

**PENGEMBANGAN APLIKASI MEDIA SOSIAL SAMAKAKI
PADA PT QATROS TEKNOLOGI NUSANTARA
MENGUNAKAN RUBY ON RAILS**



Disusun Oleh:

N a m a : Jasmine Erina Firdaus
NIM : 19523095

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2023

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PENGEMBANGAN APLIKASI MEDIA SOSIAL SAMAKAKI
PADA PT QATROS TEKNOLOGI NUSANTARA
MENGUNAKAN RUBY ON RAILS**

TUGAS AKHIR JALUR MAGANG



الجمهورية الإسلامية اندونيسية
Yogyakarta, 09 Oktober 2023

Pembimbing,

(Dr. Yudi Prayudi S.Si., M.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**PENGEMBANGAN APLIKASI MEDIA SOSIAL SAMAKAKI
PADA PT QATROS TEKNOLOGI NUSANTARA
MENGUNAKAN RUBY ON RAILS**

TUGAS AKHIR JALUR MAGANG

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 30 Oktober 2023

Tim Penguji

Dr. Yudi Prayudi, S. Si., M. Kom.

Anggota 1

Dr. Ahmad Luthfi, S. Kom, M. Kom.

Anggota 2

Ari Sujarwo, S. Kom., MIT. (HONS)

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia



(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Jasmine Erina Firdaus
NIM : 19523095

Tugas akhir dengan judul:

**PENGEMBANGAN APLIKASI MEDIA SOSIAL SAMAKAKI
PADA PT QATROS TEKNOLOGI NUSANTARA
MENGUNAKAN RUBY ON RAILS**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apa pun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 30 Oktober 2023



(Jasmine Erina Firdaus)

HALAMAN PERSEMBAHAN

Puji syukur saya panjatkan kepada Allah SWT atas segala nikmat dan kesempatan yang telah diberikan kepada saya dalam menyelesaikan tugas akhir saya yang berjudul “Pengembangan Aplikasi Media Sosial Pada PT Qatros Teknologi Nusantara Menggunakan Ruby on Rails”. Tugas akhir ini saya persembahkan kepada saya sendiri, dan keluarga serta kerabat dekat terkasih yang telah mendukung secara penuh serta membantu ketika saya mengalami kesulitan dalam pengerjaan tugas akhir ini.

HALAMAN MOTO

“Dan apabila hamba-hamba-Ku bertanya kepadamu (Muhammad) tentang Aku, maka sesungguhnya Aku dekat. Aku kabulkan permohonan orang yang berdoa apabila dia berdoa kepada-Ku. Hendaklah mereka itu memenuhi (perintah)-Ku dan beriman kepada-Ku, agar mereka memperoleh kebenaran.”

– Q.S. Al Baqarah: 186

KATA PENGANTAR

Assalamualaikum Warrahmatullahi Wabarakatuh,

Puji dan syukur penulis panjatkan kepada Allah SWT yang senantiasa melimpahkan Rahmat serta Hidayah-Nya sehingga penulis dapat menyelesaikan laporan tugas akhir dengan lancar. Serta sholawat dan salam tak lupa dipanjatkan kepada Nabi Muhammad SAW yang selalu menjadi suri teladan bagi umat-Nya.

Laporan ini disusun sebagai salah satu syarat dalam kelulusan di penjaluran Magang pada Program Studi Informatika – Program Sarjana dan menjadi bukti serta dokumentasi dari pelaksanaan magang yang telah dilakukan. Pada kesempatan ini penulis ingin mengucapkan terima kasih kepada pihak-pihak terkait di antaranya sebagai berikut:

1. Kepada Allah SWT yang senantiasa memberikan Rahmat, petunjuk, dan kekuatan dalam perjalanan penulisan skripsi ini.
2. Kedua orang tua saya, yang selalu memberi dukungan dari segi mental maupun finansial selama ini.
3. Andromeda Noval Riyanto selaku adik dari penulis, yang memberi dukungan dan menghibur penulis.
4. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Jurusan Informatika dan Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku Ketua Program Studi Informatika – Program Sarjana.
5. Bapak Dr. Yudi Prayudi S.Si., M.Kom. selaku dosen pembimbing tugas akhir yang senantiasa membantu dan memberikan arahan dalam pengerjaan tugas akhir selama saya menempuh penjaluran akhir.
6. Bapak dan Ibu dosen Program Studi Informatika yang telah memberikan ilmu yang bermanfaat dan membantu selama saya menempuh studi.
7. Seluruh civitas yang ada di Fakultas Teknologi Industri yang telah membantu saya dalam administrasi kelengkapan saya dalam mendapatkan beasiswa.
8. Bapak Lutvi Rosyady selaku CEO PT Qatros Teknologi Nusantara sekaligus mentor dan anggota tim SAMAKAKI selama magang yang mengajarkan saya banyak hal dalam dunia kerja.
9. Teman baik penulis yaitu Muhammad Riko Bediatra yang senantiasa menemani perjalanan saya serta memberikan motivasi penuh selama kegiatan perkuliahan dan magang dalam penjaluran ini.

10. Teman-teman yaitu “Jogjaa” (Tiara, Dio, Luthfi), Adis, Rania, Chika, Thalia, Sallu, Alafta, Reizi, Fardhan, Imam Sibaweh, Alfa, dan Ummi.

Laporan tugas akhir ini masih jauh dari kata sempurna sehingga penulis menerima segala bentuk saran yang nanti akan berguna untuk menyempurnakan laporan ini. Semoga laporan ini dapat bermanfaat bagi pembaca.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Yogyakarta, 19 Juli 2023



(Jasmine Erina Firdaus)

SARI

Penulis melakukan kegiatan magang sebagai *backend engineer* di PT Qatros Teknologi Nusantara atau sering disebut Qatros. Qatros sendiri adalah sebuah *startup* di bidang teknologi yang melayani jasa untuk keperluan produk digital seperti konsultasi, pengembangan, dan pemeliharaan. Salah satu aplikasi yang telah dibuat oleh Qatros adalah MovesGG.id. Kantor Qatros bertempat pada Jl. Sarirejo No. 2, Setan, Maguwoharjo, Depok Sleman, D.I Yogyakarta.

Proyek pembuatan aplikasi sosial media Samakaki dipilih untuk dikaji lebih lanjut pada laporan ini. Aplikasi ini didasari oleh minimnya sebuah media sosial yang aman digunakan untuk anak-anak. Hal ini menyebabkan kurangnya wadah bagi mereka untuk mengenal cara menggunakan internet secara bijak, baik, dan benar. Metode yang dipilih dalam pembuatan aplikasi ini adalah SCRUM. Metode ini dipilih karena dapat mengadaptasi perubahan yang terjadi selama proyek ini dikembangkan. Hasil akhir dari proyek ini adalah sebuah aplikasi media sosial dalam lingkup keluarga. Salah satu fitur dari aplikasi ini yaitu pengguna dapat membuat bagan silsilah keluarga di dalam aplikasi tersebut yang berfungsi sebagai batasan bagi konten yang akan tertampil pada aplikasi pengguna.

Dalam pelaksanaan proyek ini masih terdapat batasan dan kekurangan pada aplikasi Samakaki. Oleh karena itu untuk proyek selanjutnya diharapkan dapat merancang desain aplikasi agar terlihat lebih menarik bagi pengguna, melakukan analisis secara lebih mendalam dan detail mengenai fitur yang dibutuhkan serta aplikasi dapat dikomersialkan dan digunakan oleh masyarakat umum.

Kata kunci: Scrum, Backend, Ruby, Ruby on Rails Framework.

GLOSARIUM

Glosarium memuat daftar kata tertentu yang digunakan dalam laporan dan membutuhkan penjelasan, misalnya kata serapan yang belum lazim digunakan. Urutkan sesuai abjad. Contoh penulisannya seperti di bawah ini:

Backend	bagian dari sistem informasi yang bertugas untuk menyediakan kebutuhan yang tidak langsung berinteraksi dengan pengguna, yaitu data, logika aplikasi, dan server.
API	sebuah kumpulan kode yang membuat dua program atau lebih dapat berkomunikasi yang bertujuan untuk menghubungkan klien dengan server.
Scrum	metode pengembangan perangkat lunak yang menggunakan pendekatan agile.
Ruby	sebuah bahasa pemrograman yang bersifat open-source.
Ruby on Rails	salah satu framework yang menggunakan arsitektur MVC (Model View Controller) yang dikhususkan untuk bahasa pemrograman Ruby.
Media Sosial	sebuah platform luring yang memungkinkan pengguna untuk berinteraksi, berbagi konten, dan berkomunikasi dengan orang lain melalui teks, gambar, atau video.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI.....	ix
GLOSARIUM.....	x
DAFTAR ISI	xi
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Ruang Lingkup.....	3
1.3 Tujuan	4
1.4 Manfaat	4
1.5 Sistematika Penulisan	5
BAB II LANDASAN TEORI DAN TINJAUAN PUSTAKA	6
2.1 Backend.....	6
2.2 Rails Framework	7
2.3 PostgreSQL	8
2.4 REST API	8
2.5 <i>JavaScript Object Nation (JSON)</i>	9
2.6 SCRUM.....	10
2.7 Media Sosial.....	12
2.8 Tinjauan Pustaka	12
BAB III PELAKSANAAN MAGANG	15
3.1 Sprint Planning.....	15
3.2 Development	16
3.3 Hasil Proyek.....	28
BAB IV REFLEKSI PELAKSANAAN MAGANG	37
4.1 Relevansi Akademik	37
4.2 Pembelajaran Magang.....	39
4.3 Manfaat Magang	40
BAB V PENUTUP.....	42
5.1 Kesimpulan	42
5.2 Saran.....	42
DAFTAR PUSTAKA	43

DAFTAR TABEL

Tabel 2. 1 Tabel Tinjauan Pustaka.....	13
Tabel 3. 1 Daftar Backlog.....	15

DAFTAR GAMBAR

Gambar 1.1 Tampak depan bangunan PT Qatros Teknologi Nusantara pada Google Earth.....	2
Gambar 1.2 Lokasi PT Qatros Teknologi Nusantara pada Google Maps.....	2
Gambar 2.1 Diagram MVC (<i>Model View Controller</i>).....	6
Gambar 2.2 Perbandingan PostgreSQL dengan MySQL (Dwi Praba & Safitri, 2020).....	8
Gambar 2.3 Arsitektur REST <i>Web Service</i>	9
Gambar 2.4 Contoh penulisan dalam format JSON.....	9
Gambar 2.5 Alur kerja sebuah <i>sprint</i> dalam <i>scrum</i>	10
Gambar 3.1 Proses Scrum.....	15
Gambar 3.2 MVP Dari Mitra	17
Gambar 3.3 Alur Flowchart	19
Gambar 3.4 Rancangan Basis Data Aplikasi Samakaki	20
Gambar 3.5 API <i>login</i>	21
Gambar 3.6 Atribut data fitur <i>user</i>	21
Gambar 3.7 Fungsi <i>create</i> fitur <i>register user</i>	21
Gambar 3.8 Hash pada fitur user	22
Gambar 3.9 Atribut API <i>biodata_user</i>	23
Gambar 3.10 Fungsi <i>create</i> pada fitur <i>biodata_user</i>	23
Gambar 3.11 Respon API pada pengisian <i>biodata_users</i>	23
Gambar 3.12 Respon API ketika tidak berhasil	23
Gambar 3.13 Atribut pada fitur <i>post</i>	24
Gambar 3.14 Fungsi <i>create</i> pada fitur <i>post</i>	24
Gambar 3.15 Respon API pada fitur <i>post</i> ketika ukuran <i>content</i> tidak sesuai.....	24
Gambar 3.16 Respon API pada fitur <i>post</i> ketika berhasil.....	24
Gambar 3.17 Fungsi SMTP	25
Gambar 3.18 Fungsi <i>create</i> fitur <i>user_relations</i>	26
Gambar 3.19 Penamaan Untuk <i>Role Relation</i>	27
Gambar 3.20 Respons API ketika pengguna mengundang pengguna lainnya pada <i>family tree</i>	27
Gambar 3.21 Respons API ketika pengguna mendapatkan undangan <i>family tree</i>	27
Gambar 3.22 Respons API ketika pengguna yang diundang sudah konfirmasi	27
Gambar 3.23 Respons API jika kedua pengguna sudah saling terhubung di <i>family tree</i>	27
Gambar 3.24 Halaman Landing Page	28

Gambar 3.25 Halaman Privacy and Policy	29
Gambar 3.26 Halaman Privacy and Policy (2)	29
Gambar 3.27 Halaman Terms and Condition	29
Gambar 3.28 Halaman Terms and Condition (2).....	30
Gambar 3.29 Halaman Login.....	30
Gambar 3.30 Halaman Daftar Akun	31
Gambar 3.31 Halaman Lupa Kata Sandi	31
Gambar 3.32 Halaman Lupa Kata Sandi (2).....	32
Gambar 3.33 Halaman Lupa Kata Sandi (3).....	32
Gambar 3.34 Halaman Beranda	33
Gambar 3.35 Halaman Beranda (2)	33
Gambar 3.36 Halaman Profil	34
Gambar 3.37 Halaman Silsilah Keluarga.....	34
Gambar 3.38 Halaman Undang Keluarga.....	35
Gambar 3.39 Halaman Undang Keluarga (2)	35
Gambar 3.40 Halaman Event	36
Gambar 3.41 Halaman Pop Up <i>Create Event</i>	36
Gambar 3.42 Halaman Tentang	36

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan internet di era sekarang memberi kemudahan dalam mengakses informasi secara bebas. Di Indonesia sendiri perkembangan internet semakin meningkat dari tahun ke tahun, hal ini membuktikan bahwa internet memiliki peran yang penting dalam banyak aspek yang ada. Pada awalnya internet digunakan pada abad 19 untuk kepentingan militer, namun perkembangan dan kebutuhan akan teknologi terus menjadikan internet digunakan di berbagai banyak sektor seperti sosial, pendidikan, hingga ke sistem pemerintahan. Internet juga sudah banyak digunakan oleh berbagai kategori usia, mulai dari anak-anak sampai usia lanjut.

Perkembangan internet yang sangat cepat membuat permintaan dari berbagai sektor semakin meningkat. PT Qatros Teknologi Nusantara menyediakan jasa teknologi informasi yang dapat memenuhi permintaan tersebut, perusahaan ini merupakan sebuah perusahaan yang bergerak dalam bidang teknologi jasa teknologi informasi yang berkomitmen untuk membangun proses bisnis yang efektif dan efisien bagi klien dengan mengimplementasikan teknologi.

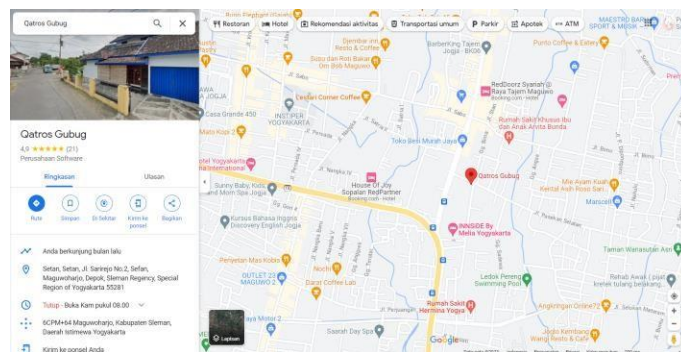
PT Qatros Teknologi Nusantara secara umum bertugas membangun kultur yang memfasilitasi karyawan dan penduduk lokal untuk berbagi ilmu serta pengalaman, berkontribusi pada pembangunan daerah terutama pada bidang teknologi dan memberdayakan masyarakat lokal dengan menyediakan informasi dan akses teknologi.

Peran utama dari PT Qatros Teknologi Nusantara yang terletak di kota Yogyakarta adalah menjadi perusahaan teknologi perusahaan teknologi dengan memiliki kualitas kelas dunia yang memfasilitasi pelayanan profesional dengan menjadikan daerah rural sebagai basis utama yang bertujuan untuk meletakkan penduduk lokal di satu lingkungan yang mendukung untuk tumbuh dengan penggunaan teknologi agar mereka dapat bekerja tanpa harus berpindah domisili.

PT Qatros Teknologi Nusantara berlokasi pada Jl. Sarirejo No. 2, Setan, Maguwoharjo, Depok Sleman, D.I Yogyakarta. Letak gedung ini berada di kawasan perumahan dusun Sarirejo. Apabila dilihat dari depan maka tampak depan bangunan akan seperti Gambar 1.1 dan Gambar 1.2.



Gambar 1.1 Tampak depan bangunan PT Qatros Teknologi Nusantara pada Google Earth



Gambar 1.2 Lokasi PT Qatros Teknologi Nusantara pada Google Maps

Penulis melaksanakan magang di PT Qatros Teknologi Nusantara selama lima bulan terhitung dari bulan Agustus sampai bulan Januari sebagai *Backend Engineer*. Penulis mendapatkan proyek bernama tim SAMAKAKI. Pada tim ini beranggotakan dua belas orang dan bertugas untuk mengembangkan sebuah proyek aplikasi media sosial berbasis web. *Backend* adalah tempat di mana proses pada suatu sistem informasi atau aplikasi berjalan, data dapat ditambahkan, diubah maupun dihapus. *Backend* mengurus segala jenis proses yang tidak berhubungan langsung dengan pengguna seperti *server* dan basis data (Pangestika & Dirgahayu, 2021).

Proyek dikerjakan dan dijalankan menggunakan metode pengembangan *scrum*. *Scrum* adalah salah satu metode rekayasa *software* dengan menggunakan prinsip *agile* yang bertumpu pada kekuatan kolaborasi tim, *incremental product* dan proses iterasi untuk mewujudkan hasil akhir. Menurut Schwaber & Sutherland *scrum* adalah sebuah kerangka kerja yang dapat mengatasi suatu masalah kompleks yang selalu berubah dan juga dinilai dapat memberikan

kualitas produk sesuai dengan keinginan pengguna secara kreatif dan produktif (Schwaber & Sutherland, 2020).

Internet memberikan dampak positif untuk kehidupan sehari-hari, akan tetapi internet juga bisa memberikan ancaman yang luar biasa baik itu untuk keluarga ataupun anak-anak (Nainggolan et al., 2022). Salah satu pengaruh negatif dari penggunaan media sosial adalah apa pun yang informan unggah bisa dapat dengan mudah dilihat oleh orang lain sehingga hal tersebut dapat membocorkan masalah-masalah pribadi dari para pengguna media sosial (Rafiq, 2020). Oleh karena itu, setiap pengguna tidak bisa membatasi konten apa saja yang telah diunggah oleh pengguna lainnya. Hal ini juga berpengaruh terhadap anak-anak yang sedang menggunakan media sosial, dikarenakan kurangnya pengawasan dan pemahaman tentang penggunaan media sosial secara baik dan benar yang diberikan oleh orang tua kepada anak. Sehingga saat ini anak bisa dengan bebas mengakses media sosial tanpa ada batasan apa pun.

Berdasarkan permasalahan di atas, proyek pengembangan aplikasi media sosial berbasis web yang bertujuan untuk menciptakan sebuah media sosial ramah anak menjadi topik tugas akhir karena menurut penulis fungsi dari media sosial sendiri adalah menjadi sebuah alat yang memudahkan dalam menyebarkan suatu berita ke massa. Namun, dengan adanya masalah pada penggunaan media sosial bagi anak menjadi sebuah tantangan baru untuk menciptakan sebuah media sosial dengan lingkup yang lebih kecil namun kegunaan dari media sosial itu sendiri masih bisa dirasakan oleh pengguna.

Oleh karena itu, media sosial ini memiliki sebuah fitur bernama *family tree* yang berfungsi untuk mengelompokkan sebuah grup dan grup ini akan terdiri dari anggota keluarga. Grup ini berperan dalam mengisi konten yang ada pada fitur *timeline*. Sehingga hanya anggota yang berada pada grup tersebut yang dapat melihat unggahan. Pengguna juga dapat mengatur siapa saja anggota dari grup tersebut karena anggota pada grup tersebut hanya bisa diundang agar bisa masuk ke dalam grup tersebut.

Selain itu, proyek ini sekaligus menjadi salah satu proyek yang di dalamnya penulis berkontribusi penuh dalam pembuatannya. Setelah mengerjakan proyek ini, penulis memahami apa saja dasar-dasar yang diperlukan dalam pengembangan aplikasi berbasis web dalam lingkup *backend*.

1.2 Ruang Lingkup

Dalam pengembangan web aplikasi media sosial *family tree*, menggunakan metode *scrum* untuk mencapai hasil yang tepat dan efektif. Proyek ini memiliki dua belas anggota tim

dalam pengembangan, salah satunya adalah penulis. Masing-masing peran dari dua belas orang tersebut ada dua *frontend engineer*, dua *backend engineer*, dua *uix*, empat *mobile engineer* dan dua *quality assurance engineer*. Proyek ini menghabiskan sebanyak empat *sprint* dengan jumlah sepuluh hari dalam satu *sprint*, sehingga total pengerjaan proyek kurang lebih dua bulan mulai dari bulan November hingga bulan Desember. Kontribusi yang dikerjakan penulis pada proyek ini dapat disimak sebagai berikut:

- a. Analisa MVP dan Penentuan Prioritas
- b. Pembuatan Database
- c. Fitur User
- d. Fitur Biodata User
- e. Fitur Post
- f. SMTP Gmail
- g. Fitur Family Tree

1.3 Tujuan

Tujuan dari proyek pengembangan web aplikasi media sosial *family tree* adalah sebagai sebuah wadah bagi anak-anak untuk menggunakan media sosial. Dengan didukung fitur *family tree* konten yang disajikan dapat dibatasi. Hal ini dikarenakan hanya lingkup keluarga yang telah diundang saja yang dapat melihat konten yang diunggah.

1.4 Manfaat

Manfaat yang diperoleh dari pengembangan aplikasi web media sosial *family tree* antara lain:

1. Memudahkan pengguna dalam melihat dan mengetahui siapa saja anggota keluarga besar.
2. Menjaga keamanan pengguna di bawah umur dikarenakan media sosial tidak memiliki iklan atau konten dewasa dengan penggunaan fitur *private* pada *post*.
3. Pengguna dapat berbagi kabar seputar kehidupan kepada seluruh anggota keluarga dengan lebih mudah.

1.5 Sistematika Penulisan

Sistematika penulisan ini menjelaskan secara keseluruhan dari setiap materi yang terdapat dalam laporan ini. Adapun susunan sistematika penulisan laporan adalah sebagai berikut:

a. BAB I: Pendahuluan

Bab ini membahas terkait gambaran umum proyek yang dikerjakan, tujuan dan manfaat terkait proyek yang telah dikerjakan oleh penulis.

b. BAB II: Landasan Teori dan Tinjauan Pustaka

Bab ini membahas teori-teori dasar yang mendukung topik utama yang diangkat pada Tugas Akhir ini sebagai landasan pengembangan aplikasi web media sosial *family tree*.

c. BAB III: Pelaksanaan Magang

Bab ini menggambarkan kegiatan yang berisi implementasi dari proses kegiatan magang.

d. BAB IV: Refleksi Pelaksanaan Magang

Bab ini menjelaskan refleksi dan hasil yang telah diperoleh selama pelaksanaan magang pada proyek pengembangan aplikasi web media sosial *family tree*.

e. BAB V: Penutup

Bab ini berisi kesimpulan yang didapatkan dari bab-bab yang telah dipaparkan sebelumnya dan terdapat sara dari penulis kepada pembaca yang akan melakukan penjaluran tugas akhir dengan jalur magang.

BAB II

LANDASAN TEORI DAN TINJAUAN PUSTAKA

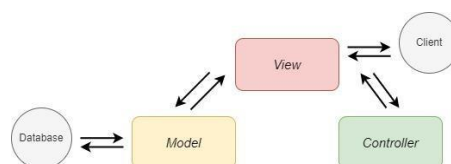
2.1 Backend

Terdapat beberapa *role* dalam suatu pengembangan sebuah sistem informasi atau aplikasi, salah satu *role* tersebut adalah *backend engineer*. Apabila dilihat dari sisi *backend* adalah tempat di mana proses suatu sistem informasi berjalan yang di dalamnya terdapat proses sebuah data ditambahkan, diubah ataupun dihapus. Oleh karena itu seorang *backend engineer* adalah orang yang pekerjaannya berfokus terhadap keamanan, desain sistem dan manajemen data pada sebuah sistem informasi. Biasanya seorang *backend engineer* tidak langsung berinteraksi kepada *user* dan tampilan pada suatu aplikasi, salah satunya adalah *database* dan *server*.

Seorang *backend engineer* dalam mengembangkan aplikasi kebanyakan menggunakan model arsitektur MVC. MVC terdiri dari tiga bagian yang saling berkaitan yaitu *Model*, *View* dan *Controller*. Yang pertama adalah *Model*, yang memiliki fungsi untuk mengelola data pada aplikasi, menerima respons yang diberikan, dan mengolah instruksi yang berkaitan dengan basis data (CRUD Data). CRUD yang dimaksud yaitu:

- a. CREATE untuk Pembuatan data.
- b. READ untuk Membaca.
- c. UPDATE untuk Memperbarui.
- d. DELETE untuk Menghapus data.

View biasanya berupa antarmuka yang melakukan interaksi secara langsung dengan pengguna yang menerjemahkan informasi dari *model* ke dalam bentuk yang sesuai untuk berinteraksi dengan *user*. *Controller* mengandung proses *input* dari pengguna, kemudian mengirim perintah kepada *model* dan *view* agar melakukan proses sesuai dengan masukan atau secara singkatnya adalah pemetaan aksi pengguna terhadap respons sistem (Oleh & Pangestika, 2021). Apabila divisualisasikan maka akan seperti pada Gambar 2.1.



Gambar 2.1 Diagram MVC (*Model View Controller*)

Dalam pengerjaannya, seorang *backend engineer* tak luput dari pemrograman atau program komputer pada suatu sistem informasi. Pada dasarnya komputer membutuhkan keberadaan program agar bisa menjalankan fungsinya sebagai komputer, hal ini dilakukan dengan mengeksekusi serangkaian instruksi program pada prosesor. Sebuah program biasanya memiliki suatu model pengeksekusian tertentu agar dapat dieksekusi oleh komputer secara langsung, salah satu bentuk model adalah kode sumber yang berisi format kode yang memungkinkan *programmer* menganalisis dan melakukan penelaahan algoritma pada suatu program kemudian kode sumber nantinya akan dikompilasi oleh utilitas bahasa pemrograman tertentu sehingga membentuk sebuah program.

Untuk pemrograman sendiri ada banyak sekali jenisnya, yaitu Pemrograman Sistem Operasi, Pemrograman Aplikasi, Pemrograman Web dan Pemrograman Game. Pada proyek yang dikerjakan penulis menggunakan Pemrograman Aplikasi, pada pemrograman ini dapat menggunakan berbagai macam bahasa pemrograman salah satunya Java, C++/C#, Python, Go dan Ruby. Bahasa pemrograman adalah sebuah instruksi standar untuk memerintah komputer agar menjalankan fungsi tertentu dan merupakan suatu himpunan dari aturan *syntax* dan semantik yang dipakai untuk mendefinisikan program komputer (Saragih, 2018).

2.2 Rails Framework

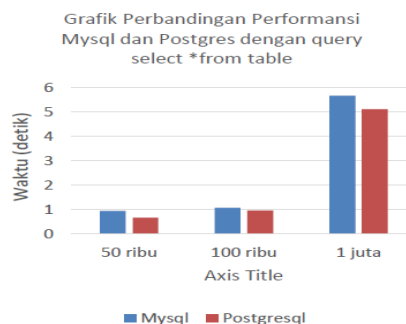
Web framework adalah kerangka kerja perangkat lunak yang dirancang untuk mendukung pengembangan aplikasi web termasuk *web service*, *web resources* dan *web API*. *Web framework* menyediakan cara standar untuk membangun dan mengembangkan aplikasi web dan membuat pekerjaan pengembang menjadi lebih mudah (Saputra & Fathoni Aji, 2018). *Framework* sendiri ada banyak jenisnya tergantung dengan jenis bahasa pemrograman yang digunakan.

Penulis menggunakan Ruby on Rails sebagai *framework*. *Framework* ini ditulis dalam bahasa pemrograman Ruby dan memiliki kelebihan yaitu ramah untuk pemula karena kesederhanaannya pada *code syntax*, lebih aman karena menyediakan fitur keamanan dari sisi klien dan server dengan enkripsi kata sandi dan kartu kredit serta mengamankan aplikasi web dari injeksi SQL dan serangan XSS, lebih sedikit kode dikarenakan lebih banyak menggunakan bahasa manusia, dan mendukung *agile project development*.

2.3 PostgreSQL

Pada pembuatan sebuah sistem informasi pasti membutuhkan sebuah basis data atau *database*, basis data terdiri dari dua kata yaitu Basis dan Data. Basis dapat diartikan sebagai kumpulan, markas, atau gudang. Sedangkan data adalah fakta, keterangan, atau peristiwa yang didapatkan melalui proses pengamatan atau pencarian pada sumber-sumber tertentu (Waruwu, 2019). Dari pengertian tersebut, basis data adalah kumpulan data-data yang saling berkaitan antar satu dengan yang lain serta dapat disimpan, diubah, dimanipulasi atau dipanggil kembali oleh pengguna sesuai dengan kebutuhan.

Salah satu di antara jenis basis data dalam pemrograman adalah PostgreSQL. PostgreSQL adalah sebuah *database* yang bersifat *opensource* dengan kemampuan yang bagus dibandingkan dengan *database-database* yang lain (Dwi Praba STMIK Nusa Mandir Jakarta, 2018). *Database* ini memiliki keunggulan dalam kelengkapan fitur yang sangat mendukung *database* sebuah aplikasi dengan skala menengah ataupun skala besar. Pada Gambar 2.2 merupakan visualisasi perbandingan data performansi dari PostgreSQL dan MySQL yang menunjukkan PostgreSQL lebih unggul daripada MySQL.

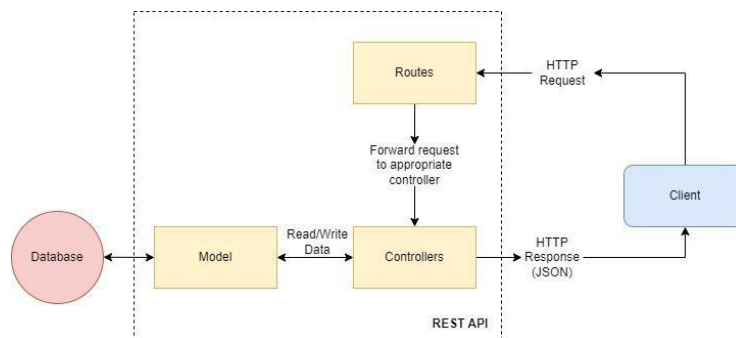


Gambar 2.2 Perbandingan PostgreSQL dengan MySQL (Dwi Praba & Safitri, 2020)

2.4 REST API

Application Programming Interface (API) adalah sebuah konsep fungsi antarmuka pemrograman aplikasi yang menjadi salah satu cara agar sebuah aplikasi dapat digunakan dan dimanfaatkan oleh pihak lain tanpa mengubah struktur kode utama pada *database* sistem (Muri et al., 2019). Dalam API sendiri memiliki beberapa jenis gaya arsitektur salah satunya adalah REST API (*REpresentational State Transfer*) yang dikenalkan pada tahun 2000 oleh Roy Fielding, istilah ini dikenal sebagai sebuah *web service* yang diimplementasikan dengan menggunakan HTTP (*Hypertext Transfer Protocol*) dengan menggunakan prinsip REST. REST sendiri merupakan arsitektur *client server* di mana *client* mengirimkan permintaan ke

server, kemudian permintaan diproses dan akan mengembalikan tanggapan. Berikut adalah visualisasi arsitektur REST API pada Gambar 2.3.



Gambar 2.3 Arsitektur REST *Web Service*

Pada REST API memiliki metode dan tipe *request* yang tersedia untuk melakukan yang menghubungkan sebuah data, antara lain GET untuk membaca sebuah data, POST untuk menambahkan sebuah data, PUT untuk mengubah *field* dalam data, DELETE untuk menghapus sebuah data.

2.5 JavaScript Object Nation (JSON)

JavaScript Object Nation (JSON) merupakan sebuah format yang ringan untuk memasukkan data ke dalam sebuah *variable* JSON sendiri merupakan bagian dari JavaScript, memiliki format teks yang sepenuhnya independen tetapi menggunakan konversi yang familier dengan bahasa pemrograman dari keluarga-C, termasuk C/++/#, Java, JavaScript, Perl, Python, Ruby, dan sebagainya (Wisnu et al., 2015).

JSON adalah sebuah struktur data berbentuk universal yang bisa digunakan dalam berbagai bahasa pemrograman. JSON memiliki dua struktur utama yaitu:

- Kumpulan pasangan atau nilai. Pada beberapa bahasa struktur ini dinyatakan sebagai *object*, *record*, *structure*, *hash table*, *dictionary*, *keyed list*, atau *associative array*.
- Daftar nilai terurutkan (*an ordered list of values*). Biasa disebut dengan *array*, *vector*, *list*, atau *sequence*.

Berikut bentuk sebuah data dari JSON pada Gambar 2.4.

```

{"id": 1,
 "name": "Kopi",
 "stock": "10",
 "type": "minuman",
 "price": "10.000"}
  
```

Gambar 2.4 Contoh penulisan dalam format JSON

2.6 SCRUM

SCRUM adalah salah satu metode rekayasa *software* yang diciptakan oleh Jeff Sutherland dan Ken Schwaber untuk mempercepat perilisan suatu produk siap pakai kepada pengguna dengan kualitas produk yang lebih tinggi sesuai dengan prinsip *agile*. Metode *agile* biasa disebut juga sebagai *sprint* yang di dalamnya pengguna *scrum* harus memecahkan suatu tugas pada suatu proyek menjadi lebih kecil yang akan mempermudah anggota tim mengetahui apa dan kapan harus mengerjakan tugas-tugas yang tersedia pada suatu *sprint*.

Penggunaan metode ini sangat memudahkan suatu tim untuk mengetahui langkah-langkah apa saja yang harus dilakukan dalam suatu proyek atau biasa disebut *workflow*. Dalam sebuah tim *scrum* memiliki beberapa peran yang terdiri dari *Scrum Master*, *Product Owner*, dan *Developers*. Pada tim ini diharuskan menyelesaikan pekerjaan penting dalam *sprint* dan bertanggung jawab atas semua aktivitas mengenai produk seperti kolaborasi dengan *stakeholders*, verifikasi, pemeliharaan, pengoperasian, eksperimen, penelitian dan pengembangan, dan hal lain yang mungkin diperlukan (Schwaber & Sutherland, 2020). *Sprint* memiliki tahapan, yaitu:

- A. *Sprint Planning*
- B. *Daily Scrum*
- C. *Sprint Review*
- D. *Sprint Retrospective*

Inti dari seluruh pelaksanaan dari SCRUM adalah *Sprint*, dengan durasi tetap selama satu bulan agar memiliki daur hidup yang konsisten dalam pengerjaannya sehingga saat *sprint* sebelumnya akan bisa dimulai kembali. Alur kerja *sprint* dapat dilihat pada Gambar 2.5 dengan tahapan-tahapan sebagai berikut



Gambar 2.5 Alur kerja sebuah *sprint* dalam *scrum*

(Sumber: <https://rasis2.github.io/inneonusa/>)

a. Product Backlog

Product Backlog merupakan sebuah daftar yang tersusun tentang apa saja yang dibutuhkan untuk meningkatkan hasil produk, dalam hal ini memiliki sebuah Product Backlog Item yang berbentuk sebuah tindakan memecah dan mendefinisikan sebuah item menjadi lebih kecil dan lebih akurat.

b. Sprint Planning

Sprint Planning merupakan sebuah tahapan untuk merencanakan pekerjaan yang akan dilakukan lebih dulu. Pada tahapan ini dilakukan dengan melakukan kolaborasi bersama seluruh anggota tim SCRUM. Durasi Sprint Planning yang dikerjakan berbeda antar satu dengan yang lain, tergantung dengan tingkat kesusahan dan ukuran masalah yang akan diselesaikan.

c. Sprint Backlog

Sprint Backlog merupakan perkiraan awal dari *development team* yang berupa sebuah daftar rincian dari sebuah *product backlog* yang akan ditambahkan ke *sprint planning* guna mencapai *sprint goal*. Untuk mengetahui apakah terjadi peningkatan secara simultan, sebuah *sprint backlog* harus memiliki satu peningkatan proses dengan prioritas tinggi yang diambil dari hasil pertemuan retrospective sebelumnya.

d. Daily Meeting

Daily Meeting adalah pertemuan singkat setiap hari, untuk membahas dan mengoptimalkan performansi dan kolaborasi sebuah tim dengan mengevaluasi pekerjaan yang telah dilakukan pada hari itu juga untuk mengetahui apa saja kesulitan yang dialami sejak pertemuan terakhir.

e. Sprint Review

Pada tahap ini tim harus melakukan tinjauan pada sebuah sprint mengenai pengerjaan atau fitur yang telah dilakukan pada saat itu dalam konteks upaya pengembangan secara keseluruhan. Tujuan Sprint Review adalah untuk memeriksa dan mengadaptasi produk yang sedang dibangun. Yang terpenting dalam kegiatan ini adalah percakapan yang terjadi di antara para pesertanya, meliputi tim *scrum*, *stakeholders*, sponsor, dan pelanggan (Septian et al., 2020).

f. Sprint Retrospective

Pada tahap ini tim mengkaji bagaimana perkembangan dan berjalannya sebuah sprint. Tim *scrum* mengidentifikasi perubahan yang terjadi untuk meningkatkan efektivitas sebuah sprint, kemudian tahap ini dilakukan guna untuk menutup sebuah sprint. Batas waktu untuk melakukan Sprint Retrospective maksimum tiga jam untuk jangka pengerjaan sprint selama satu bulan, jika jangkanya lebih pendek maka biasanya lebih pendek.

2.7 Media Sosial

Media sosial menurut Danah Boyd adalah sebuah kumpulan perangkat lunak yang memungkinkan individu maupun komunitas untuk berkumpul, berbagi, berkomunikasi, dan dalam kasus tertentu saling berkolaborasi atau bermain (Setiadi, 2016). Perkembangan media sosial di Indonesia merupakan fenomena yang tidak dapat diabaikan dalam perkembangan teknologi informasi dan komunikasi di era modern ini. Media sosial telah mengubah secara signifikan cara kita berinteraksi, berbagi informasi, dan menjalin hubungan dengan orang lain, baik di tingkat pribadi maupun profesional. Dalam beberapa tahun terakhir, media sosial telah menjadi bagian integral dari kehidupan sehari-hari masyarakat Indonesia (Putri & Romli, 2021).

Penggunaan media sosial memiliki dampak yang baik, namun jika penggunaan media sosial dilakukan secara tidak bijak maka akan menimbulkan dampak yang berbahaya bagi anak-anak yang menggunakan media sosial. Dampak negatif yang ditimbulkan dari penggunaan media sosial dalam kehidupan sehari-hari bagi remaja dan anak-anak meliputi *cyber bullying*, memicu kejahatan, pornografi, komunikasi buruk, ancaman ujaran kebencian, perkembangan emosi, perkembangan fisik, dan mengumbar rahasia (Sugeng Cahyono, 2018). Salah satu hal yang disebabkan oleh dampak tersebut akan menimbulkan perpeloncoan di kalangan pelajar dengan menggunakan media sosial atau yang diketahui sebagai fenomena *cyber bullying*. Ketika anak kecanduan menggunakan media sosial ia cenderung tidak pernah berhubungan dengan masyarakat sekitar sehingga ia akan mementingkan diri sendiri dan menunjukkan sifat tak acuh terhadap lingkungan sekitarnya.

2.8 Tinjauan Pustaka

Referensi yang digunakan pada penulisan tugas akhir ini mengacu terhadap beberapa penelitian terdahulu, seperti yang terdapat pada Tabel 2. 1

Tabel 2. 1 Tabel Tinjauan Pustaka

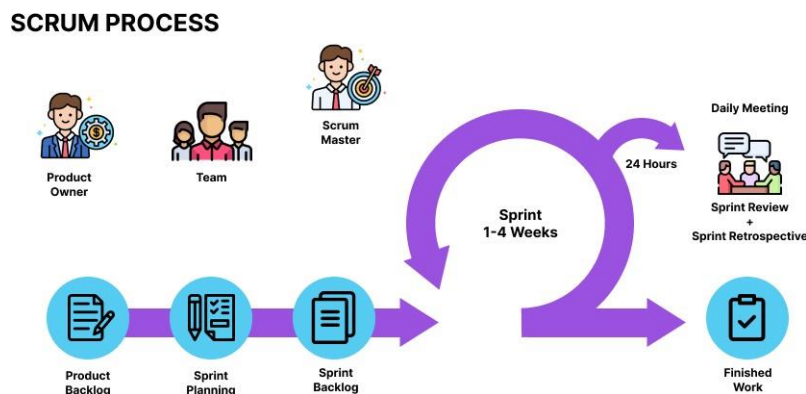
No.	Peneliti	Judul	Tahun	Tujuan	Metode	Hasil
1.	Saputra, D Fathoni Aji, R	Analisis Perbandingan Performa Web Service REST Menggunakan Framework Laravel, Django, dan Ruby on Rails untuk Akses Data Aplikasi Mobile	2018	Pengukuran performa pada <i>framework</i> dengan cara menganalisis penggunaan CPU dan RAM pada saat proses <i>request</i> berlangsung.	Pengukuran Performa	Ruby on Rails digunakan untuk membangun <i>web service</i> untuk mendukung aplikasi <i>mobile</i> yang dapat dihubungkan dengan jaringan lokal. Ruby on Rails juga digunakan dalam membangun proses <i>login</i> dan mengambil biodata mahasiswa, mengisi KRS, dan transkrip.
2.	Juliansyah Eka Putra, F Utomo, S D. Rachmanto, A	Aplikasi Quiz dengan Konsep <i>Gamification</i> Berbasis Web Ruby on Rails & React.JS	2021	Membuat aplikasi kuis yang dapat memberikan pertanyaan sesuai dengan audiens	Pembuatan aplikasi dengan menggunakan <i>framework</i> gamifikasi milik Marczewski	Ruby on Rails dapat digunakan untuk membangun sebuah aplikasi kuis yang dapat digunakan untuk berkompetisi bersama audiens lain dengan menyesuaikan kebutuhan audiens.
3.	Rizaldi, A Maria, E Wahyono, T	Analisis Penerapan Metode Scrum Pada Pengembangan Sistem Informasi	2022	Pembuatan sebuah sistem informasi web untuk memudahkan proses	Pembuatan aplikasi dengan metode Scrum.	Metode Scrum dapat digunakan untuk pembuatan sebuah aplikasi dan dapat

		Akuntansi Koperasi		pengendalian internal dalam pengawasan data anggota dan keuangan pada koperasi		mengatasi perubahan kebutuhan pada fase pembuatan sebuah sistem.
4.	Kurniawati, L Utama, A A	Dampak Penggunaan Media Sosial Youtube Terhadap Perilaku Negatif Anak	2022	Mengetahui dampak negatif media sosial terhadap perilaku anak	Analisa Wawancara dan Observasi Perilaku Anak	Salah satu hasil dari penelitian ini adalah media sosial dapat mempengaruhi pergaulan karena kurang kontrol dari Guru dan Siswa dapat mengakses semua situs-situs yang tidak baik secara bebas.
5.	Triastuti, E Adrianto, D Nurul, A	Kajian Dampak Penggunaan Media Sosial Bagi Anak dan Remaja	2017	Mengetahui dampak media sosial bagi anak dan remaja	Pengumpulan data dengan melakukan <i>Forum Group Discussion</i>	Anak-anak dan remaja melakukan hal yang disebut sebagai TMI atau <i>Too Much Information</i> pada saat mengakses media sosial hal itu secara tidak sadar mereka lakukan dan akan mengungkap terlalu banyak informasi tentang kehidupan pribadi mereka.

BAB III

PELAKSANAAN MAGANG

Pada pelaksanaan magang, penulis mengerjakan proyek pengembangan aplikasi berbasis web yang bernama Samakaki dengan menggunakan metode *scrum*. Metode ini merupakan salah satu dari bagian metodologi pengembangan perangkat lunak yang berprinsip *agile* dan mengutamakan kecepatan dan fleksibilitas dalam pengembangan rekayasa perangkat lunak yang kompleks. Tahapan dari proses pada proyek pengembangan aplikasi berbasis web SAMAKAKI dengan menggunakan metode Scrum meliputi *Sprint Planning*, *Sprint Backlog*, *Daily Meeting*, *Sprint Review*, *Sprint Retrospective*.



Gambar 3.1 Proses Scrum

4.1 Sprint Planning

Pada tahap ini, kegiatan yang dilakukan di pengerjaan proyek melibatkan seluruh anggota tim. Sprint Planning dipimpin oleh Scrum Master. Tujuan yang ingin dicapai pada tahap ini adalah identifikasi pada produk yang ada di *backlog*.

Tabel 3.1 Daftar Backlog

No.	Backlog
1.	Analisa MVP dan Penentuan Prioritas
2.	Pembuatan Database
3.	Fitur User
4.	Fitur Biodata User
5.	Fitur Post
6.	SMTP Gmail
7.	Fitur Family Tree

Proses pengembangan dari aplikasi ini memakan waktu selama tiga bulan. Durasi setiap satu *sprint* berkisar antara satu sampai dua minggu. Pada tahapan awal *sprint*, *backlog* yang

dikerjakan adalah Analisa MVP dan penentuan Prioritas. Hasil dari tahapan ini adalah sebuah rancangan awal dengan bentuk *flowchart* yang menjelaskan alur kerja dari sistem aplikasi Samakaki. Selanjutnya pada *sprint* kedua, yaitu Pembuatan *Database*. Dalam *sprint* ini menghasilkan sebuah rancangan basis data keseluruhan aplikasi dengan bentuk ERD. *Sprint* ketiga, yaitu Fitur User. Pada *sprint* ini menghasilkan sebuah fitur pada User, salah satunya adalah fitur registrasi dan *login*. *Sprint* keempat mengerjakan fitur Biodata User. Pada *sprint* ini menghasilkan sebuah fitur untuk profil pengguna. Kemudian *sprint* kelima yaitu fitur Post. Pada *sprint* ini menghasilkan fitur *posting* di mana pengguna bisa mengunggah sebuah unggahan. Lalu pada *sprint* keenam yaitu pembuatan fungsi SMTP dengan menggunakan Gmail. Hasil dari *sprint* ini adalah sistem bisa mengirim email *auto-bot* untuk verifikasi sebuah akun. Terakhir yaitu *sprint* ketujuh, fitur *family tree*. Pada *sprint* ini menghasilkan sebuah fitur utama dari aplikasi ini, ketika pengguna mengundang anggota lain untuk ditampilkan pada sebuah bagan pohon keluarga.

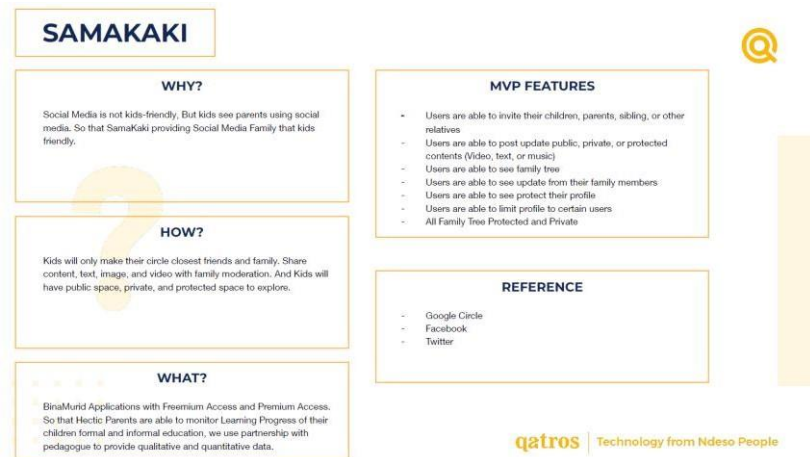
4.2 Development

Pada tahap ini disebut juga sebagai tahap pengembangan yang akan dikerjakan oleh tim pengembang sesuai dengan *sprint planning* yang telah dilakukan. Pengerjaan proyek ini harus sesuai dengan *product backlog* yang sudah ditentukan dan harus dalam pengawasan scrum master.

Untuk proyek ini, satu *sprint* biasanya dilakukan dalam jangka waktu lima belas hari kerja. Dalam satu *sprint* diadakan pertemuan untuk *sprint planning* yang bertujuan untuk penyusunan strategi dalam pengembangan aplikasi, *daily meeting* yang bertujuan mengetahui apa saja *blocker* yang dialami dari tiap anggota pengembang dari berbagai divisi sehingga bisa menyelesaikan *blocker* tersebut dengan cepat, dan *sprint review* sekaligus *retrospective* yang bertujuan untuk meninjau kembali apakah dalam satu *sprint* tersebut sudah sesuai dengan *sprint planning* sebelumnya dan *retrospective* bertujuan untuk meningkatkan kualitas dan efektivitas pada *sprint* selanjutnya.

3.2.1. Analisa MVP dan Penentuan Prioritas

Dalam hal ini penulis melakukan diskusi bersama dengan anggota tim mengenai MVP yang telah diberikan oleh mitra perusahaan, MVP yang diberikan memuat apa, mengapa dibuat, bagaimana cara menggunakan, fitur-fitur yang harus ada dalam pengembangan aplikasi, dan referensi dari aplikasi yang terkait dengan Samakaki.



Gambar 3.2 MVP Dari Mitra

Untuk menentukan prioritas tugas yang akan dikerjakan terlebih dahulu ketika sudah melakukan analisa MVP yang telah diberikan. Dari MVP yang didapat, penulis mengapa Samakaki dicetuskan yaitu diketahui media sosial sekarang sangat tidak aman untuk anak-anak tetapi anak-anak melihat para orang tua menggunakan media sosial oleh karena itu pada proyek kali ini Samakaki akan menyediakan media sosial untuk keluarga yang aman bagi anak-anak. Untuk cara kerja aplikasi ini, pengguna anak-anak akan membuat lingkup pertemanannya dengan keluarga saja, kemudian berbagi konten seperti teks, gambar, dan video dengan keluarganya.

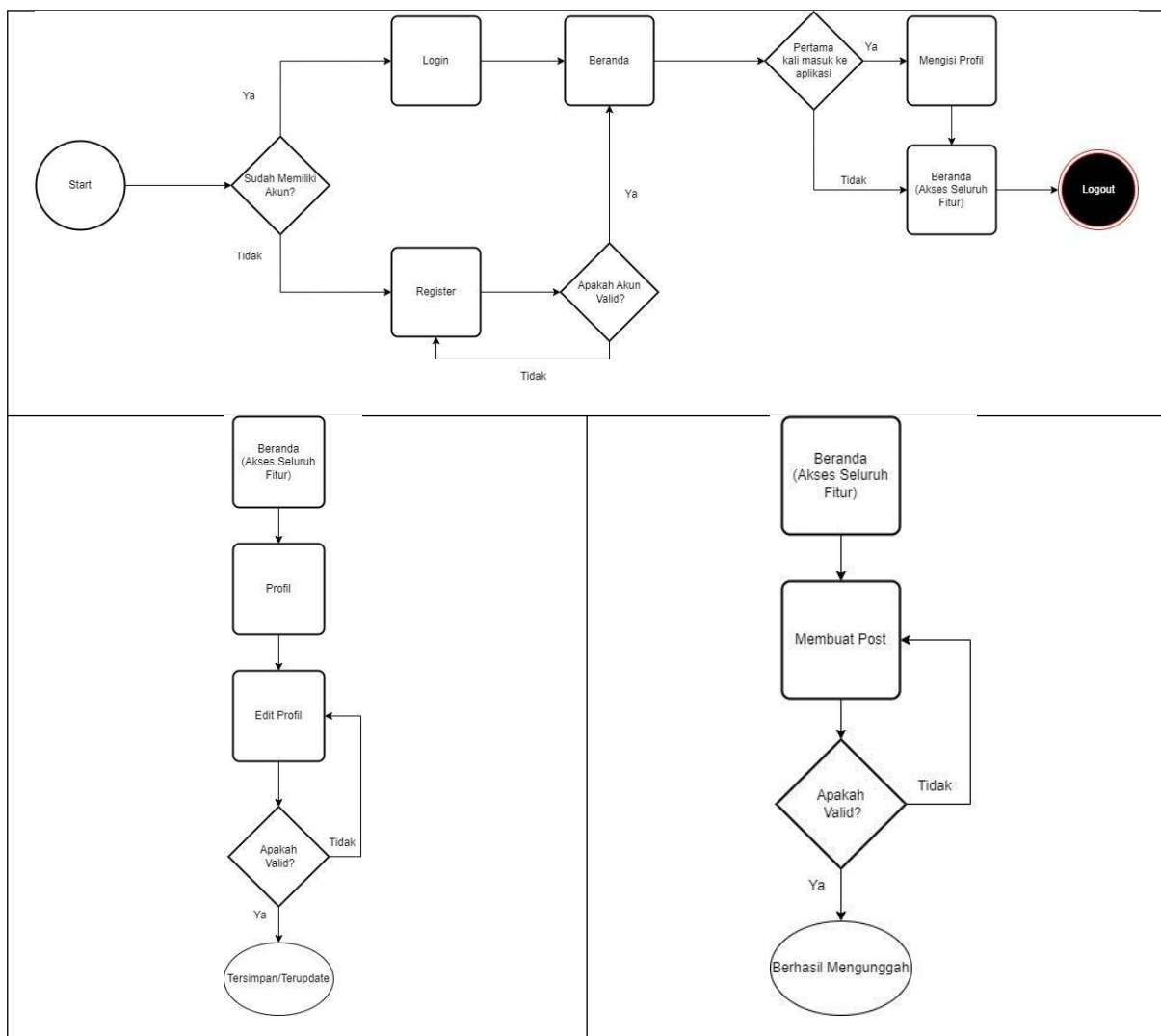
Untuk fitur yang ditulis pada MVP, Samakaki memiliki fitur di mana pengguna bisa mengundang anak, orang tua, saudara, atau hubungan relatif yang pengguna punya, mengunggah sebuah *post* ke *public*, *private*, atau *protected*, bisa melihat *family tree*, melihat pembaruan dari anggota *family tree*, bisa mengubah profil mereka ke *protected*, bisa memberikan *limit* kepada orang yang ingin melihat profil mereka, dan semua *family tree* bersifat *private* dan *protected*. Penulis sebagai *backend developer* memiliki tugas sebagai berikut:

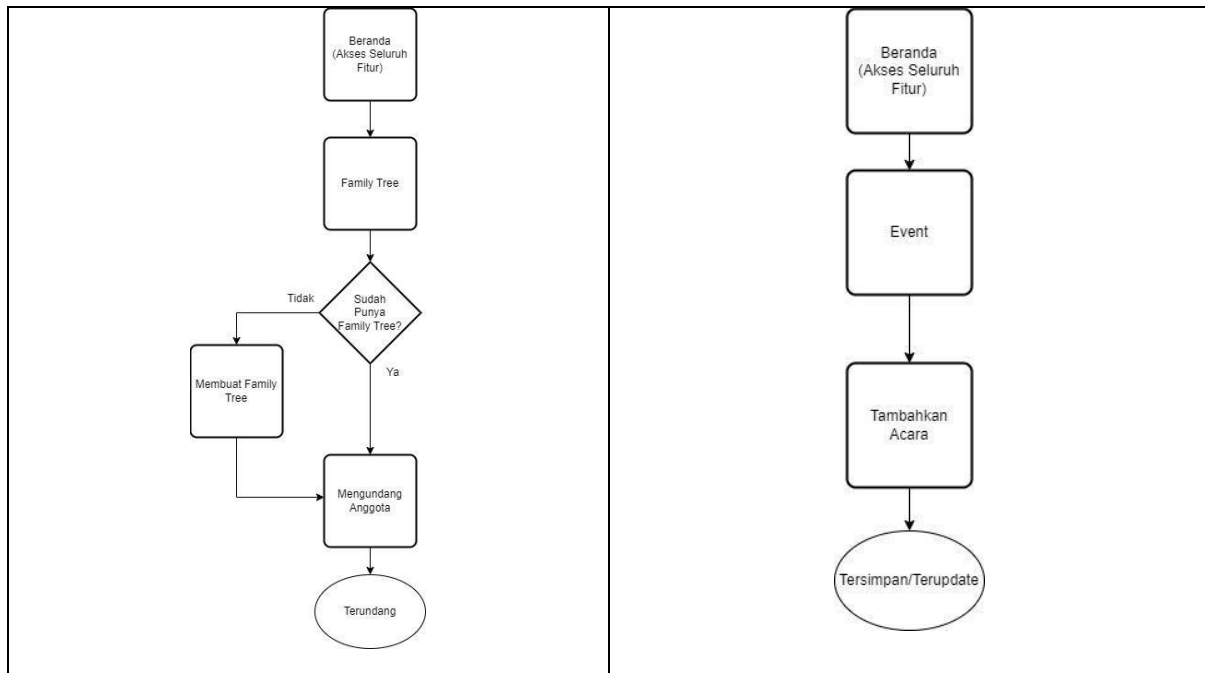
1. Membuat Rancangan *Database*
2. Membuat Logika Pemrograman
3. Membuat API
4. Membuat Dokumentasi API
5. *Monitoring Backend* pada Aplikasi

3.2.2. Analisa Alur Flowchart

Tahapan pertama dalam membangun sebuah aplikasi memerlukan sebuah flowchart sebagai pedoman utama (Fauzi, 2033). Flowchart adalah cara penulisan algoritma dengan menggunakan notasi grafis, flowchart divisualisasikan dalam bentuk gambar atau bagan yang menjelaskan alur dari sebuah program. Flowchart menjelaskan tentang alur kerja sistem yang akan dibangun dan keperluan data apa saja yang dibutuhkan. Fungsi dari *flowchart* adalah membantu untuk menganalisis untuk programmer dalam memecahkan sebuah masalah ketika ingin membangun suatu sistem.

Flowchart yang dibuat oleh *uix* menjadi acuan yang digunakan selama proses pengembangan aplikasi backend serta pembuatan basis data. Tampilan *flowchart* yang dibuat akan tampak seperti pada Gambar 3.3.





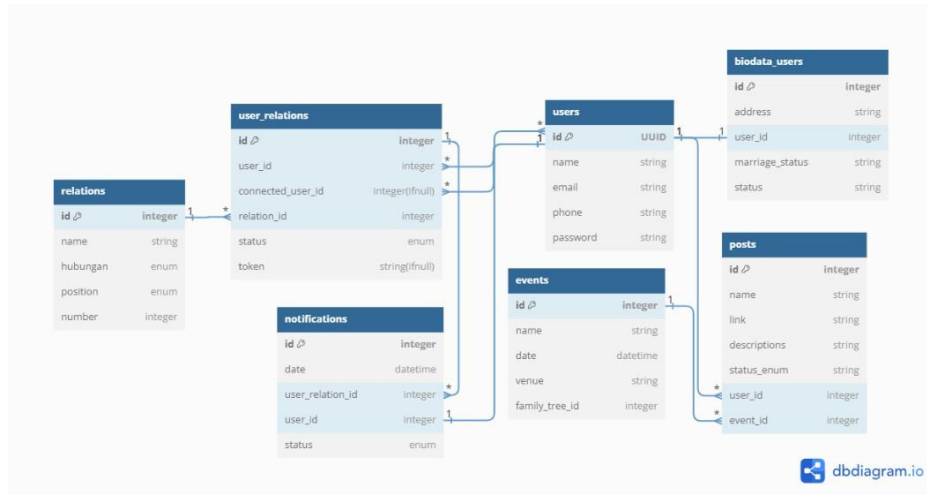
Gambar 3.3 Alur Flowchart

Seperti pada gambar di atas, alur kerja pada aplikasi ini dari sisi pengguna adalah dimulai dari kondisi apakah pengguna sudah memiliki akun atau belum. Jika belum, pengguna harus melakukan kegiatan Registrasi dengan mengisi data yang dibutuhkan seperti Nama, Email, dan Password. Ketika akun sudah valid maka akan diteruskan ke kondisi pengguna berada di Beranda. Ketika pengguna adalah pengguna baru, pengguna harus melengkapi Profil dengan mengisi data yang dibutuhkan salah satunya Foto Profil, Tanggal Lahir, dan Alamat. Setelah semua sudah dilengkapi, pengguna baru bisa menggunakan keseluruhan fitur pada aplikasi Samakaki.

Setelah pengguna mengisi semua validasi yang dibutuhkan sistem, pengguna baru bisa mengakses fitur lengkap pada aplikasi ini yaitu fitur Profil, Post, Family Tree, dan Event. Terdapat kondisi tertentu pada fitur Family Tree, yaitu ketika pengguna pertama dalam Family Tree tersebut harus membuat dan mengundang anggota lainnya terlebih dahulu.

3.2.3. Pembuatan Database

Dalam sebuah aplikasi pasti memerlukan *database* untuk mengetahui apa saja data yang dibutuhkan nanti saat pengembangan. Berikut pada Gambar 3.4 adalah tampilan rancangan *database* dalam aplikasi Samakaki.



Gambar 3.4 Rancangan Basis Data Aplikasi Samakaki

Penggunaan *database* kali ini menggunakan PostgreSQL karena memiliki keunggulan dalam penyimpanan media ke semua bahasa pemrograman berbasis laptop lainnya dan memiliki kecepatan yang mengalahkan basis data yang lain. Dalam perancangan *database* ini berfungsi sebagai pengoptimalan analisis data (Nabilla & Hidayat, 2021).

Hasil dari perancangan *database* membuahkan hasil dengan total sembilan tabel data yaitu tabel *users*, *user_relations*, *biodata_users*, *posts*, *relations*, *notifications*, *messages*, *events*, *rooms*. Dari seluruh tabel yang tersedia memiliki peran dan fungsi yang berbeda. Pada tabel *users* berfungsi untuk menyimpan data utama tentang pengguna, dan data ini akan diambil dan ditampilkan pada saat Login dan Registrasi. Tabel *relations* berfungsi untuk menyimpan data dari relasi tiap pengguna dengan pengguna lainnya, dan menampilkan data pada fitur Family Tree pada saat mengisi *form* untuk mengundang pengguna lainnya. Pada tabel *users_relations* adalah bentuk dari *join table* dari tabel *users* dan *relations*, tabel ini akan menyimpan data terkait pengguna yang terhubung dengan pengguna lainnya. Pada tabel *biodata_users* berfungsi untuk menyimpan data pengguna untuk tampilan pada fitur Profil, dan pada tabel ini harus diisi pengguna untuk validasi saat pengguna mengakses keseluruhan fitur aplikasi. Pada tabel *notifications* berfungsi sebagai notifikasi untuk pengguna saat undangan untuk Family Tree diterima atau terkirim, maka dari itu dalam tabel ini mengambil data dari *user_relation_id*. Pada tabel *post* berfungsi untuk menyimpan data pada sebuah unggahan yang diunggah dan mengambil data *user_id* untuk menampilkan informasi penulis yang mengunggah unggahan. Terdapat tabel *events* yang terdapat pada fitur Events dan berfungsi untuk menyimpan data terkait sebuah acara yang akan diselenggarakan dan akan ditampilkan pada fitur Events.

Dalam fitur User ini menggunakan sebuah *library* bernama Bcrypt yang berfungsi untuk melakukan *hashing* pada sebuah model. Pada Gambar 3.8 terdapat sebuah definisi metode *password* ketika pengguna *input* kata sandi baru maka akan dilakukan ‘password_digest’ yang nanti ketika menyimpan sebuah data dapat mencegah data dicuri dan data tersebut akan ter-enkripsi sebelum disimpan ke dalam basis data.

```
require 'bcrypt'

/*Example of Encrypt Password*/
/*
Bismillah 1243
-> '$2a$12$5kAbnTC906AcAP8ikzQAo.l4smk4BhcKIRvDz4w.NSV6.G3TnTH6a'
*/

class User < ApplicationRecord
  has_one :biodata_user, dependent: :destroy
  has_many :posts, dependent: :destroy
  has_many :events, dependent: :destroy
  validates :email, :presence => true, :uniqueness => true
  has_many :user_relations, dependent: :destroy
  has_many :relations, through: :user_relations
  has_many :notifications
  has_many :messages
  has_many :participants
  include BCrypt
  validates :email, format: { with: /\A(?:[^\s]+)@((?:[-a-z0-9]+\.)+[a-z]{2,})\z/i}
  before_create :confirmation_token

  def password
    @password ||= Password.new(password_digest)
  end
end
```

Gambar 3.8 Hash pada fitur user

3.2.5. Fitur Biodata User

Pada fitur ini, pengguna diharuskan untuk mengisi fitur *biodata_user* terlebih dahulu untuk bisa mengakses fitur lainnya. Pada fitur ini berguna untuk menyimpan data informasi dari pengguna seperti pada Gambar 3.9 yang akan ditampilkan pada halaman edit profil. Pengguna tidak bisa melewati fitur ini, dikarenakan notifikasi untuk mengisi *biodata_user* akan muncul lebih awal ketika pengguna telah melakukan register akun kemudian verifikasi akun dan melakukan *login* pertama kali pada akun tersebut, pada Gambar 3.10 merupakan logika untuk membuat sebuah *biodata_users* yang jika berhasil akan menampilkan respons seperti Gambar 3.11 yang menampilkan detail informasi dari data tabel biodata berisi nama, email, nomor telepon, tanggal lahir, alamat, status pernikahan, foto profil, status akun apakah sudah ter-verifikasi atau belum. Apabila salah satu kolom tidak diisi maka sebuah data untuk *biodata_users* tidak akan bisa disimpan dan akan menampilkan respons pesan seperti Gambar 3.12 yang akan mengharuskan pengguna untuk mengisi kolom yang kosong.

```

def new_attribute
  {
    name: self.user.name,
    email: self.user.email,
    phone: self.user.phone,
    dob: self.dob,
    address: self.address,
    marriage_status: self.marriage_status,
    status: self.status,
    avatar: self.avatar.url,
    is_verified: self.user.email_confirmed
  }
end

```

Gambar 3.9 Atribut API *biodata_user*

```

def create
  @biodata = BiodataUser.new(biodata_params.merge(user_id: @user.id))
  @biodata.save ? response_to_json({biodata:@biodata.new_attribute},
:success) : response_error(@biodata.errors, :unprocessable_entity)
end

```

Gambar 3.10 Fungsi *create* pada fitur *biodata_user*

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

```

```

{
  "data": {
    "biodata": {
      "name": "Jasmine Erina",
      "email": "jyerind@gmail.com",
      "phone": "081345990216",
      "dob": "2001-05-15T00:00:00.000+07:00",
      "address": "Gunung Guntur",
      "marriage_status": "Belum Menikah",
      "status": "Publik",
      "avatar": "https://samakaki.fly.dev/rails/active_storage/disk/
eyJicmFpbHM10nsibWVzc2FnZSI6IjBaDddV693YT3wNVNTSWHN056wJNCbWEzWj8kVFE6ZG1zNmVXbzFNMjk2WldobmMyTjBiQVlk2QntWU90QmthWE53YjN0cGRHbHZia2tpYzJsd
WJHbVhVHmNm1sc1pXWhiV1U5Sw1RMV1UVt3PRGxwM1Kau1XTX1NM1k1w1RMV1qSm1ZekUwM1aaUSURTBMBxB3w1JN01HwB1R1Z1WVcxBeqMwZWRV10T0N1b1pEvmhOVfK0T1
RwaV1t5XZek16WmpsaVpEbG1NbVpqTVRSbVptSTFNvFF1Yw5CbkJqc0dWRG95WTI5dWRHvNvKRjkwZVhCbFTSVBhVzFowjJvdmFuQmxad1k3QmxRnkYVTmxjblpwTjWzWjRnRaV69
LYkc5a11Xdz0iLCJleHAiOiIyMDIzLTA3LTEwVDE00iIw0jE5Ljc2Ni0iLCJwdX10i3ib69iX2tleSj9fQ==--9087c357fbecca47a152625a70e0da42895964b4e/
d5a56895bbb1c23f9bd9b2fc14ffb514.jpg"
    }
  },
  "is_verified": true
},
"status": "success"

```

Gambar 3.11 Respon API pada pengisian *biodata_users*

```

1
2
3
4
5
6
7

```

```

{
  "message": {
    "address": [
      "can't be blank"
    ]
  }
}

```

Gambar 3.12 Respon API ketika tidak berhasil

3.2.6. Fitur Post

Pada fitur ini setelah pengguna mengisi *biodata_user*, pengguna bisa mengakses fitur *post* di mana pengguna bisa mengunggah unggahan yang berisi teks, gambar, dan video. Dalam satu *post* terdapat data yang bisa dilihat pada Gambar 3.13 dan memiliki kapasitas penyimpanan *content* maksimal 10 *megabytes* yang jika pengguna mengunggah ukuran melebihi batas maksimal akan menampilkan respons seperti pada Gambar 3.15. Pengguna

harus melakukan konfirmasi email akun terlebih dahulu untuk bisa mengakses fitur membuat *post* dan jika berhasil akan menampilkan respons seperti pada Gambar 3.16.

```
def new_attribute
  {
    id: self.id,
    descriptions: self.descriptions,
    status: self.status,
    content: self.content.url,
    created_at: self.created_at,
    :user =>
    {
      name: self.user.name,
      avatar: self.user.biodata_user.try(:avatar).try(:url)
    }
  }
end
```

Gambar 3.13 Atribut pada fitur *post*

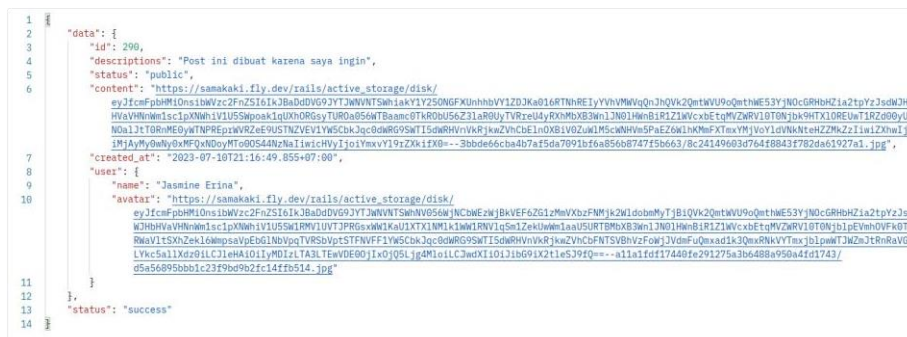
```
def create
  @post = Post.new(post_params.merge(user_id: @user.id))
  @post.save ? response_to_json(@post.new_attribute, :success) :
  response_error(@post.errors, :unprocessable_entity)
end

def post_params
  params.permit(:descriptions, :status, :content)
end
```

Gambar 3.14 Fungsi *create* pada fitur *post*



Gambar 3.15 Respon API pada fitur *post* ketika ukuran *content* tidak sesuai



Gambar 3.16 Respon API pada fitur *post* ketika berhasil

3.2.7. SMTP Gmail

Fungsi ini berguna untuk memudahkan lokal email untuk mengirimkan *token* kepada server sebelum ke server email dituju yang bertujuan sebagai pengguna untuk melakukan verifikasi akun ke email terdaftar yang aktif. Tujuan lain dalam penggunaan SMTP ini untuk meminimalisir penggunaan email palsu sebagai akses utama, sehingga tidak menimbulkan interupsi dari pihak luar. Berikut contoh penulisan fungsi untuk menggunakan SMTP dengan *framework* Ruby on Rails pada Gambar 3.17.

```
# SMTP for gmail
config.action_mailer.perform_deliveries = true
config.action_mailer.default_url_options = { host: 'localhost', port: 3000 }

config.action_mailer.delivery_method = :smtp
config.action_mailer.smtp_settings = {
  address:           'smtp.gmail.com',
  port:             587,
  domain:           'localhost:3000',
  user_name:        'samakaki.gatros@gmail.com',
  password:         '██████████',
  authentication:   'plain',
  enable_starttls_auto: true,
  open_timeout:     5,
  read_timeout:     5 }
```

Gambar 3.17 Fungsi SMTP

3.2.8. Fitur Family Tree

Pada fitur ini berfungsi bagi pengguna untuk melihat, membuat, dan mengundang pada silsilah keluarga atau *family tree*. Fitur ini memiliki dua tabel data yang saling berkaitan yaitu tabel *user* dan *relations* yang membentuk tabel baru bernama *user_relations* berisi atribut data seperti pada Gambar 3.18 yang akan memanggil fungsi Get Relations yang berisi data dari *user*.

```
def new_attribute
  {
    id: self.id,
    status: self.status,
    user: self.user_id = self.user.new_attribute
  }
end

def self.get_relation(user)
  @detail = []
  @con_relations_user = []
  @relations = UserRelation.where(user_id: user.id)
  @relations.map do |relation|

  .
  .
  .

    end
  end
end
end
```

```

return {current_user:user.name,relation:@detail}
end

```

Gambar 3.18 Fungsi *create* fitur *user_relations*

Dari fungsi di atas, terdapat *user_id* karena pada tabel ini mengambil data yang ada pada tabel *user* untuk mengambil *user_id* dan tabel *Relations* untuk mengambil *relation_id*. Alur fitur ini yakni:

1. *User* melakukan registrasi terlebih dahulu dan verifikasi email terlebih dahulu agar bisa login
2. *User* mengisi biodata user agar bisa mengakses fitur lainnya
3. *User* membuka fitur *family tree*, kemudian pada anggota pertama user harus mengisi data untuk mengundang pengguna lainnya
4. *User* mengirimkan *link* undangan kepada anggota yang dituju, pada link terdapat token yang berisi data dari *user* dan *relations*
5. Ketika link tersebut di klik, maka akan menuju ke halaman registrasi untuk pengguna baru
6. Pengguna baru akan mendapatkan notifikasi untuk mengkonfirmasi undangan dari pengguna yang lain untuk masuk ke dalam *family tree*

Penulis membuat kode untuk kategori apa saja yang ada di *family tree* yang bisa dilihat pada Gambar 3.19 dikarenakan dari pihak *Frontend* dan *Mobile* belum bisa menampilkan tata letak *family tree* secara dinamis dan bertujuan memudahkan untuk memanggil fungsi pada *relations_name*. Pada alur fitur yang telah dijelaskan di atas memiliki beberapa respons API yaitu pada alur pengguna mengundang pengguna lain yang menampilkan respons seperti Gambar 3.20 yang jika dibaca akan menjelaskan pengguna mengundang pengguna lainnya bernama Riko sebagai adik pertama dari silsilah keluarga dan memiliki status *non active* yang di mana menandakan undangan tersebut belum dikonfirmasi oleh pengguna yang dituju.

Maka untuk mengkonfirmasi undangan tersebut menampilkan respons seperti pada Gambar 3.21 yang akan muncul pada fitur notifikasi dan jika sudah dikonfirmasi akan menampilkan respons seperti pada Gambar 3.22. Setelah sudah dikonfirmasi, status akan berubah menjadi *active* dan ketika diperiksa pada data kedua *user* maka akan menampilkan respons seperti pada Gambar 3.23 yang menandakan kedua *user* sudah saling terhubung pada suatu *family tree*.


```
enum :relation_name, {bapak:0, ibu:1, adek_pertama:2, kakek_dari_bapak:3,
nenek_dari_bapak:4,kakek_dari_ibu:5,nenek_dari_ibu:6,husband:7,wife:8,adek_kedua:
9,adek_ketiga:10,anak_pertama:11,anak_kedua:12,anak_ketiga:13,kakak_pertama:14,ka
kak_kedua:15,kakak_ketiga:16}
```

Gambar 3.19 Penamaan Untuk *Role Relation*

```
1  |
2  | "data": {
3  |   "id": 184,
4  |   "name": "Riko",
5  |   "invitation_token": "eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjo5NywicmVsYXRpb25faWQ1OjE4NH0.RgJHCBeHEs1zw1p0nzKzUTqL531s2cvQhrfes30BHjg",
6  |   "relation_name": "adek_pertama",
7  |   "code": "Kn1",
8  |   "user_relation": {
9  |     "id": 184,
10 |     "connected_user_id": null,
11 |     "status": "non_active",
12 |     "user_id": 97,
13 |     "relation_id": 184
14 |   }
15 | },
16 | "status": "success"
17 |
```

Gambar 3.20 Respons API ketika pengguna mengundang pengguna lainnya pada *family tree*

```
1  |
2  | "data": {
3  |   "id": 110,
4  |   "user_relation_id": 184,
5  |   "user_id": 115,
6  |   "status": "unread",
7  |   "descriptions": "Anda di undang ke dalam family tree oleh Jasmine Erina, apakah anda mengenal Jasmine Erina?",
8  |   "created_at": "2023-07-10T21:26:20.131+07:00",
9  |   "updated_at": "2023-07-10T21:26:20.131+07:00"
10 | },
11 | "status": "success"
12 |
```

Gambar 3.21 Respons API ketika pengguna mendapatkan undangan *family tree*

```
1  |
2  | "data": {
3  |   "relation": {
4  |     "connected_user_id": 115,
5  |     "status": "active",
6  |     "id": 184,
7  |     "token": "eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjo5NywicmVsYXRpb25faWQ1OjE4NH0.RgJHCBeHEs1zw1p0nzKzUTqL531s2cvQhrfes30BHjg",
8  |     "user_id": 97,
9  |     "relation_id": 184,
10 |     "created_at": "2023-07-10T21:24:35.378+07:00",
11 |     "updated_at": "2023-07-10T21:29:18.489+07:00"
12 |   }
13 | },
14 | "status": "success"
15 |
```

Gambar 3.22 Respons API ketika pengguna yang diundang sudah konfirmasi

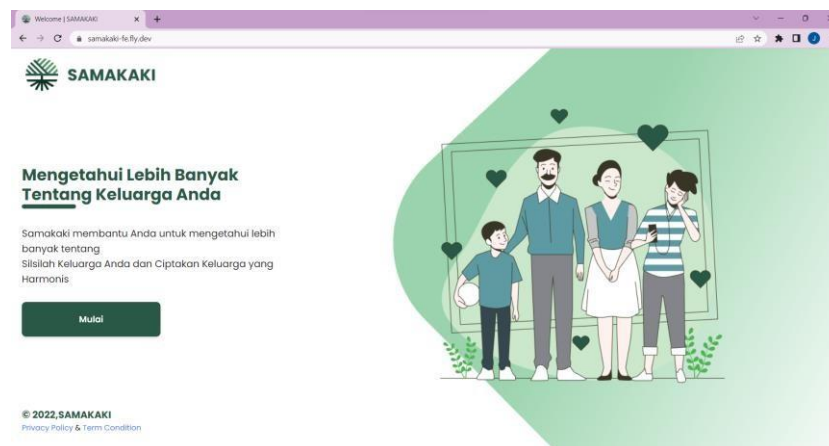
```
1  |
2  | "data": {
3  |   "relation": {
4  |     "connected_user_id": 115,
5  |     "status": "active",
6  |     "id": 184,
7  |     "token": "eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjo5NywicmVsYXRpb25faWQ1OjE4NH0.RgJHCBeHEs1zw1p0nzKzUTqL531s2cvQhrfes30BHjg",
8  |     "user_id": 97,
9  |     "relation_id": 184,
10 |     "created_at": "2023-07-10T21:24:35.378+07:00",
11 |     "updated_at": "2023-07-10T21:29:18.489+07:00"
12 |   }
13 | },
14 | "status": "success"
15 |
```

Gambar 3.23 Respons API jika kedua pengguna sudah saling terhubung di *family tree*

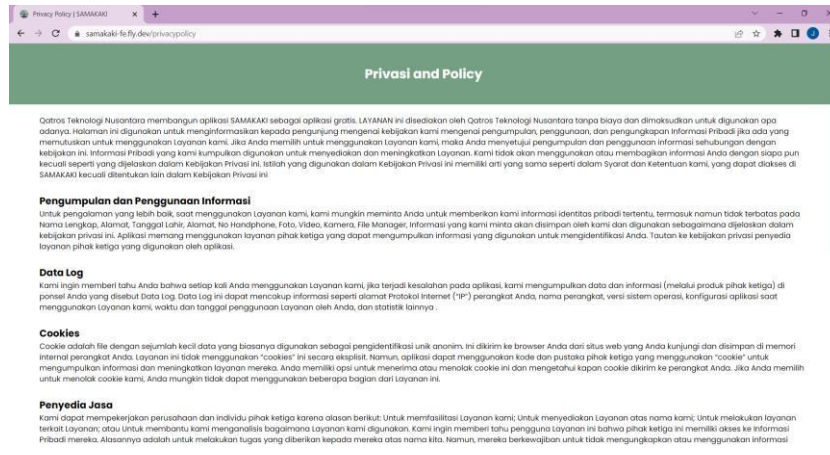
4.3 Hasil Proyek

4.3.1 Halaman *Landing Page*

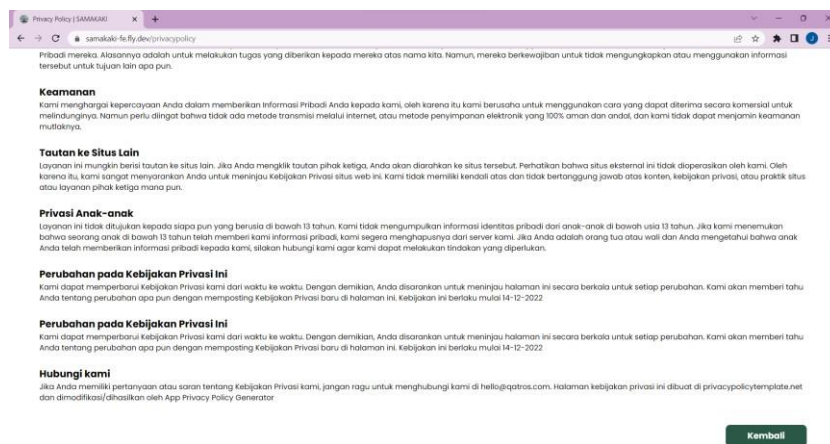
Halaman *landing page* adalah sebuah halaman awal yang akan pengguna lihat saat pertama kali mengakses sistem. Di dalam halaman ini terdapat tombol mulai yang akan menuju ke halaman daftar akun, kemudian terdapat teks yang berisi *link* untuk mengakses sebuah halaman yaitu halaman Privacy Policy dan Term Conditions. Pada halaman Privacy Policy berisikan informasi mengenai hukum yang menjelaskan tentang jaminan data dan informasi pengunjung yang diambil dari sistem ini seperti pada Gambar 3.25 dan Gambar 3.26. Pada halaman Privacy Policy terdapat tombol kembali yang akan mengembalikan pengunjung ke halaman *landing page*. Apabila pengunjung klik teks Term Conditions akan menampilkan seluruh informasi mengenai syarat dan ketentuan lisensi yang dibatasi oleh jangka waktu pemakaian tertentu seperti pada Gambar 3.27 dan pada Gambar 3.28 memiliki tombol kembali untuk kembali ke *landing page*.



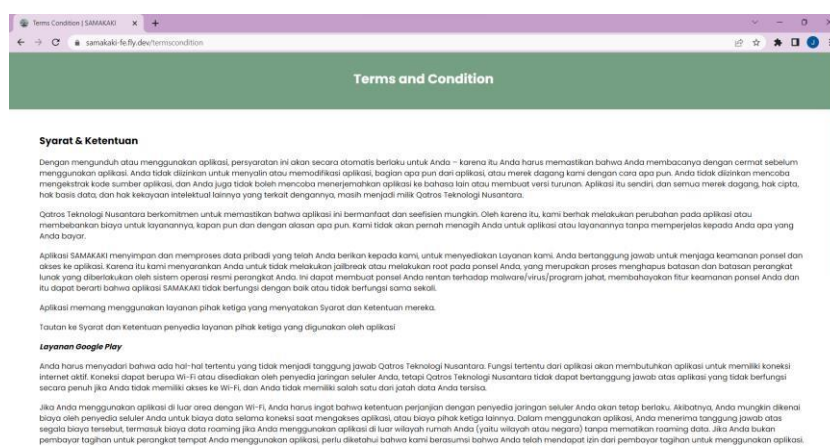
Gambar 3.24 Halaman Landing Page



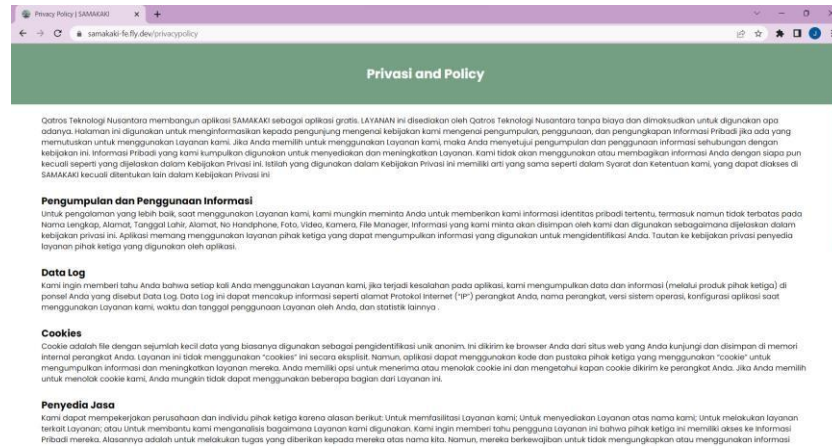
Gambar 3.25 Halaman Privacy and Policy



Gambar 3.26 Halaman Privacy and Policy (2)



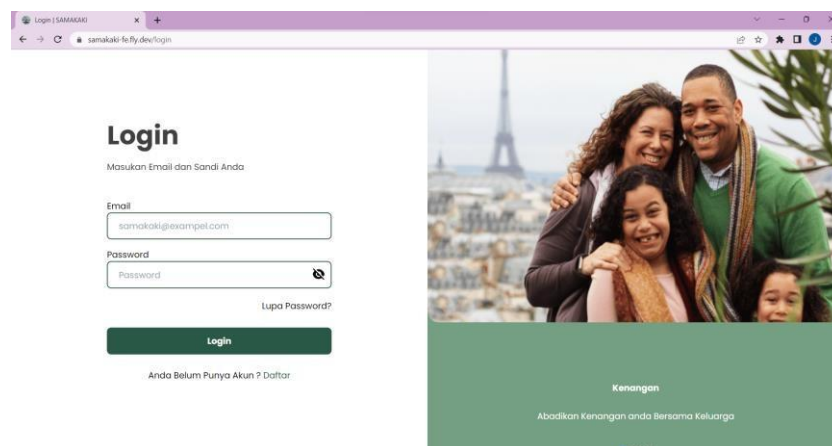
Gambar 3.27 Halaman Terms and Condition



Gambar 3.28 Halaman Terms and Condition (2)

4.3.2 Halaman Login

Halaman *login* menampilkan *form* di mana pengunjung harus mengisi data berupa email dan kata sandi untuk melakukan aktivitas *login* dengan menggunakan akun yang sudah ada ketika pengunjung belum memiliki akun bisa mengakses *link* teks *Daftar*. Terdapat *hyperlink* *Lupa Password* yang berfungsi ketika pengguna mengalami kendala lupa kata sandi untuk melakukan aktivitas reset pada kata sandi. Tampilan laman *login* bisa dilihat pada Gambar 3.29.

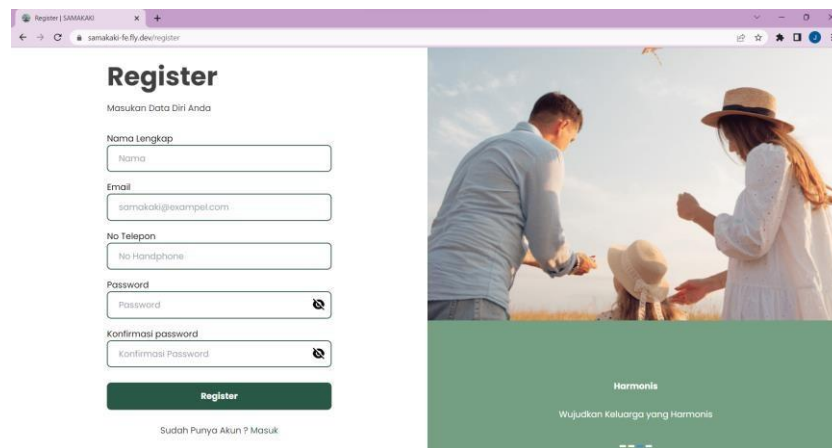


Gambar 3.29 Halaman Login

4.3.3 Halaman Daftar Akun

Halaman daftar akun bertujuan untuk pengunjung mendaftarkan akun baru ke dalam sistem. Saat halaman ini diakses, sistem akan menampilkan sebuah *form* yang harus diisi terlebih dahulu. Data yang dimasukkan ke dalam *form* berupa Nama Lengkap, Email, Nomor Telepon, Kata Sandi, dan Konfirmasi Kata Sandi. Khusus pada kolom Kata Sandi, pengguna

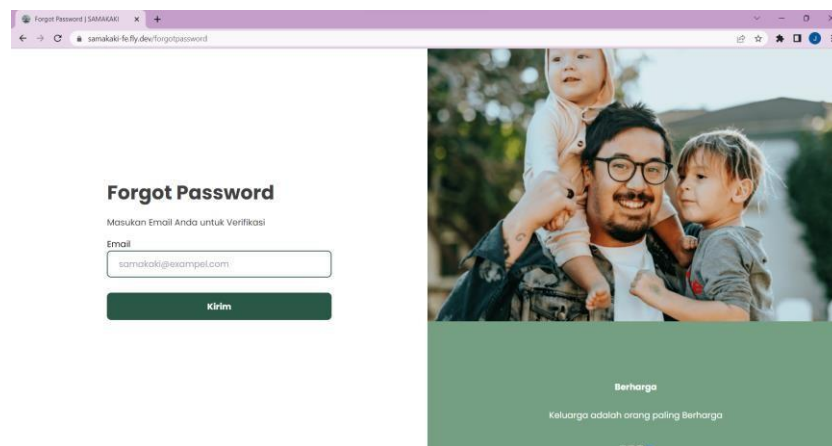
harus memasukkan kata sandi dengan ketentuan minimal delapan karakter. Halaman daftar akun ditunjukkan pada Gambar 3.30.



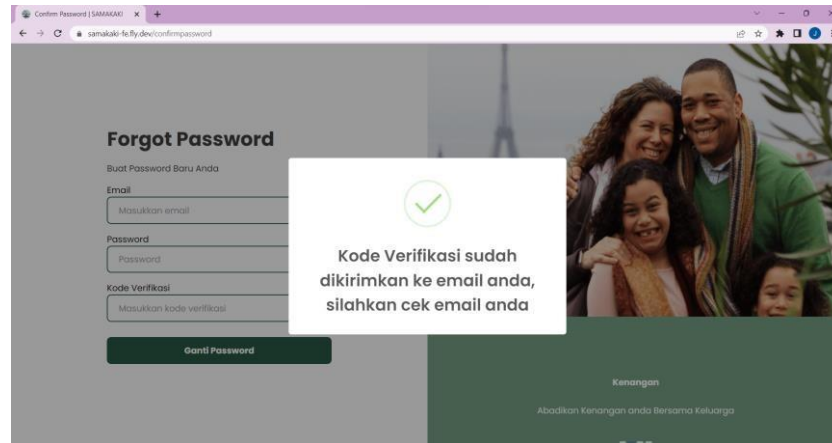
Gambar 3.30 Halaman Daftar Akun

4.3.4 Halaman Forgot Password

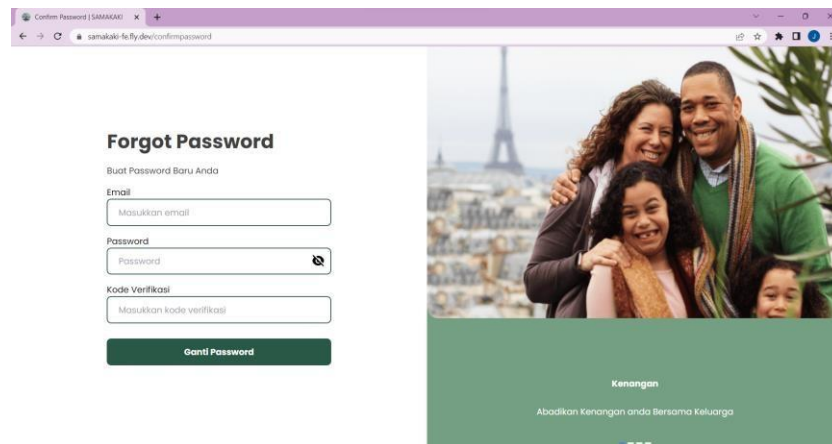
Halaman Lupa Kata Sandi menampilkan *form* untuk mengisi email terlebih dahulu agar sistem bisa mengirimkan *token* yang nantinya akan digunakan untuk melakukan reset kata sandi, tampilan tersebut ditunjukkan pada Gambar 3.31 dan ketika email telah dikirimkan oleh sistem maka sistem akan menampilkan notifikasi seperti pada Gambar 3.32. Setelah pengguna mendapatkan *token* yang telah dikirimkan ke email, maka pengguna akan mengisi *form* untuk melakukan reset kata sandi. Tampilan *form* untuk mengisi reset kata sandi ditunjukkan pada Gambar 3.33.



Gambar 3.31 Halaman Lupa Kata Sandi



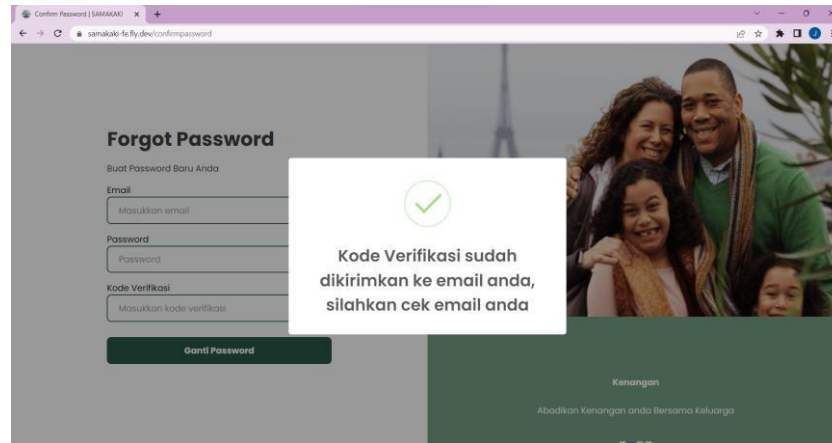
Gambar 3.32 Halaman Lupa Kata Sandi (2)



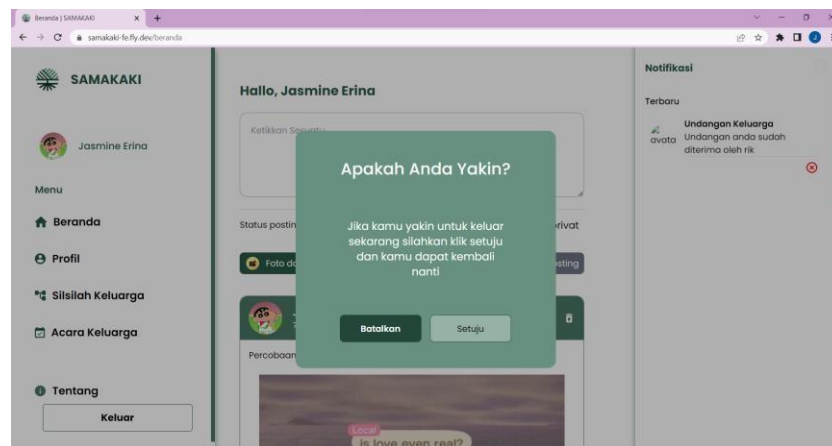
Gambar 3.33 Halaman Lupa Kata Sandi (3)

4.3.5 Halaman Beranda

Halaman beranda merupakan halaman yang ditampilkan setelah pengguna melakukan aktivitas *login*. Pada halaman ini menampilkan nama pengguna yang sedang *login* terletak di sebelah kiri *navigation bar* dan di tengah atas halaman. Pada halaman ini juga mencakupi beberapa fitur yang tampil seperti fitur *post* dan notifikasi. Di pojok kiri bawah terdapat tombol keluar yang bertujuan untuk pengguna melakukan proses *logout* dari akses beranda yang ditampilkan pada Gambar 3.35. Tampilan halaman beranda ditunjukkan pada Gambar 3.34.



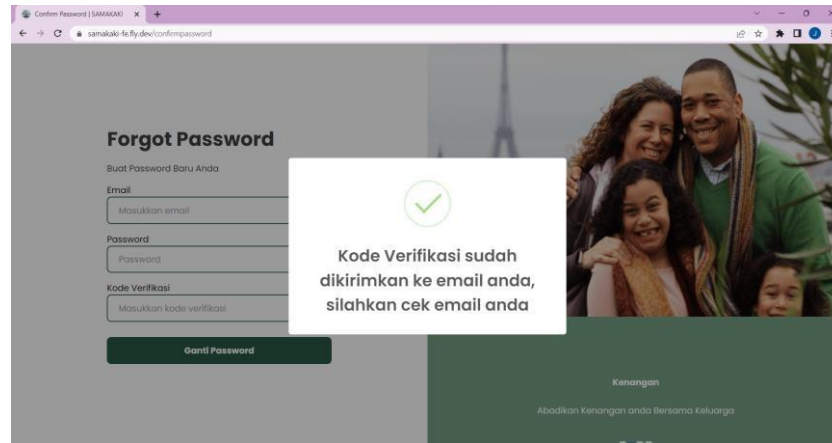
Gambar 3.34 Halaman Beranda



Gambar 3.35 Halaman Beranda (2)

4.3.6 Halaman Profil

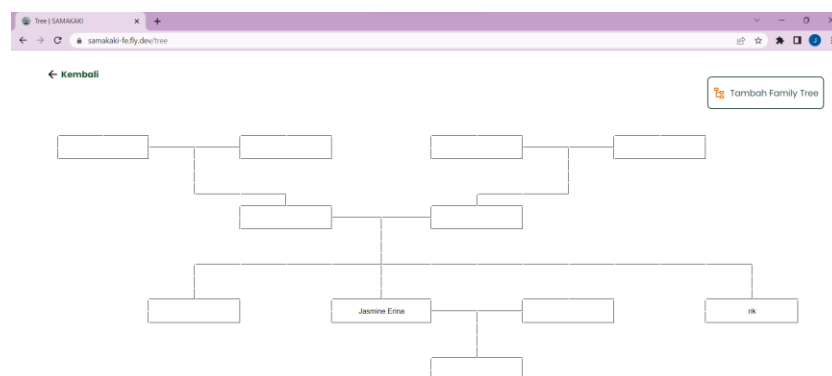
Halaman profil akan menampilkan pratinjau dari data yang telah diisi sebelumnya di laman daftar akun. Pada halaman profil pengguna bisa mengubah profil sesuai keinginan, mulai dari foto profil, nama, tanggal lahir, dan lokasi. Untuk foto profil pengguna hanya bisa mengunggah ukuran foto maksimal 10 *megabyte*. Tampilan halaman profil ditunjukkan pada Gambar 3.36.



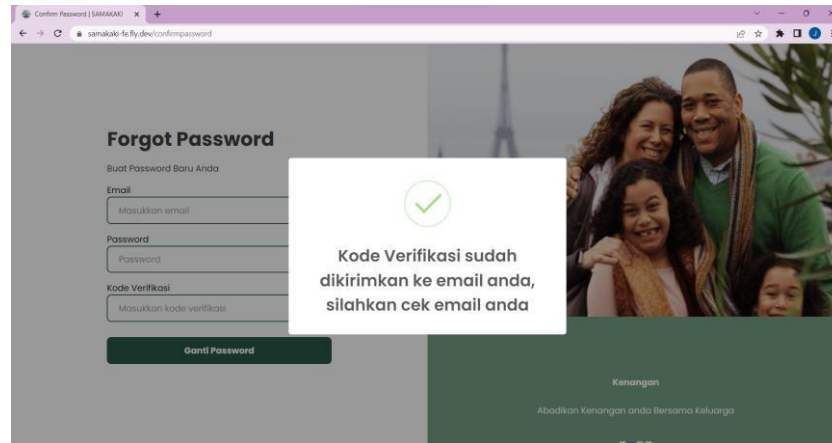
Gambar 3.36 Halaman Profil

4.3.7 Halaman Silsilah Keluarga

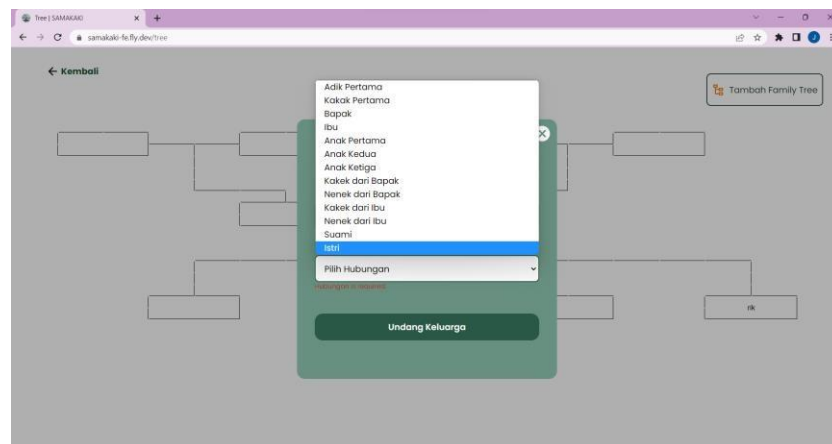
Halaman Silsilah Keluarga menampilkan struktur dari pohon keluarga yang telah dibuat. Struktur yang dibuat masih terbentuk secara statis, sehingga pengguna utama akan berada di tengah bagan struktur silsilah keluarga. Pada pojok kiri terdapat tombol kembali yang berfungsi mengembalikan pengguna ke halaman beranda dan pojok kanan atas terdapat tombol Tambah Family Tree yang berfungsi untuk mengundang keluarga yang lain, tampilan halaman ditunjukkan pada Gambar 3.37. Ketika tombol Tambah Family Tree ditekan maka akan muncul halaman *pop up* yang berisi *form* untuk mengundang anggota silsilah keluarga yang ditunjukkan pada Gambar 3.38 dan Gambar 3.39.



Gambar 3.37 Halaman Silsilah Keluarga



Gambar 3.38 Halaman Undang Keluarga

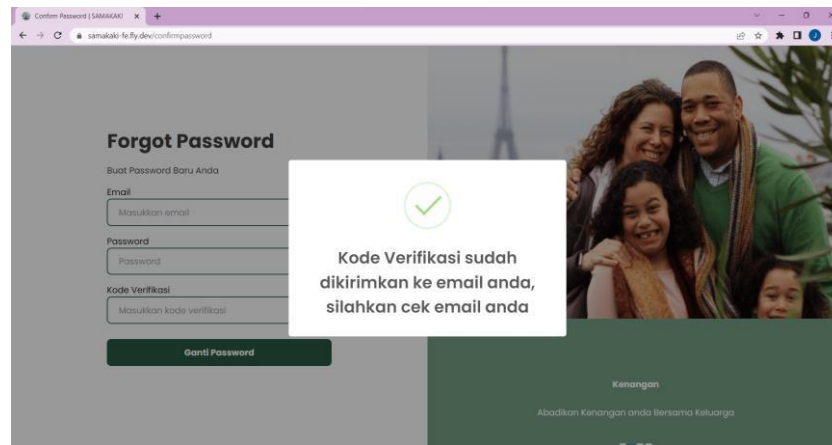


Gambar 3.39 Halaman Undang Keluarga (2)

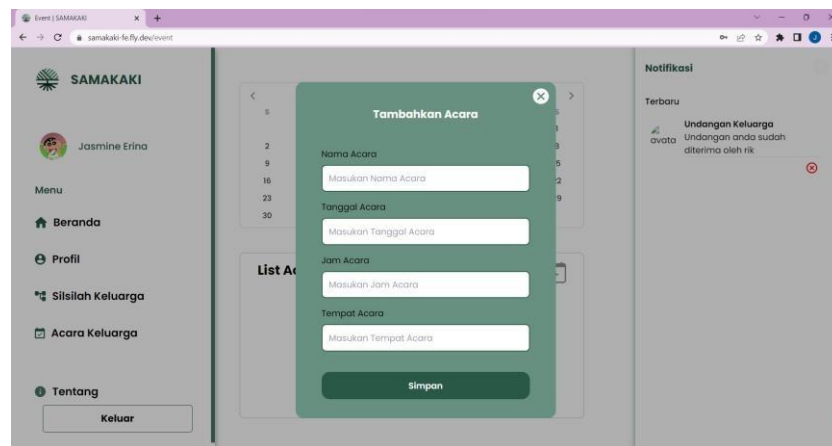
4.3.8 Halaman *Event*

Halaman *event* bertujuan untuk memudahkan anggota silsilah keluarga mengetahui apa saja agenda acara yang ada. Pada halaman ini secara umum seperti aplikasi kalender yang menampilkan tanggal, bulan, dan tahun. Ketika sebuah silsilah keluarga memiliki sebuah acara maka akan menampilkan acara yang telah diagendakan yang ditunjukkan pada Gambar 3.40. Pada laman ini terdapat ikon kalender dengan tanda plus, ketika ditekan maka akan menampilkan *pop up* yang berisi *form* yang harus diisi untuk membuat *event* seperti pada Gambar 3.41.

Form yang diisikan pada halaman ini berisi data Nama Acara, Tanggal Acara, Jam Acara, dan Tempat Acara. Ketika data telah diisi dan disimpan, maka akan ditampilkan pada List Acara. Dalam List Acara terdapat informasi dari acara yang akan datang, pengguna bisa melakukan *edit* dan *delete* seperti pada *icon* yang tersedia.



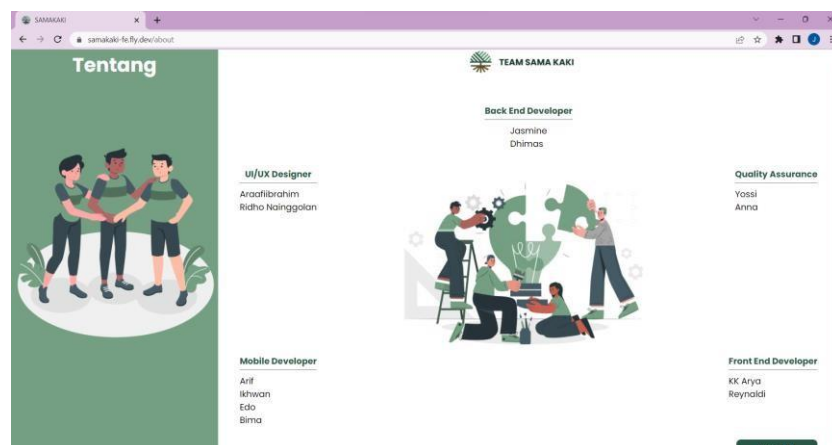
Gambar 3.40 Halaman Event



Gambar 3.41 Halaman Pop Up Create Event

4.3.9 Halaman Tentang

Halaman tentang hanya menampilkan informasi mengenai tim *developer* yang mengerjakan aplikasi ini.



Gambar 3.42 Halaman Tentang

BAB IV

REFLEKSI PELAKSANAAN MAGANG

5.1 Relevansi Akademik

Dalam pelaksanaan magang ini terdapat kesinambungan antara materi yang diberikan saat perkuliahan dengan yang ada di lapangan kerja magang. Salah satunya yaitu mata kuliah Pengembangan Aplikasi Berbasis Web, pada mata kuliah ini mahasiswa diwajibkan untuk memahami cara mengembangkan dan mengetahui kebutuhan sebuah aplikasi berbasis web. Pada lapangan kerja magang, pekerja magang ditugaskan untuk merancang dan mengembangkan sebuah aplikasi dari segi *backend*. Selain itu terdapat persamaan antara materi pada mata kuliah Pengembangan Aplikasi Berbasis Web dengan proses pelaksanaan kegiatan magang seperti Implementasi Bahasa Pada Sebuah Sistem, Penggunaan Framework, Penggunaan Metode Pengembangan.

5.1.1 Implementasi Bahasa Ruby Pada Sebuah Sistem

Selama enam bulan magang penulis harus beradaptasi dengan kebutuhan kantor, yaitu menggunakan bahasa Ruby sebagai alat utama dalam pengerjaan proyek. Ruby sendiri merupakan hal yang baru bagi penulis, sehingga penulis mengalami beberapa kendala dalam penggunaannya dikarenakan bahasa pemrograman yang dipelajari saat di kampus berbeda dengan kebutuhan kantor. Untuk menyelesaikan kendala tersebut, penulis melakukan latihan secara mandiri untuk meningkatkan keahlian dalam menggunakan bahasa tersebut agar bisa menyelesaikan proyek dengan lancar dan mudah.

Tahapan awal dalam melakukan pengenalan dengan Ruby, mentor memberikan beberapa *task* yang bertujuan untuk meningkatkan fundamental logika dalam penggunaan bahasa Ruby. Penulis juga mencoba implementasi Ruby dengan basis data PostgreSQL di mana basis data tersebut digunakan sebagai jenis basis data utama yang digunakan dalam kebutuhan proyek ini.

5.1.2 Penggunaan Framework

Penggunaan *framework* secara umum bertujuan untuk memudahkan sebuah proses pengembangan pada sebuah aplikasi. Pada dasarnya *framework* merupakan sebuah kerangka kerja yang dapat membantu developer dalam penulisan kode. Pada penggunaan bahasa Ruby, maka penggunaan *framework* pun berbeda dengan bahasa pemrograman yang lain. Jenis

framework yang digunakan adalah Ruby on Rails, pada *framework* ini menggunakan arsitektur Model View Controller (MVC).

Penggunaan *framework* sangat fleksibel sehingga hal ini memudahkan developer dalam pengembangan sebuah aplikasi atau *website* secara *full-stack* dikarenakan bisa menangani tugas *backend* dan *frontend* dalam satu kerangka proyek. Keunggulan dari Rails sendiri memiliki banyak fungsi salah satunya adalah memperbarui basis data, mengirim dan menerima email, menyimpan unggahan pada *cloud*.

Penggunaan *framework* ini termasuk ke tingkat mudah ke sulit, dikarenakan pada sisi sulit ia mengharuskan pengguna melakukan instalasi terlebih dahulu terkait *library* apa saja yang ingin digunakan. Namun, di sisi lain ia memiliki sebuah *command-line tools* yang bernama RubyGems yang berfungsi sebagai ekstensi yang memudahkan para developer menambahkan sebuah *function* tanpa harus menulis kode tambahan.

5.1.3 Pembuatan Aplikasi Dengan Metode SCRUM

Dalam pembuatan aplikasi menggunakan metode yang sebelumnya penulis belum pernah gunakan yaitu metode SCRUM. Penggunaan metode ini membuat penulis harus beradaptasi dengan aturan yang ada pada metode SCRUM. Tidak hanya itu, penulis juga harus beradaptasi dengan fleksibilitas dalam penerapan metode ini. Metode ini memiliki beberapa tahap di antaranya Sprint Planning, Sprint Backlog, Daily Meeting, Sprint Review, dan Sprint Retrospective. Bahwa metode ini dilaksanakan dalam kurun waktu sepuluh hari sekali, sehingga pada setiap sprint, *backlog* yang akan dikerjakan harus disesuaikan dengan kurun waktu yang ada, agar tidak terjadi Backlog Blooming yang dapat menghambat sebuah pengembangan aplikasi.

Selama pengerjaan proyek ini, kelebihan dari metode Scrum adalah tidak memerlukan tim yang besar untuk menghasilkan produk secara efisien. Dengan demikian, menggunakan tim dengan ukuran lebih kecil dapat mengurangi biaya dan memberikan dukungan satu sama lain. Dalam penerapan metode ini, anggota harus selalu siap untuk mengadaptasi diri terhadap perubahan yang mungkin terjadi pada setiap pertemuan karena kebutuhan berubah dengan cepat sesuai dengan perubahan yang dibutuhkan.

Pada tahapan Retrospective, membuat tim pengembang untuk merefleksikan hasil sprint yang telah berjalan sebelumnya. Keuntungan yang dirasakan adalah peningkatan proses, pembelajaran, dan perbaikan berkelanjutan pada proyek yang sedang dikerjakan. Tim dengan mudah melakukan identifikasi apa yang telah berhasil dan apa saja yang perlu diperbaiki, dan

memungkinkan adaptasi lebih cepat terhadap perubahan dan peningkatan kualitas pengerjaan pada Sprint. Selain itu, Retrospective meningkatkan kolaborasi antara anggota tim, mengurangi frustrasi, dan meningkatkan kepuasan pada anggota tim, serta memantau kemajuan proyek secara lebih efektif. Hal ini mendorong keterlibatan tim dalam pengembangan produk dan mencapai tujuan proyek dengan lebih baik melalui Tindakan perbaikan yang konkret.

5.2 Pembelajaran Magang

5.2.1 Teknis

Pelaksanaan magang yang telah berlangsung dalam kurun waktu enam bulan di PT Qatros Teknologi Nusantara menghasilkan banyak pengalaman yang baru bagi penulis yang dapat bermanfaat untuk melanjutkan karier di masa yang akan datang. Selama magang, penulis mempelajari alur proses pengerjaan suatu proyek dengan menggunakan metode SCRUM di mana pada metode ini mengusung efektivitas dan efisiensi waktu dalam pengerjaan suatu proyek. Penulis juga mempelajari keberagaman bahasa pemrograman yang tersedia, salah satunya adalah Ruby yang digunakan sebagai bahasa utama di lapangan kerja. Sebagai pendukung dalam pengerjaan proyek terdapat *tools* baru yang penulis pelajari yaitu *framework* bernama Ruby on Rails.

Selama magang, pengembangan proyek ini tidak jauh dari pembelajaran yang telah dipelajari di perkuliahan. Dalam proses perancangan aplikasi, ilmu yang diterapkan memiliki kesamaan dengan teori yang dipelajari pada mata kuliah Pengembangan Aplikasi Berbasis Web dan Rekayasa Perangkat Lunak. Teori tersebut meliputi perancangan basis data, mengolah data melalui sistem yang telah dirancang, melakukan implementasi yang telah dirancang oleh *uix*.

5.2.2 Non Teknis

1. Public Speaking Dalam Presentasi

Keahlian berbicara di depan umum menjadi sebuah tantangan baru bagi para pekerja magang dikarenakan harus menjelaskan suatu hasil di depan orang yang lebih profesional. Keahlian ini sangat penting sebagai penunjang untuk ke jenjang berikutnya, khususnya pada dunia pekerjaan. Dalam metode Scrum terdapat sebuah tahapan bernama Daily Meeting, di mana pada tahapan ini semua anggota tim harus melakukan presentasi kecil terkait hasil kerja harian yang telah dikerjakan beserta apa saja kendala yang dialami agar tidak terjadi keterlambatan dalam pengerjaan proyek. Melalui pengalaman ini penulis juga harus menyiapkan poin-poin apa saja yang akan disampaikan pada presentasi.

2. Manajemen Waktu

Selama kegiatan magang, penulis mendapatkan tantangan dalam melakukan manajemen waktu. Dikarenakan kegiatan magang dilakukan secara luring dan mengharuskan untuk datang ke lapangan kerja dalam tepat waktu. Dalam hal ini, penulis juga melakukan prioritas kegiatan yang lebih utama mana yang harus didahulukan dari kegiatan magang dan kampus secara bersamaan.

3. Adaptasi Lingkungan

Ketika memulai sesuatu yang baru, dipastikan harus melakukan penyesuaian terhadap lingkungan yang baru. Budaya yang ada di lapangan berbeda dengan budaya di lingkungan kampus, hal yang dimaksud adalah budaya di lapangan memiliki kewajiban dikarenakan pekerjaan yang dilakukan bersangkutan dengan klien. Dalam hal ini, penyesuaian pada lingkungan menjadi kewajiban untuk membantu penulis dalam melakukan kegiatan yang ada di lapangan. Hal-hal yang harus disesuaikan ada banyak sekali, mulai dari penyesuaian dengan kolega magang, penggunaan *tools* yang digunakan di lapangan, dan rutinitas harian seperti *daily meeting*. Hal ini dapat mempercepat proses pembelajaran dan implementasi proyek pada kegiatan magang.

4. Kemampuan Berkomunikasi

Dalam pengerjaan proyek sangat dibutuhkan kemampuan berkomunikasi yang baik agar menciptakan harmonisasi dikarenakan proyek dikerjakan dalam bentuk sebuah tim. Untuk menciptakan hal tersebut, penulis harus melakukan komunikasi yang baik agar mengetahui dan memberitahu apa saja kebutuhan yang dibutuhkan pada proyek ini. Dalam tim memiliki keberagaman latar belakang divisi yang berbeda, yaitu beranggotakan dari divisi *frontend*, *backend*, *mobile*, *uix*, dan *quality assurance*.

5.3 Manfaat Magang

Selama kegiatan magang yang telah dilakukan selama enam bulan di PT Qatros Teknologi Nusantara mendapatkan berbagai macam manfaat dalam mengimplementasi ilmu pengetahuan yang telah dipelajari dan mempraktikkan ke dalam kegiatan magang. Khususnya dalam pengembangan aplikasi pada bagian *backend*. Dengan menggunakan bahasa Ruby memiliki *syntax* yang mudah dipahami dan dinamis, sehingga kode yang dibuat menjadi lebih ringkas dan mudah dibaca, yang pada akhirnya mempercepat proses pengembangan dan pemeliharaan aplikasi Samakaki. Selain itu Ruby merupakan bahasa yang digunakan dalam *framework* Ruby on Rails yang sangat produktif, dengan banyak alat dan konvensi yang

mengurangi pekerjaan repetitif dalam pengembangan suatu web. Dari penggunaan bahasa Java yang pernah digunakan dalam mata kuliah Pengembangan Aplikasi Berbasis Web memiliki kesamaan dalam alur pengerjaan pada *backend* namun secara teknis berbeda. Secara keseluruhan masing-masing dari bahasa tersebut memiliki karakteristik sendiri, namun logika pemrograman atau alur pembuatan sebuah aplikasi cenderung sama.

Salah satu manfaat yang diperoleh adalah *public speaking* dalam melakukan presentasi pada saat Daily Meeting, manajemen waktu untuk menentukan prioritas pada setiap kegiatan yang ingin dilakukan, adaptasi lingkungan yang baru di mana lingkungan tersebut merupakan lingkungan pekerjaan, dan kemampuan berkomunikasi dengan individu baru yang memiliki watak dan kepribadian yang berbeda. Hal tersebut membuat penulis memiliki rasa empati, menghargai pendapat seseorang, dan keterampilan dalam mengembangkan sebuah aplikasi.

BAB V

PENUTUP

6.1 Kesimpulan

Semua kegiatan pengembangan aplikasi telah selesai dilakukan dan menghasilkan beberapa hasil yang telah dijabarkan pada bab sebelumnya. Berikut merupakan kesimpulan dari kegiatan pengembangan aplikasi yang telah dilaksanakan.

- a. Penulis menjalankan kegiatan magang selama enam bulan terhitung dari bulan Agustus sampai dengan bulan Januari, kemudian dilanjut pada satu bulan penuh di bulan Maret. Kegiatan magang dilakukan secara *hybrid* sehingga bisa dilakukan di mana saja.
- b. Ruby on Rails memudahkan pengembangan aplikasi berbasis web dikarenakan *syntax* setara dengan bahasa manusia sehingga mudah dipahami. Namun, terdapat kesulitan dalam menerapkan *environment* Ruby on Rails apabila sistem operasi yang digunakan adalah Windows.
- c. Metode yang digunakan dalam pengembangan aplikasi ini adalah SCRUM dengan tahapan Sprint Planning, Sprint Backlog, Daily Meeting, Sprint Review, dan Sprint Retrospective.
- d. Basis data yang digunakan adalah PostgreSQL, yang sangat cocok dengan pengembangan sebuah aplikasi media sosial dengan penyimpanan skala menengah.
- e. Aplikasi media sosial Samakaki berhasil dibuat dengan menggunakan Ruby on Rails dan memiliki fitur User, Post, Biodata User, Family Tree, Events.
- f. Aplikasi Samakaki dapat menjadi media sosial yang aman digunakan untuk anak dikarenakan konten yang dilihat hanya berasal dari pengguna yang telah masuk ke dalam *family tree*.

6.2 Saran

Adapun saran untuk penelitian dan pengembangan aplikasi media sosial berbasis *website*, sebagai berikut:

- a. Melakukan analisis secara lebih mendalam dan detail agar dapat memenuhi kebutuhan dalam fitur yang akan diimplementasikan.
- b. Aplikasi dapat digunakan dan dikomersialkan kepada masyarakat untuk penggunaan yang diharap bermanfaat bagi kehidupan sehari-hari.
- c. Merancang tampilan desain keseluruhan aplikasi agar lebih menarik untuk digunakan pengguna.

DAFTAR PUSTAKA

- Dwi Praba, A., & Safitri, M. (2020). *Studi Perbandingan Performansi Antara MySQL Dan PostgreSQL*. VIII(2). <https://www.adminer.org/>.
- Dwi Praba STMIK Nusa Mandir Jakarta, A. (2018). Aplikasi Rekap Mengajar Berbasis Webiste Dengan Database PostgreSQL. In *Ijns.org Indonesian Journal on Networking and Security* (Vol. 8). Online.
- Fauzi, J. R. (2033). *Algoritma Dan Flowchart Dalam Menyelesaikan Suatu Masalah*.
- Juliansyah Eka Putra, F., Utomo, S., D. Rachmanto, A., & Budiarto, S. (2021). *Aplikasi Quiz Dengan Konsep Gamification Berbasis Web Menggunakan Ruby On Rails & React.JS*.
- Muri, M. F. A., Utomo, H. S., & Sayyidati, R. (2019). Search Engine Get Application Programming Interface. *Jurnal Sains Dan Informatika*, 5(2), 88–97. <https://doi.org/10.34128/jsi.v5i2.175>
- Nabilla, N. U., & Hidayat, S. (2021). *Pengembangan Business Intelligence pada Sistem Informasi Distributor*.
- Nainggolan, N. P., Heryenzus, & Utnasari, I. (2022). Pembinaan Manajemen Internet Sehat dan Aman Bagi Keluarga Pada Masyarakat Kelurahan Sekanak Raya. *Puan Journal Indonesia*, 3(2), 139–148.
- Oleh, D., & Pangestika, R. (2021). *Pengembangan Back-End Sistem Informasi Pendataan Sekolah Desa Komunitas Pendar Foundation Yogyakarta*.
- Pangestika, R., & Dirgahayu, R. T. (2021). *Pengembangan Back-end Sistem Informasi Pendataan Sekolah Desa Komunitas Pendar Foundation Yogyakarta*.
- Putri, N. B., & Romli, N. A. (2021). *Analisis Dampak Adiksi Internet Pada Media Sosial Twitter Di Indonesia Dengan Pendekatan Teori Komunikasi* (Vol. 7, Issue 1). www.journal.uniga.ac.id
- Rafiq, A. (2020). Dampak Media Sosial Terhadap Perubahan Sosial Suatu Masyarakat. *Global Komunika*, 1(1).
- Rizaldi, A., Maria, E., Wahyono, T., Purwanto, P., & Hartomo, K. D. (2022). Analisis Penerapan Metode Scrum Pada Pengembangan Sistem Informasi Akuntansi Koperasi. *Jurnal Media Informatika Budidarma*, 6(1), 57. <https://doi.org/10.30865/mib.v6i1.3349>
- Saputra, D., & Fathoni Aji, R. (2018). Analisis Perbandingan Performa Web Service Rest Menggunakan Framework Laravel, Django Dan Ruby On Rails Untuk Akses Data

Dengan Aplikasi Mobile (Studi Kasus: Portal E-Kampus STT Indonesia Tanjungpinang).
Bangkit Indonesia, 2.

Saragih, R. R. (2018). *Pemrograman Dan Bahasa Pemrograman*.

Schwaber, K., & Sutherland, J. (2020). *Panduan Scrum*.

Septian, B., Komara Jayadi, I., Holil, M., & Handriani, I. (2020). *Sistem Human Capital Management Menggunakan Metode Scrum*. <https://jurnal.umj.ac.id/index.php/just-it>

Setiadi, A. (2016). *Pemanfaatan Media Sosial Untuk Efektifitas Komunikasi*.

Sugeng Cahyono, A. (2018). *Dampak Media Sosial Terhadap Permasalahan Sosial Anak*.

Waruwu, T. S. (2019). *Implementasi PostgreSQL Sebagai Sistem Manajemen Basis Data Pada Pendaftaran Mahasiswa Baru Berbasis Web*.

Wisnu, D., Stmik, B., & Malang, A. (2015). Perancangan Sistem KHS Mobile di STMIK ASIA Malang Menggunakan Android Programming dan JSON. In *Jurnal Ilmiah Teknologi dan Informasia ASIA (JITIKA)* (Vol. 9, Issue 2). <http://www.myusro.info/2013/01/pengertian-json-javascript-object.html>