BAB 3
PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Pelaksanaan magang sebagai *Full Stack Web Developer* yang dilakukan di PT Otak Kanan di bimbing oleh Bapak Wahyu Prasetyo yang merupakan staf *Manager Operations* di PT. Otak Kanan. Selama dua bulan pertama, semua informasi seperti tugas diberikan secara online melalui aplikasi WhatsApp. Kemudian, magang offline dilakukan di kantor Graha Pena Surabaya dimana peserta magang akan melakukan *meeting* setiap minggu untuk membahas perkembangan dari proyek yang dilakukan.

3.2 Tugas yang Dilakukan

Selama kegiatan magang sebagai *Full Stack Web Developer* di PT Otak Kanan, tugas utama peserta magang adalah untuk merencanakan dan membangun website forum Sigarda mulai dari tampilan, fitur-fitur dan skema database yang diperlukan. Pembangunan website ini dilakukan menggunakan Visual Studio Code dengan ReactJS sebagai *framework* untuk bagian *Front-End* dan Laravel untuk bagian *Back-End*. Berikut merupakan beberapa tugas yang dilakukan selama pelaksanaan magang.

3.2.1 Merancang tampilan dari website menggunakan Figma

Pada tahap ini, rancangan tampilan dari website Sigarda mulai dari halaman *login* dan *register* hingga halaman notifikasi dibuat menggunakan aplikasi Figma. Disini Figma digunakan karena menyediakan berbagai alat dan fitur yang mendukung proses desain antarmuka pengguna (UI) dan pengalaman pengguna (UX), termasuk pembuatan wireframe, mockup, dan prototipe interaktif. Selain itu, Figma juga memungkinkan pengguna untuk dapat bekerja pada dokumen yang sama secara bersamaan, memberikan komentar, dan membuat perubahan dalam waktu nyata. Perancangan tampilan dari website ini dimulai dari melakukan *wireframing* untuk menentukan letak dari konten, membuat mockup atau representasi visual yang lebih detail dari wireframe, dan *prototyping* untuk memberi alur interaksi di dalam website.

3.2.2 Merancang skema database dan tabel-tabel yang diperlukan untuk website

Pada tahap ini dilakukan perancangan skema relasi dari tabel-tabel yang diperlukan dalam database untuk website Sigarda menggunakan website *Lucid Chart*. Total dari tabel yang dibuat pada tahap ini adalah 16 tabel yang semuanya dirancang berdasarkan ide atau fitur-fitur yang ada di dalam website tersebut. Tabel yang dibuat meliputi tabel *user*, *posts*, tabel forums, tabel *comments*, tabel notifikasi dan berbagai tabel *logs* yang diperlukan untuk mencatat perubahan yang dilakukan pada setiap tabel yang ada.

3.2.3 Mengimplementasikan rancangan yang telah dibuat

Pada tahap ini dilakukan implementasi untuk pembuatan *front-end* dan *back-end* dari website berdasarkan rancangan UI/UX dan skema relasi tabel database yang telah dibuat. Implementasi pembuatan *front-end* dari website dilakukan untuuk halaman jelajahi dan notifikasi dimana implementasi tersebut dilakukan menggunakan *text editor* Visual Studio Code dengan menggunakan *framework* ReactJS. Kemudian, *back-end* dari website dilakukan implementasinya menggunakan *framework* Laravel mulai dari pembuatan tabel dengan melakukan migrasi, pembuatan model dan controller, serta pembuatan route untuk API.

3.3 Uraian Pelaksanaan Magang

Pada Tabel 3.1 berikut adalah uraian pelaksanaan kerja magang yang dilakukan selama periode magang di PT Otak Kanan.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Perkenalan, mempelajari Laravel dan membuat model CRUD sederhana, membuat video demo dari model CRUD yang telah di buat.
2	Introduksi tugas website dewisata.com, mempelajari *source code* website dan cara kerja cPanel, membuat page baru untuk melakukan *submit price offer*
3	Membuat menu dan page untuk *view offer* dalam bentuk *tabular form*
4	Introduksi tugas Jurnal Kas, melakukan *setup* pada Laravel dan memnbuat skema relasi database
5	Melanjutkan tugas Jurnal Kas dengan membuat skema tabel relasi database dan mengimplementasikannnya menggunakan migrasi Laravel
6	Mempelajari cara untuk setup dan membuat rest API dan membuat tabel database
7	Melanjutkan pembuatan skema relasi tabel dan implementasi API secara perlahan untuk setiap tabel
8	Introduksi tugas proyek website Sigarda, perkenalan kelompok, mempelajari proyek Sigarda dan desain UI/UX-nya
9	Merencanakan, merancang dan membuat skema tabel database untuk proyek website forum Sigarda
10 & 11	Merevisi dan memperbarui skema tabel/model database dan mengimplementasikannya pada database dengan melakukan migrasi pada Laravel
12	*Meeting*, revisi tabel relasi database, membuat fitur *post* dengan membuat *controller* dan menimplementasikan fungsionalitasnya serta membuat API-nya
Lanjut pada halaman berikutnya	

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang (lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
13	Membuat *controller comments/replies*, membuat route dan API-nya, membuat/merancang ulang tabel untuk *Following* dan tabel komentar, membuat tabel *Likes* untuk komentar dan tabel *Upvotes* untuk postingan, membuat tabel log aktivitas pengguna, dan tabel rekomendasi forum serta menjalankan migrasi, membuat API untuk komentar dan *likes*, *controller* dan modelnya, membuat tabel untuk logs dari postingan, meembuat tabel log untuk forum dan fungsionalistasnya
14 & 15	Membuat tabel log komentar dan notifikasi serta menerapkan fungsionalitasnya, Membuat *controller* untuk notifikasi serta membuat API dan menerapkan fungsionalitas untuk fungsi komentar, melakukan perencanaan untuk pembuatan fitur rekomendasi forum dengan mendefiniskan kriteria rekomendasi
16 & 17	Merencanakan dan membuat sistem *following* untuk pengguna dengan membuat *controller* dan mengimplementasikan API-nya, melakukan *update* untuk sistem fitur *Likes* untuk komentar serta membuat *controller* untuk fitur tersebut dan mengimplementasikan fungsionalitas dan API-nya, membuat *controller* untuk sistem *upvoting* dan mengimplemtasikan API dan fungsionalitasnyaI

3.3.1 Software yang digunakan

Selama kegiatan magang yang dilakukan di PT Otak Kanan, peserta magang mengerjakan kegiatan kerja dengan perangkat keras yang dimiliki. Selain itu, dalam pengerjaan proyek Sigarda, ada pula beberapa perangkat lunak yang digunakan oleh peserta magang selama melaksanakan kegiatannya. Berikut adalah beberapa perangkat lunak yang dibutuhkan untuk mengerjakan proyek website Sigarda.

1. Github: digunakan untuk melakukan pelacakan perubahan kode untuk bagian *Front End* dan *Back End* agar dapat melihat riwayat revisi, membandingkan versi, dan kembali ke versi sebelumnya jika diperlukan.

2. Visual Studio Code: digunakan sebagai *Text Editor* untuk pengembangan website Sigarda yang menggunakan *framework* Laravel.

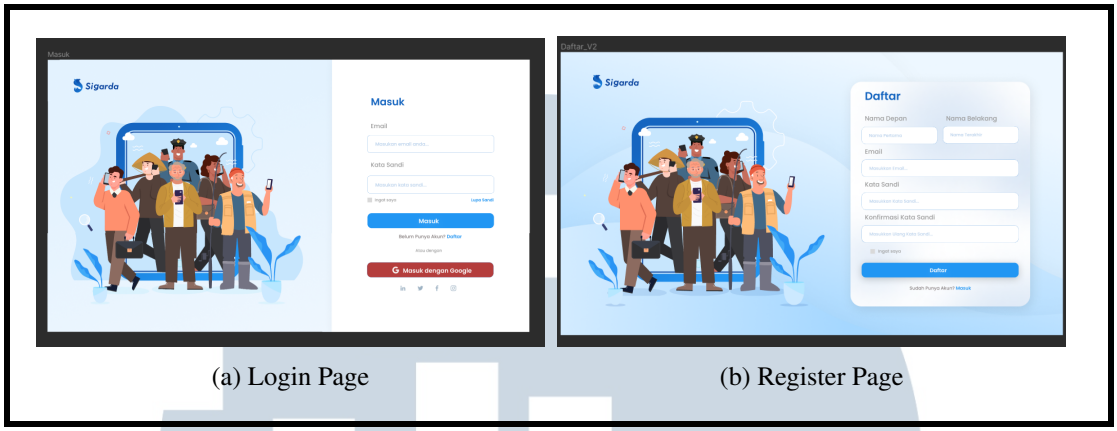
3. Figma: digunakan untuk merancang tampilan *mockup* dari website Sigarda.
4. XAMPP: sebagai penyedia server lokal termasuk web server Apache, database MySQL, dan *interpreter* PHP, yang diperlukan untuk menjalankan aplikasi Laravel.
5. Laravel: *framework back-end* yang membantu developer membangun aplikasi web yang cepat, terstruktur, aman, skalabel, dan mudah dikembangkan bersama tim.
6. Postman: digunakan untuk mengembangkan dan menguji API karena memungkinkan kita untuk mengirim permintaan seperti *GET, POST, PUT*, dan menganalisis respons dari API.

3.3.2 Merancang tampilan website Sigarda

Sebelum melakukan implementasi, dibuatlah rancangan tampilan dari website Sigarda terlebih dahulu. Rancangan ini dibuat menggunakan Figma dan kemudian dipresentasikan untuk mendapatkan saran dan arahan. Setelah rancangan tampilan disetujui, maka peserta magang mengimplementasikan rancangan tersebut melalui *front-end website*.

A. Mock-up halaman Login dan Register

Gambar 3.1 menunjukkan rancangan tampilan dari halaman *Login* dan *Register* untuk website Sigarda. Pada Gambar 3.1 bagian a ditampilkan halaman *Login* dimana pengguna dapat memasukan alamat *e-mail* dan *password* yang sudah terdaftar. Apabila pengguna belum memiliki akun, maka pengguna harus mendaftarkannya terlebih dahulu melalui halaman *Register* yang dapat dilihat pada Gambar 3.1 bagian b. Dalam halaman ini, pengguna harus memasukkan nama depan dan belakang, alamat e-mail, *password*, dan konfirmasi *password*. Apabila pengguna telah memasukkan data tersebut maka pengguna dapat mendaftarkan akun dan melakukan *Login*.

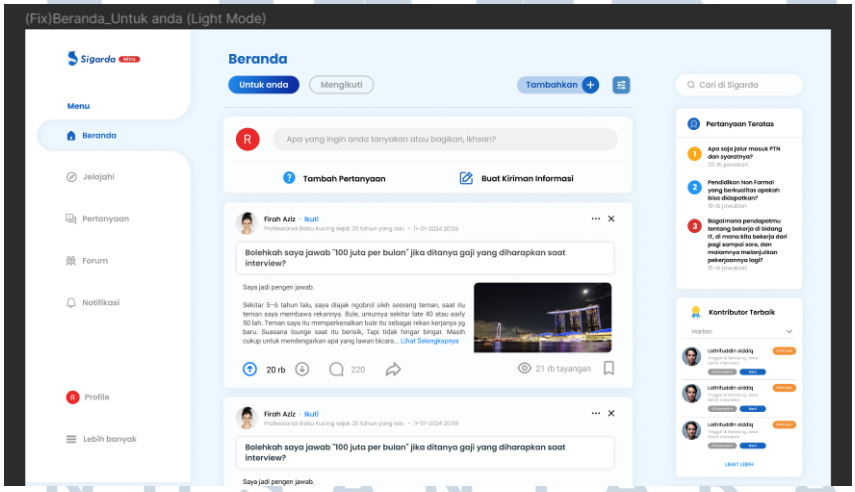


(a) Login Page (b) Register Page

Gambar 3.1. *Mockup* halaman *Login* dan *Register* Sigarda website

B. Mock-up halaman Beranda

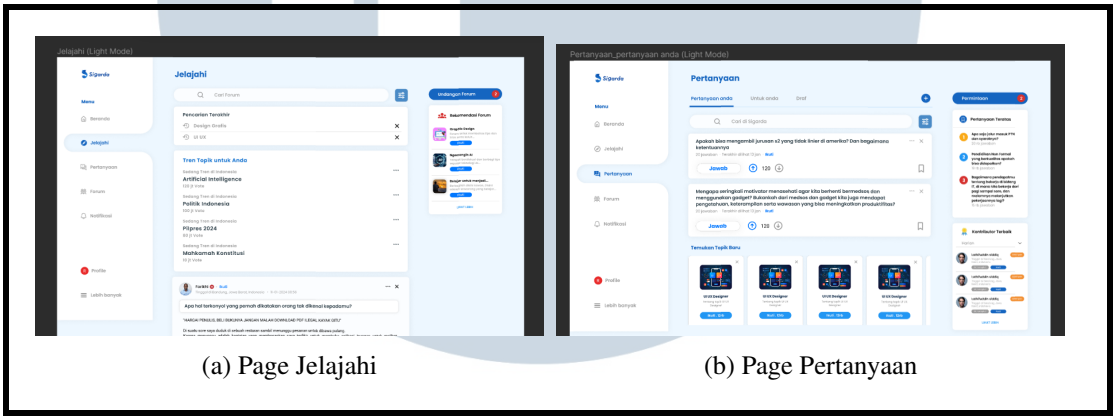
Pada Gambar 3.2 dapat dilihat rancangan untuk halaman utama beranda, dimana halaman ini merupakan halaman pertama yang dilihat oleh pengguna ketika berhasil melakukan *login*. Pada halaman ini pengguna dapat mengajukan pertanyaan dan memilih *post* mana yang ingin dilihat. Untuk pilihan ”Untuk Anda”, pengguna dapat melihat berbagai *post* yang sedang *trending* atau yang memiliki banyak *upvote*. Selain itu, pengguna juga dapat mengubahnya ke pilihan ”Mengikuti” dimana di sini pengguna dapat melihat *post* dari pengguna lain yang telah di-*follow*.



Gambar 3.2. *Mockup* halaman beranda

C. Mock-up halaman Jelajahi dan Pertanyaan

Pada halaman jelajahi yang dapat dilihat pada Gambar 3.3, pengguna dapat melakukan pencarian mengenai topik/forum apa saja yang ingin dilihat. Pengguna juga akan mendapatkan rekomendasi topik yang sedang *trending*. Sedangkan pada halaman Pertanyaan, terdapat tiga opsi yaitu ”Pertanyaan anda” dimana pengguna dapat melihat status dari pertanyaan yang telah diajukan. Selain itu ada juga ”Untuk anda” dimana pengguna dapat melihat pertanyaan-pertanyaan yang sedang populer. Yang terakhir ada ”Draf” dimana pengguna dapat melihat kembali pertanyaan-pertanyaan yang belum terkirim.

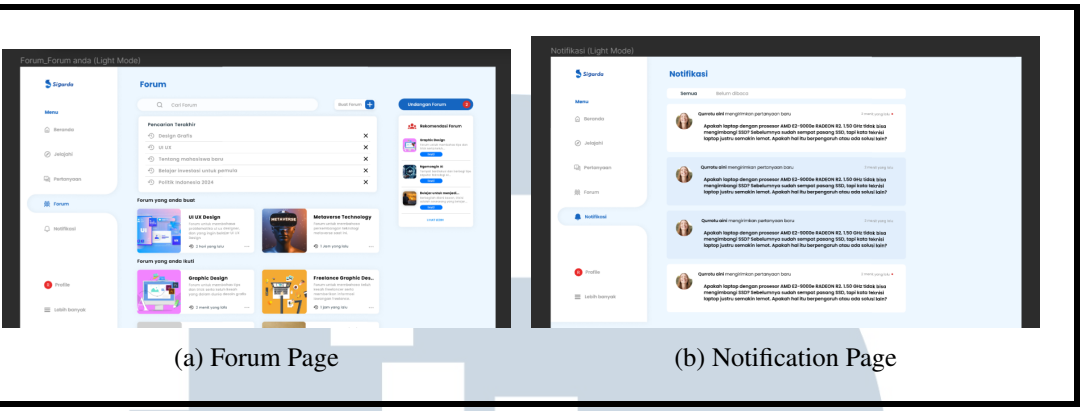


(a) Page Jelajahi (b) Page Pertanyaan

Gambar 3.3. *Mockup* halaman Jelajahi dan Pertanyaan pada Sigarda *website*

D. Mock-up page Forum dan Notifikasi

Pada halaman notifikasi yang dapat dilihat pada Gambar 3.4 bagian b, pengguna dapat melihat berbagai notifikasi yang muncul untuk akun pengguna. Notifikasi dapat berupa pertanyaan atau balasan dari pengguna lain terhadap sebuah postingan/pertanyaan yang telah di *upload*. Selain itu pengguna juga dapat mencari ataupun membuat sebuah forum pada halaman forum yang dapat dilihat pada Gambar 3.5 bagian a.

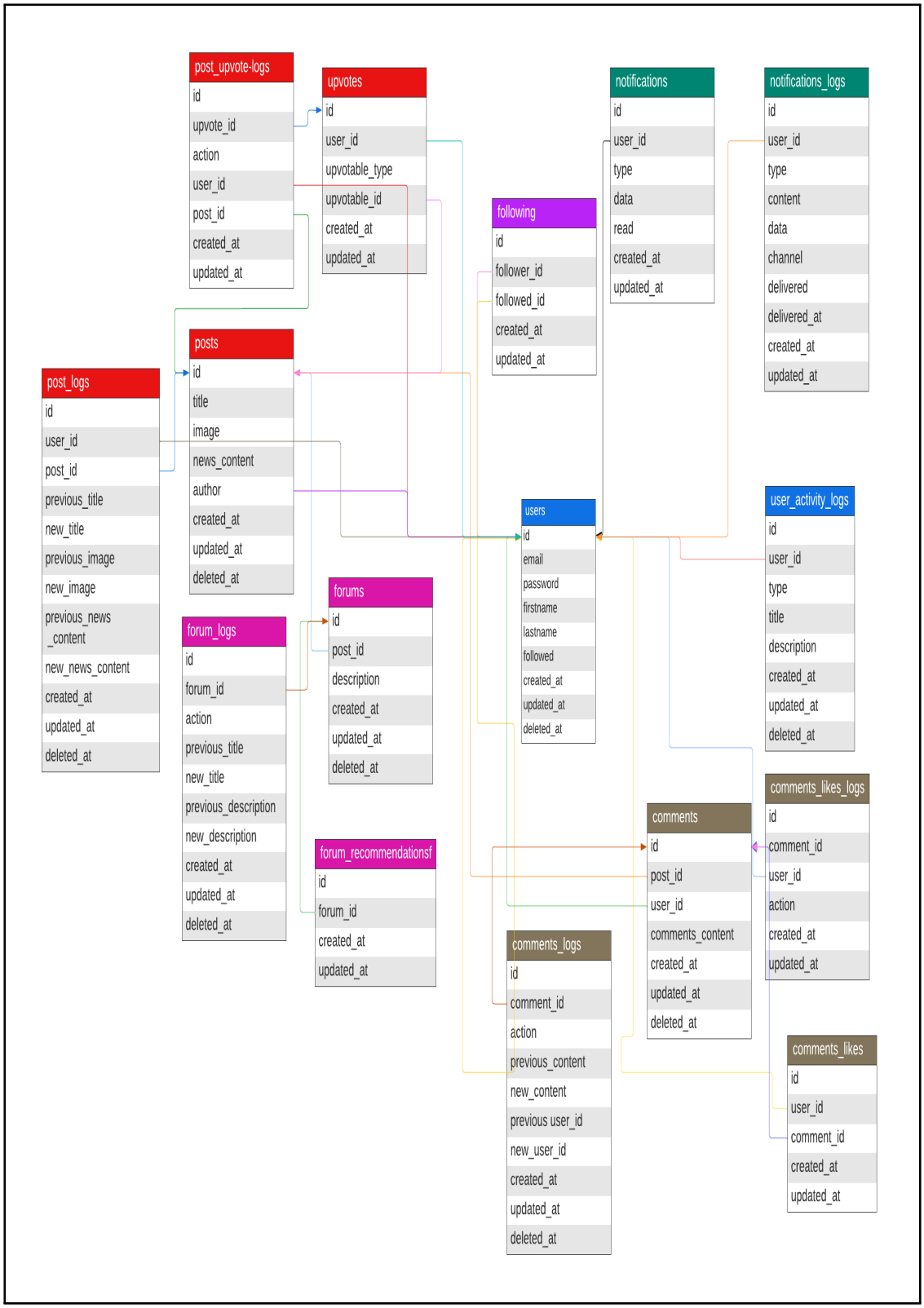


(a) Forum Page (b) Notification Page

Gambar 3.4. *Mockup* halaman Forum dan Notifikasi pada Sigarda *website*

3.3.3 Merancang Skema Tabel Relasi Database

Pada Gambar 3.5, dapat dilihat rancangan skema relasi tabel database yang digunakan untuk memberikan gambaran visual tentang struktur database, termasuk tabel, kolom, dan hubungan antar tabel. Dengan adanya skema yang jelas, maka pembuatan kode SQL akan dapat lebih mudah untuk dibuat, dipelihara, dan diakses dalam database. Selain itu, skema juga penting untuk membantu dalam memastikan kualitas data dalam database dengan mendefinisikan aturan untuk data yang dapat disimpan. Hal ini dapat membantu mencegah data yang tidak *valid* atau tidak akurat masuk ke dalam database. Dalam database ini, dibuat 16 tabel yang dibutuhkan berdasarkan ide awal, desain mockup, dan fitur-fitur dari Sigarda mulai dari tabel *user*, tabel *posts*, tabel forum, dan beberapa tabel lainnya.



Gambar 3.5. Sigarda database *relation scheme*

A. Tabel Users dan User Activity Logs

Pada Tabel 3.2, dapat dilihat tabel *users* yang menyimpan informasi penting tentang pengguna yang terdaftar dalam website Sigarda. Informasi penting pengguna dapat berupa id pengguna, email, *username*, dan *password*. Kemudian, untuk memelihara catatan aktivitas pengguna dalam Sigarda, maka dibentuklah tabel *user activity logs* yang dapat dilihat pada Tabel 3.3 untuk menyimpan data seperti tipe dan deskripsi aktivitas/perubahan, dan stempel waktu yang menunjukkan kapan aktivitas/perubahan tersebut terjadi.

Tabel 3.2. Tabel *Users*

Key (PK/FK)	Name	Type	Null	Unique	Keterangan
PK	id	bigint(20)	No		ID Pengguna
	email	varchar(255)	No	Yes	Alamat email Pengguna
	password	varchar(255)	No		Password Pengguna
	firstname	varchar(100)	No		Nama depan Pengguna
	lastname	varchar(100)	Yes		Nama belakang Pengguna
	followed	tinyint(1)	No		Status mengikuti

Tabel 3.3. Tabel *User Activity Logs*

Key (PK/FK)	Name	Type	Null	Keterangan
PK	id	bigint(20)	No	ID Aktivitas pengguna
FK	user_id	bigint(20)	No	ID Pengguna
	type	varchar(255)	No	Tipe aktivitas yang dilakukan
	title	varchar(255)	Yes	Judul aktivitas
	description	longtext	No	Deskripsi aktivitas

B. Tabel Posts, Post Logs, Upvotes, dan Post Upvote Logs

Tabel *Posts* yang dapat dilihat pada Tabel 3.4, menyimpan informasi tentang postingan yang dibuat oleh pengguna di dalam Sigarda seperti judul dan deskripsi postingan. Sedangkan pada Tabel 3.6, dapat dilihat tabel *Upvotes* yang digunakan untuk menyimpan informasi tentang *upvote* yang diberikan pada postingan oleh pengguna. Segala perubahan yang terjadi dalam tabel *Posts* dan *Upvotes* kemudian di catat dalam tabel *Post Logs*, yang dapat dilihat pada Tabel 3.5 dan tabel *Post Upvote Logs*, yang dapat dilihat pada Tabel 3.7.

Tabel 3.4. Tabel *Posts*

Key (PK/FK)	Name	Type	Null	Keterangan
PK	id	bigint(20)	No	ID postingan
	title	varchar(255)	No	Judul postingan
	image	varchar(255)	Yes	Gambar pada postingan
	news_content	text	No	Tulisan konten dalam postingan
FK	author	bigint(20)	No	Penulis postingan (merujuk kepada user id)

Tabel 3.5. Tabel *Posts Logs*

Key (PK/FK)	Name	Type	Null	Keterangan
PK	id	bigint(20)	No	ID postingan
FK	user_id	bigint(20)	Yes	ID penulis postingan yang melakukan perubahan (merujuk kepada user id)
FK	post_id	bigint(20)	No	ID postingan yang dilakukan perubahan (merujuk kepada post id)
	action	varchar(255)	No	jenis perubahan yang dilakukan
	previous_title	text	Yes	Judul sebelum diubah
	new_title	text	Yes	Judul sesudah diubah
	previous_image	varchar(255)	Yes	Gambar sebelum diubah
	new_image	varchar(255)	Yes	Gambar sesudah diubah
	previous_news_content	text	Yes	Konten tulisan postingan sebelum diubah
	new_news_content	text	Yes	Konten tulisan postingan sesudah diubah

Tabel 3.6. Tabel *Upvotes*

Key (PK/FK)	Name	Type	Null	Keterangan
PK	id	bigint(20)	No	ID upvote
FK	user_id	bigint(20)	No	ID pengguna yang melakukan upvote
	action	varchar(255)	No	jenis perubahan yang dilakukan
FK	user_id	bigint(20)	Yes	ID pengguna yang melakukan upvote (merujuk kepada user id)
FK	post_id	bigint(20)	Yes	ID postingan yang dilakukan upvote (merujuk kepada post id)

Tabel 3.7. Tabel *Post Upvote Logs*

Key (PK/FK)	Name	Type	Null	Keterangan
PK	id	bigint(20)	No	ID aktivitas upvote
FK	upvote_id	bigint(20)	No	merujuk ke ID upvote
	upvotable_type	enum('post')	No	Tipe upvote (berupa postingan)
	upvotable_id	bigint(20)	No	ID postingan yang dilakukan upvote

C. Tabel Forums, Forum Recommendations, dan Forum Logs

Tabel *Forums* yang dapat dilihat pada Tabel 3.8, menyimpan informasi tentang forum yang dibuat di dalam Sigarda. Informasi tersebut meliputi nama/judul forum, deskripsi forum dan kategorinya. Segala perubahan yang dilakukan dalam forum akan kemudian tercatat dan disimpan dalam tabel *Forum*

Logs yang dapat dilihat pada Tabel 3.10. Kemudian, pada Tabel 3.9 dapat dilihat tabel *Forum Recommendations* yang digunakan untuk menyimpan rekomendasi forum berdasarkan preferensi pengguna atau algoritma sistem.

Tabel 3.8. Tabel *Forums*

Key (PK/FK)	Name	Type	Null	Keterangan
PK	id	bigint(20)	No	ID forum
FK	post_id	bigint(20)	No	ID postingan
	title	varchar(255)	No	Judul forum
	description	text	No	Deskripsi forum

Tabel 3.9. Tabel *Forum Recommendations*

Key (PK/FK)	Name	Type	Null	Keterangan
PK	id	bigint(20)	No	ID rekomendasi forum
FK	forum_id	bigint(20)	No	ID forum

Tabel 3.10. Tabel *Forum Logs*

Key (PK/FK)	Name	Type	Null	Keterangan
PK	id	bigint(20)	No	ID aktivitas forum
FK	forum_id	bigint(20)	No	ID forum
	action	varchar(255)	No	jenis perubahan yang dilakukan
	previous_title	text	Yes	Judul sebelum diubah
	new_title	text	Yes	Judul sesudah diubah
	previous_description	text	Yes	Deskripsi sebelum diubah
	new_description	text	Yes	Deskripsi sesudah diubah
	created_at	timestamp	Yes	cap waktu forum dibuat
	updated_at	timestamp	Yes	cap waktu forum diperbarui
	deleted_at	timestamp	Yes	cap waktu forum dihapus

D. Tabel Comments, Comment Logs, Comment Likes, dan Comment Likes Logs

Pada Tabel 3.11, dapat dilihat tabel *Comments* yang menyimpan informasi tentang komentar seperti konten dari komentar tersebut yang dibuat oleh pengguna

terhadap sebuah postingan yang dari pengguna lainnya. Komentar yang telah diberikan dapat diberi *like* dari pengguna lain dan informasi seperti cap waktu yang menunjukkan kapan komentar disukai akan disimpan di tabel *Comment Likes* yang dapat dilihat pada Tabel 3.13. Segala perubahan informasi seperti pembaruan atau penghilangan deskripsi komentar dan *likes* akan tercatat dan tersimpan pada tabel *Comment Logs* dan *Comment Likes Logs* yang masing-masing dapat dilihat pada Tabel 3.12 dan Tabel 3.14,

Tabel 3.11. Tabel *Comments*

Key (PK/FK)	Name	Type	Null	Keterangan
PK	id	bigint(20)	No	ID komentar
FK	user_id	bigint(20)	No	ID penulis komentar
FK	post_id	bigint(20)	No	ID postingan yang dikomentar
	comments_content	text	No	Isi komentar

Tabel 3.12. Tabel *Comment Logs*

Key (PK/FK)	Name	Type	Null	Keterangan
PK	id	bigint(20)	No	ID aktivitas komentar
FK	comment_id	bigint(20)	No	ID komentar
	action	varchar(255)	No	jenis perubahan yang dilakukan
	previous_content	text	Yes	Isi komentar sebelum diubah
	new_content	text	Yes	Isi komentar sesudah diubah
	previous_user_id	bigint(20)	Yes	ID pengguna sebelum komentar diubah
	new__user_id	bigint(20)	Yes	ID pengguna sesudah komentar diubah
	created_at	timestamp	Yes	cap waktu komentar dibuat
	updated_at	timestamp	Yes	cap waktu komentar diperbarui
	deleted_at	timestamp	Yes	cap waktu komentar dihapus

Tabel 3.13. Tabel *Comment Likes*

Key (PK/FK)	Name	Type	Null	Keterangan
PK	id	bigint(20)	No	ID like pada komentar
FK	user_id	bigint(20)	No	ID penulis komentar
FK	post_id	bigint(20)	No	ID postingan yang dikomentar

Tabel 3.14. Tabel *Comment Likes Logs*

Key (PK/FK)	Name	Type	Null	Keterangan
PK	id	bigint(20)	No	ID aktivitas like
FK	user_id	bigint(20)	No	ID penulis komentar
FK	comment_id	bigint(20)	No	ID komentar
	action	varchar(255)	No	jenis perubahan yang dilakukan
	created_at	timestamp	Yes	cap waktu like dibuat
	deleted_at	timestamp	Yes	cap waktu like dihapus

E. Tabel Following

Pada Tabel 3.15, dapat dilihat tabel *Following* yang digunakan untuk menyimpan informasi tentang hubungan ”mengikuti” antar pengguna. Ini dapat membantu memfasilitasi fitur seperti umpan berita yang dipersonalisasi, rekomendasi pengguna, dan notifikasi. Tabel ini menyimpan data seperti *follower* id yang mereferensikan id pengguna yang mengikuti (pengguna yang memulai hubungan mengikuti) dan *following id* yang mereferensikan id pengguna yang diikuti (pengguna yang diikuti oleh *follower* id).

Tabel 3.15. Tabel *Following*

Key (PK/FK)	Name	Type	Null	Keterangan
PK	id	bigint(20)	No	ID following
FK	follower_id	bigint(20)	No	ID pengguna yang mengikuti
FK	followed_id	bigint(20)	No	ID pengguna yang diikuti
	created_at	timestamp	Yes	cap waktu dilakukan pengikutan
	deleted_at	timestamp	Yes	cap waktu pengikutan dihapus

F. Tabel Notification dan Notification Logs

Pada Tabel 3.16 dapat dilihat tabel *Notifications* yang berfungsi untuk menyimpan informasi tentang pemberitahuan yang dibuat di dalam website Sigarda. Informasi tersebut dapat berupa jenis pemberitahuan (seperti postingan baru, komentar baru, peringatan sistem), osi utama pesan pemberitahuan, dan cap waktu yang menunjukkan kapan pemberitahuan dibuat dan telah dibaca. Segala informasi tindakan/perubahan yang dilakukan pada notifikasi seperti jenis tindakan dan cap

waktu yang menunjukkan kapan tindakan dilakukan akan tercatat dan tersimpan dalam tabel *Notification Logs* yang dapat dilihat pada Tabel 3.17.

Tabel 3.16. Tabel *Notifications*

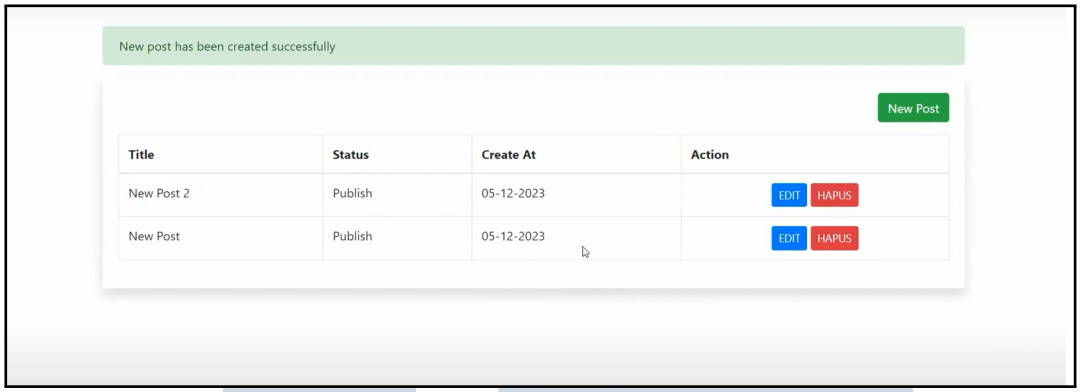
Key (PK/FK)	Name	Type	Null	Keterangan
PK	id	bigint(20)	No	ID notifikasi
FK	user_id	bigint(20)	No	ID pengguna
	type	enum('user_followed', dll.	No	Tipe notifikasi
	data	longtext	No	Data notifikasi
	read	tinyint(1)	No	Status dibaca

Tabel 3.17. Tabel *Notification Logs*

Key (PK/FK)	Name	Type	Null	Keterangan
PK	id	bigint(20)	No	ID aktivitas notifikasi
FK	user_id	bigint(20)	No	ID pengguna
	type	enum('email', dll.	No	Data notifikasi
	content	text	Yes	Isi notifikasi
	data	text	Yes	Data khusus notifikasi tambahan
	channel	varchar(255)	Yes	Saluran pengiriman (misalnya, alamat email, ID perangkat).
	delivered	tinyint(1)	No	Menunjukkan apakah notifikasi terkirim
	delivered_at	timestamp	Yes	cmencatat waktu pengiriman

3.3.4 Mempelajari dan Memahami framework Laravel

Setelah melakukan rancangan skema database yang diperlukan, selanjutnya yang diperlukan adalah mengimplementasikannya. Untuk proyek ini, Laravel digunakan sebagai *framework* untuk bagian *back-end*. Sebelum dapat menggunakan Laravel, peserta magang harus mempelajari dan memahami cara penggunaan Laravel untuk melakukan migrasi, membuat *controller* dan model, dan membuat *route* untuk API. Sebagai awal, peserta magang juga diarahkan untuk membuat aplikasi CRUD sederhana dan mempresentasikannya. Gambar 3.6 merupakan contoh dari aplikasi CRUD sederhana yang telah dibuat. Dalam aplikasi CRUD sederhana ini, pengguna dapat membuat atau menulis sebuah *title* yang nantinya akan dibaca. *Title* tersebut kemudian dapat dihapus dan diperbarui sesuai keiinginan pengguna.



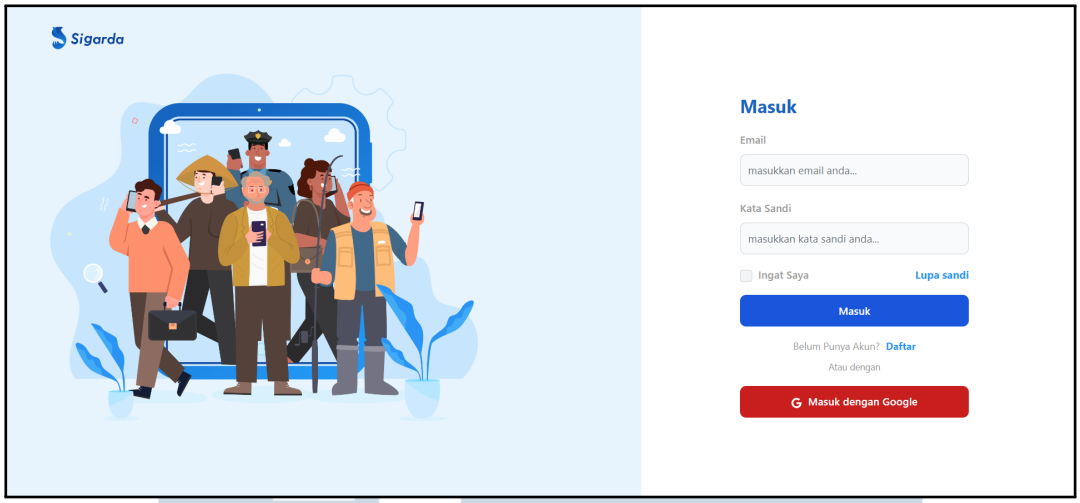
Gambar 3.6. Contoh aplikasi CRUD sederhana

3.3.5 Implementasi

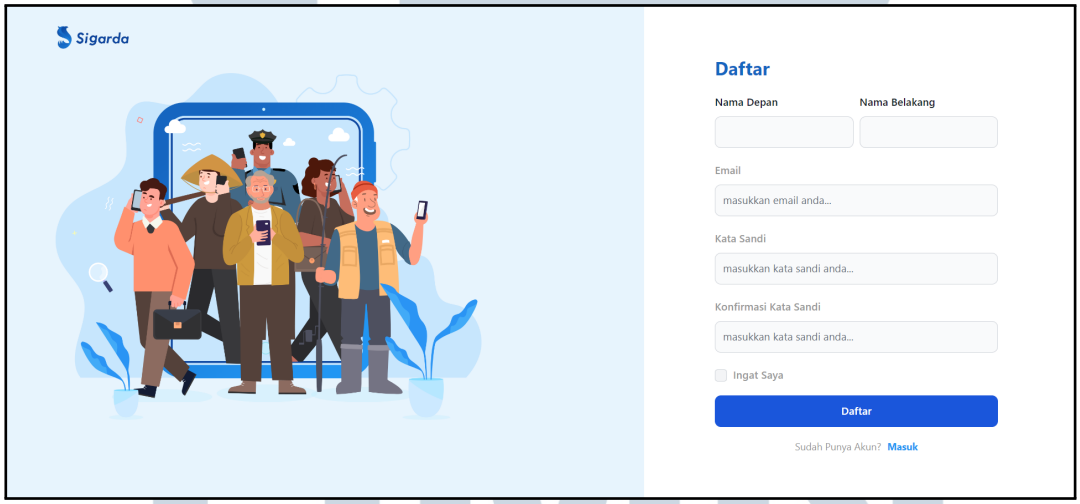
Pada bagian ini, dilakukan implementasi terhadap *front-end* dan *back-end* website berdasarkan rancangan tampilan dan skema relasi database yang telah dibuat. Dalam melakukan implementasi, untuk bagian *front-end* dilakukan menggunakan library ReactJS dan untuk bagian *back-end* dilakukan menggunakan *framework* Laravel.

A. Login dan Register

Pada Gambar 3.7 dan Gambar 3.8 dapat dilihat halaman *login* dan *register* yang merupakan halaman pertama yang dilihat oleh pengguna apabila ingin memasuki website Sigarda. Halaman *login* dan *register* dibuat menggunakan JSX dan untuk halaman *login*, digunakan form dengan kolom input email dan *password*. Ini juga mencakup kotak centang untuk mengingat pengguna, tautan untuk *password* yang terlupa, dan tombol untuk masuk dan mendaftar. Selain itu, halaman *register* juga menggunakan sebuah formulir dengan bidang input untuk nama depan, nama belakang, email, *password*, dan konfirmasi *password* serta kotak centang untuk *"remember user"* dan tombol untuk mengirimkan formulir tersebut.



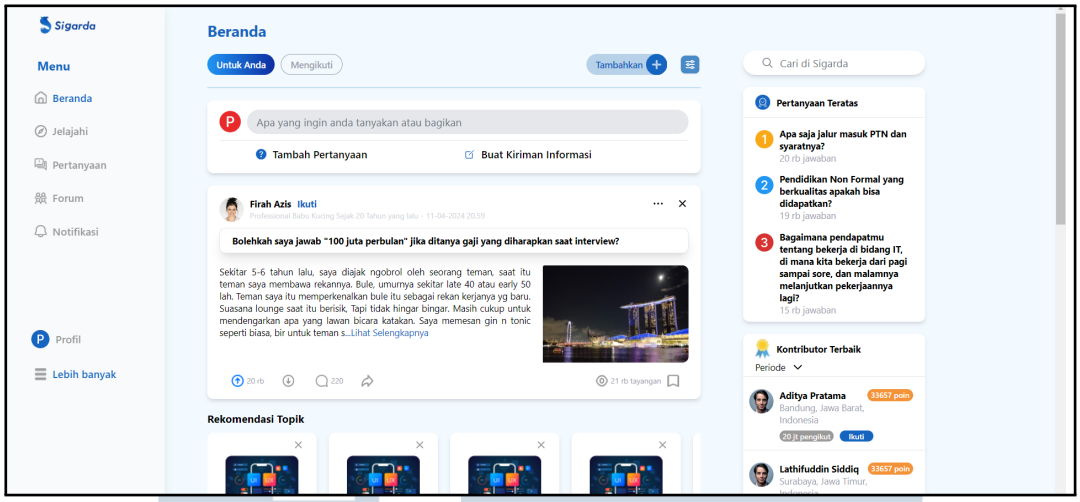
Gambar 3.7. Halaman *login*



Gambar 3.8. Halaman *register*

B. Beranda

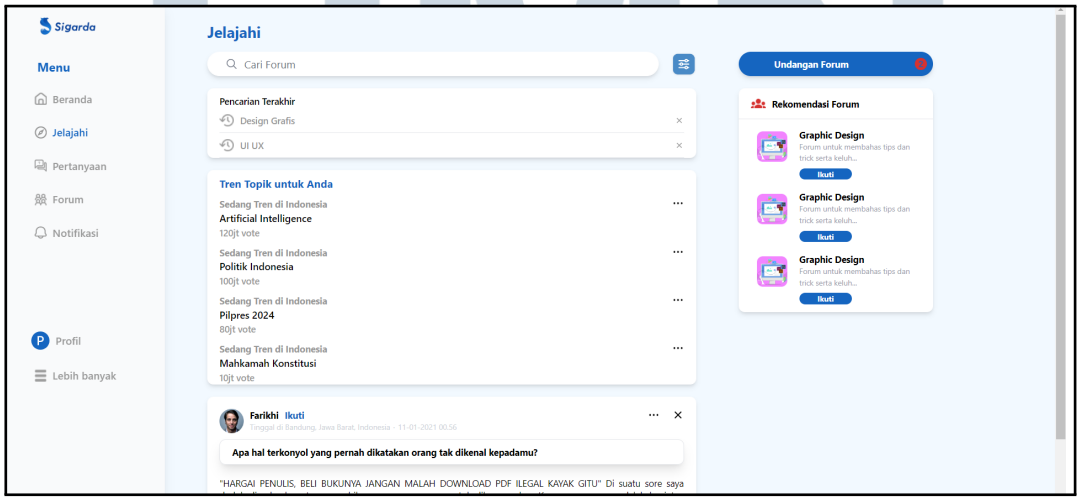
Halaman ini merupakan halaman pertama yang dilihat oleh pengguna ketika berhasil melakukan login. Pada halaman ini dapat dilihat berbagai postingan yang sedang populer maupun postingan dari pengguna yang telah diikuti. Untuk *dashboard*, dilakukan *filtering* pada array *dummyCard* berdasarkan properti status. Hal ini dilakukan untuk mengetahui status ”Mengikuti” dan ”Ikuti”. Selain itu, komponen ini juga menampilkan struktur UI yang terdiri dari *SidebarLayout, NavbarPage, HeaderMobile, ButtonHandlerFollow, Button*, dan beberapa elemen lainnya. Halaman beranda dapat dilihat pada Gambar 3.9.



Gambar 3.9. Halaman beranda

C. Jelajahi

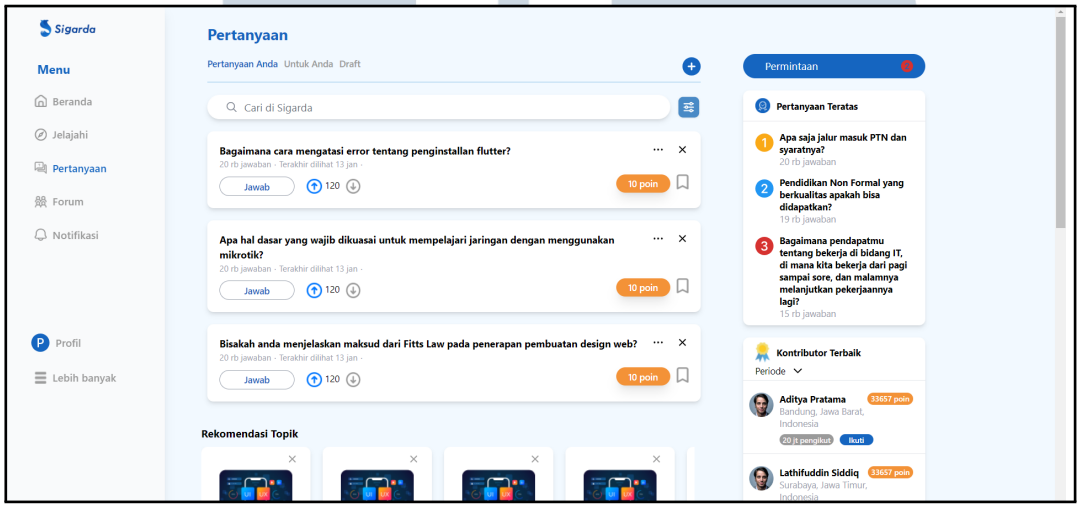
Pada Gambar 3.10, dapat dilihat halaman pertanyaan yang merupakan halaman dimana pengguna dapat melakukan pencarian mengenai topik atau forum yang ingin dicari. Di dalam area konten utama, terdapat sebuah bagian *header* dengan komponen *NavbarPage* dan sebuah *input field* pencarian. *Input field* pencarian diberi border abu-abu dan diberi ikon pencarian. Selain itu, ada juga sebuah tombol yang berisi gambar *filter* untuk melakukan penyaringan topik secara spesifik.



Gambar 3.10. Halaman jelajahi

D. Pertanyaan

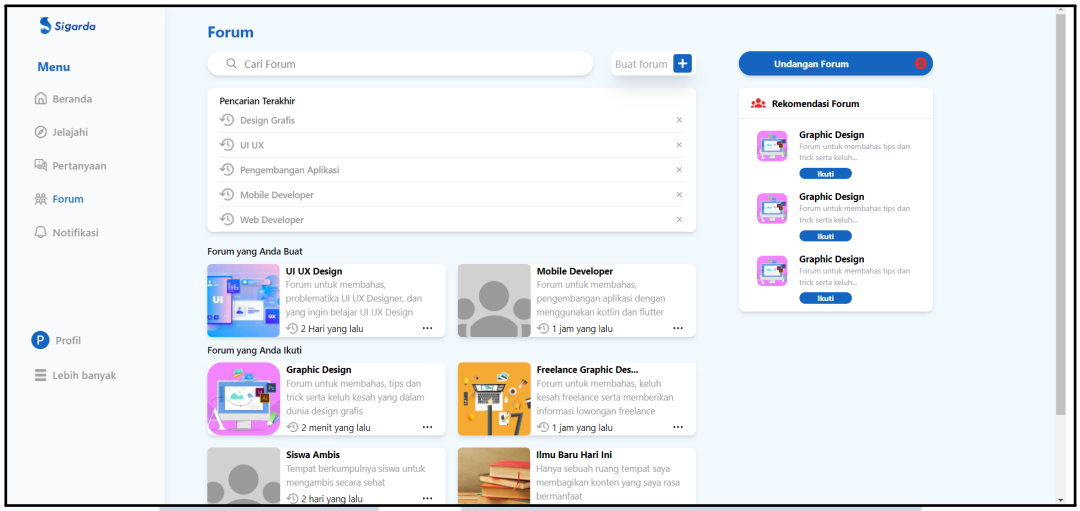
Halaman ini merupakan halaman dimana pengguna dapat mengajukan dan melihat status sebuah pertanyaan mengenai sebuah topik dalam forum. Dalam halaman ini ada *navbar* dengan pilihan ”Pertanyaan anda”, ”Untuk anda”, dan ”Draf” dimana pengguna dapat melihat pertanyaan yang telah diajukan, pertanyaan yang diajukan pengguna lainnya, dan draf untuk jawaban dari pertanyaan yang diajukan. Halaman pertanyaan dapat dilihat pada Gambar 3.11.



Gambar 3.11. Halaman pertanyaan

E. Forum

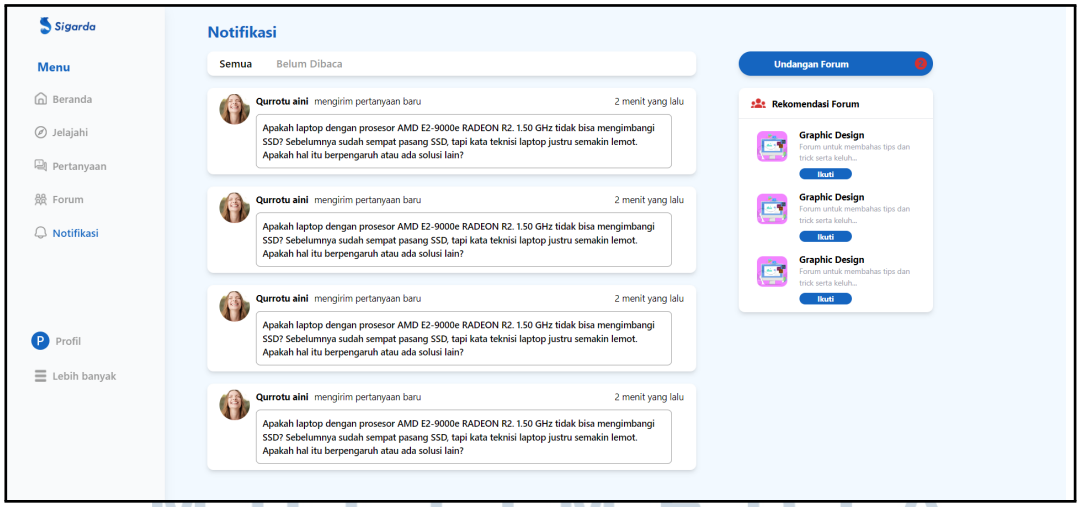
Pada Gambar 3.12, dapat dilihat halaman forum yang dibungkus dalam sebuah komponen *SidebarLayout* dan berisi berbagai elemen seperti *navbar*, input pencarian, tombol untuk membuat forum baru, bagian-bagian untuk menampilkan pencarian terakhir, forum yang dibuat oleh pengguna, dan forum yang diikuti oleh pengguna. Forum-forum tersebut ditampilkan menggunakan komponen *CardForum*, yang menerima properti seperti gambar, judul, deskripsi, dan tanggal.



Gambar 3.12. Halaman forum

F. Notifikasi

Pada Gambar 3.13, dapat dilihat halaman notifikasi yang merupakan halaman dimana pengguna dapat melihat notifikasi yang didapat. Pada halaman ini, digunakan komponen *CardNotifikasi* untuk menampilkan kartu notifikasi dengan gambar, nama, deskripsi, tanggal, dan pertanyaan.

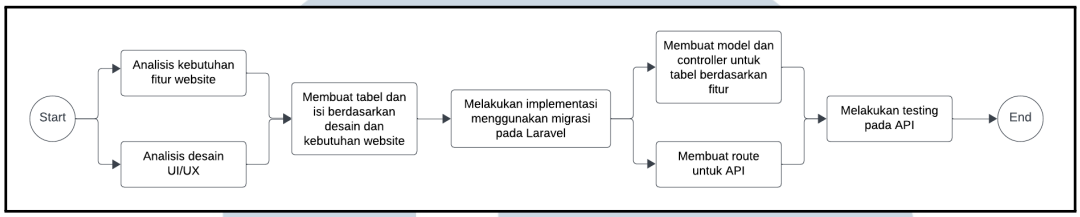


Gambar 3.13. Halaman notifikasi

G. Database dan API

Setelah skema relasi database telah dibuat, tugas *back-end* selanjutnya adalah untuk mengimplementasikannya. Pada Gambar 3.14, dapat dilihat tahapan

proses yang dilakukan untuk membuat *back-end* dari website Sigarda mulai dari merancang skema relasi tabel database hingga melakukan implementasi dan *testing* pada API.

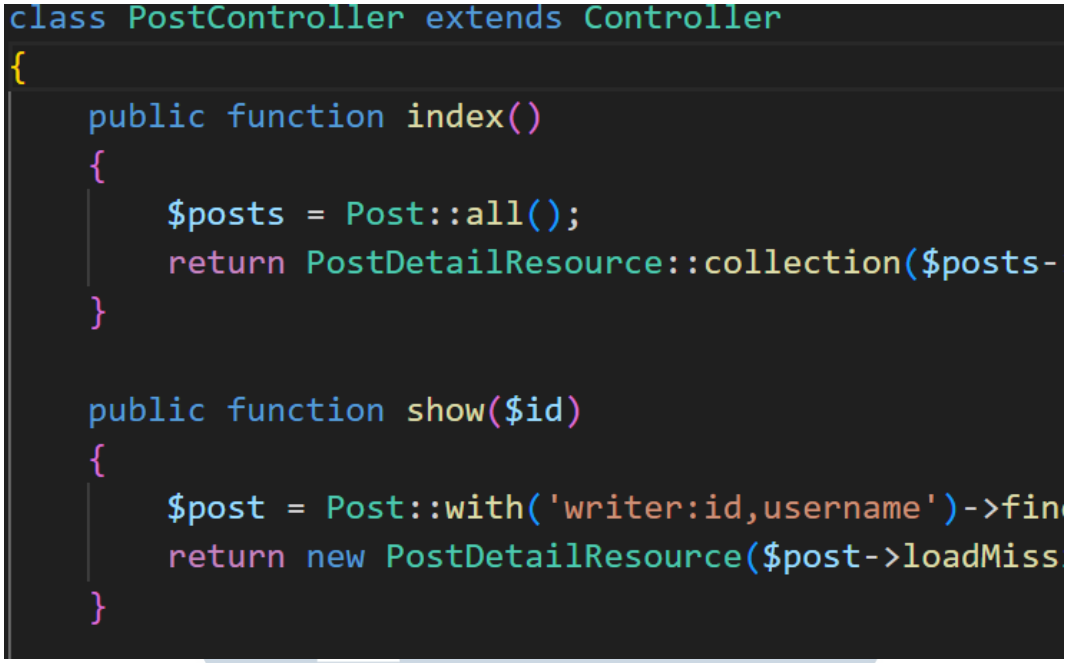


Gambar 3.14. Tahapan pembuatan *back-end* website Sigarda

Implementasi dilakukan menggunakan migrasi pada Laravel, dimana penggunaan fitur migrasi ini akan membantu untuk mengelola perubahan struktur database dengan mudah dan terorganisir. Dengan adanya fitur migrasi pembuatan, pengubahan, dan penghapusan yang dilakukan terhadap tabel database dapat dilakukan secara terkontrol, serta riwayat perubahan tersebut dapat dilacak. Migrasi dilakukan untuk penciptaan setiap tabel yang ada pada database berdasarkan skema yang telah dibuat.

Selanjutnya, setelah setiap tabel telah dibuat di dalam database, maka langkah selanjutnya adalah untuk membuat model yang digunakan untuk mewakili representasi data dalam aplikasi Laravel. Model terhubung dengan database dan menyediakan sebuah abstraksi untuk berinteraksi dengan data. Pembuatan model dilakukan untuk setiap data yang akan direpresentasikan dalam Laravel. Hal ini dilakukan dengan mendefinisikan struktur tabel menggunakan kelas *Schema* dalam *file* model.

Kemudian, sebagai penghubung antara *request* API dan model dibuatlah sebuah *controller* yang dapat menerima *request* dari API, memprosesnya, dan kemudian mengembalikan respons ke klien. Sebagai contoh, pada Gambar 3.15 dapat dilihat *PostController* menangani request API yang terkait dengan postingan. Pada bagian ini, *method index()* mengambil semua postingan dari database dan mengembalikannya dalam format JSON. Selain itu *method show()* mengambil postingan dengan ID dan *username* tertentu dan mengembalikannya dalam format JSON.



Gambar 3.15. Contoh *controller*

Setelah dibuatnya *controller*, dibuatlah *route* untuk mendefinisikan pemetaan antara URL dan *controller*. *Route* menentukan *controller* mana yang harus dipanggil ketika klien mengakses URL tertentu. Selain itu, *Route* mendefinisikan URL yang dapat diakses oleh klien untuk menggunakan API dan menentukan *method* HTTP yang diizinkan untuk setiap URL, seperti *GET, POST, PUT,* dan *DELETE*. Sebagai contoh, pada Gambar 3.16 dapat dilihat **route** tersebut mendefinisikan dua end-point yaitu "*/users*" yang diakses dengan *method* HTTP *GET* dan memanggil *method index()* pada *UserController*, dan "*/users/id*" yang diakses dengan *method* HTTP *GET* dan memanggil *method show()* pada *UserController*. *Method* ini akan mengambil pengguna dengan ID yang ditentukan dalam parameter id dan mengembalikannya dalam format JSON.

```
Route::get('/users', [UserController::class, 'index']);
Route::get('/users/{id}', [UserController::class, 'show']);
```

Gambar 3.16. Contoh *route*

Dalam penggunaannya, model, *controller*, dan *route* bekerja sama untuk membangun API yang kuat dan terstruktur dalam Laravel. Model mengelola data, *controller* memproses data dan request API, sedangkan *route* menghubungkan URL dengan *controller*.

3.4 Kendala dan Solusi yang Ditemukan

Dalam pelaksanaan kerja magang yang dilakukan di PT Otak Kanan ada beberapa kendala yang dialami, berikut merupakan beberapa kendala tersebut.

1. Kurangnya pemahaman peserta magang mengenai bagian *back-end* website termasuk cara penggunaan *framework* Laravel dan pembuatan API.
2. Kurangnya anggota tim di bagian *back-end* sehingga ada keterbatasan dalam penanganan tugas dan keahlian dimana adanya kolaborasi atau pertukaran ide dan informasi tidak memungkinkan.

Berdasarkan kendala-kendala tersebut, berikut merupakan solusi yang dapat dilakukan untuk mengatasinya.

1. Karena adanya pemahaman yang kurang mengenai *framework* Laravel, maka peserta magang harus melakukan pembelajaran dan eksplorasi mengenai *framework* tersebut dan *back-end* website secara mandiri melalui berbagai sarana seperti video tutorial ataupun dokumentasi.
2. Karena kekurangannya anggota tim, maka peserta magang harus dapat melakukan tugas secara mandiri dan melakukan diskusi dengan *project leader* untuk bertukar informasi dan ide.