        <Div className="py-x-sm px-sm">
          <Label
            className="fw-semibold label-l16"
            color={tabState === "Activity" ? "v2-pr-500" : "ne-400"}
            text="Activity"
          />
        </Div>
      </Div>
    </Div>
  </Container>
</Layout>
```



Monica Narda Davita

Kode Sumber 64 Notification: index.jsx 5/10


```

(tabState === "Activity" 65 (
  <
    <Spacer size={16} />
    <Div flex between width="100%">
      <Div flex className="gap-x-sm">
        <Div
          flex
          center
          style={{ borderRadius: "40px" }}
          onClick={() => setTabStateActivity(tabActivity[0])}
          pointer
          className={classNames(
            "py-x-sm px-big",
            tabStateActivity === "joined" ? "bg-v2-pr-50" : "bg-ne-25"
          )}
          width="fit-content"
        >
          <Div className="py-x-sm px-sm">
            <Label
              className="fw-semibold label-l16"
              color={
                tabStateActivity === "joined" ? "v2-pr-500" : "ne-500"
              }
              text="Joined Community"
            />
          </Div>
        </Div>
        <Div
          flex
          center
          style={{ borderRadius: "40px" }}
          onClick={() => setTabStateActivity(tabActivity[1])}
          pointer
          className={classNames(
            "py-x-sm px-big",
            tabStateActivity === "managed" ? "bg-v2-pr-50" : "bg-ne-25"
          )}
          width="fit-content"
        >
          <Div className="py-x-sm px-sm">
            <Label
              className="fw-semibold label-l16"
              color={
                tabStateActivity === "managed" ? "v2-pr-500" : "ne-400"
              }
              text="Managed Community"
            />
          </Div>
        </Div>
      </Div>
    <Paragraph
      className="paragraph-p16 fw-semibold v2-pr-500"
      text="Mark all as read"
      onClick={
        tabStateActivity === "joined"
          ? handleJoinedMarkAll
          : handleManagedMarkAll
      }
    />
  )
)

```



Monica Narda Davita



</Div>



Kode Sumber 65 Notification: index.jsx 6/10

```

{tabState === "Activity" &&
  tabStateActivity === "joined" &&
  (!isEmpty(joinedCommunityNotif.reminder) ||
  !isEmpty(joinedCommunityNotif.invitation) ||
  !isEmpty(joinedCommunityNotif.new_activity)) ? (
  <
  <Spacer size={24} />
  <Div
    id="activity-joined-reminderForYou"
    width="100%"
    flex
    column
  >
  <Paragraph
    marginless
    text="Reminder for You"
    className="fw-semibold paragraph-p18 ne-900 pl-big"
  />
  <Spacer size={8} />
  {joinedCommunityNotif.reminder.map((notif) => (
    <JoinedCommunity
      notifType="page"
      id={notif.id_notif}
      onRead={({id} =>
        handleReadNotification(id, notif?.section)
      )}
      section={notif.section}
      context={notif.content}
      date={notif.created_at}
      status={notif.status}
      source={notif?.notif_from}
    />
  )))}
  </Div>

  <Spacer size={12} />
  <Div id="activity-joined-newForYou" width="100%" flex column>
  <Paragraph
    marginless
    text="Invitation"
    className="fw-semibold paragraph-p18 ne-900 pl-big"
  />
  <Spacer size={8} />
  {joinedCommunityNotif.invitation.slice(0, 5).map((notif) => (
    <JoinedCommunity
      notifType="page"
      id={notif.id_notif}
      onRead={({id} =>
        handleReadNotification(id, notif?.section)
      )}
      showModalInvitation={() => setShowModalInvitation(true)}
      section={notif.section}
      date={notif.created_at}
      status={notif.status}
      context={notif.content}
      modal={notif?.additional_data?.modal}
    />
  )))}
  </Div>
}

```


Monica Nardà Davita


Kode Sumber 66 Notification: index.jsx 7/10

```

<Spacer size={12} />
<Div id="activity-joined-invitation" width="100%" flex column>
  <Paragraph
    marginless
    text="New For You"
    className="fw-semibold paragraph-p18 ne-900 pl-big"
  />
  <Spacer size={8} />
  {joinedCommunityNotif.new_activity.map((notif) => (
    <JoinedCommunity
      notifType="page"
      id={notif.id_notif}
      onRead={() =>
        handleReadNotification(id, notif?.section)
      }
      section="new_for_you"
      date={notif.created_at}
      status={notif.status}
      context={notif?.additional_data?.type}
      content={notif?.content}
      profilePicture={notif?.community?.image}
      postTitle={notif?.additional_data?.post?.description}
      postPicture={notif?.additional_data?.post?.image}
      postDate="16 April 2023"
      postTime="16.30"
      post={notif?.additional_data?.post}
      url={notif?.url}
    />
  ))}
</Div>
) : (
  tabState === "Activity" &&
  tabStateActivity === "joined" &&
  isEmpty(joinedCommunityNotif.reminder) &&
  isEmpty(joinedCommunityNotif.invitation) &&
  isEmpty(joinedCommunityNotif.new_activity) && (
    <Div flex column width="100%" center>
      <Image src={NotifEmpty} width={300} height={300} />
      <Spacer size={40} />
      <Paragraph
        size="h5"
        fontFamily="poppins"
        weight="bold"
        marginless
        text="No Notification Yet"
      />
      <Spacer size={8} />
      <Paragraph
        centered
        className="label-l16 font-roboto-regular px-x-big ne-600"
        marginless
        alignCenter
        text="Go Active! Join a community that suits your interest and the notification will show up here."
      />
      <Spacer size={32} />
      <Button
        // onClick={()
        width="148px"
        type="submit"
        variant="primary"
      />
      <Button>
        Start to Explore
      </Button>
    </Div>
  )
)
}

```



Monica Narda Davita

Kode Sumber 67 Notification: index.jsx 8/10

```

<table className="Activity" >
  <tbody>
    <tr>
      <td>
        <div className="Activity" >
          <div className="Header" >
            <div className="Title" >
              <h3>Activity: Manage</h3>
            </div>
            <div className="Description" >
              <p>Manage the community members and actions</p>
            </div>
          </div>
          <div className="Content" >
            <div className="List" >
              <ul>
                <li>
                  <div className="Item" >
                    <div className="Header" >
                      <h4>Manage Members</h4>
                    </div>
                    <div className="Table" >
                      <table>
                        <thead>
                          <tr>
                            <th>Name</th>
                            <th>Email</th>
                            <th>Role</th>
                            <th>Actions</th>
                          </tr>
                        </thead>
                        <tbody>
                          <tr>
                            <td>John Doe</td>
                            <td>john.doe@example.com</td>
                            <td>Admin</td>
                            <td>
                              <button type="button">Edit</button>
                              <button type="button">Delete</button>
                            </td>
                          </tr>
                        </tbody>
                      </table>
                    </div>
                  </div>
                </li>
                <li>
                  <div className="Item" >
                    <div className="Header" >
                      <h4>Manage Actions</h4>
                    </div>
                    <div className="Table" >
                      <table>
                        <thead>
                          <tr>
                            <th>Action Name</th>
                            <th>Status</th>
                            <th>Created At</th>
                            <th>Updated At</th>
                            <th>Actions</th>
                          </tr>
                        </thead>
                        <tbody>
                          <tr>
                            <td>Post a new post</td>
                            <td>Active</td>
                            <td>2023-10-27 10:00:00</td>
                            <td>2023-10-27 10:00:00</td>
                            <td>
                              <button type="button">Edit</button>
                              <button type="button">Delete</button>
                            </td>
                          </tr>
                        </tbody>
                      </table>
                    </div>
                  </div>
                </li>
                <li>
                  <div className="Item" >
                    <div className="Header" >
                      <h4>Manage Engagement</h4>
                    </div>
                    <div className="Table" >
                      <table>
                        <thead>
                          <tr>
                            <th>Engagement Type</th>
                            <th>Status</th>
                            <th>Created At</th>
                            <th>Updated At</th>
                            <th>Actions</th>
                          </tr>
                        </thead>
                        <tbody>
                          <tr>
                            <td>Like a post</td>
                            <td>Active</td>
                            <td>2023-10-27 10:00:00</td>
                            <td>2023-10-27 10:00:00</td>
                            <td>
                              <button type="button">Edit</button>
                              <button type="button">Delete</button>
                            </td>
                          </tr>
                        </tbody>
                      </table>
                    </div>
                  </div>
                </li>
              </ul>
            </div>
          </div>
        </div>
      </td>
    </tr>
  </tbody>
</table>

```

Kode Sumber 68 Notification: index.jsx 9/10

```

import React, { useState, useEffect } from "react";
import { isEmpty, set } from "lodash-es";
import {
  Div,
  Heading,
  Label,
  Button,
  Paragraph,
  Span,
  Spacer,
  UpdateComponent,
  ManagedCommunity,
  ModalInvitationNotif,
  ModalRequestNotif,
} from "@components";
import { Offcanvas } from "react-bootstrap";
import classNames from "classnames";
import { GoPrimitiveDot } from "react-icons/go";
import { BsChevronDown, BsChevronUp } from "react-icons/bs";
import BellIcon from "../../public/assets/icons/BellIcon.svg";
import ProfilePict from "../../public/assets/images/Person.svg";
import NotifPostAsset from "../../public/assets/images/NotifPostAsset.svg";
import NotifEmpty from "../../public/assets/images/img-empty-notification.png";
import JoinedCommunity from "../JoinedCommunity";
import { useRouter } from "next/router";
import ModalMemberRequestNotif from "@components/modal/ModalMemberRequestNotif";
import {
  getNotificationActivityJoined,
  getNotificationActivityManaged,
  getNotificationUpdates,
  postReadAllNotification,
  postReadNotification,
  postInvitationConfirmation,
} from "@services/actions/Api";
import { getUserToken } from "@services/utills/Helper";
import { date } from "yup";
import { cookie } from "next/cookie";
import Image from "next/image";

```

Kode Sumber 69 Notification: index.jsx 10/10

5.1.6.2. NotificationSidebar Component

```

import React, { useState, useEffect } from "react";
import { isEmpty, set } from "lodash-es";
import {
  Div,
  Heading,
  Label,
  Button,
  Paragraph,
  Span,
  Spacer,
  UpdateComponent,
  ManagedCommunity,
  ModalInvitationNotif,
  ModalRequestNotif,
} from "@components";
import { Offcanvas } from "react-bootstrap";
import classNames from "classnames";
import { GoPrimitiveDot } from "react-icons/go";
import { BsChevronDown, BsChevronUp } from "react-icons/bs";
import BellIcon from "../../public/assets/icons/BellIcon.svg";
import ProfilePict from "../../public/assets/images/Person.svg";
import NotifPostAsset from "../../public/assets/images/NotifPostAsset.svg";
import NotifEmpty from "../../public/assets/images/img-empty-notification.png";
import JoinedCommunity from "../JoinedCommunity";
import { useRouter } from "next/router";
import ModalMemberRequestNotif from "@components/modal/ModalMemberRequestNotif";
import {
  getNotificationActivityJoined,
  getNotificationActivityManaged,
  getNotificationUpdates,
  postReadAllNotification,
  postReadNotification,
  postInvitationConfirmation,
} from "@services/actions/Api";
import { getUserToken } from "@services/utills/Helper";
import { date } from "yup";
import { cookie } from "next/cookie";
import Image from "next/image";

```

Kode Sumber 70 Notification: NotificationSidebar.jsx 1/8

```

const NotificationSidebar = ({ onClose, onShow }) => {
  const { push } = useRouter();

  const [isActive, setIsActive] = useState("Update");
  const [showDropdown, setShowDropdown] = useState(false);
  const [showModalInvitation, setShowModalInvitation] = useState(false);
  const [showModalRequestNotif, setShowModalRequestNotif] = useState(false);
  const [showModalMemberRequest, setShowModalMemberRequest] = useState(false);
  const [dropdownActive, setDropdownActive] = useState("Joined Community");
  const [idNotifModal, setIdNotifModal] = useState()
  const [updatesNotif, setUpdatesNotif] = useState();
  const [joinedCommunityNotif, setJoinedCommunityNotif] = useState();
  const [managedCommunityNotif, setManagedCommunityNotif] = useState();
  const [invitationNotif, setInvitationNotif] = useState(DUMMY_DATA_INVITATION);
  const [requestNotif, setRequestNotif] = useState(DUMMY_DATA_REQUEST);
  const [memberRequestNotif, setMemberRequestNotif] = useState();
  const [communityId, setCommunityId] = useState();
  const [communityName, setCommunityName] = useState();
  const [dateModal, setDateModal] = useState();
  const userToken = getUserToken();

  useEffect(() => {
    getNotificationUpdates(userToken).then((response) => {
      setUpdatesNotif(response.result);
      // setInvitationNotif(response.result.invitation[0]?.additional_data?.modal)
      // setRequestNotif(response.result.invitation[1]?.additional_data?.modal)
    });

    getNotificationActivityJoined(userToken).then((response) => {
      setJoinedCommunityNotif(response.result);
    });

    getNotificationActivityManaged(userToken).then((response) => {
      setManagedCommunityNotif(response.result);
      let indexWithModal = -1;
      for (let i = 0; i < response.result.reminder.length; i++) {
        if (response.result.reminder[i].additional_data.modal !== null) {
          indexWithModal = i;
          break;
        }
      }
      // console.log('iki, ', response.result?.reminder[indexWithModal].additional_data?.modal)
      setMemberRequestNotif(
        response.result?.reminder[indexWithModal]?.additional_data?.modal
      );
      if (response.result?.reminder[indexWithModal]?.additional_data?.modal) {
        setCommunityName(response.result?.reminder[indexWithModal].content);
        setCommunityId(response.result?.reminder[indexWithModal].community?.id_community);
        setDateModal(response.result?.reminder[indexWithModal].created_at);
        setIdNotifModal(response.result?.reminder[indexWithModal].id_notif)
      }
    });
  });
};

```



Monica Narda Davita

Kode Sumber 71 Notification: NotificationSidebar.jsx 2/8

```
const handleJoinedMarkAll = () => {
  postReadAllNotification(userToken, { tab: "joined" }).then((response) => {
    const { code, result } = response;
    if (code === 200) {
      const updatedData = { ...joinedCommunityNotif };
      Object.keys(updatedData).forEach((notificationType) => {
        updatedData[notificationType] = updatedData[notificationType].map(
          (notification) => {
            return { ...notification, status: "read" };
          }
        );
      });
      setJoinedCommunityNotif(updatedData);
    }
  });
};

const handleManagedMarkAll = () => {
  postReadAllNotification(userToken, { tab: "managed" }).then((response) => {
    const { code, result } = response;
    if (code === 200) {
      const updatedData = { ...managedCommunityNotif };
      Object.keys(updatedData).forEach((notificationType) => {
        updatedData[notificationType] = updatedData[notificationType].map(
          (notification) => {
            return { ...notification, status: "read" };
          }
        );
      });
      setManagedCommunityNotif(updatedData);
    }
  });
};

const handleReadNotification = (id, section) => {
  postReadNotification(userToken, { id_notif: id }).then((response) => {
    const { code, result } = response;
    if (code === 200) {
      if (isActive === "Activity") {
        if (dropdownActive === "Joined Community") {
          var tempArr = { ...joinedCommunityNotif };
          const itemIndex = joinedCommunityNotif[section].findIndex(
            (notif) => notif.id_notif === id
          );
          if (itemIndex !== -1) {
            tempArr[section][itemIndex].status = "read";
            setJoinedCommunityNotif(tempArr);
          }
        } else if (dropdownActive === "Managed Community") {
          var tempArr = { ...managedCommunityNotif };
          const itemIndex = managedCommunityNotif[section].findIndex(
            (notif) => notif.id_notif === id
          );
          if (itemIndex !== -1) {
            tempArr[section][itemIndex].status = "read";
            setManagedCommunityNotif(tempArr);
          }
        }
      } else if (isActive === "Updates") {
        var tempArr = { ...updatesNotif };
        const itemIndex = updatesNotif[section].findIndex(
          (notif) => notif.id_notif === id
        );
        if (itemIndex !== -1) {
          tempArr[section][itemIndex].status = "read";
          setUpdatesNotif(tempArr);
        }
      }
    }
  });
};
```

Kode Sumber 72 Notification: NotificationSidebar.jsx 3/8

```
const handleMemberAccepted = (id) => {
  console.log('ini id user' + id);
  postInvitationConfirmationUserToken, {
    id_community: communityId,
    id_user: id,
    tab: "managed",
    confirmation: "accept",
    id_notif: idNotifModal
  })
  .then((response) => {
    const { code, result } = response;
    if (code === 200) {
      console.log('success')
    }
  })
  const updatedData = memberRequestNotif.map(member => {
    if (member.id_user === id) {
      return {
        ...member,
        status: "accepted"
      };
    }
    return member;
  });
  console.log('update data', updatedData)
  setMemberRequestNotif(updatedData);
}

const handleMemberDenied = (id) => {
  postInvitationConfirmationUserToken, { id_community: communityId, id_user: id, tab: 'managed', confirmation: 'denied', id_notif: idNotifModal })
  .then((response) => {
    const { code, result } = response;
    if (code === 200) {
      console.log('success')
    }
  })
  const updatedData = memberRequestNotif.map(member => {
    if (member.id_user === id) {
      return {
        ...member,
        status: "denied"
      };
    }
    return member;
  });
  console.log('update data', updatedData)
  setMemberRequestNotif(updatedData);
}
```

Kode Sumber 73 Notification: NotificationSidebar.jsx 4/8

5.1.6.6. dateNotificationFormatter Helper

```
export const dateNotificationFormatter = (date) => {
  const currentDate = new Date();
  const inputDate = new Date(date);

  // Calculate the time difference in milliseconds
  const timeDiff = currentDate - inputDate;

  // Convert milliseconds to seconds, minutes, hours, and days
  const seconds = Math.floor(timeDiff / 1000);
  const minutes = Math.floor(seconds / 60);
  const hours = Math.floor(minutes / 60);
  const days = Math.floor(hours / 24);
  const weeks = Math.floor(days / 7);

  if (seconds < 60) {
    return 'Just Now';
  } else if (minutes < 60) {
    if(minutes === 1) return `${minutes} minute ago`;
    return `${minutes} minutes ago`;
  } else if (hours < 24) {
    if(hours === 1) return `1 hour ago`;
    return `${hours} hours ago`;
  } else if (days < 7) {
    if(days === 1) return `1 day ago`;
    return `${days} days ago`;
  } else {
    if(weeks === 1) return `1 week ago`;
    return `${weeks} weeks ago`;
  }
}
```



Monica Narda Davita



Kode Sumber 81 Edit Profile: Helper dateNotificationFormatter

5.2.1.2. Introduction Component

```
import React, { useEffect, useState } from 'react'
import { Div, Span, Heading, Paragraph, Link, Layout, Spacer, I from '@components'
import Image from 'next/image'
import classNames from 'classnames'
import LogoTelkom from './../public/assets/images/LogoTelkom.svg'
import Line from './../public/assets/images/Line.svg'
import AmoebaB from './../public/assets/logo-LogoAmoeba/Amoeba-4.svg'
import AmoebaP from './../public/assets/logo-LogoAmoeba/Amoeba-2.svg'
import AmoebaH from './../public/assets/logo-LogoAmoeba/Amoeba-3.svg'
import AmoebaL from './../public/assets/logo-LogoAmoeba/Amoeba-5.svg'
import { NavBox } from '@components'

const Introduction = ({ video }) => {
  const headlineA = ['Digital Amoeba', 'The Big Things', 'Real Game Changer']
  const logosA = [AmoebaB, AmoebaP, AmoebaH, AmoebaL]
  }

  const [subHeadline, setSubheadline] = useState(headlineA[0]);
  const [logo, setLogo] = useState(logosA[0]);
  const [triggerFade, setTriggerFade] = useState(false);
  const [currentIndex, setCurrentIndex] = useState(0);
  const [currentIndexLogo, setCurrentIndexLogo] = useState(0);

  useEffect(() => {
    const interval = setInterval(() => {
      setCurrentIndex((prevIndex) => (prevIndex + 1) % headlineA.length);
    }, 4000);

    return () => clearInterval(interval);
  }, []);

  useEffect(() => {
    const intervalLogo = setInterval(() => {
      setCurrentIndexLogo((prevIndex) => (prevIndex + 1) % logosA.length);
    }, 2000);

    return () => clearInterval(intervalLogo);
  }, []);

  useEffect(() => {
    setTriggerFade(false);
    setTimeout(() => {
      setTriggerFade(true);
    }, 80);
    setHeadline(headlineA[currentIndex]);
  }, [currentIndex]);

  useEffect(() => {
    setSubheadline(headlineA[currentIndex + 1] % headlineA.length);
  }, [currentIndex]);

  useEffect(() => {
    setLogo(logosA[(currentIndexLogo + 1) % logosA.length]);
  }, [currentIndexLogo]);

  useEffect(() => {
    setTriggerFade(false);
    setTimeout(() => {
      setTriggerFade(true);
    }, 8);
  }, [headline, subHeadline]);

  return (
    <
      <div className="x-introduction">
        <h2>Logo (Logo) </>
        <div className="x-liquid-background">
          <div flex column className="container w-100 h-100 gap-1g">
            <Spacer size=20 >
            <Spacer size=50 >
            <div flex alignItems className="pl-xxx-lg width=100% zIndex=99">
              <heading fontSize="60px" fontWeight="500" color="pr-00" className="font-amoeba-sans-normal">Aw Aw.</heading>
              <div flex center width="200px" height="60px" className="pl-xxx-lg px-x-lg px-headline-quotes">
                <heading marginless fontSize="20px" fontWeight="400" text="Imagine. Design. Develop. Crazy Ideas Into Reality." color="pr-00" />
              </div>
              <div relative>
                <heading marginless fontSize="10px" className="x-powered-by" color="pr-00">
                  Powered by
                </heading>
                <Image src={LogoTelkom} alt="logo telkom" />
              </div>
            </div>
          </div>
          <div flex alignCenter gap="xxx-big" zIndex=99 width="fit-content" className={classNames('py-xxx-big px-xxx-lg x-hero-headline')}>
            <heading marginless color="ffffff" text={headline} className={classNames('font-amoeba-sans-normal pr-00', triggerFade & "x-fade-in-headlines")} fontSize="60px" />
            <Image src={line} alt="line" />
          </div>
          <div>
            <div flex alignCenter gap="xxx-big" width="fit-content" className={classNames('font-amoeba-sans-normal pl-xxx-lg x-hero-incoming', triggerFade & "x-fade-in-subheadlines")}>
              <subHeadline />
            </div>
            <div flex alignCenter gap="xxx-big" width="fit-content" height="100px" center>
              <div className="embed-responsive embed-responsive-4by3 x-video-container">
                <video className="embed-responsive-item x-video-frame" autoPlay muted playsinline loop>
                  <source src={video} type="video/mp4" />
                  { /* Provide fallback content for browsers that do not support the video element */ }
                  Your browser does not support the video tag.
                </video>
              </div>
              <div className="x-video-cta">
                <heading text="Imagine. Design. Develop. Crazy Ideas Into Reality." lineHeight="40px" color="pr-00" fontize="60px" fontWeight="500" className="x-biggest" />
                <heading text="We're help employees turning their crazy ideas into reality, and make them even crazier." lineHeight="20px" color="pr-00" className="x-smaller" fontize="20px" fontWeight="400" />
              </div>
            </div>
          </div>
        </div>
      </div>
    </
  );
};
```

Kode Sumber 83 Landing Page DigitalAmoeba: Introduction.jsx

```

<code>
</code>

```

Kode Sumber 84 Landing Page Digital Amoeba: Introduction.scss


```
.x-history {
  width: 100%;
  height: 100%;
  .x-radius-72{
    border-radius: 72px;
  }

  .x-section-default {
    border: 2px solid rgba(255, 255, 255, 0.24);
    border-radius: 72px;
  }

  .x-expand-transition {
    transition: max-width 500ms ease;
    max-width: 138px;
  }
  .x-expand {
    max-width: 538px;
    border: 3px solid #461CCB;
  }

  .x-fade-in-text{
    opacity: 1;
    animation: fadeIn 1s;
  }

  @keyframes fadeIn {
    0% { opacity: 0; }
    100% { opacity: 1; }
  }
}
```



Monica Narda Davita



Kode Sumber 86 Landing Page Digital Amoeba: History.scss

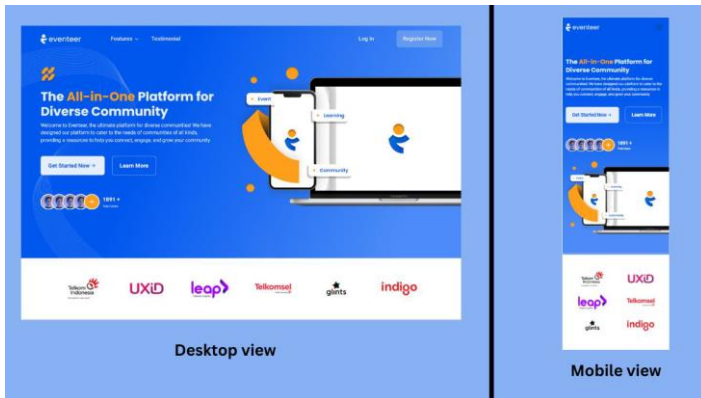
[Halaman ini sengaja dikosongkan]

BAB VI HASIL Pengerjaan

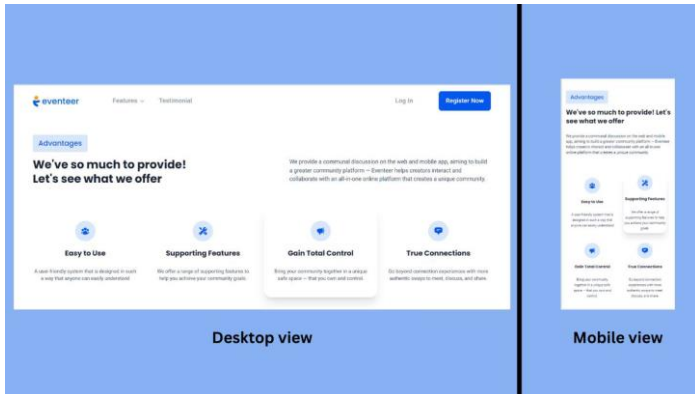
Bab ini menampilkan hasil akhir pengerjaan pengembangan ulang platform Eventeer berbasis website dan Landing Page Digital Amoeba.

6.1. Pengembangan Ulang Eventeer

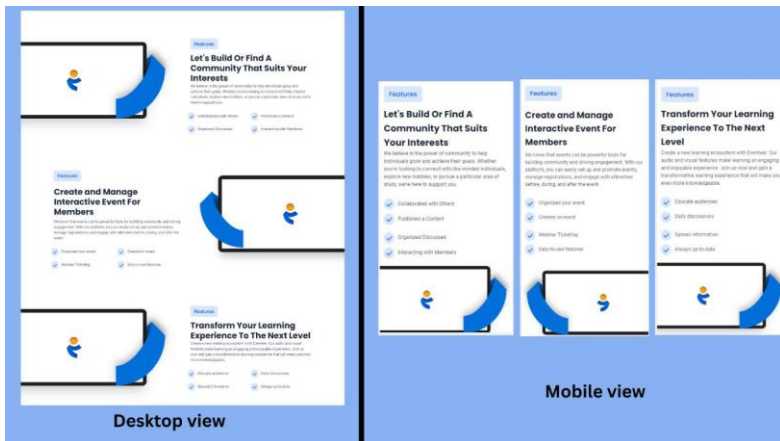
6.1.1. Landing Page



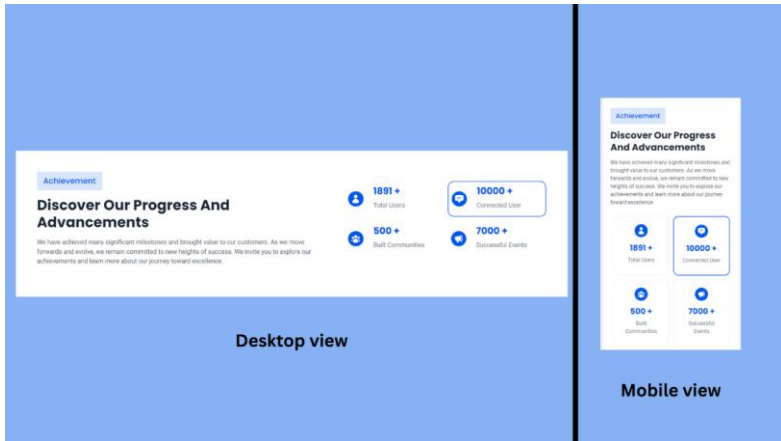
Gambar 12 Tampilan: Landing Page Hero Section



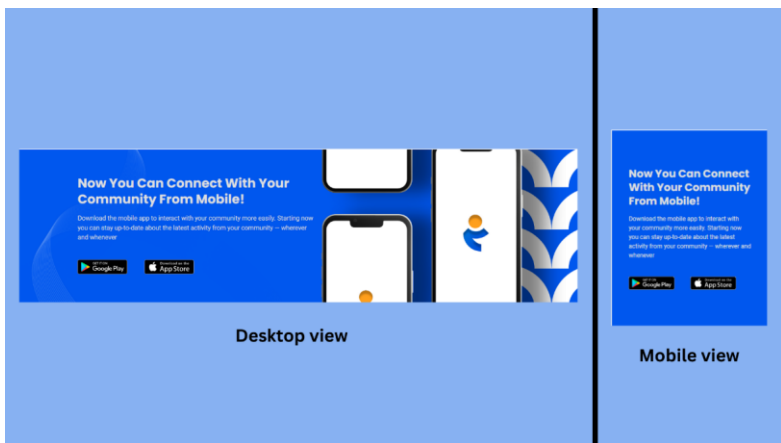
Gambar 13 Tampilan: Landing Page Advantages Section



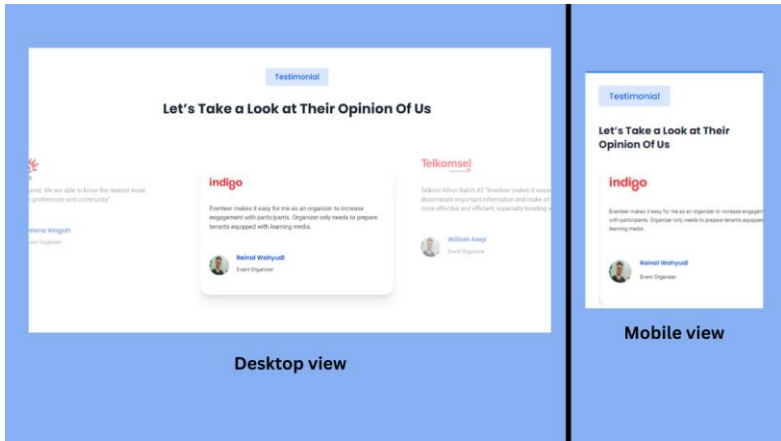
Gambar 14 Tampilan: Landing Page Features Section



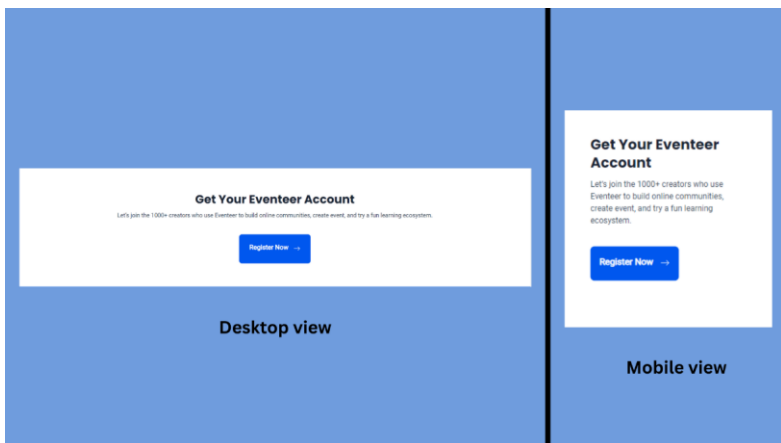
Gambar 15 Tampilan: Landing Page Achievement Section



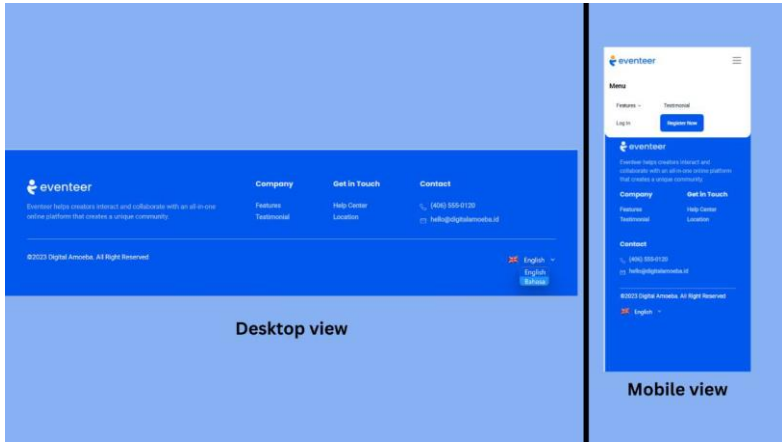
Gambar 16 Tampilan: Landing Page Download Section



Gambar 17 Tampilan: Landing Page Testimonial Section

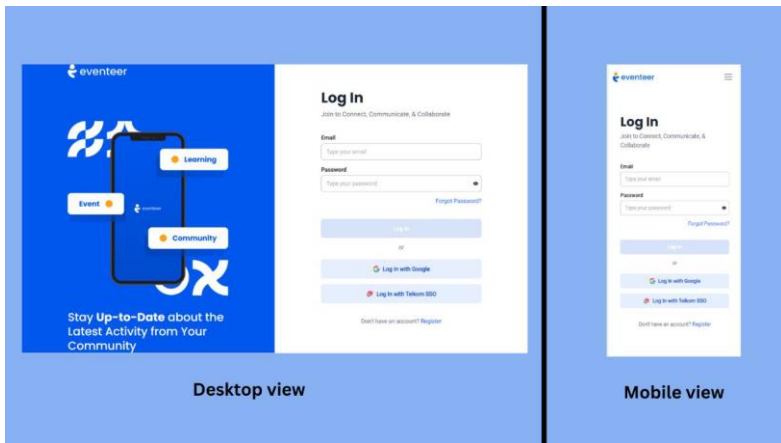


Gambar 18 Tampilan: Landing Page Get AccountSection

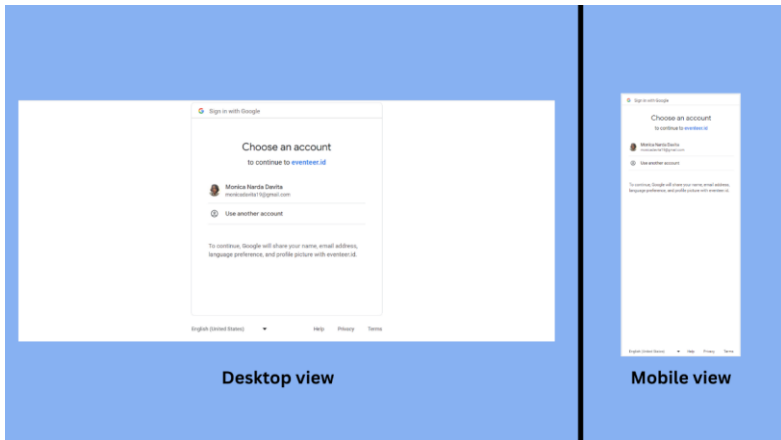


Gambar 19 Tampilan: Landing Page Footer Section

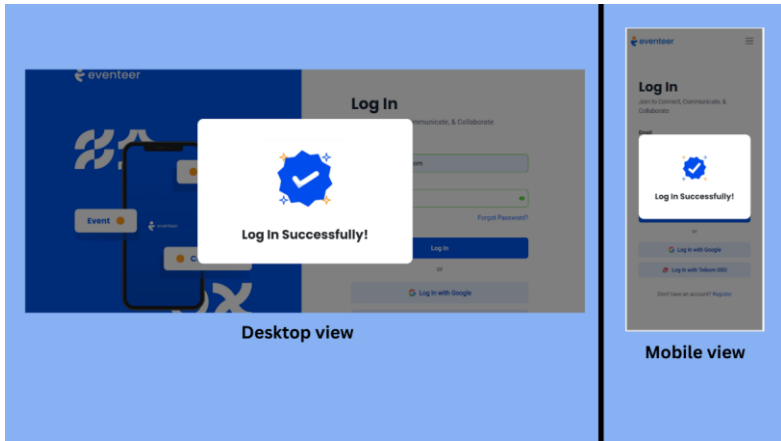
6.1.2. Login



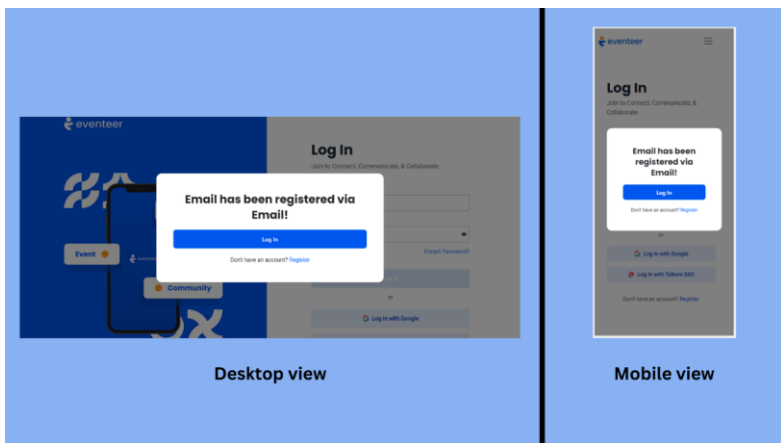
Gambar 20 Tampilan: Login Default View



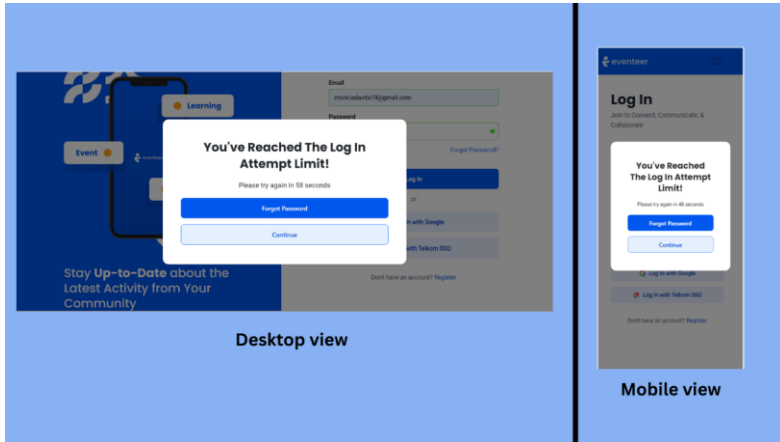
Gambar 21 Tampilan: Login with Google



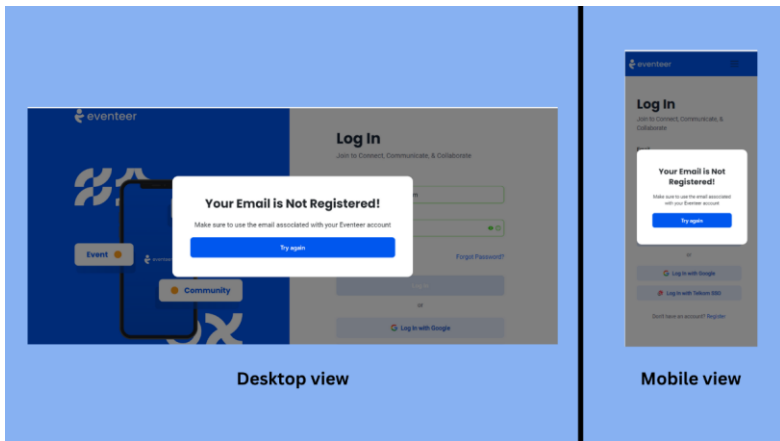
Gambar 22 Tampilan: Modal Login Sukses



Gambar 23 Tampilan: Modal Email Telah Terdaftar

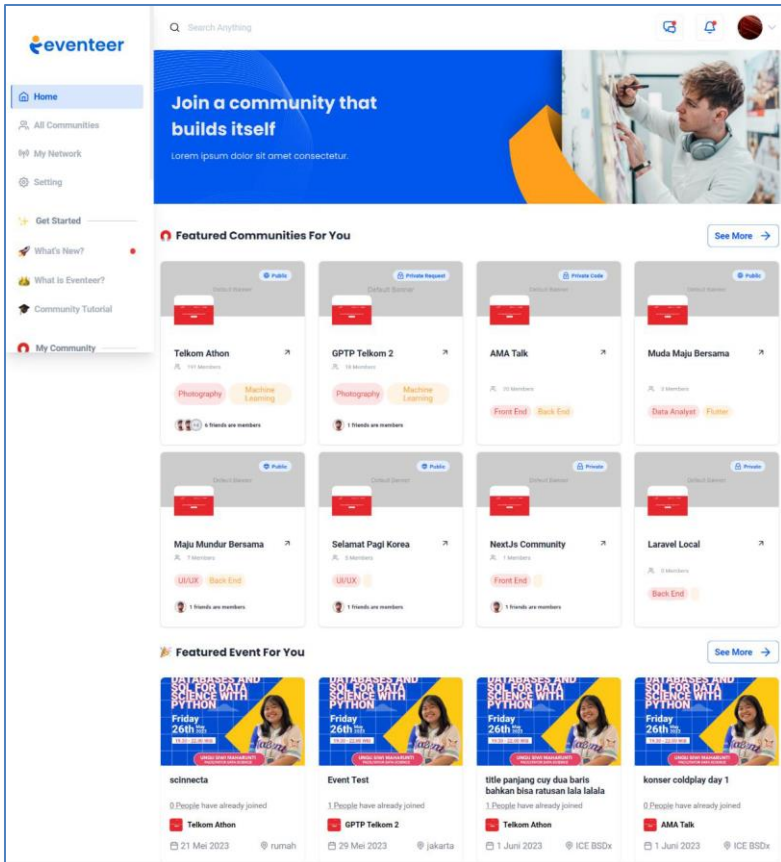


Gambar 24 Tampilan: Modal Percobaan Login Maksimum

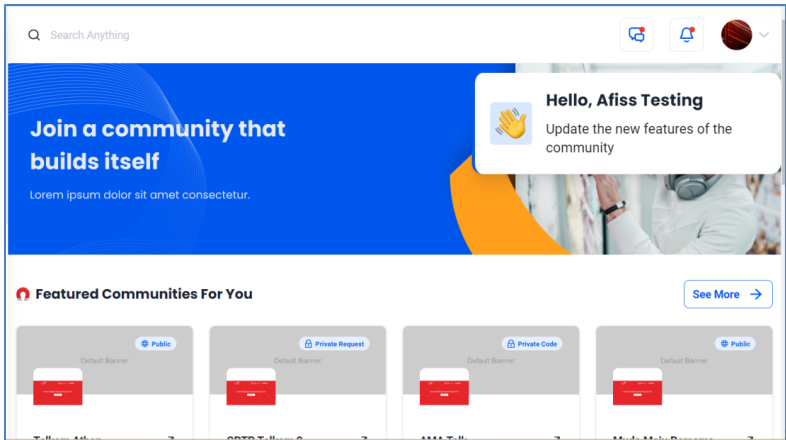


Gambar 25 Tampilan: Modal Email Belum Terdaftar

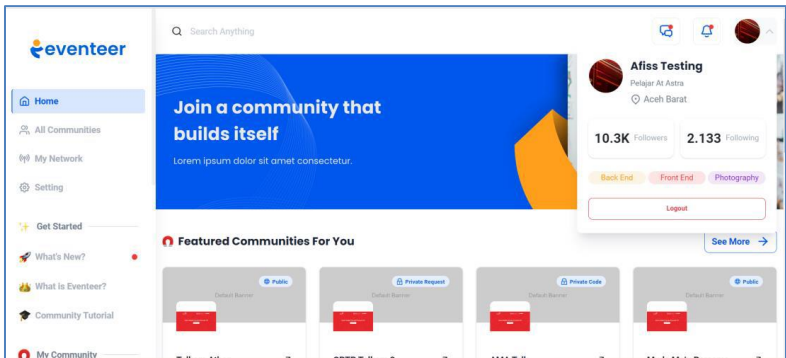
6.1.3. Homepage



Gambar 26 Tampilan: Homepage Default View



Gambar 27 Tampilan: Modal Hola Homepage



Gambar 28 Tampilan: Modal Profile

6.1.4. Edit Profile

The screenshot shows the 'Edit Profile' interface with the 'Personal Profile' tab selected. The left sidebar contains navigation options: Home, All Communities, My Network, Setting, Get Started, What's New?, What is Eventeer?, Community Tutorial, My Community, and Telkom Athlon. The main content area is titled 'PERSONAL PROFILE' and contains the following fields:

- Fullname:** A text input field containing 'Afas Testing'.
- Bio:** A text input field containing 'Big journey start from small step uhuy'.
- City/Location:** A dropdown menu with 'Aceh Barat' selected.
- Gender:** Two radio button options, 'Male' and 'Female', with 'Male' selected.

A blue 'Save' button is located in the top right corner of the form area.

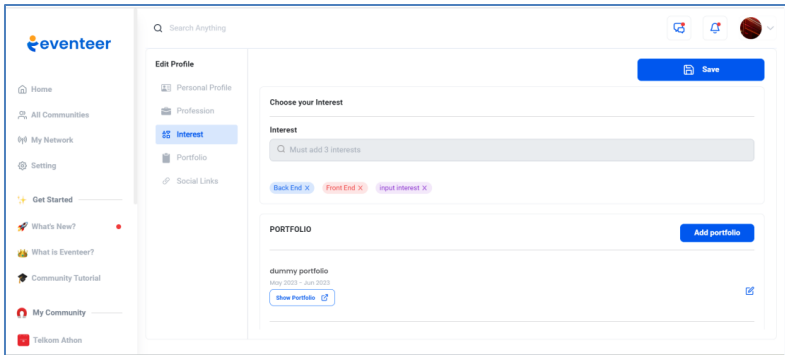
Gambar 29 Tampilan: Edit Profile Personal Profile

The screenshot shows the 'Edit Profile' interface with the 'Profession' tab selected. The left sidebar is identical to the previous image. The main content area is titled 'PROFESSION' and contains the following fields:

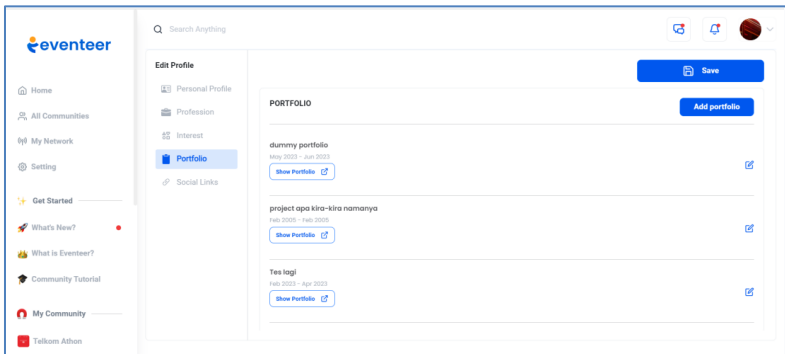
- Company/Institution Name:** A text input field containing 'Astra'.
- Profession:** A dropdown menu with 'Pelajar' selected.
- Choose your Interest:** A section with a heading 'Choose your Interest' and a sub-heading 'Interest'. Below it is a search input field with the text 'Must add 3 interests'.

A blue 'Save' button is located in the top right corner of the form area.

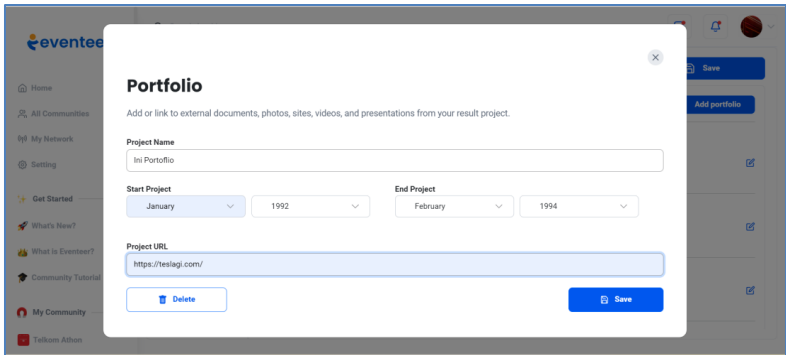
Gambar 30 Tampilan: Edit Profile Profession



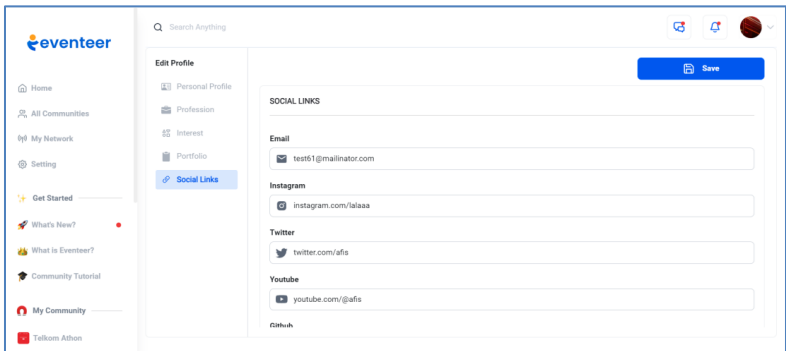
Gambar 31 Tampilan: Edit Profile Interest



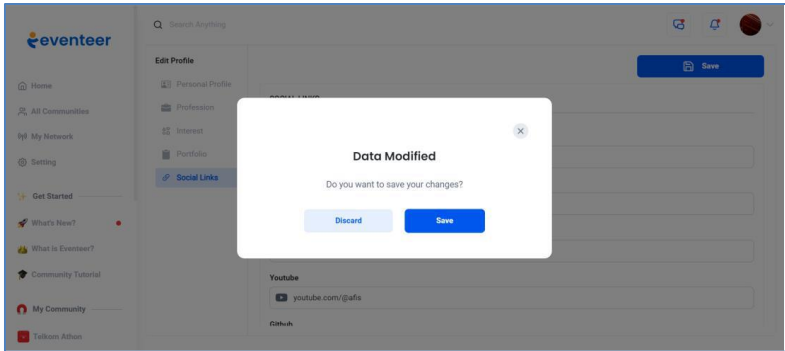
Gambar 32 Tampilan: Edit Profile Portfolio



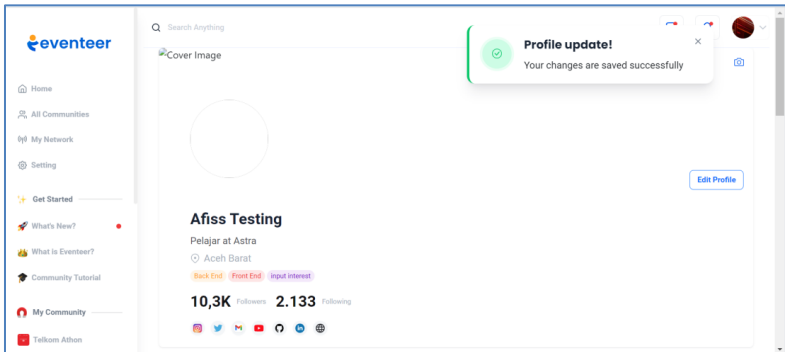
Gambar 33 Tampilan: Modal Add Portfolio



Gambar 34 Tampilan: Edit Profile Social Links

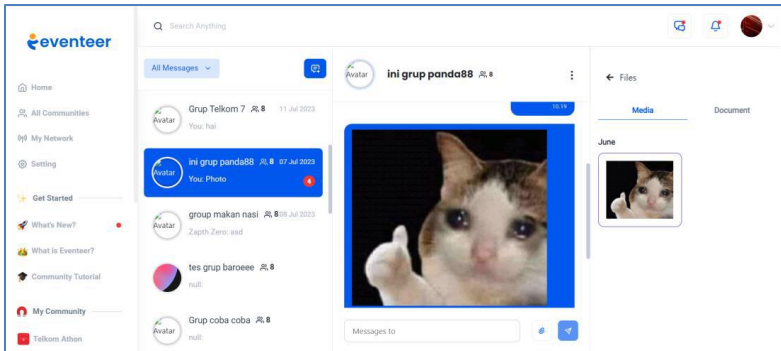


Gambar 35 Tampilan: Modal Save Profile Confirmation

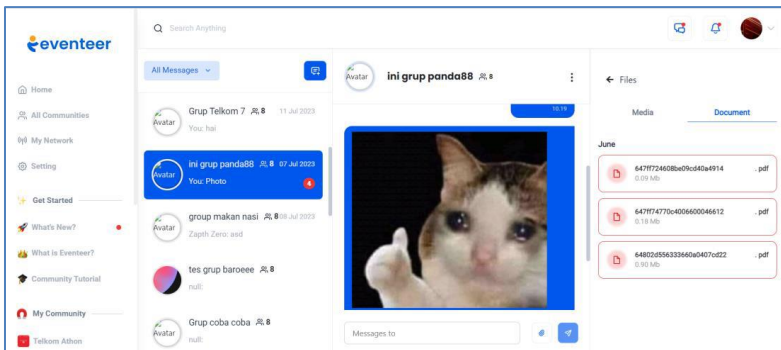


Gambar 36 Tampilan: Toast Edit Profile Sukses

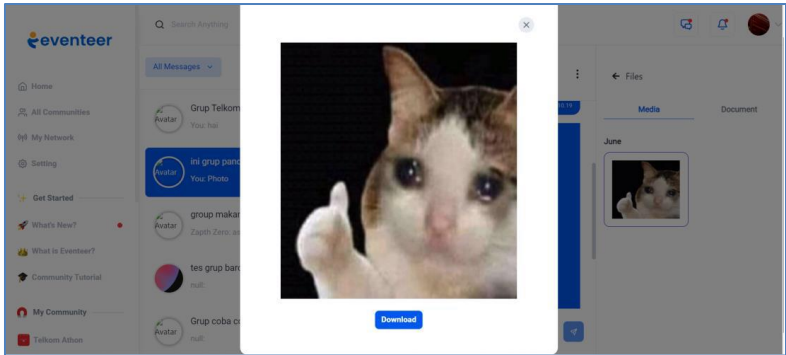
6.1.5. List Chat



Gambar 37 Tampilan: List Files Media

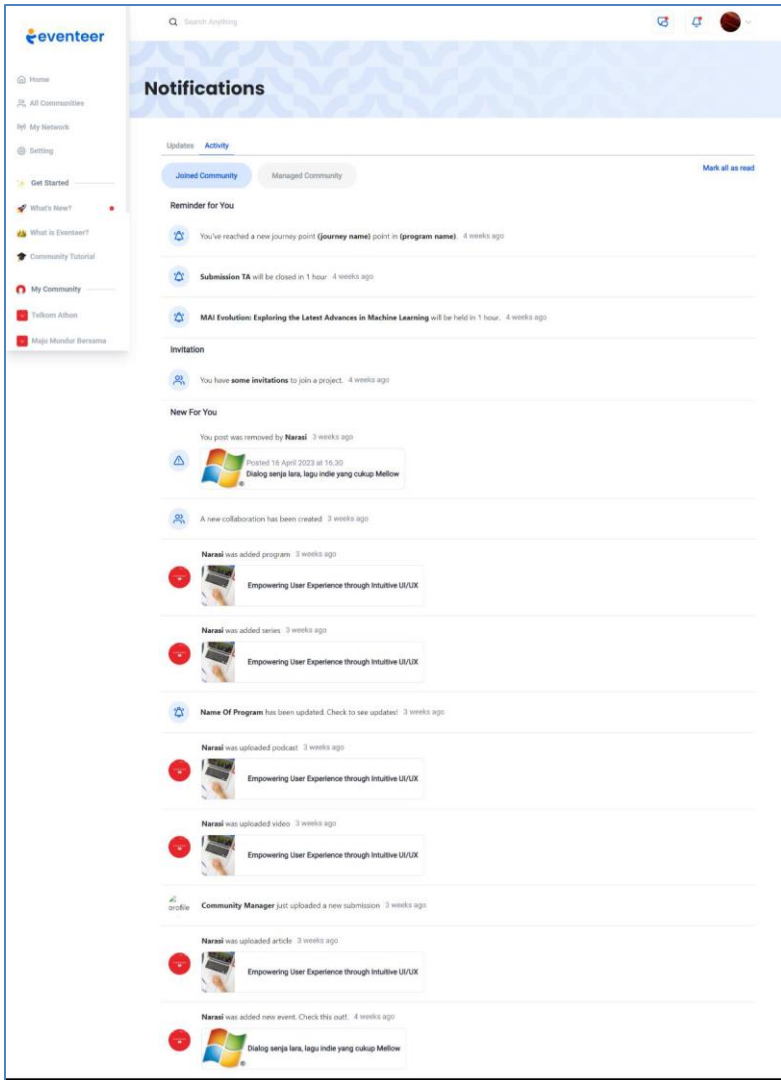


Gambar 38 Tampilan: List Files Document

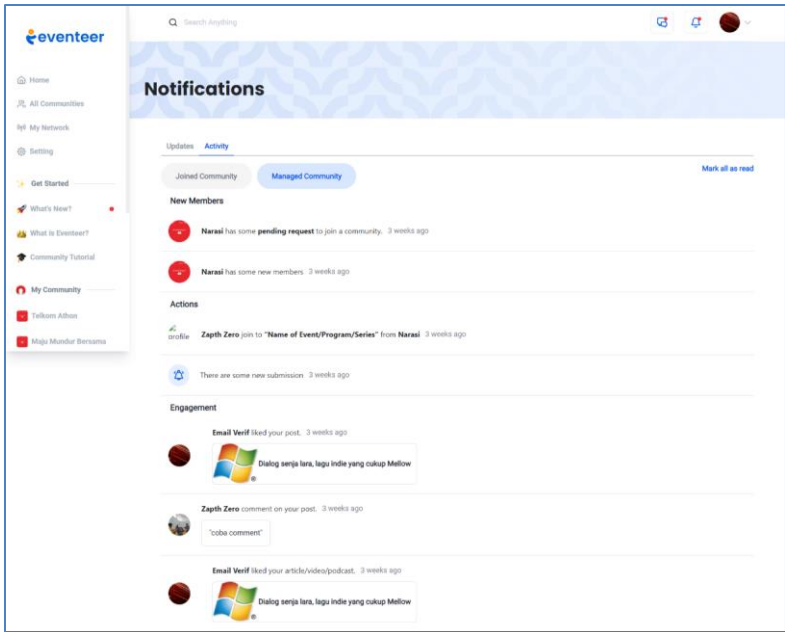


Gambar 39 Tampilan: Modal Show Photo

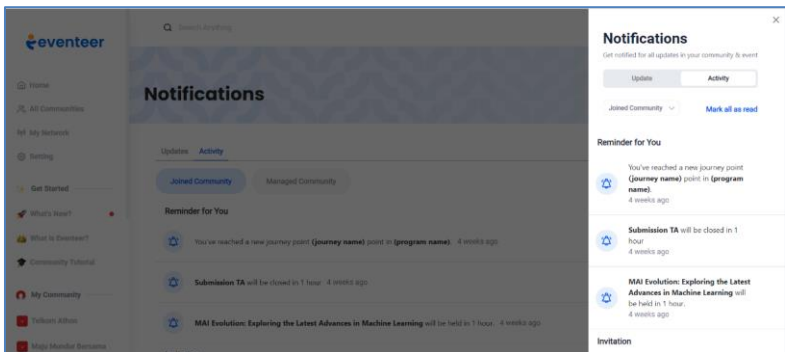
6.1.6. Notification



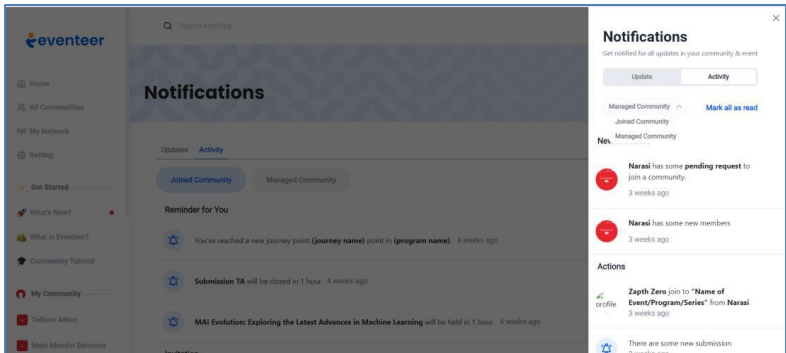
Gambar 40 Tampilan: Notification Page Tab Activity Joined Community



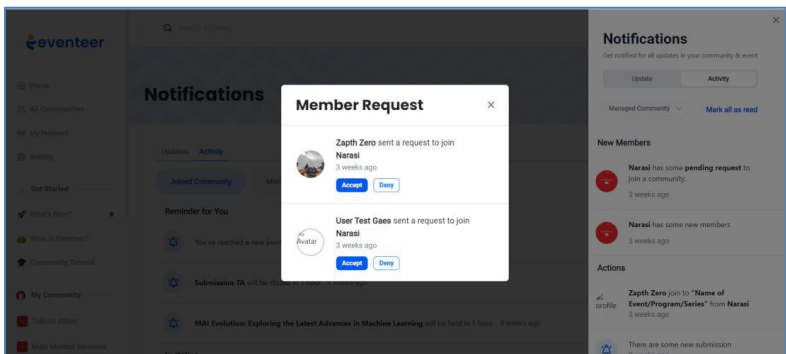
Gambar 41 Tampilan: Notification Page Tab Activity Managed Community



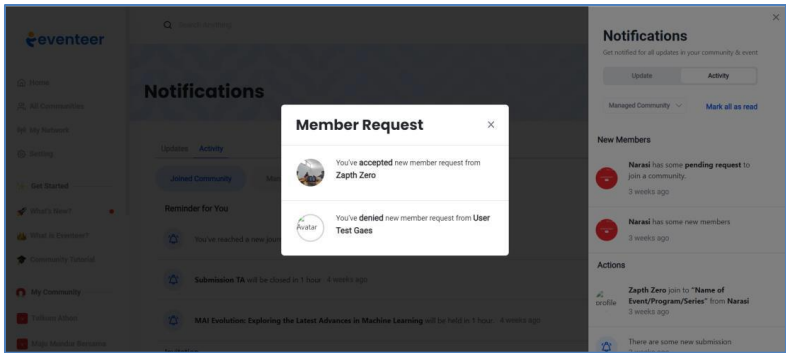
Gambar 42 Tampilan: Notification Sidebar Tab Activity Joined Community



Gambar 43 Tampilan: Notification Sidebar Tab Activity Managed Community



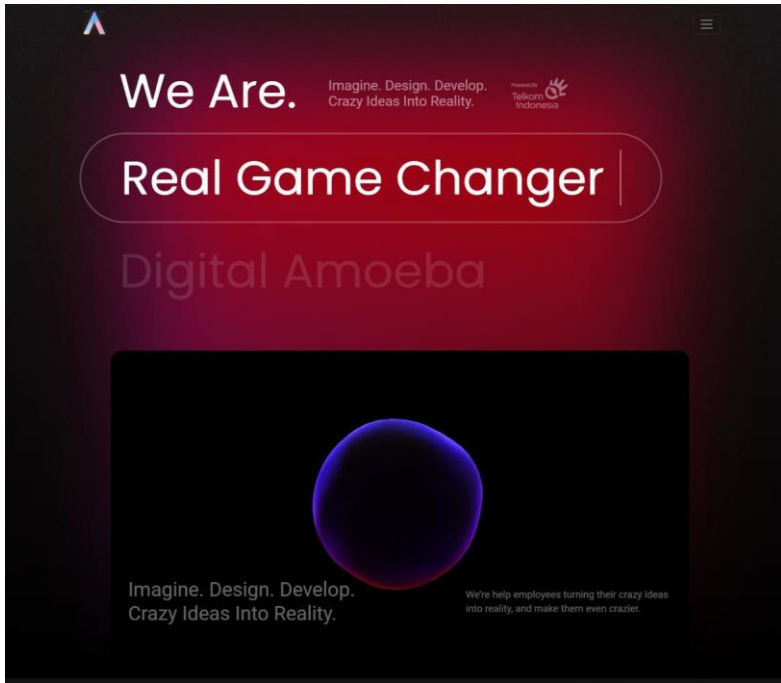
Gambar 44 Tampilan: Modal Member Request



Gambar 45 Tampilan: Modal Member Request Respond

6.7. Landing Page Digital Amoeba

6.7.1. Introduction

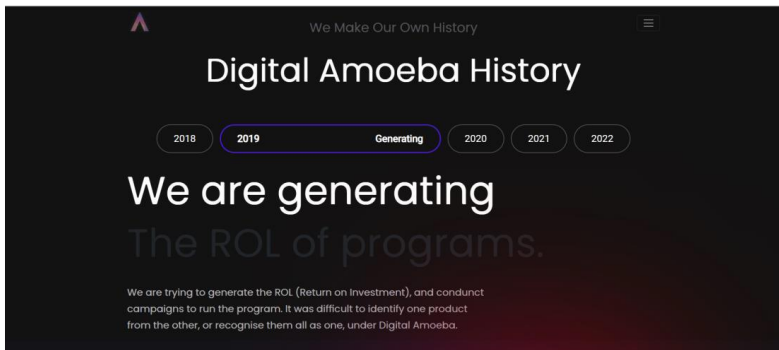


Gambar 46 Tampilan: Landing Page Digital Amoeba Hero Section

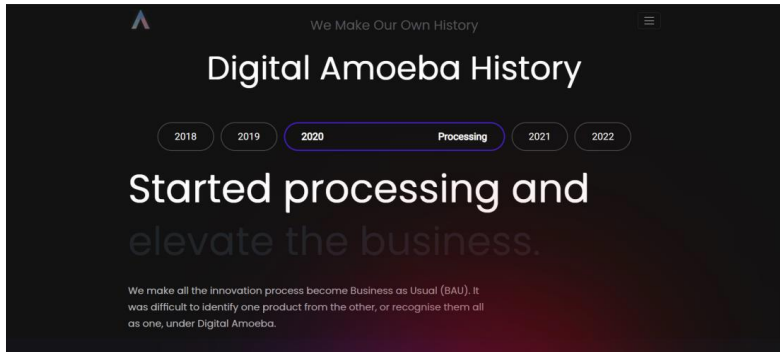
6.7.2. History



Gambar 47 Tampilan: Landing Page Digital Amoeba History 2018



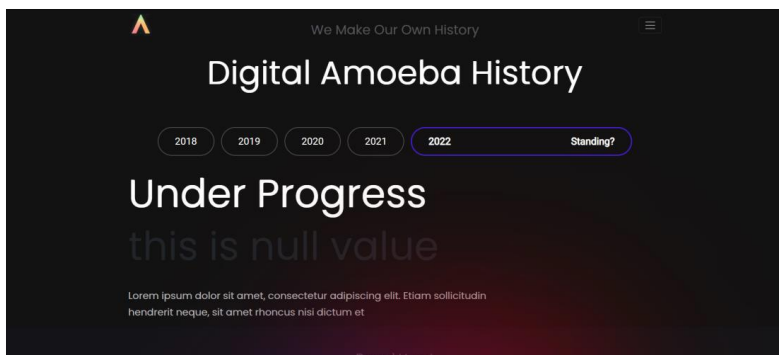
Gambar 48 Tampilan: Landing Page Digital Amoeba History 2019



Gambar 49 Tampilan: Landing Page Digital Amoeba History 2020



Gambar 50 Tampilan: Landing Page Digital Amoeba History 2021



Gambar 51 Tampilan: Landing Page Digital Amoeba History 2022

[Halaman ini sengaja dikosongkan]

BAB VII

KESIMPULAN DAN SARAN

7.1. Kesimpulan

Program Magang Bersertifikat Kampus Merdeka 4 yang diselenggarakan oleh Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi (Kemendikbudristek) secara keseluruhan telah berjalan dengan lancar walaupun terdapat keterlambatan dalam mobilisasi keberangkatan. Namun, selama pelaksanaan program segala sesuatunya berjalan dengan lancar dan terstruktur.

Mitra penulis yang mana ialah PT. Telekomunikasi Indonesia Tbk, terlebih khusus ialah Digital Amoeba, merupakan mitra yang menerapkan Agile dengan metode Scrum sebagai budaya kerjanya. Metode tersebut membantu penulis dalam pengembangan perangkat lunak modern. Penulis merasa mitra sudah sangat baik dalam mengondisikan peserta magang dan memiliki lingkungan kerja yang sehat.

Penulis yang selama kurang lebih empat (4) bulan melakukan magang di Digital Amoeba telah melakukan *bug fixing* dan mengembangkan dua platform website yaitu Eventeer versi kedua dan Landing Page Digital Amoeba. Masih ada beberapa fitur yang belum dapat diselesaikan, tetapi secara umum platform telah dikembangkan dengan baik.

7.2. Saran

1. Bagi pihak penyelenggara Magang dan Studi Independen Bersertifikat periode 4:
 - a. Lebih mendisiplinkan proses mobilisasi keberangkatan dengan cara memberi wewenang bagi tiap mitra secara mandiri untuk memesan tiket mobilisasi keberangkatan demi efisiensi.
2. Bagi PT. Telekomunikasi Indonesia Tbk (Digital Amoeba):
 - a. Memberikan fasilitas kantor yang lebih aman untuk kegiatan magang agar peserta lebih nyaman dalam bekerja.
3. Saran bagi penulis:
 - a. Lebih disiplin dalam memajemen waktu.
 - b. Memperluas pengetahuan mengenai Frontend dengan kerangka kerja ReactJS, NextJS, dan Bootstrap.
4. Saran bagi sistem yang dikembangkan:
 - a. Melakukan penyesuaian tampilan pada berbagai ukuran layar (*responsiveness*) untuk beberapa halaman selain Landing Page dan Login.
 - b. Melengkapi beberapa fitur lain yang belum dikerjakan seperti 'Media Learning' dan 'Submission' bagi peserta magang periode selanjutnya.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- Al-Saqqa, S., Sawalha, S. and Abdelnabi, H. (2020) 'Agile software development: Methodologies and trends', *International Journal of Interactive Mobile Technologies*. doi: 10.3991/ijim.v14i11.13269.
- Masood, Z., Hoda, R. and Blincoe, K. (2022) 'Real World Scrum A Grounded Theory of Variations in Practice', *IEEE Transactions on Software Engineering*. doi: 10.1109/TSE.2020.3025317.
- Mildawati, T. (2016) 'TEKNOLOGI INFORMASI DAN PERKEMBANGANNYA DI INDONESIA', *EKUITAS (Jurnal Ekonomi dan Keuangan)*. doi: 10.24034/j25485024.y2000.v4.i2.1904.
- Mufti Prasetyo, S. et al. (2022) 'Pembahasan Mengenai Front-End Web Developer dalam Ruang Lingkup Web Development', *Jurnal Multidisiplin Ilmu*.
- Pengembang Web Digital Amoeba. "Home - Digital Amoeba" <https://digitalamoeba.id/>. Diakses pada Kamis 22 Juni 2023.
- A. Faradilla. (2022) 'Apa Itu HTML? Fungsi dan Cara Kerja HTML', *Hostinger.Co.Id*.
- Durganath, Y. and Edreena, P. (2016) 'Role of bootstrap in web page creation', *International Journal of Pharmacy and Technology*.
- Fariz, M., Lazuardy, S. and Anggraini, D. (2022) 'Modern Front End Web Architectures with React.Js and Next.Js', *International Research Journal of Advanced Engineering and Science*.
- Khairul Kawistara, J. and Hidayatullah, P. (2020) *Pemrograman WEB Edisi Revisi, Informatika*.
- Rawat, P. and Mahajan, A. N. (2020) 'ReactJS: A Modern Web Development Framework', *International Journal of Innovative Science and Research Technology*.
- Reactor, H. (2021) 'What is JavaScript Used For ? What is JavaScript ? What is JavaScript used for ?', *Hackreactor*.

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS

Nama : Monica Narda Davita
Tempat, Tanggal Lahir : Surabaya, 19 Oktober 2001
Jenis Kelamin : Perempuan
Telepon : +62895342292555
Email : monicadavita19@gmail.com

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC , ITS
Angkatan : 2020
Semester : 6 (Enam)

