

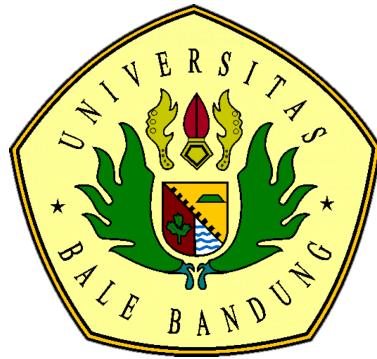
**PENGEMBANGAN APLIKASI FORUM TANYA JAWAB
TUGAS SEKOLAH (YUKITANYA) MENGGUNAKAN
FRAMEWORK NEXT.JS DAN POSTGRESQL**

SKRIPSI

**Karya Tulis sebagai Syarat untuk Memperoleh
Gelar Sarjana Komputer dari Fakultas Teknologi Informasi
Universitas Bale Bandung**

Disusun oleh :

**TAUFIK HIDAYAT
NPM. 301200032**



**PROGRAM STRATA 1
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG
BANDUNG
2024**

LEMBAR PERSETUJUAN PEMBIMBING

PENGEMBANGAN APLIKASI FORUM TANYA JAWAB TUGAS SEKOLAH
(YUKITANYA) MENGGUNAKAN FRAMEWORK NEXT.JS DAN
POSTGRESQL

Disusun oleh :

TAUFIK HIDAYAT
NPM. 301200032

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar
SARJANA KOMPUTER

Pada

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG

Baleendah, Juli 2024

Disetujui oleh :

Pembimbing Utama

Pembimbing Pendamping

Yusuf Muharam, S.Kom., M.Kom. Mohammad Bayu Anggara, S.Kom., M.Kom.
NIK. 04104820003 NIK. 04104823002

LEMBAR PERSETUJUAN PENGUJI

PENGEMBANGAN APLIKASI FORUM TANYA JAWAB TUGAS SEKOLAH
(YUKITANYA) MENGGUNAKAN FRAMEWORK NEXT.JS DAN
POSTGRESQL

Disusun oleh :

TAUFIK HIDAYAT
NPM. 301200032

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar
SARJANA KOMPUTER

Pada

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG

Baleendah, Juli 2024

Disetujui oleh :

Pengaji 1

Pengaji 2

Yudi Herdiana, S.T., M.T.
NIK. 04104808008

Sutiyono, S.T., M.Kom.
NIK. 01043180002

LEMBAR PENGESAHAN PROGRAM STUDI

PENGEMBANGAN APLIKASI FORUM TANYA JAWAB TUGAS SEKOLAH
(YUKITANYA) MENGGUNAKAN FRAMEWORK NEXT.JS DAN
POSTGRESQL

Disusun oleh :

TAUFIK HIDAYAT
NPM. 301200032

SKRIPSI ini telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar
SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Juli 2024

Mengetahui,
Dekan,

Mengesahkan,
Ketua Program Studi,

Yudi Herdiana, S.T., M.T. Yusuf Muhamam, S.Kom., M.Kom.
NIK. 04104808008 NIK. 04104820003

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini, saya:

Nama : Taufik Hidayat
NPM : 301200032
Jurusan : Teknik Informatika
Fakultas : Teknologi Informasi
Judul : Pengembangan Aplikasi Forum Tanya Jawab
Tugas Sekolah (Yukitanya) Menggunakan
Framework Next.js dan PostgreSQL

Dengan ini saya menyatakan bahwa skripsi ini benar-benar karya saya sendiri. Sepengetahuan saya tidak terdapat karya yang ditulis atau diterbitkan orang lain, kecuali sebagai acuan atau kutipan dengan mengikuti tata penulisan karya ilmiah yang lazim.

Bandung, Juli 2024

Taufik Hidayat
NPM. 301200032

ABSTRAK

Penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi forum tanya jawab tugas sekolah Yukitanya, menggunakan *framework* Next.js dan *database* PostgreSQL. Yukitanya adalah *platform* yang menghubungkan pengguna dalam forum diskusi untuk menyelesaikan tugas sekolah bersama. Penelitian ini dilaksanakan di PT. Forest Interactive selama program MSIB 5.

Berdasarkan pengamatan dan survei, ditemukan beberapa masalah pada Yukitanya, seperti banyaknya pertanyaan yang mengandung kata-kata tidak sopan, kurangnya perlindungan terhadap serangan spam, serta rendahnya minat pengguna. Oleh karena itu, penelitian ini bertujuan mengembangkan fitur *filter* kata terlarang, mengimplementasikan metode CAPTCHA untuk mencegah *spam*, dan menambahkan fitur-fitur menarik guna meningkatkan minat pengguna. Pendekatan gabungan kualitatif dan kuantitatif digunakan dalam penelitian ini, dengan model pengembangan *Agile* untuk mengembangkan aplikasi. Algoritma Knuth-Morris-Pratt diterapkan pada fitur *filter* kata terlarang untuk mendeteksi kata-kata tidak sopan dalam pertanyaan dan jawaban. Algoritma Cloudflare Turnstile digunakan sebagai metode CAPTCHA untuk mencegah serangan *spam* oleh bot. Selain itu, Google Gemini API diintegrasikan untuk menjawab pertanyaan secara otomatis menggunakan AI.

Hasil pengujian pengguna menunjukkan capaian untuk fitur autentikasi 86%, halaman beranda, fitur CAPTCHA, pengelolaan pertanyaan, fitur tanyakan pada AI, pengelolaan jawaban, dan halaman profil sebesar 100%, *filtering* kata terlarang 75%, fitur pencarian 89%, dan fitur notifikasi 89%, dengan rata-rata 93,9%. Secara keseluruhan, hasil pengujian menunjukkan bahwa aplikasi berfungsi dengan baik, tetapi masih perlu perbaikan pada fitur *filtering* kata terlarang. Penelitian ini diharapkan menghasilkan aplikasi forum tanya jawab yang membantu siswa menyelesaikan tugas sekolah dengan cara yang lebih menarik dan tidak membosankan dengan memanfaatkan teknologi terkini.

Kata kunci: Algoritma Knuth-Morris-Pratt, Aplikasi Forum Tanya Jawab, Google Gemini API, Metode CAPTCHA Turnstile

ABSTRACT

This research aims to design and develop Yukitanya school assignment question and answer forum application, using Next.js framework and PostgreSQL database. Yukitanya is a platform that connects users in a discussion forum to complete school assignments together. This research was conducted at PT Forest Interactive during the MSIB 5 program.

Based on observations and surveys, several problems were found in Yukitanya, such as the number of questions containing impolite words, lack of protection against spam attacks, and low user interest. Therefore, this research aims to develop a forbidden word filter feature, implement a CAPTCHA method to prevent spam, and add interesting features to increase user interest. A combined qualitative and quantitative approach was used in this research, with an Agile development model to develop the application. The Knuth-Morris-Pratt algorithm is applied to the forbidden word filter feature to detect impolite words in questions and answers. Cloudflare Turnstile algorithm is used as a CAPTCHA method to prevent spam attacks by bots. In addition, Google Gemini API is integrated to automatically answer questions using AI.

User test results show achievements for the authentication feature of 86%, home page, CAPTCHA feature, question management, ask AI feature, answer management, and profile page of 100%, forbidden word filtering of 75%, search feature of 89%, and notification feature of 89%, with an average of 93.9%. Overall, the test results show that the application works well, but still needs improvement on the forbidden word filtering feature. This research is expected to produce a question and answer forum application that helps students complete school assignments in a more interesting and less boring way by utilizing the latest technology.

Keywords: *Google Gemini API, Knuth-Morris-Pratt Algorithm, Q&A Forum Application, Turnstile CAPTCHA Method.*

KATA PENGANTAR

Puji dan Syukur kami panjatkan ke Hadirat Allah SWT, karena berkat limpahan Rahmat dan Karunia-Nya sehingga penulis dapat menyelesaikan laporan skripsi ini dengan baik dan tepat pada waktunya. Skripsi ini merupakan salah satu syarat untuk menyelesaikan program studi Teknik Informatika jenjang Strata-1 Universitas Bale Bandung. Dalam skripsi ini membahas mengenai Pengembangan Aplikasi Forum Tanya Jawab Tugas Sekolah (Yukitanya) Menggunakan Framework Next.js dan PostgreSQL.

Laporan skripsi ini dapat dibuat dengan bantuan dari berbagai pihak untuk menyelesaikan tantangan dan hambatan selama pengeraannya. Oleh karena itu, pada kesempatan kali ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT yang telah memberikan karunia-Nya selama proses pengeraaan laporan skripsi ini.
2. Kedua Orang Tua yaitu Bapak Maman Rohman dan Ibu Iis Ismatu Sholihat serta saudara kandung saya yang telah memberikan doa dan dukungan selama proses pembuatan laporan skripsi ini.
3. Bapak Yudi Herdiana, S.T., M.T. selaku Dekan Fakultas Teknologi Informasi Universitas Bale Bandung dan dosen penguji 1.
4. Bapak Yusuf Muharam, S.Kom., M.Kom. selaku Ketua Program Studi Teknik Informatika Universitas Bale Bandung dan dosen pembimbing utama.
5. Bapak Mohammad Bayu A., S.Kom., M.Kom. selaku dosen pembimbing pendamping.
6. Bapak Sutiyono, S.T., M.Kom. selaku dosen penguji 2
7. Seluruh Dosen beserta staf Fakultas Teknologi Informasi Universitas Bale Bandung.
8. Rekan-rekan FTI angkatan 2020 yang senantiasa saling membantu dan memberikan semangat dalam proses penelitian maupun penulisan laporan.

9. Semua pihak yang memberikan dukungan dan bantuannya kepada penulis untuk menyelesaikan skripsi ini namun tidak bisa disebutkan satu persatu.

Penulis menyadari bahwa masih banyak kekurangan dalam penyusunan skripsi ini, namun penulis telah melakukan semua yang diperlukan dengan yang terbaik. Oleh karena itu, segala kritik dan saran yang membangun akan penulis terima dengan senang hati. Akhir kata, semoga laporan skripsi ini dapat diterima dan bermanfaat bagi kita semua.

Bandung, Juli 2024

Penulis

DAFTAR ISI

ABSTRAK.....	vi
ABSTRACT.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xvi
DAFTAR LAMPIRAN.....	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Metodologi Penelitian.....	4
1.5.1 Metode Pengumpulan Data.....	4
1.5.2 Metode Pengembangan Sistem.....	5
1.6 Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA.....	8
2.1 Landasan Teori.....	8
2.2 Dasar Teori.....	13
2.2.1 <i>Web Application</i>	13
2.2.2 Forum Diskusi.....	15
2.2.3 <i>Full Stack Development</i>	15
2.2.4 <i>Relational Database Management System (RDBMS)</i>	17
2.2.5 <i>Object Relational Mapping (ORM)</i>	19
2.2.6 <i>Application Programming Interface (API)</i>	20

2.2.7 Metode <i>Agile</i>	22
2.2.8 Algoritma Knuth-Morris-Pratt (KMP).....	23
2.2.9 Metode CAPTCHA.....	24
2.2.10 Metode pengujian <i>Black Box</i>	25
2.2.11 Metode pengujian penelitian.....	26
BAB III METODOLOGI.....	27
3.1 Kerangka Pikir.....	27
3.2 Deskripsi.....	27
3.2.1 Identifikasi Masalah.....	28
3.2.2 Pengumpulan Data.....	28
3.2.3 Pengembangan Sistem.....	29
3.2.4 Pelaporan.....	33
BAB IV ANALISIS DAN PERANCANGAN.....	34
4.1 Analisis.....	34
4.1.1 Analisis Masalah.....	34
4.1.2 Analisis <i>Software</i>	34
4.1.3 Analisis Pengguna.....	35
4.1.4 <i>User Interface</i>	35
4.1.5 Fitur-fitur.....	36
4.1.6 Analisis Data.....	37
4.1.7 Analisis Biaya.....	38
4.2 Perancangan.....	38
4.2.1 UML (<i>Unified Modelling Language</i>).....	39
4.2.2 Struktur Tabel.....	54
4.2.3 Desain Antarmuka.....	60
BAB V IMPLEMENTASI DAN PENGUJIAN.....	67
5.1. Implementasi.....	67

5.1.1 Listing Program.....	67
5.1.2 Implementasi Sistem.....	80
5.1.3 Spesifikasi Sistem.....	81
5.1.4 Instalasi Sistem.....	82
5.1.5 Menjalankan Sistem.....	82
5.2 Pengujian.....	93
BAB VI KESIMPULAN.....	102
6.1 Kesimpulan.....	102
6.2 Saran.....	103
DAFTAR PUSTAKA.....	104
LAMPIRAN.....	109
RIWAYAT HIDUP.....	130

DAFTAR GAMBAR

Gambar 2.1 Arsitektur <i>Client-side</i> dan <i>Server-side</i>	14
Gambar 2.2 Arsitektur <i>Full Stack Development</i>	16
Gambar 2.3 Arsitektur RDBMS.....	17
Gambar 2.4 Arsitektur <i>Agile</i>	22
Gambar 2.5 Arsitektur KMP.....	23
Gambar 2.6 Arsitektur CAPTCHA.....	25
Gambar 2.7 Arsitektur <i>Black Box Testing</i>	25
Gambar 3.1 Kerangka pikir.....	27
Gambar 4.1 <i>Use Case Diagram</i> Yukitanya.....	39
Gambar 4.2 <i>Activity Diagram Register</i>	47
Gambar 4.3 <i>Activity Diagram Login</i>	48
Gambar 4.4 <i>Activity Diagram</i> Halaman Pencarian.....	49
Gambar 4.5 <i>Activity Diagram</i> Membuat Pertanyaan.....	50
Gambar 4.6 <i>Activity Diagram</i> Fitur Tanyakan pada AI.....	51
Gambar 4.7 <i>Activity Diagram</i> Mengakses Halaman Premium.....	52
Gambar 4.8 <i>Class Diagram</i> Aplikasi Yukitanya.....	53
Gambar 4.9 Desain Halaman <i>Register</i>	61
Gambar 4.10 Desain Halaman <i>Login</i>	62
Gambar 4.11 Desain Halaman Beranda.....	62
Gambar 4.12 Desain Dialog Buat Pertanyaan Baru.....	63
Gambar 4.13 Desain Halaman Detail Pertanyaan.....	64
Gambar 4.14 Desain Dialog Buat Jawaban Baru.....	64
Gambar 4.15 Desain Halaman Notifikasi.....	65
Gambar 4.16 Desain Halaman Profil.....	66
Gambar 5.1 Halaman <i>Register</i>	83

Gambar 5.2 Halaman <i>Login</i>	84
Gambar 5.3 Halaman Beranda di <i>Desktop</i>	85
Gambar 5.4 Halaman Beranda di <i>Mobile</i>	85
Gambar 5.5 Dialog Buat Pertanyaan.....	86
Gambar 5.6 Notifikasi <i>Filtering</i> Kata Terlarang.....	86
Gambar 5.7 Halaman Detail Pertanyaan di <i>Desktop</i>	87
Gambar 5.8 Halaman Detail Pertanyaan di <i>Mobile</i>	87
Gambar 5.9 Dialog Buat Jawaban.....	88
Gambar 5.10 Fitur Tanyakan Pada AI.....	88
Gambar 5.11 Halaman Pencarian.....	89
Gambar 5.12 Halaman Notifikasi.....	90
Gambar 5.13 Halaman Mata Pelajaran.....	90
Gambar 5.14 Halaman Premium.....	91
Gambar 5.15 Dialog Pembayaran.....	91
Gambar 5.16 Halaman Profil.....	92
Gambar 5.17 Grafik pertanyaan kuisioner 1.....	94
Gambar 5.18 Grafik pertanyaan kuisioner 2.....	94
Gambar 5.19 Grafik pertanyaan kuisioner 3.....	94
Gambar 5.20 Grafik pertanyaan kuisioner 4.....	95
Gambar 5.21 Grafik pertanyaan kuisioner 5.....	95
Gambar 5.22 Grafik pertanyaan kuisioner 6.....	95
Gambar 5.23 Grafik pertanyaan kuisioner 7.....	96
Gambar 5.24 Grafik pertanyaan kuisioner 8.....	96
Gambar 5.25 Grafik pertanyaan kuisioner 9.....	96
Gambar 5.26 Grafik pertanyaan kuisioner 10.....	97
Gambar 5.27 Grafik pertanyaan kuisioner 11.....	97
Gambar 5.28 Grafik pertanyaan kuisioner 12.....	97

Gambar 5.29 Grafik pertanyaan kuisioner 13.....	98
Gambar 5.30 Grafik pertanyaan kuisioner 14.....	98
Gambar 5.31 Grafik pertanyaan kuisioner 15.....	98
Gambar 5.32 Grafik pertanyaan kuisioner 16.....	99
Gambar 5.33 Grafik pertanyaan kuisioner 17.....	99
Gambar 5.34 Grafik pertanyaan kuisioner 18.....	99
Gambar 5.35 Grafik pertanyaan kuisioner 19.....	100
Gambar 5.36 Grafik pertanyaan kuisioner 20.....	100
Gambar 5.37 Grafik pertanyaan kuisioner 21.....	100
Gambar 5.38 Grafik pertanyaan kuisioner 22.....	101
Gambar 5.39 Grafik pertanyaan kuisioner 23.....	101

DAFTAR TABEL

Tabel 2.1 Acuan Penelitian.....	8
Tabel 4.1 Analisis Biaya.....	38
Tabel 4.2 Deskripsi Aktor.....	40
Tabel 4.3 Deskripsi <i>Use Case Register</i>	40
Tabel 4.4 Deskripsi <i>Use Case Login</i>	41
Tabel 4.5 Deskripsi <i>Use Case Mengakses Halaman Beranda</i>	41
Tabel 4.6 Deskripsi <i>Use Case Mengakses Halaman Pencarian</i>	43
Tabel 4.7 Deskripsi <i>Use Case Mengakses Halaman Notifikasi</i>	43
Tabel 4.8 Deskripsi <i>Use Case Mengakses Halaman Mata Pelajaran</i>	44
Tabel 4.9 Deskripsi <i>Use Case Mengakses Halaman Premium</i>	45
Tabel 4.10 Deskripsi <i>Use Case Mengakses Halaman Profil</i>	46
Tabel 4.11 Perancangan tabel <i>user</i>	54
Tabel 4.12 Perancangan tabel <i>question</i>	55
Tabel 4.13 Perancangan tabel <i>old slug</i>	56
Tabel 4.14 Perancangan tabel <i>question image</i>	56
Tabel 4.15 Perancangan tabel <i>subject</i>	57
Tabel 4.16 Perancangan tabel <i>answer</i>	57
Tabel 4.17 Perancangan tabel <i>favorite</i>	58
Tabel 4.18 Perancangan tabel <i>rating</i>	58
Tabel 4.19 Perancangan tabel <i>membership</i>	59
Tabel 4.20 Perancangan tabel <i>notification</i>	59
Tabel 5.1 Spesifikasi Perangkat Keras.....	81
Tabel 5.2 Spesifikasi Perangkat Lunak.....	81
Tabel 5.3 Hasil Uji oleh Pengguna.....	93

DAFTAR LAMPIRAN

Lampiran 1 TOR	109
Lampiran 2 Data Penurunan Jumlah Pengguna Aplikasi dari <i>Database</i>	111
Lampiran 3 Hasil Wawancara dengan Narasumber	113
Lampiran 4 Dokumentasi	115
Lampiran 5 Dataset Kata-kata Terlarang dalam format JSON	117
Lampiran 6 Data Pengujian dalam format CSV	119
Lampiran 7 Listing Program	122
Lampiran 8 Hasil Pengujian Black Box	126

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pendidikan merupakan kunci utama dalam membangun generasi muda yang cerdas, kreatif, dan berkarakter (Saputri *et al*, 2023). Namun, terdapat beberapa tantangan dalam proses pembelajaran, seperti kurangnya variasi dalam metode pembelajaran, minat belajar yang rendah, dan kesulitan mendapatkan bantuan saat belajar di rumah (Hamida & Putra, 2021). Tantangan-tantangan tersebut dapat berakibat pada kurangnya interaksi dan kolaborasi antar siswa, rasa bosan dan penurunan motivasi belajar (Zam, 2021), serta berpotensi menimbulkan kesenjangan pengetahuan antar siswa (Rofingah, 2021). Upaya untuk mencapai pendidikan yang berkualitas menjadi hal yang krusial dalam rangka mewujudkan generasi penerus bangsa yang unggul. Perkembangan teknologi saat ini memunculkan berbagai macam aplikasi yang berguna untuk membantu pembelajaran siswa, salah satunya adalah aplikasi Brainly (Nugroho, 2021).

Pada program Kampus Merdeka *Batch 5*, peneliti mengikuti kegiatan Studi Independen di perusahaan bernama *Forest Interactive*. *Forest Interactive* merupakan perusahaan yang bergerak di bidang teknologi informasi, khususnya telekomunikasi. Kantor pusat perusahaan ini berlokasi di Megan Avenue II Unit C-7-2, 12, Jalan Yap Kwan Seng, Kampung Baru, 50450, Kuala Lumpur, Malaysia. *Forest Interactive* pun memiliki beberapa kantor cabang di Indonesia salah satunya di Jl. Suryanagara No.27, RT.06/RW.05, Cibaduyut Wetan, Kec. Bojongloa Kidul, Kota Bandung. Selama kegiatan Studi Independen di *Forest Interactive* ini, peneliti membuat sebuah aplikasi bernama Yukitanya. Aplikasi ini merupakan aplikasi forum tanya jawab untuk menyelesaikan tugas sekolah secara bersama yang menjadikan Brainly sebagai referensi.

Hasil penelitian terhadap aplikasi yang telah dibuat sebelumnya menunjukkan bahwa masih terdapat beberapa masalah di aplikasi Yukitanya ini. Yang pertama adalah belum adanya sistem *filtering* kata pada fitur pembuatan

pertanyaan dan jawaban sehingga memungkinkan pengguna untuk membuat pertanyaan dan jawaban yang tidak sopan. Yang kedua yaitu belum adanya fitur *rate limit* yang menyebabkan pengguna dapat membuat pertanyaan dan jawaban sebanyak mungkin sehingga banyak sekali pertanyaan dan jawaban yang bersifat *spam*. Dan yang terakhir adalah kurangnya minat orang untuk menggunakan aplikasi ini. Berdasarkan data yang diambil dari *database* aplikasi Yukitanya, menunjukan bahwa selama bulan Desember 2023 - Mei 2024 hanya terdapat 109 pengguna, dan rata-rata hanya ada 12 pertanyaan baru/bulan.

Penelitian terdahulu yang memiliki kesamaan dengan topik penelitian ini yaitu dilakukan oleh Almasik *et al* (2020) yang membuat aplikasi forum diskusi dengan tujuan agar UMKM dapat saling berinteraksi, bertukar informasi, dan mendapatkan dukungan dari pemerintah dan sesama UMKM untuk meningkatkan kinerja dan daya saing mereka dalam pasar global. Berikutnya penelitian dari Ramadhan *et al* (2020) yang membahas tentang aplikasi forum diskusi yang dikembangkan untuk Himpunan Mahasiswa Teknik Universitas Jenderal Soedirman bertujuan untuk berbagi pengetahuan tanpa batasan bagi pengguna lain untuk melihat diskusi. Dan terakhir penelitian dari Aziz *et al* (2023) yang membahas tentang aplikasi forum diskusi yang berfokus pada game esport, yang dikembangkan dengan menggunakan metode *Extreme Programming* dan berbasis web. Perbedaan jurnal acuan pertama dengan penelitian yang akan dibuat adalah jurnal ini menggunakan metode *Rational Unified Process* (RUP), sedangkan peneliti akan menggunakan metode *Agile*. Perbedaan jurnal acuan kedua dengan penelitian yang akan dibuat adalah jurnal ini menggunakan metode *Waterfall*, sedangkan peneliti akan menggunakan metode *Agile*. Perbedaan lain yaitu jurnal ini menggunakan bahasa pemrograman PHP, *framework* Laravel, dan *database* MySQL, sedangkan peneliti akan menggunakan bahasa pemrograman TypeScript, *framework* Next.js dan *database* PostgreSQL. Perbedaan jurnal acuan ketiga dengan penelitian yang akan dibuat adalah jurnal ini menggunakan metode *Extreme Programming*, sedangkan peneliti akan menggunakan metode *Agile*.

Oleh karena itu, dalam pembuatan aplikasi ini akan diterapkan Algoritma Knuth-Morris-Pratt untuk melakukan *filtering* kata pada proses pembuatan pertanyaan dan jawaban untuk mencegah adanya kata-kata yang tidak sopan.

Selanjutnya akan diterapkan juga metode CAPTCHA untuk mencegah terjadinya *spam* dalam pembuatan pertanyaan dan jawaban. Untuk menambah minat pengguna, akan ditambahkan beberapa fitur menarik seperti *User Interface* yang lebih *user-friendly*, integrasi ke Google Gemini AI, dan *Leaderboard* yang menampilkan pengguna teraktif minggu ini dan pertanyaan terpopuler minggu ini. Dan akan dilakukan juga *SEO Optimization* agar aplikasi ini dapat terindeks oleh Google sehingga dapat dikenal oleh lebih banyak orang. Pada pengembangan aplikasi ini akan menggunakan bahasa pemrograman TypeScript, *framework* Next.js, dan *database* PostgreSQL. Metode pengembangan aplikasi yang digunakan yaitu *Agile*. Berdasarkan latar belakang tersebut, penulis tertarik untuk meneliti serta mengembangkan aplikasi forum tanya jawab dalam bentuk penulisan ilmiah yang berjudul: “PENGEMBANGAN APLIKASI FORUM TANYA JAWAB TUGAS SEKOLAH (YUKITANYA) MENGGUNAKAN FRAMEWORK NEXT.JS DAN POSTGRESQL”.

1.2 Rumusan Masalah

Dari latar belakang yang telah dikemukakan di atas, maka perumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana mencegah adanya kata-kata tidak sopan pada pertanyaan dan jawaban di aplikasi forum tanya jawab Yukitanya?
2. Bagaimana mengatasi serangan *spam* pada aplikasi forum tanya jawab Yukitanya?
3. Bagaimana cara meningkatkan minat orang untuk menggunakan aplikasi forum tanya jawab ini?

1.3 Batasan Masalah

Batasan masalah terhadap penelitian ini adalah sebagai berikut:

1. Aplikasi ini dibangun menggunakan bahasa pemrograman TypeScript, *framework* Next.js, dan *database* PostgreSQL.
2. Aplikasi ini menggunakan algoritma Knuth-Morris-Pratt untuk mendeteksi adanya kata-kata tidak sopan.

3. Aplikasi ini menggunakan metode CAPTCHA milik Cloudflare yaitu Turnstile untuk mencegah terjadinya serangan *spam*.
4. Pada aplikasi ini hanya mencakup aplikasi web saja tanpa adanya aplikasi mobile maupun desktop.
5. Aplikasi ini mengambil referensi dari Brainly.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dengan dibuatnya tugas akhir skripsi berjudul “Yukitanya” adalah sebagai berikut:

1. Mencegah adanya kata-kata tidak sopan pada pertanyaan dan jawaban di aplikasi forum tanya jawab Yukitanya.
2. Mengatasi serangan *spam* pada aplikasi forum tanya jawab Yukitanya.
3. Meningkatkan minat orang untuk menggunakan aplikasi forum tanya jawab ini.

1.5 Metodologi Penelitian

Penelitian ini menggunakan metode gabungan antara pendekatan kualitatif dan kuantitatif, yang dipilih untuk menyesuaikan dengan teknik pengolahan data yang akan digunakan. Pendekatan kualitatif memungkinkan peneliti untuk memahami konteks dan nuansa dari data yang diperoleh, sementara pendekatan kuantitatif memberikan struktur yang lebih sistematis dan memungkinkan analisis statistik yang lebih mendalam. Metode *agile* akan diterapkan sebagai kerangka kerja untuk pengembangan sistem dalam penelitian ini. *Agile* dipilih karena fleksibilitas, kemampuan beradaptasi, serta peningkatan kolaborasi dan efisiensi pengembang dalam menghadapi tantangan.

1.5.1 Metode Pengumpulan Data

1. Observasi

Survei secara langsung terhadap aplikasi forum tanya jawab tugas sekolah yang sudah ada sebelumnya seperti Yukitanya, dan Brainly.

2. Survei

Metode ini melibatkan pengumpulan data melalui kuesioner atau wawancara dengan responden yang relevan dengan topik penelitian. Dalam konteks pengembangan aplikasi Yukitanya, metode survei dapat dilakukan dengan mengumpulkan data dari web developer mengenai masalah apa yang masih terjadi di Yukitanya dan apa penyebabnya.

3. Studi Pustaka

Survei studi literatur pustaka seperti artikel, jurnal, buku, dll.

1.5.2 Metode Pengembangan Sistem

Perancangan dalam pembuatan aplikasi ini menggunakan metode *Agile* yang terdiri dari tahapan berikut:

1. Perencanaan (*planning*), merupakan tahap awal dalam proses kategorisasi yang membutuhkan langkah-langkah tertentu.
2. Implementasi, yaitu tahap persiapan menu untuk pengguna yang dihasilkan dari perancangan sistem baru yang telah disetujui ke dalam bahasa pemrograman.
3. Pengujian (*testing*), adalah prasyarat utama dalam sebuah sistem. Pada tahap ini, pengembang menerjemahkan sistem ke dalam bentuk kode, kemudian melakukan uji coba perangkat lunak.
4. Dokumentasi, adalah proses pencatatan perangkat lunak, dilakukan dengan merekam setiap langkah dari sistem yang dibangun.
5. Penyebaran (*deployment*), pengembang menginformasikan pembaruan layanan kepada pengguna.
6. Pemeliharaan (*maintenance*), memastikan sistem tetap dalam kondisi optimal.

1.6 Sistematika Penulisan

Pada bab ini, penulis membahas sistematika laporan skripsi pada bab ini dengan tujuan untuk memudahkan pembaca dalam memahami isi laporan ini. Penulis menyusun laporan skripsi ini ke dalam beberapa bab yang merupakan satu kesatuan dan saling melengkapi, seperti yang terlihat di bawah ini.

BAB I (PENDAHULUAN)

Bab ini menjadi landasan penelitian tugas akhir dengan memaparkan latar belakang penelitian, gambaran kesulitan yang dihadapi, rumusan dan batasan masalah, tujuan penelitian, metode yang digunakan, dan sistematika penulisan. Hal ini bertujuan untuk memberikan gambaran menyeluruh mengenai penelitian yang akan dilakukan.

BAB II (TINJAUAN PUSTAKA)

Tinjauan literatur ini memberikan pemahaman mendalam mengenai berbagai teori dan konsep dasar yang relevan dengan topik yang dibahas. Beberapa di antaranya mencakup definisi dan karakteristik dari aplikasi web, yang merujuk pada perangkat lunak yang berjalan di atas jaringan internet dan dapat diakses melalui peramban web. Selain itu, dibahas pula konsep forum diskusi, yaitu *platform online* yang memungkinkan pengguna untuk berinteraksi dan berdiskusi mengenai berbagai topik. Di samping itu, tinjauan ini juga menguraikan konsep *full stack development*, yang mengacu pada pengembangan aplikasi yang mencakup semua lapisan teknologi, mulai dari *front-end* hingga *back-end*. Lebih lanjut, tinjauan ini juga meninjau laporan penelitian sebelumnya yang telah dilakukan dalam bidang ini, yang memberikan wawasan dan konteks historis terhadap perkembangan dan penerapan teori-teori tersebut dalam praktik. Dengan demikian, tinjauan literatur ini berfungsi sebagai landasan teoritis yang komprehensif untuk mendukung analisis lebih lanjut dalam penelitian ini.

BAB III (METODOLOGI PENELITIAN)

Dalam proses penelitian ini, metodologi yang diterapkan mencakup serangkaian langkah yang komprehensif yang melibatkan pembuatan sistem, termasuk pengembangan bahan, identifikasi metode yang relevan, serta tahapan kerja yang diperlukan guna menyelesaikan tugas yang ditetapkan. Penjelasan atas proses ini didukung dengan penerapan sistem dan alat yang sesuai, memberikan fondasi yang kuat bagi keseluruhan pendekatan penelitian yang diambil oleh penulis.

BAB IV (ANALISIS, PERANCANGAN, DAN HASIL BIDANG INFORMATIKA ANALISIS)

Bab ini menguraikan secara rinci berbagai metode yang digunakan oleh penulis dalam merancang dan mengembangkan sistem yang direncanakan. Dalam pembahasan ini, penulis menjelaskan langkah-langkah yang ditempuh untuk membangun rangkaian alat dan fungsi yang terintegrasi dalam sistem tersebut. Setiap alat dan fungsi yang menjadi bagian dari sistem dijelaskan dengan detail, termasuk bagaimana masing-masing komponen ini bekerja sama untuk mencapai tujuan yang diinginkan. Selain itu, bab ini juga menyajikan rangkaian hasil pengujian yang telah dilakukan terhadap sistem yang dibuat. Hasil pengujian ini memberikan gambaran tentang kinerja dan efektivitas sistem, serta mengidentifikasi area yang memerlukan perbaikan atau penyesuaian lebih lanjut. Dengan demikian, bab ini tidak hanya menggambarkan proses pengembangan sistem, tetapi juga menyediakan bukti empiris tentang keberhasilan dan kekurangan dari metode yang digunakan melalui hasil pengujian yang komprehensif.

BAB V (IMPLEMENTASI DAN PENGUJIAN TEKNIK INFORMATIKA)

Bab ini mencakup berbagai aspek penting dalam proses implementasi sistem, mulai dari penggabungan komponen, spesifikasi teknis, instalasi, hingga pengoperasian dan pengujian sistem. Proses menjalankan sistem yang telah dirancang juga diuraikan, mencakup pengoperasian dan pemeliharaan agar sistem berfungsi optimal. Selain itu, hasil pengujian yang dilakukan terhadap sistem yang dirangkai disajikan untuk memastikan kinerja yang baik dan mengidentifikasi area yang memerlukan perbaikan. Dengan demikian, bab ini memberikan gambaran komprehensif tentang keseluruhan proses dari penggabungan hingga evaluasi kinerja sistem melalui pengujian yang teliti.

BAB VI (KESIMPULAN DAN SARAN)

Bab ini menarik kesimpulan dari hasil analisis data pada bab-bab sebelumnya. Saran-saran yang diberikan akan sangat bermanfaat karena dapat membantu dalam membuat dan mengembangkan isi laporan sesuai dengan tujuan penelitian skripsi ini.

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan teori ini berisi mengenai informasi-informasi penelitian yang berkaitan dengan topik penelitian ini yang pernah dilakukan sebelumnya. Hal ini dimaksudkan untuk melakukan perbandingan mengenai kelebihan dan kekurangan yang sudah ada. Berbagai informasi landasan teori didapatkan dari berbagai jurnal-jurnal penelitian yang ada.

Tabel 2.1 Acuan Penelitian

No	Judul	Masalah	Metode	Solusi
1	Rancang Bangun Aplikasi Forum Diskusi Usaha Mikro Kecil dan Menengah Berbasis Web. (Atmadja, A. R., & Pariyatin, Y., 2020).	Penelitian yang dibuat untuk menghadapi tantangan globalisasi dan masalah internal yang dihadapi oleh UMKM	Rational Unified Process (RUP)	Dengan menggunakan aplikasi forum Diskusi, UMKM dapat saling berinteraksi, bertukar informasi, dan mendapatkan dukungan Dari pemerintah dan sesama UMKM untuk meningkatkan kinerja dan daya saing mereka dalam pasar global.

2	Aplikasi Forum Diskusi Himpunan Mahasiswa Teknik Universitas Jenderal Soedirman Menggunakan Laravel. (Satria, F., Chasanah, N., & Iskandar, D., 2020).	Forum online merupakan platform Untuk berbagi pengetahuan tanpa batasan bagi pengguna lain untuk melihat diskusi, sehingga penyebaran informasi menjadi lebih luas. Namun, aplikasi lain memiliki keterbatasan dimana pengguna harus menjadi anggota terlebih dahulu untuk membaca diskusi, yang mengakibatkan proses penyebaran informasi kurang efisien.	Metode Waterfall	Dengan menggunakan aplikasi ini, mahasiswa tidak perlu menjadi anggota terlebih dahulu untuk membaca diskusi, sehingga penyebaran informasi menjadi lebih luas dan efisien
3	Aplikasi Forum Diskusi Seputar	Peneliti menggarisbawa	Metode Extreme	Aplikasi forum diskusi ini

	<p>Game Esport Berbasis Web Dengan Metode Extreme Programming. (Aziz, F. A., & Ikasari, I. H., 2023).</p>	<p>hi pentingnya informasi terkini seputar Game esport di Indonesia, mengingat pertumbuhan industri ini dan minat yang meningkat dari generasi muda.</p> <p>Meskipun ada pertumbuhan pesat dalam industri esport di Indonesia, kekurangan dalam media yang menyediakan informasi terupdate menjadi masalah yang perlu diatasi.</p>	<p>Programming</p>	<p>bertujuan Untuk menjadi Solusi atas kebutuhan akan informasi terkini seputar game esport di Indonesia. Dengan menggunakan metode Extreme Programming, aplikasi ini dibangun dengan fokus pada interaksi pengguna dan responsif terhadap perkembangan game esport.</p>
--	---------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Jurnal yang ditulis oleh Sipa Almasik, Aldy Rialdy Atmadja, Yeni Pariyatni, mahasiswa Sekolah Tinggi Teknologi Garut pada tahun 2020 membahas tentang pembangunan aplikasi forum diskusi untuk usaha mikro, kecil, dan

menengah (UMKM) berbasis web sebagai solusi dalam menghadapi tantangan globalisasi dan masalah internal yang dihadapi oleh UMKM. Pemberdayaan UMKM dalam menghadapi persaingan global membutuhkan pengembangan sumber daya manusia, inovasi produk, dan ekspansi pasar. Namun, UMKM juga menghadapi masalah internal seperti manajemen sumber daya manusia, keterampilan, keuangan, dan pemasaran. Aplikasi yang dirancang dan dikembangkan menggunakan metodologi *Rational Unified Process* (RUP), melalui tahapan *inception, elaboration, construction, dan transition*. Tujuan dari aplikasi ini adalah untuk membantu pegiat UMKM dan pemerintah di Kabupaten Garut dalam berdiskusi mengenai permasalahan yang dihadapi oleh UMKM serta untuk memajukan UMKM di daerah tersebut. Dengan menggunakan aplikasi forum diskusi ini, UMKM dapat saling berinteraksi, bertukar informasi, dan mendapatkan dukungan dari pemerintah dan sesama UMKM untuk meningkatkan kinerja dan daya saing mereka dalam pasar global.

Berikutnya, jurnal yang ditulis oleh Fisal Satria Nur Ramadhan, Nur Chasanah, Dadang Iskandar, mahasiswa Teknik Informatika dari Universitas Jenderal Soedirman pada tahun 2020 membahas tentang aplikasi forum diskusi yang dikembangkan khusus untuk Himpunan Mahasiswa Teknik Universitas Jenderal Soedirman menggunakan *framework* Laravel. Forum *online* merupakan *platform* untuk berbagi pengetahuan tanpa batasan bagi pengguna lain untuk melihat diskusi, sehingga penyebaran informasi menjadi lebih luas. Namun, aplikasi lain memiliki keterbatasan dimana pengguna harus menjadi anggota terlebih dahulu untuk membaca diskusi, yang mengakibatkan proses penyebaran informasi kurang efisien. Aplikasi forum diskusi yang dikembangkan untuk Himpunan Mahasiswa Teknik Universitas Jenderal Soedirman bertujuan untuk mengatasi keterbatasan tersebut. Dengan menggunakan aplikasi ini, mahasiswa tidak perlu menjadi anggota terlebih dahulu untuk membaca diskusi, sehingga penyebaran informasi menjadi lebih luas dan efisien. Metode pengembangan yang digunakan adalah *Waterfall*, dengan bahasa pemrograman PHP, *framework* Laravel, dan *database* MySQL.

Selanjutnya, jurnal yang ditulis oleh Fuad Abdul Aziz, Ines Heidiani Ikasari, mahasiswa Teknik Informatika dari Universitas Pamulang pada tahun

2023 membahas tentang aplikasi forum diskusi yang berfokus pada *game esport*, yang dikembangkan dengan menggunakan metode *Extreme Programming* dan berbasis web. Peneliti menggarisbawahi pentingnya informasi terkini seputar *game esport* di Indonesia, mengingat pertumbuhan industri ini dan minat yang meningkat dari generasi muda. Meskipun ada pertumbuhan pesat dalam industri *esport* di Indonesia, kekurangan dalam media yang menyediakan informasi *ter-update* menjadi masalah yang perlu diatasi. Aplikasi forum diskusi ini bertujuan untuk menjadi solusi atas kebutuhan akan informasi terkini seputar *game esport* di Indonesia. Dengan menggunakan metode *Extreme Programming*, aplikasi ini dibangun dengan fokus pada interaksi pengguna dan responsif terhadap perkembangan *game esport*. Penelitian ini menunjukkan bahwa aplikasi tersebut berhasil memenuhi kebutuhan tersebut dengan memberikan wadah bagi penggemar game esport untuk mendiskusikan topik-topik yang relevan. Tanggapan dari responden menunjukkan tingkat kepuasan yang tinggi terhadap aplikasi, dengan mayoritas responden menyatakan setuju atau sangat setuju terhadap penggunaan aplikasi ini.

Perbedaan jurnal acuan pertama dengan penelitian yang akan dibuat adalah jurnal ini menggunakan metode *Rational Unified Process* (RUP), sedangkan peneliti akan menggunakan metode *Agile*. Perbedaan jurnal acuan kedua dengan penelitian yang akan dibuat adalah jurnal ini menggunakan metode *Waterfall*, sedangkan peneliti akan menggunakan metode *Agile*. Perbedaan lain yaitu jurnal ini menggunakan bahasa pemrograman PHP, *framework* Laravel, dan *database* MySQL, sedangkan peneliti akan menggunakan bahasa pemrograman TypeScript, *framework* Next.js dan *database* PostgreSQL. Perbedaan jurnal acuan ketiga dengan penelitian yang akan dibuat adalah jurnal ini menggunakan metode *Extreme Programming*, sedangkan peneliti akan menggunakan metode *Agile*. Kelebihan yang dimiliki penelitian yang akan dibuat dibanding dengan jurnal acuan lain adalah penelitian ini akan memiliki fitur *filtering* kata terlarang dengan algoritma *Knuth-Morris-Pratt (KMP)*. Kelebihan selanjutnya adalah penelitian ini akan memiliki fitur pencegahan *spam* dengan menggunakan metode CAPTCHA. Selanjutnya adalah penelitian ini akan terintegrasi dengan Gemini AI untuk menjawab pertanyaan yang dibuat pengguna secara otomatis jika diperlukan.

2.2 Dasar Teori

2.2.1 *Web Application*

Aplikasi web, atau yang sering disebut sebagai *web app*, merupakan sebuah program komputer yang dijalankan melalui *web browser*. Aplikasi ini diakses oleh pengguna melalui jaringan internet dan disajikan lewat *World Wide Web*. *Web app* bisa menyajikan halaman web yang bersifat statis, yaitu konten yang tidak berubah, atau dinamis, yaitu konten yang dapat berubah sesuai dengan interaksi pengguna atau data yang diterima (Fowler *et al*, 2004).

Web app dapat berinteraksi dengan pengguna dengan menerima input, seperti formulir atau data. Aplikasi ini menyimpan dan mengambil data sesuai kebutuhan, memastikan akses informasi relevan. Hasil interaksi tersebut ditampilkan kembali dalam bentuk yang mudah dipahami dan digunakan. Dengan kemampuan ini, *web app* dapat meningkatkan efisiensi dan kenyamanan pengguna dalam berbagai aktivitas *online*.

Aplikasi web memiliki berbagai kegunaan. Dalam *e-commerce*, *web app* memungkinkan pembelian barang secara *online*, melihat katalog, dan melakukan pembayaran yang aman. Untuk pengolahan data, *web app* membantu analisis, penyimpanan, dan pelaporan data. Di bidang komunikasi, *web app* mendukung interaksi *real-time* seperti pesan instan dan *video call*. Secara keseluruhan, *web app* penting dan serbaguna dalam kehidupan dan bisnis modern.

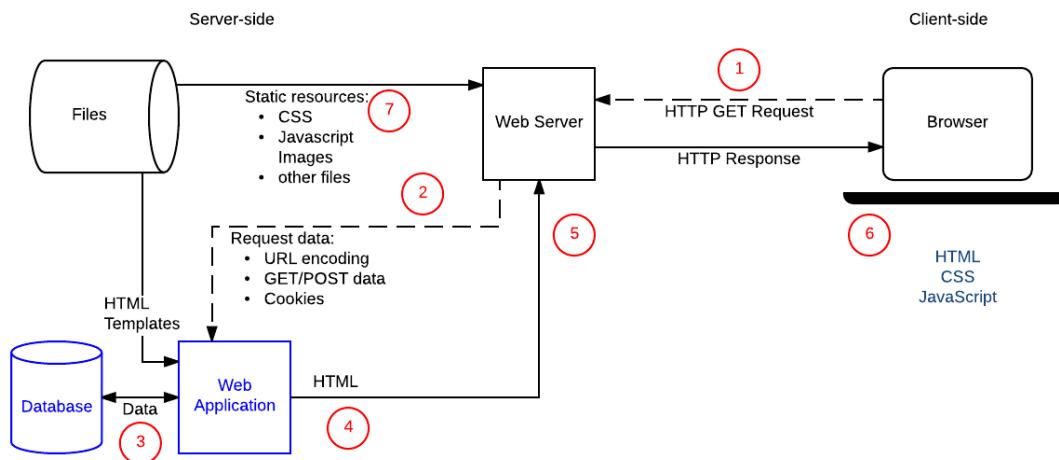
Web apps dapat memiliki beberapa keuntungan, seperti:

- a. Aksesibilitas: *Web apps* dapat diakses dari semua *web browser* dan berbagai perangkat personal dan bisnis.
- b. Efisiensi: Pengembangan *web apps* relatif mudah dan murah, dapat memiliki tahap pengembangan yang singkat, dan dapat digunakan di berbagai perangkat modern tanpa perlu membuat versi terpisah.
- c. Kemudahan: *Web apps* tidak perlu diunduh, dan dapat diakses dengan mudah melalui URL.

Contoh dari web apps antara lain:

- a. Aplikasi pengolahan data, seperti Google Sheets atau Microsoft Excel Online.

- b. Aplikasi komunikasi, seperti Gmail atau Outlook Online.
- c. Aplikasi *e-commerce*, seperti Amazon atau eBay.
- d. Aplikasi sosial media, seperti Facebook atau Twitter.



Gambar 2.1 Arsitektur Client-side dan Server-side

Sumber: developer.mozilla.org

Web application memiliki arsitektur *client-server*. Kodenya dibagi menjadi 2 komponen yaitu *client-side script* dan *server-side script*:

a. *Client-side architecture*

Client-side script berhubungan dengan fungsionalitas *user interface* seperti tombol dan *dropdown box*. Ketika pengguna mengklik pada link *web app*, web browser memuat *client-side script* dan merender elemen grafis dan teks untuk interaksi pengguna. Sebagai contoh, pengguna dapat membaca konten, menonton video, atau mengisi detail pada form kontak. Tindakan seperti mengklik tombol submit masuk ke server sebagai sebuah *client request*

b. *Server-side architecture*

Server-side script berhubungan dengan pemrosesan data. *Server web application* memproses *client request* dan mengirim kembali sebuah *response*. *Request* biasanya untuk mendapatkan lebih banyak data atau untuk mengedit atau menyimpan data baru. Sebagai contoh, ketika

pengguna mengklik pada tombol “*Read more*”, *server web application* akan mengirim konten kepada pengguna. Jika pengguna mengklik tombol submit, server aplikasi akan menyimpan data pengguna ke dalam *database*. Dalam beberapa kasus, server menyelesaikan *request* data dan mengirimkan halaman HTML lengkap kepada *client*, ini disebut *server-side rendering*.

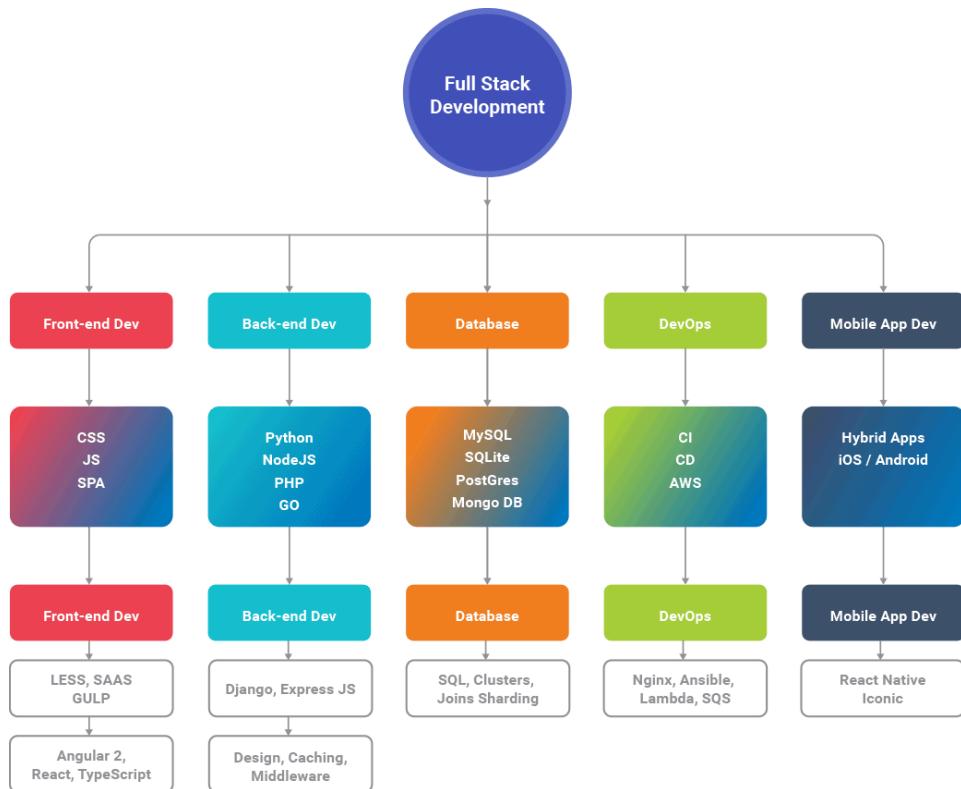
2.2.2 Forum Diskusi

Forum diskusi adalah sebuah media *online* di mana orang dapat berkomunikasi dan berbagi informasi, ide, atau pertanyaan melalui pesan yang diposting (Pendry *et al*, 2015). Forum ini berbeda dari ruang obrolan karena pesan biasanya lebih panjang daripada satu baris teks dan sering diarsipkan untuk sementara waktu. Pesan yang diposting perlu disetujui oleh moderator sebelum menjadi terlihat secara publik. Forum diskusi memiliki struktur hierarkis atau mirip pohon, di mana forum dapat berisi beberapa subforum, masing-masing dengan beberapa topik. Setiap topik dalam forum dapat dijawab oleh sebanyak orang yang diinginkan. Pengguna biasanya dapat tetap anonim atau harus mendaftar dan masuk untuk memposting pesan. Forum diskusi berasal dari papan buletin dan sistem konferensi komputer, yang merupakan evolusi teknologi dari sistem papan buletin dial-up (BBS). Forum internet umumnya dikelola oleh aplikasi web yang mengelola konten yang dihasilkan pengguna.

2.2.3 Full Stack Development

Full Stack Development adalah proses pengembangan perangkat lunak yang mencakup kedua sisi, yaitu *front end* (bagian *client*) dan *back end* (bagian *server*) dari sebuah aplikasi web (Wijaya *et al*, 2021). Seorang pengembang *full stack* memiliki keterampilan untuk membuat aplikasi web yang lengkap mulai dari desain *front end* hingga pemecahan masalah *back end*, serta melakukan *debugging*. Mereka menggunakan berbagai teknologi seperti HTML, CSS, dan JavaScript untuk mengembangkan bagian *front end*, dan bahasa pemrograman seperti PHP, Python, dan Java untuk mengelola bagian *back end* aplikasi. Kemampuan mereka memungkinkan mereka untuk memahami secara

komprehensif alur kerja aplikasi web dan menyediakan solusi terbaik bagi pengguna.



Gambar 2.2 Arsitektur Full Stack Development

Sumber: devskiller.com

Front end merujuk pada apa yang pengguna ketika mereka membuka sebuah browser atau aplikasi. Dalam kata lain, *front end* berhubungan dengan *user interface*. Ketika pengguna berinteraksi dengan sebuah website, pengguna tersebut sedang menggunakan *front end*.

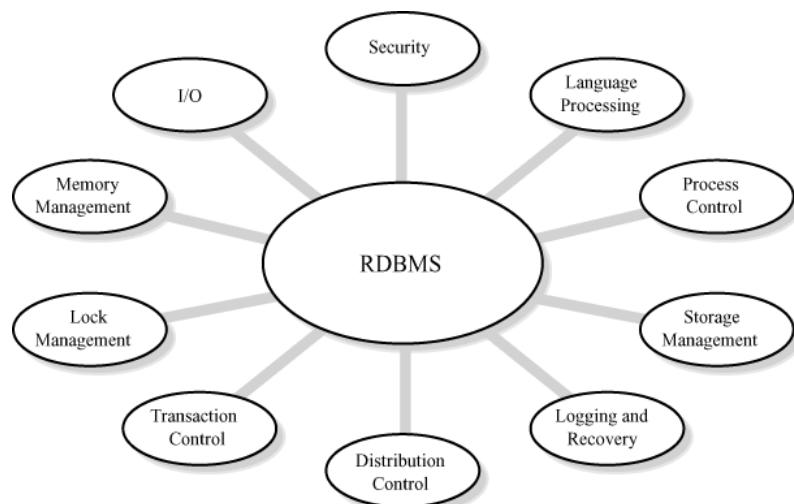
Back end merujuk pada bagian dari sebuah situs atau aplikasi dimana pengguna tidak lihat. *Back end developer* menulis kode yang membuat komunikasi di antara database dan browser untuk memastikan situs bekerja dengan baik. Komponen *back end* lain termasuk *framework automated testing*, *cybersecurity*, konfigurasi server dan skalabilitas jaringan.

Back end developer fokus kepada fungsionalitas dari situs atau aplikasi melalui coding. Mereka menggunakan berbagai bahasa pemrograman seperti Java, PHP, Ruby, Python, Golang, Rust, dan SQL untuk menyelesaikan pekerjaan. Pada

akhirnya, *back end* menciptakan pengalaman yang mulus bagi pengguna aplikasi atau situs.

2.2.4 Relational Database Management System (RDBMS)

Relational database management system (RDBMS) adalah sebuah sistem manajemen *database* yang mengorganisir data dan hubungannya dalam bentuk tabel. RDBMS memungkinkan pengguna untuk mendefinisikan, membuat, mempertahankan, dan mengontrol akses ke database relasional. RDBMS menggunakan SQL (*Structured Query Language*) untuk mengelola data, dan data disimpan dalam tabel yang terdiri dari baris dan kolom (Mustafidah *et al*, 2021). RDBMS menyediakan operasi relasional untuk memanipulasi data dalam tabel, dan memungkinkan hubungan antar tabel yang terkait. RDBMS juga menyediakan fitur untuk mempertahankan keamanan, akurasi, integritas, dan konsistensi data.



Gambar 2.3 Arsitektur RDBMS

Sumber: technicgang.com

Pada umumnya, database menyimpan kumpulan data yang dapat diminta untuk digunakan dalam aplikasi lain. RDBMS mendukung pengembangan, administrasi, dan penggunaan platform database.

RDBMS adalah jenis sistem manajemen basis data (DBMS) yang menyimpan data dalam struktur tabel berbaris baris yang menghubungkan elemen-elemen data terkait. RDBMS mencakup fungsi-fungsi yang menjaga keamanan, keakuratan, integritas, dan konsistensi data. Hal ini berbeda dengan penyimpanan *file* yang digunakan dalam DBMS.

Perbedaan lain antara sistem manajemen basis data dan sistem manajemen basis data relational meliput:

- a. Jumlah pengguna yang diizinkan: Sementara DBMS hanya dapat menerima satu pengguna dalam satu waktu, RDBMS dapat beroperasi dengan banyak pengguna.
- b. Kebutuhan *hardware* dan *software*: DBMS membutuhkan lebih sedikit *software* dan *hardware* daripada RDBMS.
- c. Jumlah data: RDBMS dapat menangani sejumlah data, dari yang kecil sampai yang besar, sementara DBMS hanya dapat mengelola jumlah yang kecil.
- d. Struktur basis data: Dalam DBMS, data disimpan dalam bentuk hirarkis, sedangkan RDBMS menggunakan tabel di mana tajuk digunakan sebagai nama kolom dan baris berisi nilai yang sesuai.
- e. Implementasi ACID: DBMS tidak menggunakan model *atomicity*, *consistency*, *isolation*, dan *durability* (ACID) untuk menyimpan data. Di sisi lain, RDBMS mendasarkan struktur data mereka pada model ACID untuk memastikan konsistensi.
- f. Basis data terdistribusi: Sementara RDBMS menawarkan dukungan lengkap untuk *database* terdistribusi, DBMS tidak akan memberikan dukungan.
- g. Jenis program yang dikelola: Sementara DBMS membantu mengelola hubungan antara tabel-tabel data yang digabungkan, DBMS berfokus pada pemeliharaan database yang ada di dalam jaringan komputer dan hard disk sistem.
- h. Dukungan normalisasi basis data: RDBMS dapat dinormalisasi, tetapi DBMS tidak.

2.2.5 Object Relational Mapping (ORM)

Object Relational Mapping (ORM) adalah teknik pemrograman yang digunakan untuk mengkonversi data antara sistem tipe yang tidak kompatibel, seperti antara data di *database* dengan kelas/objek dalam bahasa pemrograman. ORM memungkinkan *developer* untuk melakukan *query* dan memanipulasi data dari *database* dengan menggunakan paradigma objek, sehingga data di *database* dapat direpresentasikan sebagai kelas/objek dan sebaliknya. Hal ini memudahkan *developer* dalam melakukan operasi seperti *select*, *insert*, *update*, dan *delete* pada *database*, tanpa harus menggunakan query SQL langsung (Riyanto *et al*, 2023).

ORM meningkatkan efisiensi serta membantu memudahkan orang-orang dengan kemampuan SQL yang minim dalam menghubungkan *database* ke aplikasi. Selain itu, manfaat dan kelebihan ORM termasuk:

- a. Mengurangi biaya dan waktu dalam mengembangkan serta menguji aplikasi, karena kode dalam library ORM sudah lulus uji.
- b. ORM menyediakan fitur lanjutan yang bisa diakses, seperti mendukung transaksi, migrasi, dan *connection pooling*.
- c. Manajemen data *interface* mudah karena tidak perlu membuat skema database.
- d. *Maintenance* mudah karena ketika ada perubahan pada sumber data atau API, hanya ORM yang perlu diubah bukan aplikasinya.
- e. Model program bisa untuk menghubungkan aplikasi dengan kode SQL tanpa menulis ulang kode sehingga menghemat waktu *developer*.
- f. Meningkatkan keamanan aplikasi karena tools ORM didesain untuk menghindari serangan injeksi SQL.

ORM juga memiliki kekurangan, antara lain:

- a. Mempelajari *tools* ORM membutuhkan waktu yang tidak sebentar.
- b. Terkadang ORM dapat membuat kesalahan *mapping* antara objek dan database.
- c. Meski bisa diakses dengan bahasa pemrograman lain, ORM tak bisa 100% diandalkan dan *developer* tetap harus mengetahui SQL dan sintaks ORM.

- d. Jika *query* yang digunakan terlalu kompleks, performa aplikasi yang menggunakan ORM tidak maksimal.
- e. ORM dapat memperlambat performa aplikasi karena tambahan kode yang dituliskan.
- f. ORM yang tidak ditulis dengan baik dapat menghambat skema database dan mengelola migrasi *database*.
- g. Tingkat abstraksi ORM terkadang menghambat *debugging*.

2.2.6 Application Programming Interface (API)

Application Programming Interface (API) adalah sebuah mekanisme yang memungkinkan dua komponen perangkat lunak untuk saling berkomunikasi menggunakan serangkaian aturan, protokol, dan alat yang terstandarisasi. API bertindak sebagai perantara antara berbagai komponen perangkat lunak, memungkinkan mereka untuk saling berbagi data, fungsi, dan sumber daya dengan cara yang terstruktur dan terstandarisasi (Triawan *et al*, 2021). API memainkan peran penting dalam pengembangan perangkat lunak modern, memungkinkan integrasi fitur tanpa harus menambahkan data secara manual, serta memungkinkan aplikasi untuk berinteraksi satu sama lain dengan cara yang aman dan terkontrol. Dengan menggunakan API, pengembang dapat memanfaatkan fungsionalitas yang sudah ada, membangun aplikasi yang lebih kompleks, dan mempercepat siklus pengembangan, sehingga memungkinkan peluncuran produk atau layanan lebih cepat ke pasar.

Arsitektur API biasanya dijelaskan dalam istilah *client* dan *server*. Aplikasi yang mengirimkan permintaan disebut *client*, dan aplikasi yang mengirimkan respons disebut *server*. Jadi dalam kasus website prediksi cuaca, basis data biro cuaca adalah *server*, website atau aplikasi *mobile* adalah *client*.

Ada 4 cara kerja API yang berbeda, bergantung pada waktu dan alasan pembuatannya:

a. SOAP API

API ini menggunakan Simple Object Access Protocol (SOAP). Client dan server bertukar pesan menggunakan XML. Ini adalah API yang kurang fleksibel yang lebih populer di masa lalu.

b. RPC API

API ini disebut Remote Procedure Calls (RPC). Client menyelesaikan suatu fungsi (atau prosedur) di *server*, dan *server* mengirimkan output kembali ke *client*.

c. Websocket API

Websocket API adalah pengembangan API web modern lainnya yang menggunakan objek JSON untuk mengirimkan data. WebSocket API mendukung komunikasi dua arah antara aplikasi *client* dan *server*. *Server* dapat mengirim pesan panggilan balik ke *client* yang terhubung, sehingga lebih efisien daripada REST API.

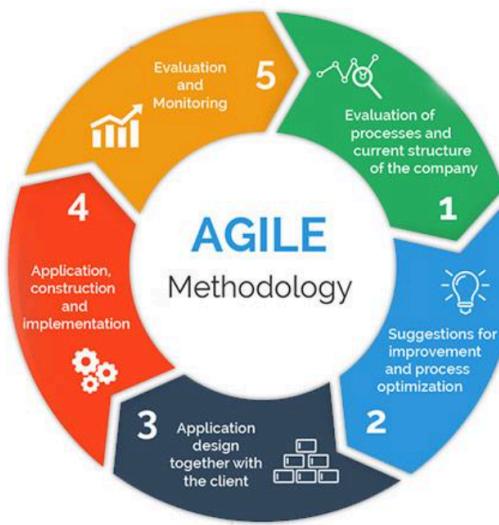
d. REST API

Ini adalah API yang paling populer dan fleksibel yang digunakan di web saat ini. *Client* mengirimkan *request* ke *server* sebagai data. *Server* menggunakan input dari *client* ini untuk memulai fungsi internal dan mengembalikan data output kembali ke *client*.

REST, yang merupakan singkatan dari *Representational State Transfer*, adalah sebuah arsitektur gaya komunikasi dalam pengembangan web yang menetapkan seperangkat prinsip dan fungsi standar. REST ini mendefinisikan serangkaian fungsi atau metode yang digunakan oleh klien untuk berinteraksi dengan dan mengakses data di *server*. Fungsi-fungsi ini termasuk di antaranya adalah metode GET, PUT, DELETE, POST, dan beberapa metode lainnya, yang masing-masing memiliki peran khusus dalam proses pengambilan, pengiriman, penghapusan, dan pembaruan data.

REST API memiliki beberapa fitur utama, salah satu yang paling penting adalah sifatnya yang *stateless*. *Statelessness*, atau ketidakbergantungan pada status sebelumnya, berarti bahwa setiap permintaan (*request*) dari klien ke *server* diproses secara mandiri tanpa mengandalkan informasi atau data dari permintaan sebelumnya. *Server* tidak menyimpan informasi tentang status atau data klien antara satu permintaan dan permintaan berikutnya. Dengan kata lain, setiap kali klien mengirimkan permintaan ke *server*, permintaan tersebut harus memuat semua informasi yang diperlukan untuk diproses, seolah-olah *server* tidak memiliki ingatan tentang interaksi sebelumnya.

2.2.7 Metode *Agile*



Gambar 2.4 Arsitektur Agile

Sumber: blogs.upv.es

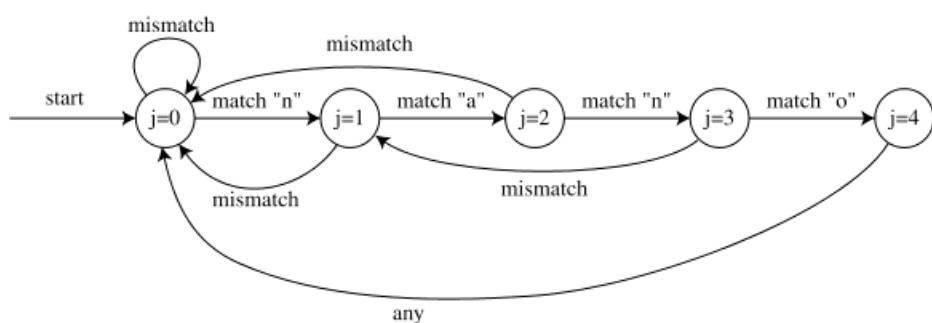
Metode *Agile* adalah metode ekspansi perangkat lunak yang berlandaskan kaidah yang sama atau pengembangan sistem dalam waktu yang singkat dengan mendahuluikan adanya interaksi cepat dari pengembangan terhadap perbedaan yang terjadi dalam bentuk apapun. Tahapan yang digunakan dalam Metode *Agile* adalah *planning*, *implementasi*, *testing*, dokumentasi, *deployment* dan *maintenance* (Handayani *et al*, 2023).

- a. Perencanaan (*planning*) sistem, merupakan salah satu tahapan awal ketika proses kategorisasi yang memerlukan suatu langkah atau tahapan. Pada tahap ini pengembang dan pengguna membuat rancangan atas kesepakatan bersama. Kegiatan yang dilakukan berupa *interview*, observasi serta melihat arsip dokumen yang ada.
- b. Implementasi, yaitu persiapan menu untuk pengguna yang mana dihasilkan dari perancangan sistem baru yang disetujui ke dalam bahasa pemrograman. Pada langkah ini dilakukan dan pendesainan web dan percodingan.
- c. *Testing/pengujian*, merupakan prasyarat utama dari sebuah *system*. Pada langkah ini pengembang melakukan pengembangan sistem ke dalam

bentuk coding, lalu dilakukan uji coba perangkat lunak. *System* baru yang sudah diimplementasikan akan dilakukan pengujian, dimana agar tidak ada *error* atau *bug* saat *system* tersebut dijalankan.

- d. Dokumentasi adalah proses pendokumentasian suatu perangkat, dilakukan dengan merekam langkah demi langkah sistem yang dibangun. Pada langkah ini, hasil pengujian didokumentasikan untuk memudahkan pemeliharaan di masa mendatang.
- e. *Deployment*, dalam tahap ini, pengembang menyebarkan informasi tentang pembaruan layanan kepada pengguna. Proses ini melibatkan pengujian kembali sistem dengan tujuan untuk memverifikasi apakah sistem telah memenuhi persyaratan yang ditetapkan sebelumnya atau belum.
- f. *Maintenance*, memelihara sistem agar dalam kondisi terbaik. Pada langkah ini proses *maintenance* dilakukan secara rutin supaya *software* tetap berjalan dan terjaga sesuai kualitas terbaik dengan seharusnya (Lutfiani *et al*, 2020).

2.2.8 Algoritma Knuth-Morris-Pratt (KMP)



Gambar 2.5 Arsitektur KMP

Sumber: ics.uci.edu

Algoritma KMP adalah salah satu algoritma pencarian string, dikembangkan secara terpisah oleh Donald E. Knuth pada tahun 1967 dan James H. Morris bersama Vaughan R. Pratt pada tahun 1966, namun keduanya mempublikasikannya secara bersamaan pada tahun 1977 (Astuti, 2017). Jika kita melihat algoritma *brute force* lebih mendalam, kita mengetahui bahwa dengan mengingat beberapa perbandingan yang dilakukan sebelumnya kita dapat

meningkatkan besar pergeseran yang dilakukan. Hal ini menghemat perbandingan, yang selanjutnya akan meningkatkan kecepatan pencarian. Secara Sistematis, langkah-langkah yang dilakukan algoritma KMP pada saat pencocokan string adalah sebagai berikut (Nursobah & Pahrudin. P, 2019):

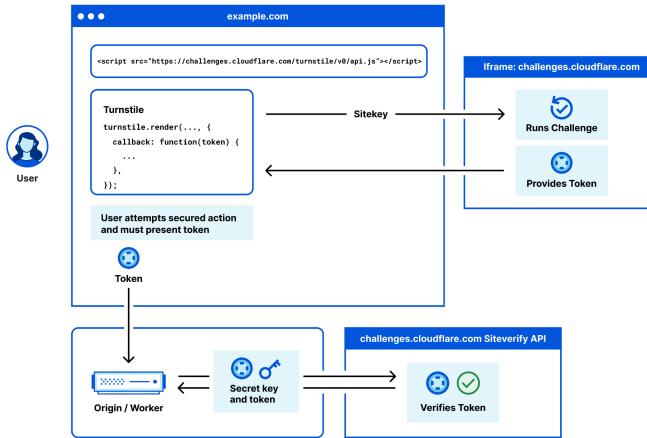
- a. Algoritma KMP mulai mencocokan *pattern* pada awal teks.
- b. Dari kiri ke kanan, algoritma ini akan mencocokan karakter per karakter *pattern*, dengan karakter di teks yang bersesuaian sampai salah satu kondisi berikut terpenuhi.
 1. Karakter di *pattern* dan di teks yang dibandingkan tidak cocok (*mismatch*)
 2. Semua Karakter di *pattern* cocok. Kemudian algoritma akan memberitahu penemuan posisi ini.
- c. Algoritma kemudian menggeser *pattern* berdasarkan table next, lalu menghitung langkah 2 sampai *pattern* berada di ujung teks.

Algoritma Knuth-Morris-Pratt (KMP) merupakan salah satualgoritma pencocokan pola. Metode pencarian KMP bekerja dengan melewatan perbandingan-perbandingan yang tidak diperlukan untuk menghindari besarnya jumlah perbandingan, dengan demikian mencapai waktu berjalan $O(n+m)$ yang optimal dalam kasus terburuk (*worst case*) pencocokan pola algoritma harus memeriksa semua karakter teks dan semua karakter dari pola setidaknya sekali. Ide utama algoritma KMP adalah untuk preprocess string pola P sehingga untuk menghitung fungsi kegagalan f yang menunjukan pergeseran P yang tepat sehingga kita dapat menggunakan kembali perbandingan yang dilakukan sebelumnya (Goodrich *et al*, 2011).

2.2.9 Metode CAPTCHA

Completely Automated Public Turing test to tell Computers and Humans Apart atau yang lebih dikenal dengan CAPTCHA adalah sebuah metode untuk membedakan penggunaan antara pengguna manusia atau bukan (Prasetyo *et al*, 2015). Bertujuan untuk mengidentifikasi dan memisahkan antara akses yang valid oleh pengguna manusia dan dengan akses yang tidak valid dari apapun yang bukan pengguna manusia misalnya robot atau perintah *script* (Radiyah, 2017). Ini

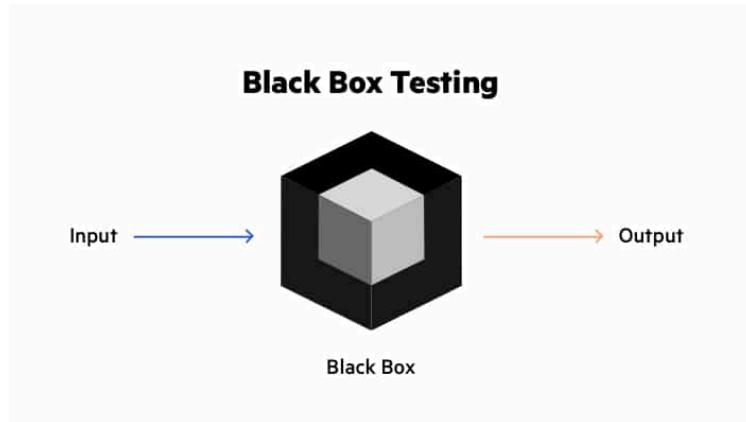
membantu mencegah akses otomatis yang digunakan dalam serangan seperti serangan DDoS, spam, dan pencurian data.



Gambar 2.6 Arsitektur CAPTCHA

Sumber: developers.cloudflare.com

2.2.10 Metode pengujian *Black Box*



Gambar 2.7 Arsitektur Black Box Testing

Sumber: imperva.com

Black Box Testing adalah bentuk pengujian desain yang berfokus pada apakah unit program yang ditentukan dalam spesifikasi terpenuhi atau tidak. Teknik pengujian pada *black box testing* hanya dilakukan dengan menjalankan atau mengeksekusi sebuah unit atau modul, kemudian memonitor apakah hasil

dari unit tersebut sesuai dengan proses bisnis yang diantisipasi. Pengujian yang dilakukan memonitor hasil eksekusi data uji, memeriksa operasi program, dan *black box* hanya melihat tampilan luar (*interface*) saja; dengan kata lain tidak perlu mengetahui cara kerja internal (Achmad *et al*, 2020).

2.2.11 Metode pengujian penelitian

Ada beberapa metode pengujian yang dapat digunakan untuk menguji hasil penelitian tentang aplikasi forum tanya jawab, di antaranya:

- 1. Uji Validitas**

Metode ini digunakan untuk mengukur seberapa validnya instrumen penelitian yang digunakan dalam penelitian. Ada beberapa jenis uji validitas, antara lain uji validitas isi, konstruk, dan kriteria (Jefri *et al*, 2022).

- 2. Uji Reliabilitas**

Metode ini digunakan untuk mengukur seberapa konsisten hasil pengukuran yang diperoleh dari instrumen penelitian. Ada beberapa jenis uji reliabilitas, antara lain uji reliabilitas internal, eksternal, dan test-retest berupa kuisioner. Pengujian reliabilitas pada kuisioner mengukur sejauh mana alat ukur tersebut konsisten dalam mengukur apa yang seharusnya diukur (Sarie *et al*, 2023).

- 3. Analisis Perbandingan**

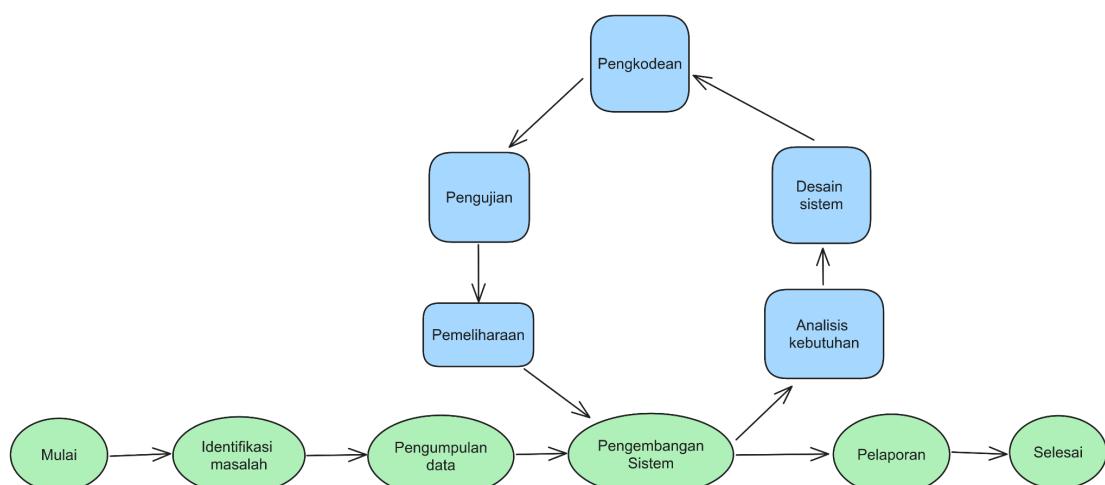
Metode ini digunakan untuk membandingkan hasil penelitian dengan penelitian sebelumnya atau dengan standar yang telah ditetapkan. Analisis perbandingan dapat membantu dalam memperoleh pemahaman yang lebih baik tentang hasil penelitian dan mengevaluasi keefektifan dari strategi pengelolaan persediaan barang yang telah diimplementasikan (Wisnuwardhana *et al*, 2024).

BAB III

METODOLOGI

3.1 Kerangka Pikir

Dalam bab ini, kerangka pikir yang dipilih adalah metodologi *Agile*. Metodologi *Agile* merupakan metodologi pengembangan perangkat lunak yang berfokus pada kolaborasi, adaptasi, dan penyampaian nilai secara berkelanjutan. Metodologi ini dipilih karena sesuai dengan sifat penelitian ini yang bersifat dinamis dan berulang.



Gambar 3.1 Kerangka pikir

3.2 Deskripsi

Deskripsi adalah sebuah aturan atau kaidah dalam komunikasi yang melibatkan proses kompleks untuk mengubah data mentah menjadi sesuatu yang dapat diartikulasikan dan disampaikan dengan cara yang jelas dan tepat. Proses ini melibatkan langkah-langkah yang hati-hati dalam menggambarkan atau menjelaskan informasi sehingga dapat dipahami dengan baik oleh pembaca atau pendengar. Tujuannya adalah untuk menyajikan informasi dengan cara yang memudahkan pemahaman dan memastikan bahwa pesan yang ingin disampaikan mencapai tujuan yang ditetapkan.

3.2.1 Identifikasi Masalah

Tahap pertama dalam pembuatan aplikasi forum tanya jawab yang ada di kerangka pikir penulis adalah mencari permasalahan atau rumusan masalah. Dalam hal ini penulis melakukan observasi terhadap aplikasi yang telah dibuat sebelumnya yaitu Yukitanya agar bisa mengetahui permasalahan apa saja yang masih ada di aplikasi Yukitanya. Beberapa masalah yang terdapat di dalam aplikasi Yukitanya adalah belum ada sistem *filtering* kata pada fitur pembuatan pertanyaan dan jawaban sehingga memungkinkan pengguna untuk membuat pertanyaan dan jawaban yang tidak sopan. Yang kedua yaitu belum adanya fitur rate limit yang menyebabkan pengguna dapat membuat pertanyaan dan jawaban sebanyak mungkin sehingga banyak sekali pertanyaan dan jawaban yang bersifat *spam*. Dan yang terakhir adalah kurangnya minat orang untuk menggunakan aplikasi ini. Berdasarkan data yang diambil dari *database* aplikasi Yukitanya, menunjukan bahwa selama bulan Desember 2023 - Mei 2024 hanya terdapat 109 pengguna, dan rata-rata hanya ada 12 pertanyaan baru/bulan.

3.2.2 Pengumpulan Data

1. Observasi

Metode yang digunakan dalam penelitian ini berfokus pada proses pengumpulan data dengan cara melakukan observasi langsung terhadap aplikasi forum tanya jawab yang telah dikembangkan sebelumnya, yaitu Yukitanya. Dalam proses observasi ini, penulis tidak hanya sekadar melihat bagaimana aplikasi tersebut berfungsi, tetapi juga secara mendalam meneliti berbagai masalah yang mungkin masih terjadi dalam penggunaannya. Pengamatan ini bertujuan untuk mengidentifikasi dan memahami isu-isu yang belum terselesaikan dalam aplikasi Yukitanya, sehingga dapat memberikan wawasan yang lebih komprehensif mengenai area yang memerlukan perbaikan atau pengembangan lebih lanjut.

2. Survei

Survei merupakan proses akuisisi data melalui interaksi wawancara dengan responden yang memiliki relevansi dengan fokus penelitian. Dalam kerangka penelitian terkait aplikasi forum tanya jawab tugas

sekolah di Forest Interactive, pendekatan wawancara diimplementasikan melalui pengumpulan data yang diperoleh dari narasumber yang memiliki kompetensi di *web development*, serta *stakeholder* lain yang berkaitan.

3. Studi Pustaka

Studi kasus dalam penelitian ini melibatkan analisis mendalam dari suatu kasus atau beberapa kasus yang terkait dengan topik penelitian. Dalam penelitian ini penulis menggunakan 3 jurnal yang terdiri dari Rancang Bangun Aplikasi Forum Diskusi Usaha Mikro Kecil dan Menengah Berbasis Web, Aplikasi Forum Diskusi Himpunan Mahasiswa Teknik Universitas Jenderal Soedirman Menggunakan Laravel, serta Aplikasi Forum Diskusi Seputar Game Esport Berbasis Web Dengan Metode *Extreme Programming*.

3.2.3 Pengembangan Sistem

Implementasi metodologi Siklus Hidup Pengembangan Perangkat Lunak (*Software Development Life Cycle*) model *Agile* pada proses pengembangan aplikasi forum tanya jawab dilakukan dengan menjalani serangkaian tahap yang dijalankan secara berurutan, sebagaimana tertera berikut ini:

1. Analisis Kebutuhan

Fase ini melibatkan perencanaan penelitian secara menyeluruh, yang mencakup alokasi sumber daya, penjadwalan, dan estimasi anggaran yang diperlukan. Selain itu, penulis diharapkan untuk membuat rencana terperinci untuk pengembangan aplikasi forum tanya jawab yang mencakup prosedur pengembangan, pemilihan tim yang terlibat, dan penentuan tanggung jawab setiap anggota tim. Setelah data diperoleh, langkah berikutnya adalah merencanakan dan merancang aplikasi forum tanya jawab. Dimulai dengan melakukan analisis menyeluruh terhadap kebutuhan *hardware* dan *software* yang diperlukan untuk membuat aplikasi forum tanya jawab.

2. Desain Sistem

Setelah analisis kebutuhan selesai, tahap desain dimulai untuk membuat arsitektur sistem dan antarmuka pengguna untuk aplikasi forum

tanya jawab. Pada tahap ini, perencanaan basis data, perancangan struktur program, dan desain antarmuka pengguna (UI) yang mudah dipahami dan ramah pengguna dilakukan dengan menggunakan alat desain seperti Balsamiq.

Sistem desain berfokus pada pembuatan desain basis data yang sesuai untuk menyimpan dan mengelola data dari forum tanya jawab. Ini mencakup identifikasi tabel dan kolom serta hubungan antara entitas dalam aplikasi. Struktur program juga dirancang untuk memastikan bahwa komponen perangkat lunak berinteraksi dengan baik, yang mencakup bagaimana proses bisnis akan diterapkan dalam kode.

Merancang antarmuka pengguna (UI) yang dapat dipahami dengan mudah oleh pengguna akhir adalah bagian yang sangat penting dari tahap desain. Dalam hal ini, alat desain seperti Balsamiq digunakan untuk membuat desain *Wireframe* yang memberikan gambaran visual tentang tata letak dan komponen antarmuka. Tujuan utamanya adalah membuat antarmuka yang mudah digunakan dan ramah untuk pengguna aplikasi.

Tahap desain dapat menghasilkan pedoman yang kuat untuk membangun aplikasi forum tanya jawab yang berfungsi dengan kebutuhan yang telah diidentifikasi sebelumnya dengan melakukan proses ini dengan hati-hati.

3. Pengkodean

Tahap ini melibatkan aspek perencanaan komprehensif dari penelitian secara menyeluruh, yang mencakup identifikasi dan alokasi sumber daya yang diperlukan, penjadwalan kegiatan, serta estimasi anggaran yang relevan. Selain itu, dalam tahap ini penulis juga diharapkan untuk merumuskan rencana detail terkait pengembangan aplikasi forum tanya jawab. Rencana ini mencakup berbagai tahapan dalam proses pengembangan serta aspek perancangan yang melibatkan:

a. Perancangan Diagram

Pada tahap ini dilakukan dengan membuat desain menggunakan bahasa pemodelan *Unified Modelling Language* (UML). UML yang digunakan dalam penelitian ini terdiri dari:

1) *Class Diagram*

Class Diagram, untuk menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Class diagram yang akan dibuat adalah sebagai berikut:

- a) User
- b) *Subject*
- c) *Question*
- d) *Answer*
- e) *Favorite*
- f) *Membership*
- g) *Notification*
- h) *Old Slug*
- i) *Question Image*
- j) *Rating*

2) *Use case Diagram*

Use case diagram digunakan untuk menggambarkan secara umum hubungan antara aktor (*actor*) dan kasus penggunaan (*use case*) dalam suatu sistem aplikasi. Diagram ini membantu dalam pemahaman tentang interaksi antara aktor (entitas yang berinteraksi dengan sistem) dan berbagai fungsi (*use case*) yang ada dalam sistem tersebut. Dalam penelitian ini, terdapat satu aktor yaitu “User” dan 10 kasus penggunaan (*use case*) yang mencakup:

- a) *Auth*: Mengatur proses autentikasi user seperti proses Registrasi, Login, dan Logout
- b) Pertanyaan: Mengelola pertanyaan seperti membuat pertanyaan baru, melihat daftar pertanyaan, memfilter pertanyaan berdasarkan mata pelajaran, mencari pelajaran berdasarkan kata kunci, mengedit pertanyaan, menghapus pertanyaan, memfavoritkan pertanyaan, dan membagikan pertanyaan.

- c) Jawaban: Mengelola jawaban seperti memberi jawaban baru, memberi rating ke jawaban lain, menandai jawaban sebagai jawaban terbaik, dan menyalin jawaban.

3) *Activity Diagram*

Activity diagram digunakan untuk memodelkan rangkaian aktivitas dalam suatu sistem, mencakup representasi visual dari urutan aktivitas yang dilakukan oleh pengguna dalam aplikasi. Diagram ini memberikan pandangan keseluruhan tentang bagaimana aktivitas berlangsung, berinteraksi, dan mengalir dalam sistem. Dalam penelitian ini, penulis akan membuat beberapa *activity diagram* yang berkaitan dengan sistem aplikasi.

Di antaranya:

- a) Proses registrasi
- b) Proses login
- c) Proses membuat pertanyaan baru
- d) Proses tanyakan pertanyaan pada AI
- e) Proses menambahkan jawaban
- f) Proses mencari pertanyaan
- g) Proses pembelian membership
- h) Proses akses menu profil

b. Pembuatan aplikasi

Setelah melakukan perancangan diagram dan antarmuka, tahap selanjutnya adalah melakukan implementasi perancangan yang telah dibuat sebelumnya dengan cara menerjemahkan ke dalam bentuk pengkodean secara nyata dengan memanfaatkan bahasa pemrograman TypeScript, *framework* Next.js, Tailwind CSS, dan *database* PostgreSQL.

4. Pengujian

Sedangkan pada tahap pengujian untuk memastikan apakah hasil sesuai dengan rancangan yang diharapkan atau belum. Apabila masih ada kekurangan maka kembali ke tahap implementasi untuk diperbaiki sampai

benar-benar sesuai dengan rancangan pengujian pada sistem menggunakan metode pengujian *Black Box*, validitas, dan reliabilitas.

5. Pemeliharaan

Pada tahap ini, kinerja aplikasi dipantau secara teratur, masalah ditangani, dan penyempurnaan atau koreksi dilakukan jika diperlukan. Untuk memastikan integritas dan kinerja optimal dari aplikasi forum tanya jawab, serta untuk menanggapi perubahan kebutuhan dan dinamika yang mempengaruhi kinerjanya, tindakan ini dilakukan.

3.2.4 Pelaporan

Tahap akhir dari perjalanan adalah penyusunan laporan skripsi, yang merupakan bagian esensial dalam persyaratan kelulusan, dan penyusunannya diarahkan sesuai dengan pedoman yang diuraikan dalam panduan Penulisan Skripsi Fakultas Teknologi Informasi Universitas Bale Bandung. Laporan skripsi ini adalah hasil akhir dari upaya penelitian yang mencerminkan kesesuaian dengan standar ilmiah dan kepatuhan terhadap ketentuan yang ditetapkan oleh panduan Penulisan Skripsi.

BAB IV

ANALISIS DAN PERANCANGAN

4.1 Analisis

Penelitian dimulai dengan analisis untuk menentukan kebutuhan dalam merancang aplikasi. Penulis telah menganalisis program untuk membantu pengembangan aplikasi forum tanya jawab. Analisis ini mencakup masalah, perangkat lunak, pengguna, antarmuka pengguna, fitur, dan biaya.

4.1.1 Analisis Masalah

Langkah pertama adalah menganalisis aplikasi forum tanya jawab Yukitanya untuk mengidentifikasi masalah yang ada. Analisis langsung pada aplikasi tersebut menemukan beberapa kekurangan, yaitu tidak adanya sistem filter kata pada fitur pembuatan pertanyaan dan jawaban, yang memungkinkan pengguna membuat konten yang tidak sopan. Kedua, tidak ada fitur *rate limit*, sehingga pengguna dapat membuat pertanyaan dan jawaban dalam jumlah banyak, menghasilkan banyak *spam*. Terakhir, minat pengguna terhadap aplikasi ini rendah. Data dari *database* Yukitanya menunjukkan bahwa dari Desember 2023 hingga Mei 2024, hanya terdapat 109 pengguna dengan rata-rata 12 pertanyaan baru per bulan.

4.1.2 Analisis *Software*

Berdasarkan analisis kebutuhan, untuk memenuhi persyaratan dalam pengembangan aplikasi forum tanya jawab diperlukan perangkat lunak sebagai alat penunjang. *Software* yang dibutuhkan untuk mendukung penelitian ini adalah sebagai berikut:

1. Laptop Lenovo Thinkpad T470
2. Sistem Operasi NixOS
3. VSCodium sebagai *text editor* digunakan untuk melakukan pengkodean dalam pembuatan aplikasi.

4. Bahasa Pemrograman TypeScript digunakan untuk membuat aplikasi ini mulai dari sisi *front-end* sampai ke *back-end*.
5. Node.js digunakan sebagai *runtime environment* untuk menjalankan aplikasi ini di *server*.
6. Next.js digunakan sebagai *web framework* dalam pengembangan aplikasi ini.
7. PostgreSQL digunakan sebagai *database* untuk menyimpan data-data dalam aplikasi ini.
8. Brave yaitu *web browser* yang digunakan untuk melihat halaman web aplikasi yang sudah dibuat.
9. Draw.io digunakan untuk merancang diagram UML.
10. Balsamiq digunakan untuk membuat rancangan antarmuka aplikasi yang akan dibuat.

4.1.3 Analisis Pengguna

Penganalisaan pengguna berkaitan dengan siapa yang akan menggunakan aplikasi forum tanya jawab ini. Pengguna aplikasi forum tanya jawab ini adalah siswa SD - SMA. Supaya nantinya aplikasi ini dapat digunakan dengan nyaman oleh para siswa, maka tampilan aplikasi ini harus dibuat sederhana dan *user-friendly*.

4.1.4 User Interface

User Interface dari aplikasi forum tanya jawab sangat mempengaruhi pengalaman dan kenyamanan pengguna. *User Interface* ini dibuat mirip dengan media sosial agar pengguna merasa lebih familiar saat menggunakannya. Tujuan dari desain ini adalah agar pengguna dapat lebih cepat beradaptasi dengan *user interface* tersebut.

Pada aplikasi forum tanya jawab ini, akan terdapat 9 *user interface* yang ditampilkan sebagai berikut:

1. Halaman Registrasi
2. Halaman Login
3. Halaman Beranda

4. Halaman Detail Pertanyaan
5. Halaman Cari
6. Halaman Notifikasi
7. Halaman Mata Pelajaran
8. Halaman Premium
9. Halaman Profil

4.1.5 Fitur-fitur

Fitur digunakan dalam aplikasi forum tanya jawab ini dimaksudkan agar pengguna dapat membuat pertanyaan dan pengguna lain dapat menjawab pertanyaan tersebut. Pengguna lain pun dapat memberi nilai kepada jawab yang ada mulai dari bintang 1 - bintang 5. Khusus untuk pengguna pembuat pertanyaan, dapat menandai salah satu jawaban menjadi “jawaban terbaik” yang akan muncul di paling atas jawaban yang lain. Berikut adalah fitur-fitur yang disediakan aplikasi forum tanya jawab:

1. Halaman Beranda yang berfungsi untuk menampilkan semua pertanyaan dari yang terbaru ke terlama dalam bentuk *timeline* seperti di sosial media pada umumnya.
2. Adanya fitur untuk membuat pertanyaan baru, pengguna juga bisa memasukkan gambar dan memilih mata pelajaran yang cocok dengan pertanyaan tersebut.
3. Adanya fitur CAPTCHA untuk mencegah pengguna membuat pertanyaan terlalu sering.
4. Adanya fitur *filtering* kata pada proses pembuatan pertanyaan dan jawaban untuk mencegah adanya kata-kata terlarang.
5. Halaman Detail Pertanyaan yang berfungsi untuk menampilkan detail pertanyaan mulai dari isi pertanyaan, tanggal dibuat pertanyaan, jumlah favorit, jumlah jawaban, dan jawaban-jawaban yang telah dibuat.
6. Adanya fitur untuk menandai sebuah pertanyaan sebagai pertanyaan favorit, fungsinya mirip seperti fitur *like* pada sosial media.
7. Adanya fitur untuk membuat jawaban untuk pertanyaan yang dipilih.

8. Adanya fitur rating untuk memberi nilai kepada sebuah jawaban mulai dari bintang 1 - 5.
9. Khusus untuk pemilik pertanyaan, dia dapat menandai sebuah jawaban sebagai “jawaban terbaik” yang akan selalu muncul di paling atas jawaban lainnya.
10. Adanya fitur “Tanyakan pada AI” untuk menjawab pertanyaan secara otomatis menggunakan Google Gemini API.
11. Adanya fitur *Share* untuk membagikan pertanyaan ke sosial media lain seperti Facebook, X, WhatsApp, dan Telegram.
12. Halaman pencarian yang berfungsi untuk mencari pertanyaan berdasarkan kata kunci dan mata pelajaran. Fitur ini bisa juga digunakan untuk mencari profil pengguna lain berdasarkan *username* dan nama.
13. Halaman mata pelajaran yang berfungsi untuk melihat daftar pertanyaan berdasarkan mata pelajaran tertentu.
14. Halaman profil yang berfungsi untuk melihat informasi yang berkaitan dengan pengguna tersebut seperti foto profil, nama, *username*, tanggal bergabung, jumlah pertanyaan, jumlah jawaban, jumlah favorit, pertanyaan, jawaban yang telah dia pernah buat.

4.1.6 Analisis Data

Analisis data adalah proses penting untuk memahami data yang digunakan dalam penelitian skripsi ini, mencakup berbagai pertanyaan dan jawaban yang telah dibuat oleh pengguna. Dalam proses ini, data dikumpulkan, disortir, dan diinterpretasikan untuk memastikan semua informasi yang diperlukan tersedia dan relevan. Setiap pertanyaan yang dikirimkan oleh pengguna akan ditampilkan di halaman beranda aplikasi dalam bentuk *timeline*, memungkinkan pengguna lain untuk melihat dan menelusuri pertanyaan tersebut secara kronologis. Ini memberikan gambaran umum tentang topik dan masalah yang sedang dibahas. Jawaban dari setiap pertanyaan akan ditampilkan di halaman detail pertanyaan, memungkinkan pengguna untuk melihat semua tanggapan yang telah diberikan dan mengikuti diskusi dengan lebih mendalam, serta menemukan solusi atau informasi yang mereka butuhkan. Melalui analisis data pertanyaan dan jawaban

ini, peneliti dapat memperoleh wawasan berharga tentang interaksi pengguna dengan aplikasi, topik yang paling sering dibahas, dan cara meningkatkan pengalaman pengguna secara keseluruhan.

4.1.7 Analisis Biaya

Berikut adalah daftar biaya yang diperlukan pada penelitian skripsi ini adalah sebagai berikut:

Tabel 4.1 Analisis Biaya

No	Jenis Kebutuhan	Biaya
1	Biaya Internet	Rp1.560.000
2	Biaya ATK	Rp634.000
3	Programming	Rp3.500.000
4	Analisis	Rp1.500.000
5	Biaya Bahan Bakar	Rp400.000
Jumlah		Rp7.594.000

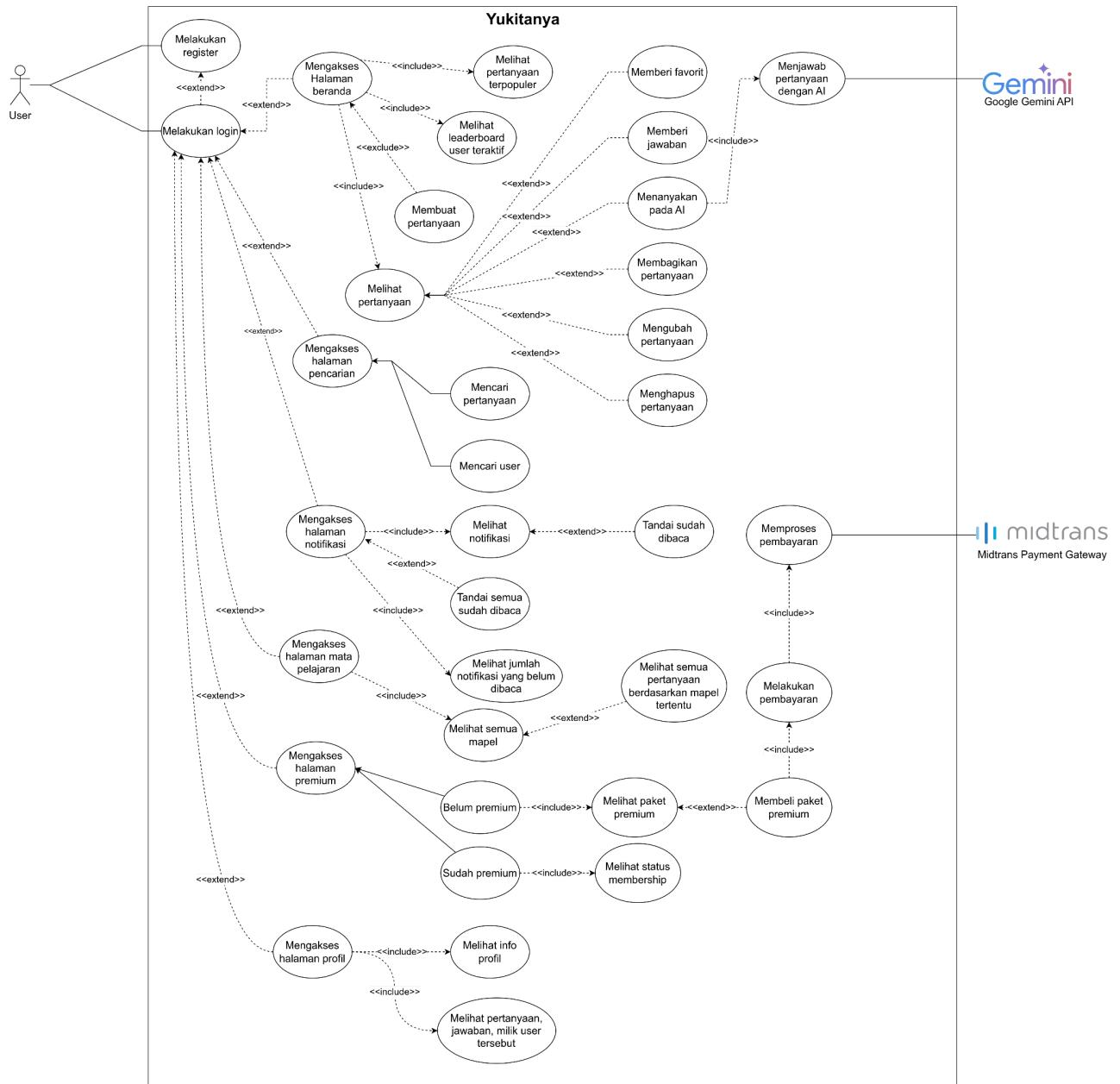
4.2 Perancangan

Sebelum memulai pembuatan aplikasi, diperlukan tahap perancangan yang mendalam. Perancangan ini mencakup pembuatan desain *software* yang biasanya diwujudkan dalam bentuk *Unified Modeling Language* (UML), bahasa pemodelan standar untuk menggambarkan, merancang, dan mendokumentasikan sistem perangkat lunak. UML menggunakan beberapa jenis diagram, seperti *use case diagram* yang menunjukkan interaksi antara pengguna dan sistem, *activity diagram* yang menggambarkan alur kerja dalam sistem, dan *class diagram* yang memetakan struktur dan hubungan antar kelas. Dengan perancangan ini, pengembang mendapatkan gambaran jelas tentang cara kerja aplikasi dan interaksi komponennya, sehingga dapat mengurangi risiko kesalahan dan memastikan semua persyaratan pengguna terpenuhi, menghasilkan aplikasi yang berfungsi sesuai harapan dan kebutuhan.

4.2.1 UML (*Unified Modelling Language*)

1. Use Case Diagram

Pada rancangan *Use Case Diagram* bertujuan untuk memberikan gambaran interaksi antara aktor dengan aplikasi forum tanya jawab tugas sekolah. Berikut adalah penjelasan *use case diagram* aplikasi forum tanya jawab tugas sekolah:



Gambar 4.1 Use Case Diagram Yukitanya

Penjelasan mengenai *Use Case* Diagram yang terdapat di atas dapat secara terinci diuraikan dalam sebuah tabel deskripsi yang telah disediakan:

a. Deskripsi Aktor

Tabel 4.2 Deskripsi Aktor

No	Aktor	Deskripsi
1	User	User merupakan semua pengguna yang dapat menggunakan aplikasi ini.

b. Deskripsi *Use Case*

1) Skenario *use case register*

Tabel 4.3 Deskripsi Use Case Register

Nama	Register
Aktor	User
Skenario Utama	
Kondisi awal	Aktor belum memiliki akun yang diperlukan untuk mengakses aplikasi ini.
Aktor	Sistem
Mengakses halaman register	Menampilkan halaman register
Mengisi form register	Memverifikasi <i>email</i> dan <i>username</i> , meng- <i>hashing password</i> , memasukkan data <i>user</i> ke <i>database</i>
Kondisi akhir	<i>User</i> memiliki akun yang

	dapat digunakan untuk <i>login</i>
--	------------------------------------

2) Skenario *use case login*

Tabel 4.4 Deskripsi Use Case Login

Nama	Login
Aktor	User
Skenario Utama	
Kondisi awal	Aktor belum memiliki akses pada aplikasi
Aktor	Sistem
Mengakses halaman login	Menampilkan halaman login
Mengisi form login	Memvalidasi <i>username</i> dan <i>password</i> . Jika berhasil, maka akan menampilkan halaman beranda.
Kondisi akhir	<i>User</i> masuk ke dalam halaman beranda, dan user dapat melihat pertanyaan yang ada di beranda.

3) Skenario *use case* mengakses halaman beranda

Tabel 4.5 Deskripsi Use Case Mengakses Halaman Beranda

Nama	Mengakses Halaman Beranda
Aktor	User
Skenario Utama	

Kondisi awal	Aktor berhasil login dan mengakses halaman beranda
Aktor	Sistem
User mengisi form <i>login</i> terlebih dahulu	Setelah berhasil <i>login</i> , sistem menampilkan halaman <i>dashboard</i>
Mengakses <i>sidebar</i>	Menampilkan pertanyaan yang paling sering ditanyakan dan daftar peringkat pengguna yang paling aktif dalam 1 musim terakhir.
Buat pertanyaan, masukkan ke <i>input field</i> , dan pilih mata pelajaran.	Melakukan validasi kepada pertanyaan dan memasukan pertanyaan ke dalam <i>database</i>
Melihat pertanyaan	Mengambil data-data pertanyaan dari <i>database</i> dan menampilkannya di beranda
Mengubah pertanyaan	Melakukan validasi kepada pertanyaan dan memasukan pertanyaan ke dalam <i>database</i>
Menghapus pertanyaan	Menghapus pertanyaan pada <i>database</i>
Kondisi akhir	Menampilkan pertanyaan-pertanyaan dan fitur-fitur lain

- 4) Skenario *use case* mengakses halaman pencarian

Tabel 4.6 Deskripsi Use Case Mengakses Halaman Pencarian

Nama	Mengakses Halaman Pencarian
Aktor	User
Skenario Utama	
Kondisi awal	Aktor berhasil login dan mengakses halaman pencarian
Aktor	Sistem
Mencari pertanyaan, memasukkan <i>keyword</i>	Mencari data pertanyaan berdasarkan <i>keyword</i> yang relevan di <i>database</i>
Mencari pengguna, memasukkan nama atau <i>username</i>	Mencari data pengguna berdasarkan nama dan <i>username</i> yang relevan di <i>database</i>
Kondisi Akhir	Data pengguna dan pertanyaan yang dicari ditampilkan

- 5) Skenario *use case* mengakses halaman notifikasi

Tabel 4.7 Deskripsi Use Case Mengakses Halaman Notifikasi

Nama	Mengakses Halaman Notifikasi
Aktor	User
Skenario Utama	

Kondisi awal	Aktor berhasil login dan mengakses halaman notifikasi
Aktor	Sistem
User masuk ke halaman notifikasi	Menampilkan data-data notifikasi dari <i>database</i>
Menandai notifikasi sudah dibaca, menekan tombol ceklis	Mengubah status notifikasi menjadi sudah dibaca di <i>database</i>

- 6) Skenario *use case* mengakses halaman mata pelajaran

Tabel 4.8 Deskripsi Use Case Mengakses Halaman Mata Pelajaran

Nama	Mengakses Halaman Mata Pelajaran
Aktor	User
Skenario Utama	
Kondisi awal	Aktor berhasil login dan mengakses halaman mata pelajaran
Aktor	Sistem
User masuk ke halaman mata pelajaran	Menampilkan daftar lengkap mata pelajaran yang tersedia untuk dipilih.

User memilih salah satu mata pelajaran	Menampilkan pertanyaan-pertanyaan berdasarkan mata pertanyaan yang dipilih
Kondisi akhir	Jenis mata pelajaran dan kumpulan pertanyaan yang telah dipilih berhasil ditampilkan dengan lengkap.

7) Skenario *use case* mengakses halaman premium

Tabel 4.9 Deskripsi Use Case Mengakses Halaman Premium

Nama	Mengakses Halaman Premium
Aktor	User
Skenario Utama	
Kondisi awal	Aktor berhasil login dan mengakses halaman premium
Aktor	Sistem
User masuk ke halaman premium	Sistem menampilkan paket premium yang tersedia
User memilih paket premium yang dipilih	Sistem menampilkan halaman pembayaran
User melakukan pembayaran	Sistem menyimpan data <i>membership</i> di <i>database</i>
Kondisi akhir	Paket premium yang tersedia ditampilkan dan user dapat melakukan pembelian

8) Skenario *use case* mengakses halaman profil

Tabel 4.10 Deskripsi Use Case Mengakses Halaman Profil

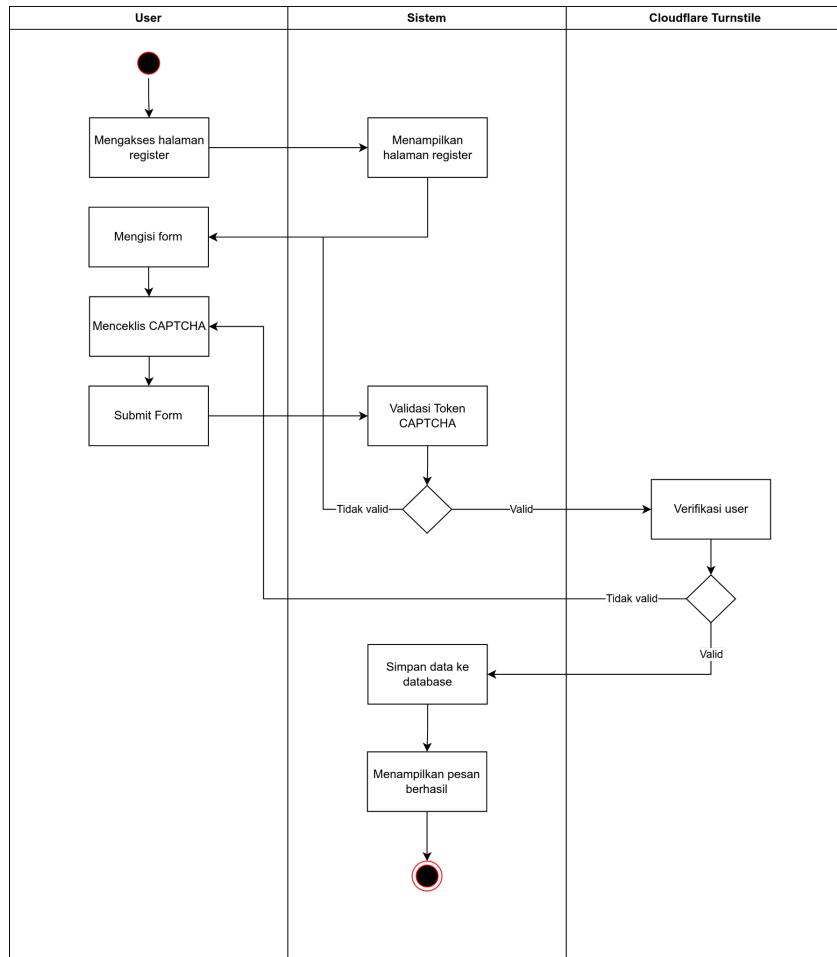
Nama	Mengakses Halaman Profil
Aktor	User
Skenario Utama	
Kondisi awal	Aktor berhasil login dan mengakses halaman profil
Aktor	Sistem
User membuka profil sendiri atau orang lain	Sistem menampilkan data yang relevan dengan user tersebut
Kondisi akhir	Data profil user berhasil ditampilkan

2. *Activity Diagram*

Activity diagram mengilustrasikan urutan dan aliran aktivitas dalam proses sebuah sistem secara visual. Berikut ini adalah contoh *activity diagram* yang digunakan dalam aplikasi Yukitanya untuk menggambarkan langkah-langkah yang terlibat dalam berbagai proses.

a. *Activity Diagram Register*

Dalam *activity diagram* yang dirancang untuk proses registrasi, terdapat serangkaian langkah yang terinci yang menunjukkan bagaimana pengguna dapat mendaftarkan akun mereka dalam aplikasi dengan efisien. Diagram ini memberikan gambaran visual tentang tahapan-tahapan yang harus dilalui pengguna mulai dari awal proses hingga akhir, mencakup verifikasi informasi dan pembuatan akun yang sesuai dengan kebutuhan aplikasi, sebagaimana terlihat dalam ilustrasi di bawah ini:



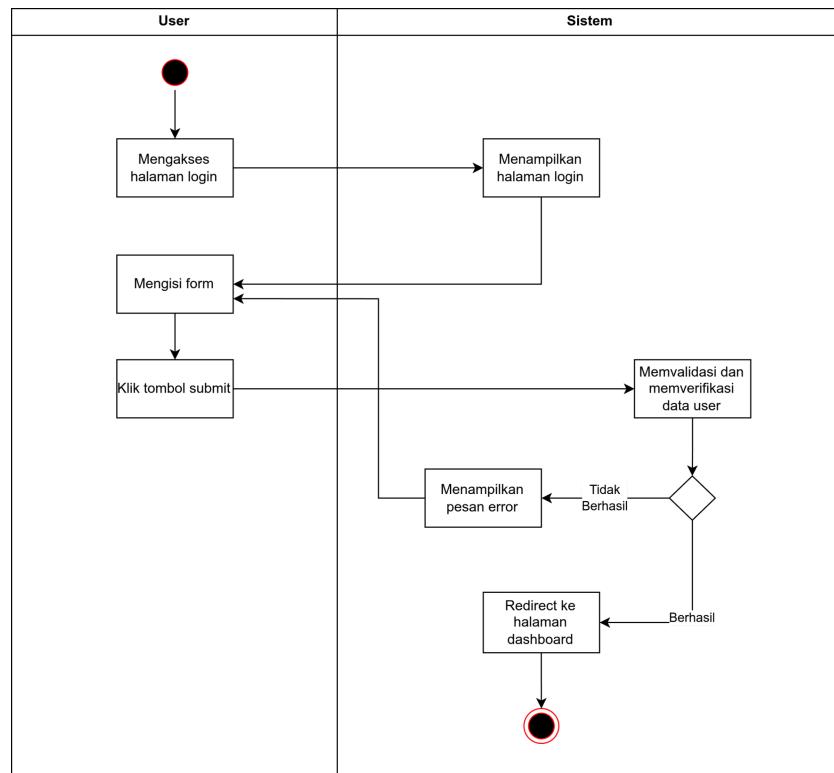
Gambar 4.2 Activity Diagram Register

Gambar 4.2 menjelaskan alur registrasi pada aplikasi forum tanya jawab tugas sekolah, di mana pengguna memulai dengan mengisi data seperti Nama, *Username*, *Email*, dan *Password*. Setelah semua data diisi, pengguna diminta untuk mencentang CAPTCHA sebagai langkah verifikasi tambahan. Sistem kemudian akan memvalidasi token CAPTCHA untuk memastikan bahwa proses dilakukan oleh manusia, bukan bot. Setelah validasi CAPTCHA berhasil, sistem akan melanjutkan dengan memeriksa apakah data yang dimasukkan, khususnya *Email* dan *Username*, sudah pernah digunakan sebelumnya. Jika semua data dinyatakan valid, sistem akan menyimpan informasi tersebut ke dalam *database* dan menampilkan pesan "Registrasi berhasil" kepada pengguna. Sebaliknya, jika

terdapat kesalahan atau data tidak valid, sistem akan menampilkan pesan error yang memberi tahu pengguna mengenai masalah tersebut, sehingga pengguna dapat melakukan perbaikan pada data yang dimasukkan.

b. *Activity Diagram Login*

Dalam *activity diagram login*, terdapat alur yang menggambarkan proses untuk melakukan autentikasi masuk ke dalam aplikasi, seperti yang terlihat pada gambar di bawah ini:

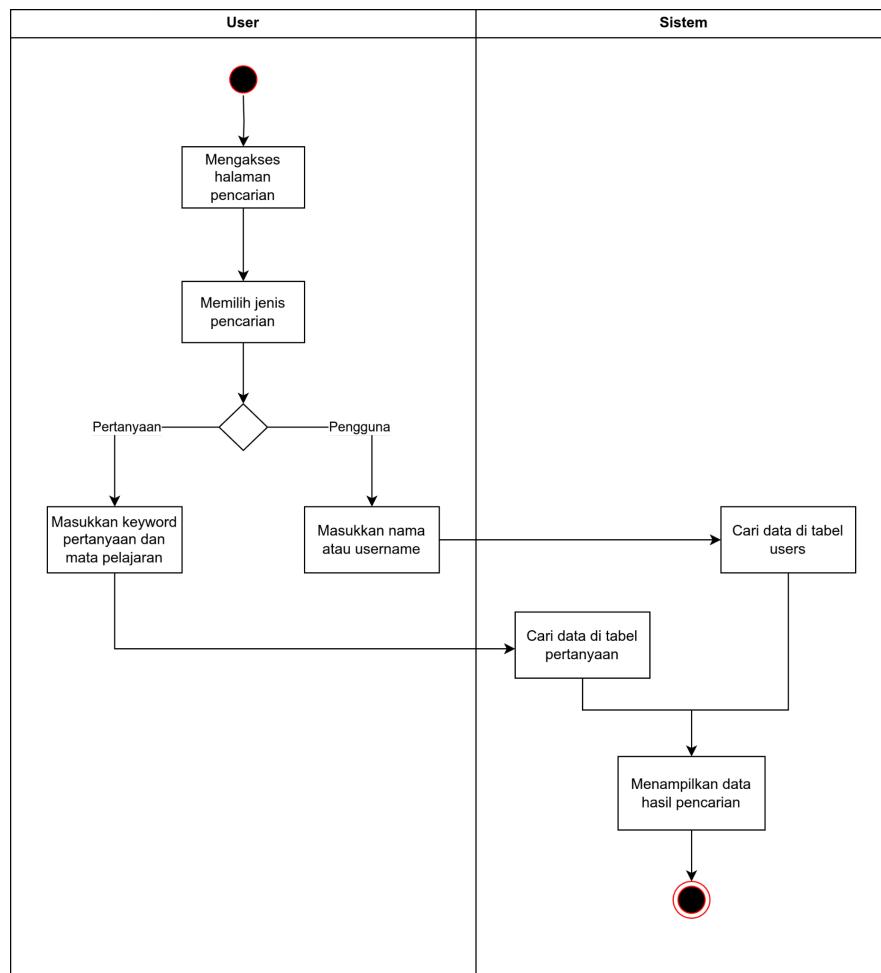


Gambar 4.3 Activity Diagram Login

Pada gambar 4.3 menunjukkan bahwa *user* akan melakukan *login* aplikasi forum tanya jawab tugas sekolah dengan memasukkan *username* dan *password*, kemudian sistem akan melakukan validasi terhadap input *login* yang dimasukkan oleh *user*, apabila data sudah sesuai maka sistem akan menampilkan halaman beranda namun apabila data tidak sesuai sistem akan menampilkan pesan *error* dan kembali ke form *login*.

c. *Activity Diagram* Mengakses Halaman Pencarian

Dalam *activity diagram* yang menggambarkan proses mengakses halaman pencarian, terdapat urutan langkah-langkah yang menunjukkan tahapan-tahapan dalam pencarian data. Diagram ini memberikan ilustrasi yang jelas tentang setiap langkah yang harus diambil pengguna saat mencari data, dimulai dari awal hingga hasil pencarian ditampilkan. Langkah-langkah ini diatur secara berurutan, mencerminkan alur kerja sistem saat menangani permintaan pencarian data. Diagram ini sangat membantu dalam memahami bagaimana proses pencarian data terjadi dan bagaimana setiap komponen dalam sistem berinteraksi untuk menghasilkan informasi yang diinginkan.

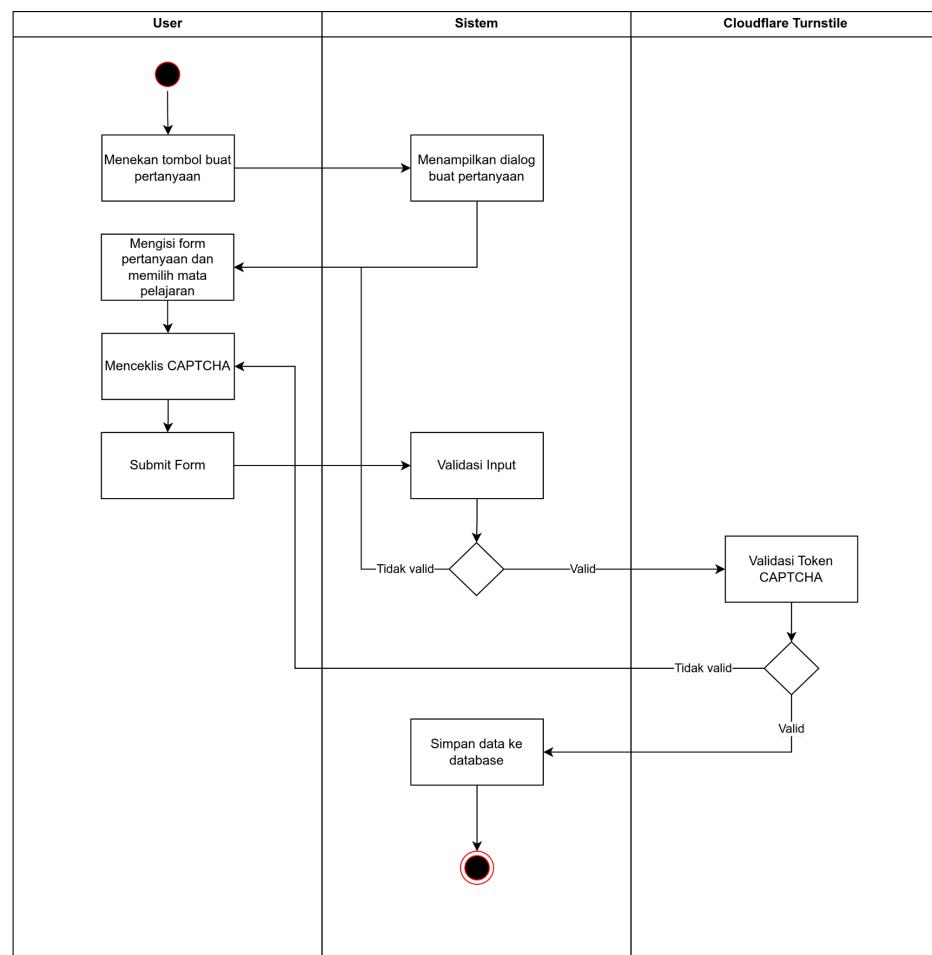


Gambar 4.4 Activity Diagram Halaman Pencarian

Gambar 4.4 menunjukkan bahwa halaman pencarian menawarkan dua opsi utama: mencari pertanyaan atau mencari pengguna lain. Untuk mencari pertanyaan, pengguna memasukkan kata kunci dan mata pelajaran terkait, sedangkan untuk mencari pengguna lain, mereka memasukkan nama atau *username*. Halaman ini memudahkan pengguna menemukan informasi atau individu yang diinginkan di *platform*.

d. *Activity Diagram* Membuat Pertanyaan

Dalam *activity diagram* membuat pertanyaan, terdapat urutan proses yang mengilustrasikan tahapan pembuatan pertanyaan, sebagaimana tercermin dalam gambar berikut ini:

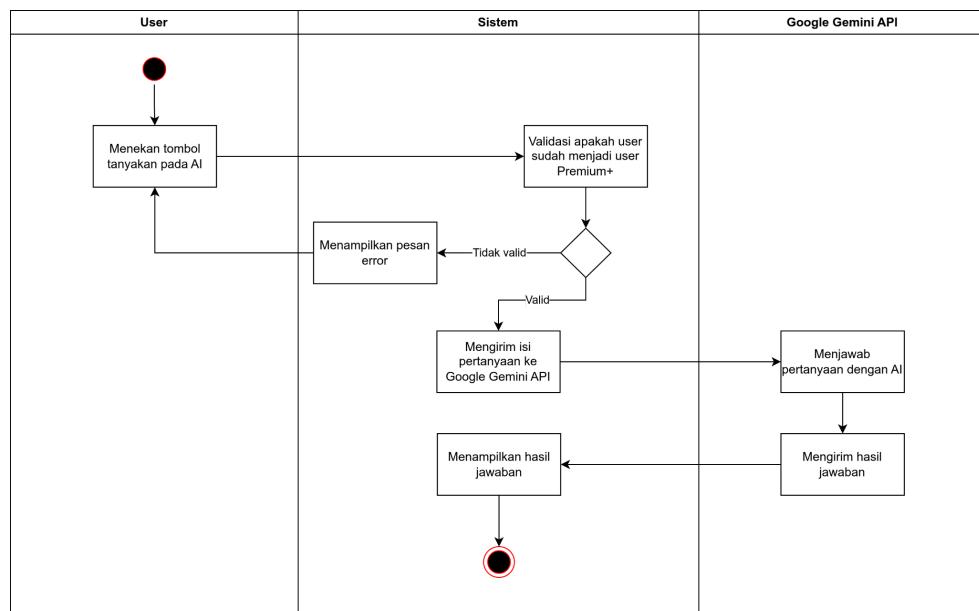


Gambar 4.5 Activity Diagram Membuat Pertanyaan

Pada gambar 4.5 menjelaskan bahwa setelah *user* menekan tombol buat pertanyaan, maka sistem akan menampilkan dialog yang berisi form untuk membuat pertanyaan. Setelah *user* mengisi pertanyaan dan memilih mata pelajaran, *user* diharuskan untuk menceklis CAPTCHA untuk bisa lanjut ke proses seterusnya. Setelah menekan tombol submit, sistem akan melakukan validasi input, termasuk proses *filtering* kata terlarang. Jika berhasil maka lanjut ke proses validasi token CAPTCHA yang akan dilakukan oleh server Cloudflare Turnstile. Jika token valid maka isi pertanyaan tadi akan disimpan di database.

e. *Activity Diagram* Fitur Tanyakan pada AI

Dalam *activity diagram* fitur tanyakan pada AI, terdapat urutan proses yang mengilustrasikan tahapan integrasi dengan Google Gemini API, sebagaimana tercermin dalam gambar berikut ini:



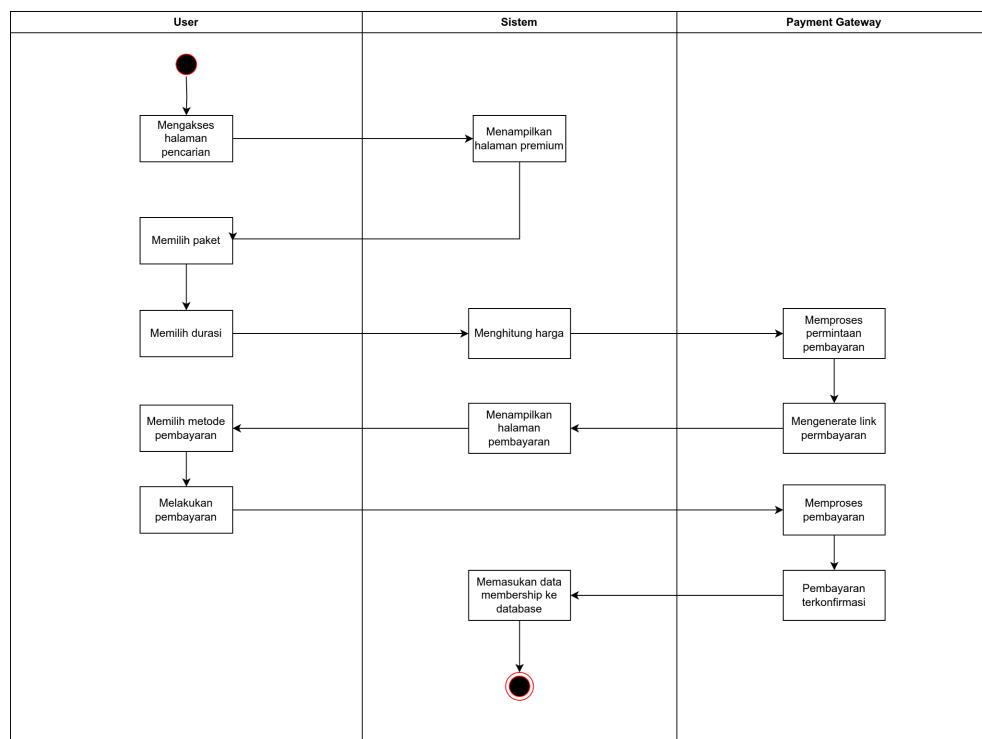
Gambar 4.6 *Activity Diagram* Fitur Tanyakan pada AI

Pada gambar 4.6 menjelaskan bahwa setelah *user* menekan tombol “Tanyakan pada AI”, maka sistem melakukan validasi apakah user telah menjadi anggota membership Premium+ atau belum. Jika sudah

maka sistem akan mengirimkan pertanyaan yang diminta tadi ke Google Gemini API. Setelah itu Google Gemini akan berusaha untuk menjawab pertanyaan tadi lalu mengirimkan jawaban kembali kepada sistem. Saat sistem telah menerima jawabannya, maka akan ditampilkan kepada *user*

f. *Activity Diagram* Mengakses Halaman Premium

Dalam *activity diagram* mengakses halaman premium, terdapat sebuah proses yang mengilustrasikan langkah-langkah pembelian paket premium yang dapat dilakukan oleh *user*; sebagaimana yang tergambar dalam gambar di bawah ini:



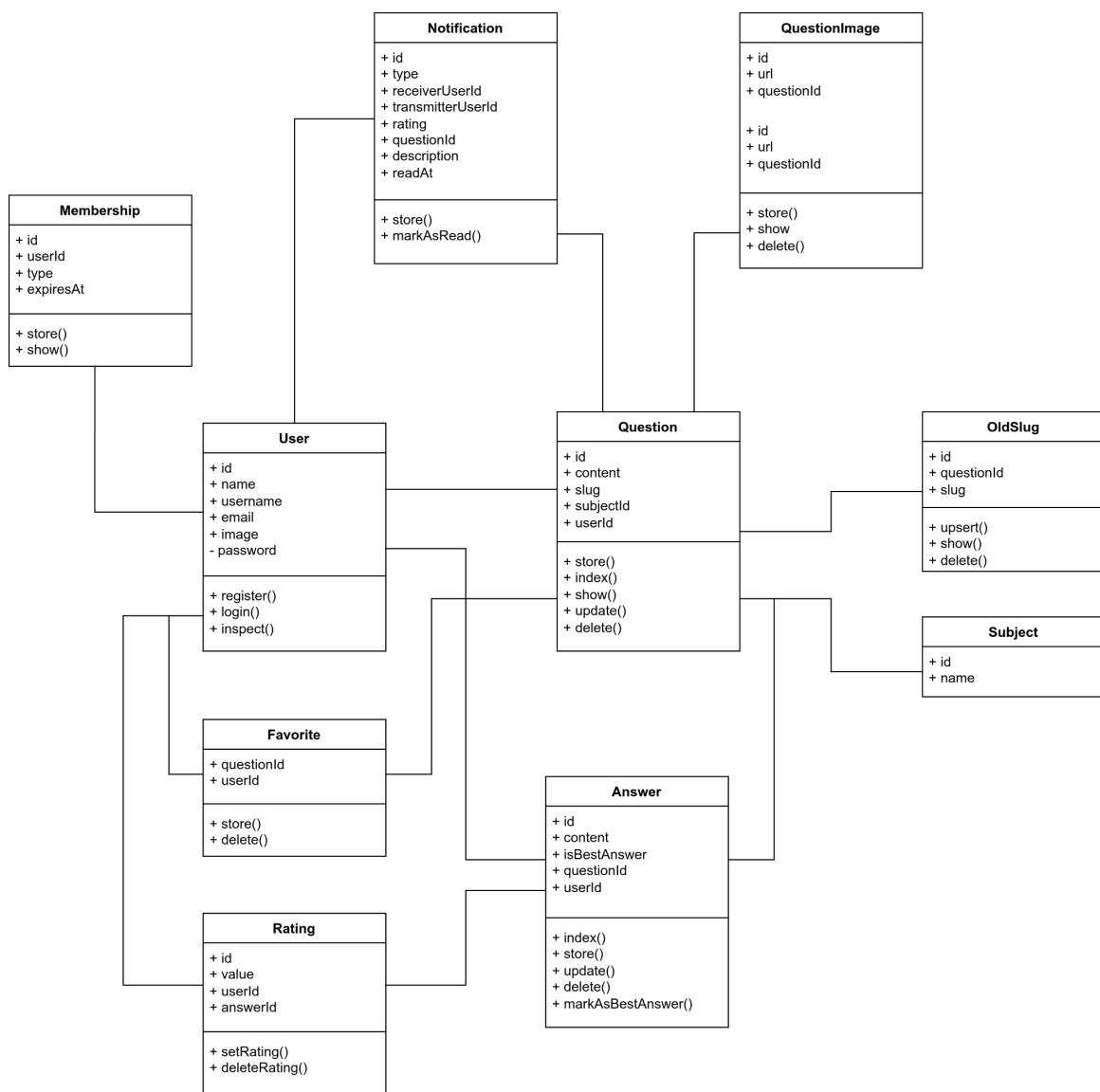
Gambar 4.7 *Activity Diagram* Mengakses Halaman Premium

Gambar 4.7 menjelaskan proses pembelian *membership* premium melalui halaman premium. Pertama, pengguna memilih paket dan durasi langganan. Sistem kemudian menghitung harga dan mengirim permintaan pembayaran ke *Payment Gateway*, yang mengembalikan link pembayaran. Pengguna membuka halaman pembayaran melalui

link tersebut, memilih metode pembayaran, dan melakukan pembayaran. Setelah pembayaran terkonfirmasi, *Payment Gateway* memberi tahu sistem, yang kemudian menyimpan informasi langganan ke *database*.

3. Class Diagram

Class diagram adalah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut adalah *class diagram* dari aplikasi Yukitanya:



Gambar 4.8 Class Diagram Aplikasi Yukitanya

Class Diagram ini menggambarkan aplikasi forum tanya jawab untuk tugas sekolah. Diagram ini mencakup kelas utama seperti pengguna, pertanyaan, jawaban, mata pelajaran, favorit, *membership*, notifikasi, dan rating, masing-masing dengan atribut dan metode yang mendefinisikan karakteristik dan fungsinya. Relasi antar kelas menunjukkan hubungan seperti satu pengguna dapat mengajukan banyak pertanyaan dan satu pertanyaan dapat memiliki banyak jawaban. Aplikasi memungkinkan pengguna untuk mengajukan pertanyaan, memberikan jawaban, memfavoritkan pertanyaan, bergabung dengan grup atau subjek, menerima notifikasi, dan memberikan rating pada jawaban.

4.2.2 Struktur Tabel

Tabel-tabel yang terdapat dalam basis data yang digunakan dalam aplikasi forum tanya jawab tugas sekolah menggunakan PostgreSQL adalah sebagai berikut:

1. Perancangan tabel *user*

Dalam perancangan tabel untuk menyimpan informasi pengguna (*user*) dalam basis data, berikut perancangan tabel untuk pengguna:

Tabel 4.11 Perancangan tabel user

Field	Type	Size	Index	Deskripsi
id	VARCHAR	25	PK	ID User
name	VARCHAR	50		Nama
username	VARCHAR	15	UNIQUE	Username
email	VARCHAR	50	UNIQUE	Email
image	VARCHAR	100		URL Foto Profil
password	VARCHAR	100		Password

				yang telah di-hash
created_at	TIMESTAMPTZ			Tanggal akun dibuat
updated_at	TIMESTAMPTZ			Tanggal terakhir akun di-update

2. Perancangan tabel *question*

Dalam perancangan tabel untuk menyimpan informasi pertanyaan (*question*) dalam basis data, berikut perancangan tabel untuk pertanyaan:

Tabel 4.12 Perancangan tabel question

Field	Type	Size	Index	Deskripsi
id	VARCHAR	25	PK	ID Pertanyaan
content	TEXT			Isi Pertanyaan
slug	VARCHAR	50	UNIQUE	Unique identifier
subject_id	VARCHAR	25	FK	ID dari mata pelajaran
user_id	VARCHAR	25	FK	ID dari pemilik pertanyaan
created_at	TIMESTAMPTZ			Tanggal pertanyaan

				dibuat
updated_at	TIMESTAMPTZ			Tanggal terakhir pertanyaan di-update

3. Perancangan tabel *old slug*

Dalam perancangan tabel untuk menyimpan informasi slug pertanyaan lama (*old slug*) dalam basis data, berikut perancangan tabel untuk *old slug*:

Tabel 4.13 Perancangan tabel old slug

Field	Type	Size	Index	Deskripsi
id	VARCHAR	25	PK	ID old slug
slug	VARCHAR	50	UNIQUE	Slug lama
question_id	VARCHAR	25	FK	ID dari pertanyaan

4. Perancangan tabel question_image

Dalam perancangan tabel untuk menyimpan informasi gambar pertanyaan (*question image*) dalam basis data, berikut perancangan tabel untuk gambar pertanyaan:

Tabel 4.14 Perancangan tabel question image

Field	Type	Size	Index	Deskripsi
id	VARCHAR	25	PK	ID gambar pertanyaan
url	TEXT			URL dari gambar

question_id	VARCHAR	25	FK	ID dari pertanyaan
-------------	---------	----	----	--------------------

5. Perancangan tabel *subject*

Dalam perancangan tabel untuk menyimpan informasi mata pelajaran (*subject*) dalam basis data, berikut perancangan tabel untuk mata pelajaran:

Tabel 4.15 Perancangan tabel subject

Field	Type	Size	Index	Deskripsi
id	VARCHAR	25	PK	ID mata pelajaran
name	VARCHAR	25		Nama mata pelajaran

6. Perancangan tabel *answer*

Dalam perancangan tabel untuk menyimpan informasi jawaban (*answer*) dalam basis data, berikut perancangan tabel untuk jawaban:

Tabel 4.16 Perancangan tabel answer

Field	Type	Size	Index	Deskripsi
id	VARCHAR	25	PK	ID jawaban
content	TEXT			Isi jawaban
is_best_answer	BOOLEAN	1		Apakah jawaban terbaik
question_id	VARCHAR	25	FK	ID dari pertanyaan
user_id	VARCHAR	25	FK	ID dari

				pengguna
created_at	TIMESTAMPTZ			Tanggal jawaban dibuat
updated_at	TIMESTAMPTZ			Tanggal terakhir jawaban di-update

7. Perancangan tabel *favorite*

Dalam perancangan tabel untuk menyimpan informasi favorit (*favorite*) dalam basis data, berikut perancangan tabel untuk favorite:

Tabel 4.17 Perancangan tabel favorite

Field	Type	Size	Index	Deskripsi
question_id	VARCHAR	25	FK	ID dari pertanyaan
user_id	VARCHAR	25	FK	ID dari pengguna

8. Perancangan tabel *rating*

Dalam perancangan tabel untuk menyimpan informasi nilai (*rating*) dalam basis data, berikut perancangan tabel untuk nilai:

Tabel 4.18 Perancangan tabel rating

Field	Type	Size	Index	Deskripsi
id	VARCHAR	25	PK	ID nilai
value	INTEGER	1		Nilai

answer_id	VARCHAR	25	FK	ID dari jawaban
user_id	VARCHAR	25	FK	ID dari pengguna

9. Perancangan tabel *membership*

Dalam perancangan tabel untuk menyimpan informasi langganan (*membership*) dalam basis data, berikut perancangan tabel untuk langganan:

Tabel 4.19 Perancangan tabel membership

Field	Type	Size	Index	Deskripsi
id	VARCHAR	25	PK	ID membership
type	VARCHAR	10		Jenis membership
expires_at	TIMESTAMPTZ			Tanggal kadaluarsa
user_id	VARCHAR	25	FK	ID dari pengguna

10. Perancangan tabel *notification*

Dalam perancangan tabel untuk menyimpan informasi notifikasi (*notification*) dalam basis data, berikut perancangan tabel untuk notifikasi:

Tabel 4.20 Perancangan tabel notification

Field	Type	Size	Index	Deskripsi
id	VARCHAR	25	PK	ID notifikasi

type	VARCHAR	10		Jenis notifikasi
receiver_use r_id	VARCHAR	25	FK	ID dari pengguna penerima
transmitter_u ser_id	VARCHAR	25	FK	ID dari pengguna pengirim
rating	INTEGER	3		Nilai dari jawaban
question_id	VARCHAR	25	FK	ID dari pertanyaan
description	VARCHAR	100		Deskripsi notifikasi
created_at	TIMESTAMPTZ			Tanggal notifikasi dibuat
read_at	TIMESTAMPTZ			Tanggal pertanyaan dibaca

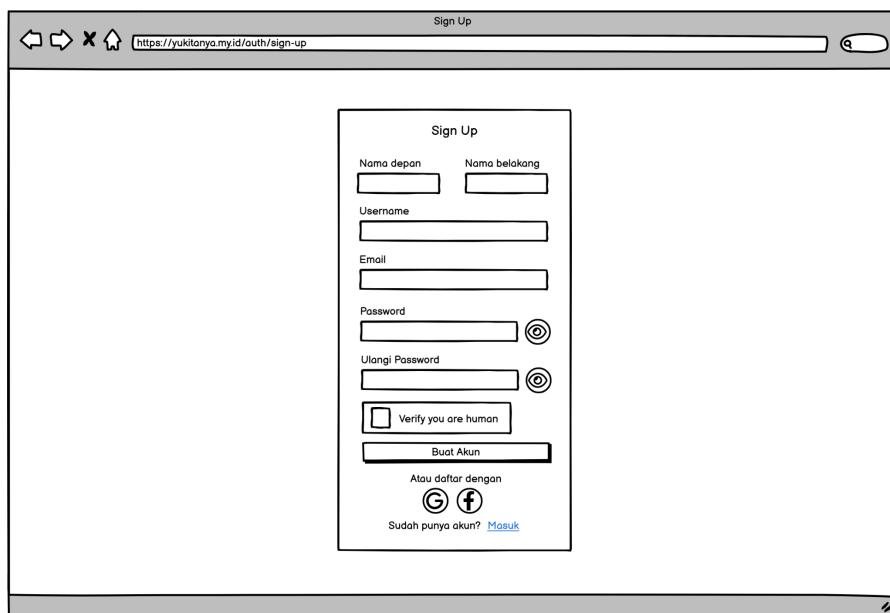
4.2.3 Desain Antarmuka

1. Desain tampilan antarmuka *register*

Pada antarmuka register, terdapat berbagai elemen tampilan yang mencakup halaman dengan kolom input untuk memasukkan informasi penting seperti Nama, *Username*, *Email*, dan *Password*. Halaman ini dirancang dengan baik untuk memudahkan pengguna dalam mengisi data yang diperlukan. Setelah pengguna menyelesaikan pengisian informasi yang

diperlukan dan menekan tombol "Buat akun", proses registrasi akun pengguna akan dimulai. Proses ini mencakup verifikasi informasi yang dimasukkan dan pembuatan akun baru di sistem. Selain itu, pengguna juga memiliki opsi untuk mendaftar menggunakan akun media sosial mereka, seperti Facebook dan Google, yang memberikan kemudahan dan mempercepat proses pendaftaran. Dengan opsi ini, pengguna tidak perlu mengisi semua informasi secara manual, sehingga prosesnya menjadi lebih cepat dan efisien.

Berikut di bawah ini merupakan tampilan antarmuka register aplikasi forum tanya jawab yang akan dirancang:

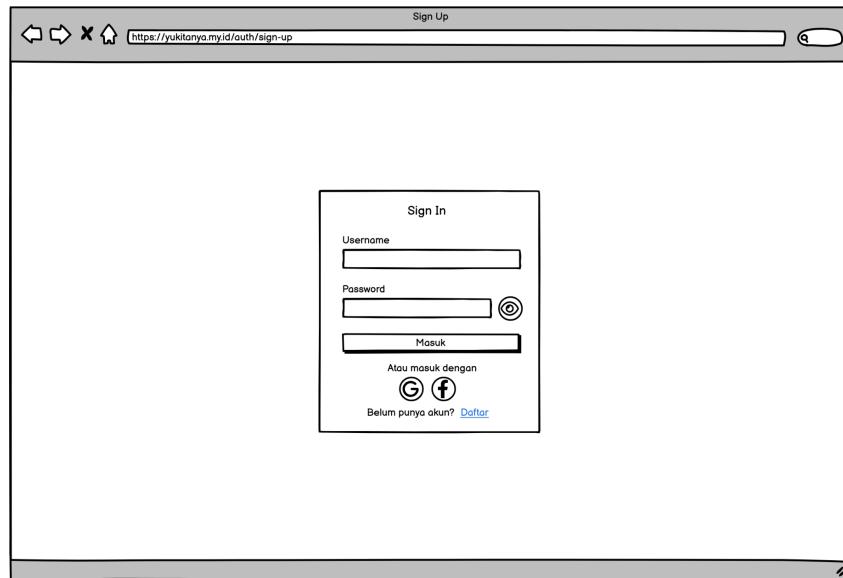


Gambar 4.9 Desain Halaman Register

2. Desain tampilan antarmuka *login*

Antarmuka login aplikasi ini dirancang untuk memudahkan akses akun dengan dua kolom untuk *username* dan *password*. Setelah mengisi kolom dan menekan tombol "Masuk," sistem akan memverifikasi informasi yang dimasukkan. Pengguna juga bisa login melalui Facebook atau Google dengan mengklik tombol yang sesuai, yang akan mengarahkan ke halaman login media sosial untuk otorisasi. Desain antarmuka yang sederhana dan *user-friendly* memastikan pengalaman pengguna yang efisien. Ilustrasi

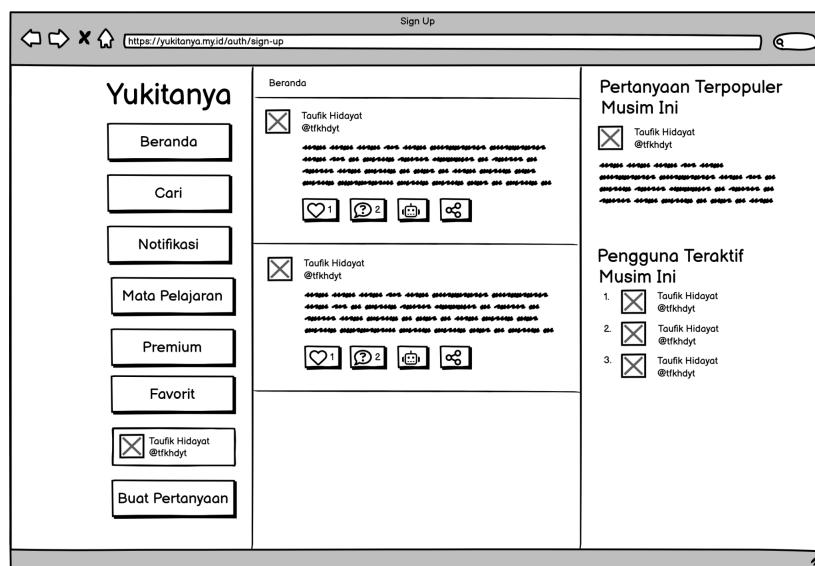
antarmuka login menunjukkan kolom untuk *username* dan *password*, serta tombol login dan opsi media sosial.



Gambar 4.10 Desain Halaman Login

3. Halaman Beranda

Di bawah ini terdapat representasi visual dari antarmuka halaman beranda aplikasi forum tanya jawab yang sedang direncanakan.

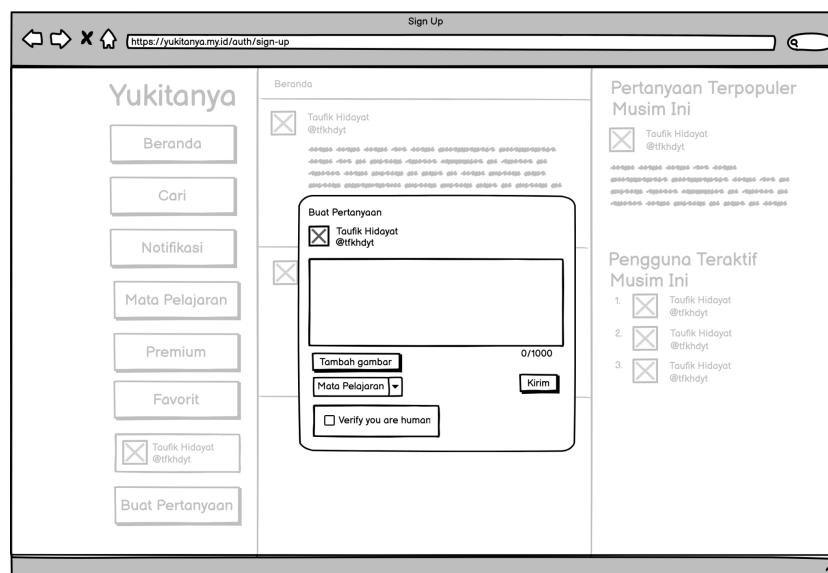


Gambar 4.11 Desain Halaman Beranda

Antarmuka beranda menampilkan *timeline* pertanyaan terbaru, tombol untuk membuat pertanyaan baru, dan *sidebar* di sebelah kiri untuk mengakses halaman lain.

4. Dialog Buat Pertanyaan Baru

Berikut di bawah merupakan tampilan antarmuka dialog buat pertanyaan baru aplikasi forum tanya jawab yang akan dirancang:



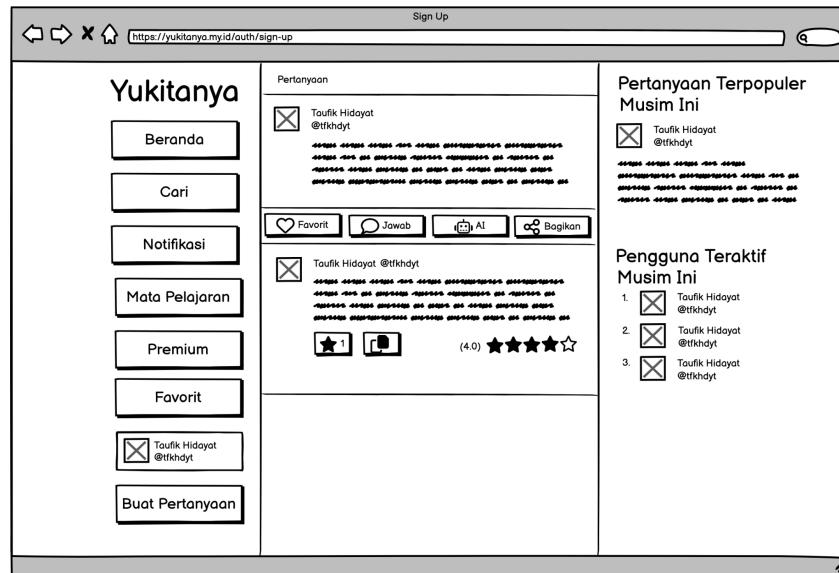
Gambar 4.12 Desain Dialog Buat Pertanyaan Baru

Dialog ini muncul setelah pengguna menekan tombol “Buat pertanyaan”. Dialog ini berfungsi sebagai input untuk proses pembuatan pertanyaan baru. Pengguna akan diminta untuk memasukan isi pertanyaan yang ingin ditanyakan.

5. Halaman Detail Pertanyaan

Pada antarmuka halaman detail pertanyaan, pengguna dapat melihat informasi lengkap tentang pertanyaan, seperti tanggal pembuatan, mata pelajaran terkait, dan jawaban yang telah diberikan. Halaman ini memfasilitasi diskusi efektif dan memungkinkan pengguna untuk berkontribusi dengan menyediakan jawaban mereka sendiri, sehingga mereka dapat berbagi pengetahuan dalam bidang yang relevan.

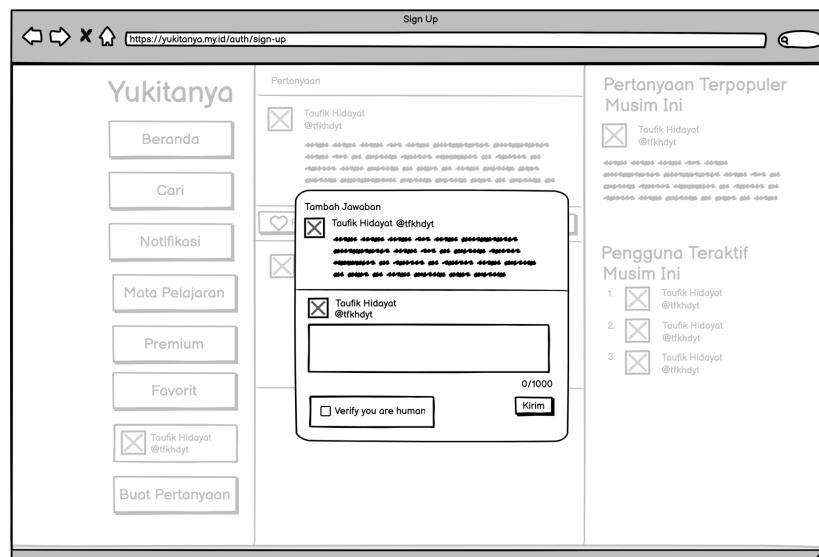
Berikut di bawah ini merupakan tampilan antarmuka halaman detail pertanyaan aplikasi forum tanya jawab yang akan dirancang:



Gambar 4.13 Desain Halaman Detail Pertanyaan

6. Dialog Buat Jawaban Baru

Berikut di bawah merupakan tampilan antarmuka dialog buat jawaban baru aplikasi forum tanya jawab yang akan dirancang:

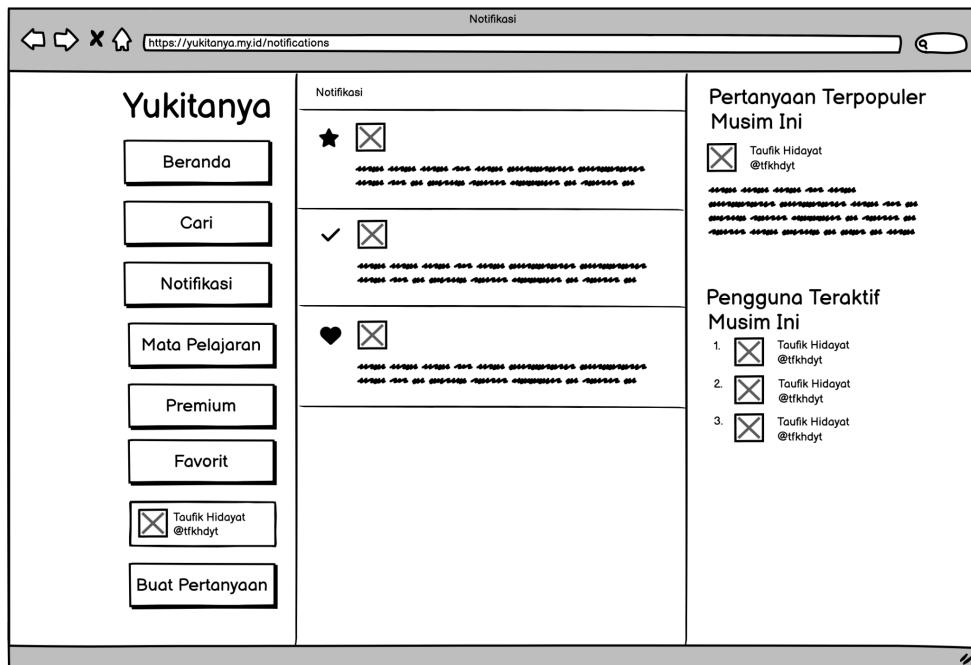


Gambar 4.14 Desain Dialog Buat Jawaban Baru

Dialog ini muncul setelah pengguna menekan tombol “Tambah jawaban”. Dialog ini berfungsi sebagai input untuk proses pembuatan jawaban baru.

7. Halaman Notifikasi

Berikut di bawah ini merupakan tampilan antarmuka halaman notifikasi aplikasi forum tanya jawab yang akan dirancang:

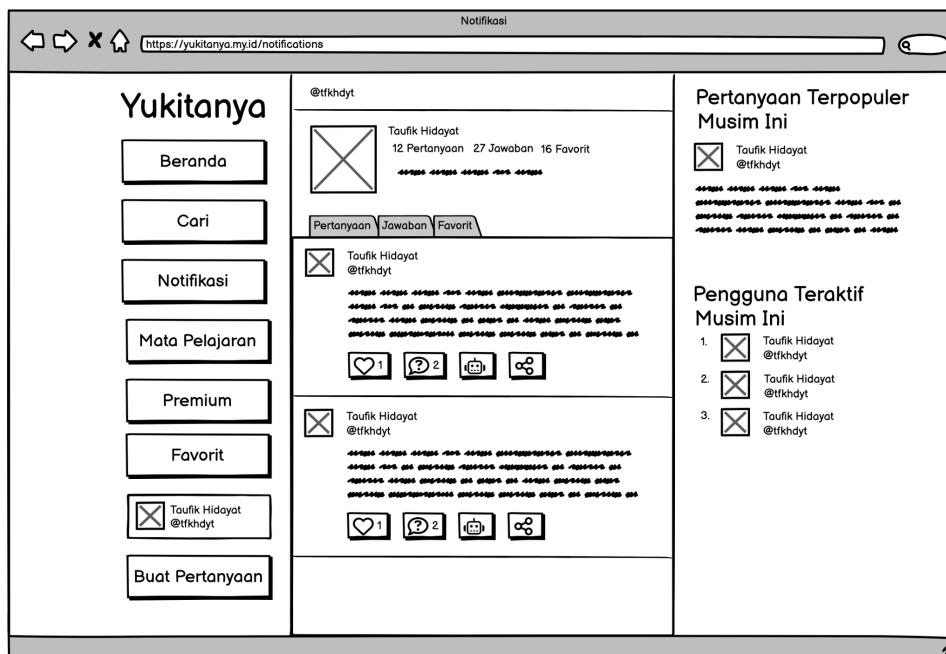


Gambar 4.15 Desain Halaman Notifikasi

Dalam antarmuka halaman notifikasi, pengguna akan melihat daftar peringatan yang relevan dengan aktivitas mereka di aplikasi. Notifikasi yang ditampilkan meliputi jawaban baru pada pertanyaan yang diajukan pengguna, nilai baru yang diberikan pada jawaban atau pertanyaan mereka, dan pemberitahuan saat pertanyaan atau jawaban mereka difavoritkan oleh pengguna lain. Selain itu, halaman notifikasi juga akan menampilkan pembaruan lain yang penting, seperti pesan pribadi dari pengguna lain, pengumuman dari admin, dan notifikasi tentang aktivitas atau event khusus yang sedang berlangsung dalam forum. Dengan demikian, pengguna dapat tetap terinformasi tentang semua interaksi yang terjadi terkait dengan aktivitas mereka di aplikasi.

8. Halaman Profil

Berikut ini adalah tampilan antarmuka halaman profil untuk aplikasi forum tanya jawab yang akan dirancang.



Gambar 4.16 Desain Halaman Profil

Antarmuka halaman profil menampilkan berbagai informasi pribadi dari pengguna. Informasi ini mencakup foto profil, nama, username, tanggal bergabung, dan data lainnya. Selain itu, halaman profil juga dapat menampilkan riwayat aktivitas pengguna, seperti pertanyaan yang telah diajukan, jawaban yang diberikan, serta jumlah poin atau reputasi yang telah diperoleh. Desain ini memastikan bahwa pengguna dapat dengan mudah melihat dan mengelola informasi pribadi serta aktivitas mereka dalam forum.

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1. Implementasi

Setelah melaksanakan analisis dan perancangan maka selanjutnya adalah pengimplementasian untuk menjalankan analisis dan perancangan yang sudah dibuat ke dalam bentuk aplikasi.

5.1.1 Listing Program

1. Halaman Register

```
import { Footer } from '@/components/footer';
import { Card,CardContent, CardHeader, CardTitle } from
'@/components/ui/card';
import { redirect } from 'next/navigation';

import { getServerAuthSession } from '@/server/auth';
import { SignupForm } from './signup-form';

export const metadata = {
  title: 'Sign Up - Yukitanya',
};

export default async function SignUp() {
  const session = await getServerAuthSession();
  if (session) return redirect('/home');

  return (
    <>
      <section className='mb-64 min-h-[100svh]
      bg-[url(/img/hero_bg.svg)] bg-cover lg:bg-bottom'>
        <div className='container px-5 pt-5 md:pt-12'>
          <Card className='mx-auto w-fit rounded-3xl border-2
          border-black p-0 md:p-6'>
            <CardHeader>
              <CardTitle className='text-center text-3xl
              font-extrabold uppercase'>
                Sign up
              </CardTitle>
            </CardHeader>
            <CardContent>
```

```

        <SignupForm />
    </CardContent>
</Card>
</div>
</section>
<Footer scroll={false} />
</>
);
}
}

```

2. Halaman Login

```

import { Footer } from '@/components/footer';
import { Card, CardContent, CardHeader, CardTitle } from
'@/components/ui/card';
import { redirect } from 'next/navigation';

import { getServerAuthSession } from '@/server/auth';
import { SigninForm } from './signin-form';

export const metadata = {
  title: 'Sign In - Yukitanya',
};

export default async function SignIn() {
  const session = await getServerAuthSession();
  if (session) return redirect('/home');

  return (
    <>
      <section className='min-h-[100svh] bg-[url(/img/hero_bg.svg)]'
bg-cover lg:bg-bottom>
        <div className='container px-5 pt-5 md:pt-12'>
          <Card className='mx-auto w-fit rounded-3xl border-2
border-black p-0 md:p-6'>
            <CardHeader>
              <CardTitle className='text-center text-3xl
font-extrabold uppercase'>
                Sign in
              </CardTitle>
            </CardHeader>
            <CardContent>
              <SigninForm />
            </CardContent>
          </Card>
        </div>
      </section>
      <Footer />
    </>
  );
}

```

```

    </>
);
}
}
```

3. Halaman Beranda

```

'use client';

import { useIntersectionObserver } from '@uidotdev/usehooks';
import type { Session } from 'next-auth';
import { useEffect } from 'react';

import { PertanyaanKosong } from '@/components/pertanyaan-kosong';
import { QuestionPost } from '@/components/question/question-post';
import { SkeletonQuestionPost } from
'@/components/question/skeleton-question-post';
import { createInitial } from '@/lib/utils';
import { api } from '@trpc/react';
import dayjs from 'dayjs';

export function Timeline({ session }: { session?: Session }) {
  const { isLoading, data, fetchNextPage, isFetchingNextPage } =
    api.question.findAllQuestions.useInfiniteQuery(
      {
        limit: 10,
      },
      {
        getNextPageParam: (lastPage) => lastPage.nextCursor,
      },
    );

  const [reference, entry] = useIntersectionObserver({ threshold: 0 });
  useEffect(() => {
    if (entry?.isIntersecting) {
      void fetchNextPage();
    }
  }, [entry, fetchNextPage]);

  const questions = data?.pages.flatMap((page) => page.data);

  if (isLoading) {
    return <SkeletonQuestionPost />;
  }

  if (questions?.length === 0 || !questions) {
    return <PertanyaanKosong user={session?.user} />;
  }
}
```

```

return (
  <>
    {questions.map((question, index) => {
      const bestAnswerRatings =
        question.answers.find((answer) =>
          answer.isBestAnswer)?.ratings ?? [];
      const totalRating =
        bestAnswerRatings?.reduce(
          (accumulator, rating) => accumulator + rating.value,
          0,
        ) ?? 0;
      const averageRating = bestAnswerRatings.length
        ? totalRating / bestAnswerRatings.length
        : NaN;
      const membership =
        question.owner.memberships.find((membership) =>
          dayjs().isBefore(membership.expiresAt),
        );
      return (
        <div
          ref={index === questions.length - 1 ? reference :
undefined}
          key={question.id}
        >
          <QuestionPost
            question={{
              content: question.content,
              createdAt: question.createdAt,
              id: question.id,
              isFavorited: question.favorites.some(
                (favorite) => favorite.userId ===
session?.user.id,
              ),
              numberOfAnswers: question.answers.length,
              numberOfFavorites: question.favorites.length,
              rating: Number.isNaN(averageRating) ? undefined :
averageRating,
              subject: question.subject,
              updatedAt: question.updatedAt,
              slug: question.slug,
              owner: {
                ...question.owner,
                membership,
                initial: createInitial(question.owner.name ??
undefined),
              },
              images: question.images,
            }}
          </QuestionPost>
        </div>
      );
    });
  )
);

```

```

        )}
      session={session}
    />
  </div>
);
})
{isFetchingNextPage && <SkeletonQuestionPost />}
/>
);
}

```

4. Detail Pertanyaan

```

import { redirect } from 'next/navigation';
import { P, match } from 'ts-pattern';
import { createInitial } from '@/lib/utils';
import { getServerAuthSession } from '@/server/auth';
import { api } from '@trpc/server';
import dayjs from 'dayjs';
import { AnswerList } from './answer-list';
import { DetailedQuestion } from './detailed-question';

export async function generateMetadata({
  params,
}: {
  params: { slug: string };
}) {
  const { slug } = params;

  const question = await
api.question.findQuestionContentBySlug.query(slug);
  if (!question) return {};

  return {
    title: `${match(question.content.length)
      .with(P.number.gte(50), () => `${question.content.slice(0,
50)}...`)
      .otherwise(() => question.content)} - Yukitanya`,
    description: question.content,
  };
}

export default async function Question({
  params,
}: {
  params: { slug: string };
}) {
  const session = await getServerAuthSession();
}

```

```
const question = await
api.question.findQuestionBySlug.query(params.slug);
if (!question) {
    return redirect('/home');
}

const membership = question.owner.memberships.find((membership)
=>
    dayjs().isBefore(membership.expiresAt),
);

return (
    <div>
        <DetailedQuestion
            question={{
                content: question.content,
                createdAt: question.createdAt,
                id: question.id,
                subject: question.subject,
                updatedAt: question.updatedAt,
                slug: question.slug,
                owner: {
                    ...question.owner,
                    membership,
                    initial: createInitial(question.owner.name ?? undefined),
                },
                images: question.images,
            }}
            session={session ?? undefined}
        />
        <div>
            <AnswerList
                question={{
                    ...question,
                    owner: {
                        ...question.owner,
                        membership,
                        initial: createInitial(question.owner.name ?? undefined),
                    },
                }}
                session={session ?? undefined}
            />
        </div>
    </div>
);
}
```

5. Halaman Pencarian

```
'use client';

import { useDebounce } from '@uidotdev/usehooks';
import type { Session } from 'next-auth';
import { useRouter, useSearchParams } from 'next/navigation';
import { useEffect, useState } from 'react';
import { match } from 'ts-pattern';

import { Input } from '@/components/ui/input';
import {
  Select,
  SelectContent,
  SelectItem,
  SelectTrigger,
  SelectValue,
} from '@/components/ui/select';
import { mapel } from '@/constants/mapel';

import { QuestionList } from './question-list';
import { UserList } from './user-list';

export function SearchForm({ session }: { session?: Session }) {
  const searchParameters = useSearchParams();
  const router = useRouter();

  const q = searchParameters.get('query');
  const s = searchParameters.get('subject');
  const t = searchParameters.get('type');

  const [query, setQuery] = useState(q ?? '');
  const [subject, setSubject] = useState(s ?? 'all');
  const [searchType, setSearchType] = useState(t ?? 'question');

  const debouncedQuery = useDebounce(query, 500);
  const debouncedSubject = useDebounce(subject, 500);
  const debouncedType = useDebounce(searchType, 500);

  useEffect(() => {
    if (debouncedType === 'question') {
      router.push(
        `/search?type=${debouncedType}&subject=${debouncedSubject}&query=${debouncedQuery}`,
      );
    } else if (debouncedType === 'user') {
      router.push(`/search?type=${debouncedType}&query=${debouncedQuery}`);
    }
  }, [debouncedType, debouncedQuery, debouncedSubject, router]);
}
```

```

);
}

}, [debouncedQuery, debouncedSubject, debouncedType, router]);

return (
<>
<form className='flex w-full items-center space-x-1 p-4'>
<Input
  className='rounded-l-full'
  onChange={(event) => {
    setQuery(event.target.value);
 }}
  placeholder='Cari...'
  type='text'
  value={query}
/>
{searchType === 'question' && (
  <Select onValueChange={setSubject} value={subject}>
    <SelectTrigger className='max-w-2/4 w-fit
rounded-none'>
      <SelectValue placeholder='Mata Pelajaran' />
    </SelectTrigger>
    <SelectContent>
      <SelectItem value='all'>Semua</SelectItem>
      {mapel.map((mpl) => (
        <SelectItem key={mpl.id} value={mpl.id}>
          {mpl.name}
        </SelectItem>
      )))
    </SelectContent>
  </Select>
)}
<Select onValueChange={setSearchType} value={searchType}>
  <SelectTrigger className='max-w-2/4 w-max
rounded-r-full'>
    <SelectValue placeholder='Jenis' />
  </SelectTrigger>
  <SelectContent>
    <SelectItem value='question'>Pertanyaan</SelectItem>
    <SelectItem value='user'>Pengguna</SelectItem>
  </SelectContent>
</Select>
</form>
<div>
  {match(Boolean(debouncedQuery))
    .with(true, () =>
      match(debouncedType)
        .with('question', () => (
          <QuestionList

```

```

        query={debouncedQuery}
        subjectId={debouncedSubject}
        session={session}
      />
    ))
    .with('user', () => <UserList query={debouncedQuery}>
/>
      .otherwise(() => ''),
    )
    .otherwise(() => '')
  </div>
</>
);
}
}

```

6. Fitur Notifikasi

```

'use client';

import { PertanyaanKosong } from '@/components/pertanyaan-kosong';
import { createInitial } from '@/lib/utils';
import { api } from '@trpc/react';

import { useIntersectionObserver } from '@uidotdev/usehooks';
import dayjs from 'dayjs';
import { useEffect } from 'react';
import { NotificationItem } from './notification-item';
import { SkeletonNotificationItem } from
'./skeleton-notification-item';

export function NotificationList({
  receiverUserId,
}: {
  receiverUserId: string;
}) {
  const { isLoading, fetchNextPage, data, isFetchingNextPage } =

api.notification.findAllNotificationByReceiverUserId.useInfiniteQuery(
  {
    receiverUserId,
    limit: 10,
  },
  {
    getNextPageParam: (lastPage) => lastPage.nextCursor,
  },
);

```

```

    const [reference, entry] = useIntersectionObserver({ threshold: 0
});

useEffect(() => {
  if (entry?.isIntersecting) {
    void fetchNextPage();
  }
}, [entry, fetchNextPage]);

const notifications = data?.pages.flatMap((page) => page.data);

if (isLoading) {
  return <SkeletonNotificationItem />;
}

if (notifications?.length === 0 || !notifications) {
  return (
    <PertanyaanKosong
      title='Notifikasi masih kosong'
      showTanyakanButton={false}
    />
  );
}

return (
  <>
    {notifications.map((notif, index) => {
      const membership = notif.transmitterUser.memberships.find(
        (membership) => dayjs().isBefore(membership.expiresAt),
      );

      return (
        <div
          key={notif.id}
          ref={index === notifications.length - 1 ? reference : undefined}
        >
          <NotificationItem
            id={notif.id}
            createdAt={notif.createdAt}
            hasBeenRead={Boolean(notif.readAt)}
            question={notif.question}
            transmitterUser={{
              ...notif.transmitterUser,
              membership,
              initial: createInitial(notif.transmitterUser.name
?? undefined),
            }}
            type={notif.type}
            rating={notif.rating ?? undefined}
          </NotificationItem>
        </div>
      );
    });
  </>
);
}

```

```

        description={notif.description}
    />
  </div>
);
})}
{isFetchingNextPage && <SkeletonNotificationItem />}
/>
);
}

```

7. Halaman Premium

```

import type { Metadata } from 'next';

import { getServerAuthSession } from '@/server/auth';

import { formatLongDateTime } from '@/lib/datetime';
import { redirect } from 'next/navigation';
import { PremiumType } from './premium-type';

export const metadata: Metadata = {
  title: 'Premium - Yukitanya',
};

export default async function PremiumPage() {
  const session = await getServerAuthSession();
  if (!session) {
    return redirect('/home');
  }

  return (
    <div className='p-4'>
      {session.user.membership ? (
        <>
          <h2 className='text-center text-2xl my-4 font-bold text-[#F48C06] uppercase'>
            Terima kasih telah berlangganan
          </h2>
          {session.user.membership.expiresAt && (
            <p className='text-center text-sm font-medium text-[#696969]'>
              Membership{' '}
              {session.user.membership.type === 'standard'
                ? 'Premium'
                : 'Premium+'}{' '}
              mu akan berlaku sampai{' '}
            </p>
            {formatLongDateTime(session.user.membership?.expiresAt)}
          )}
        </>
      ) : null}
    </div>
  );
}

```

```

        </p>
    )}
</>
) : (
<>
    <h2 className='text-center text-2xl my-4 font-bold text-[#F48C06] uppercase'>
        Pilih paket yang cocok untuk Anda
    </h2>
    <PremiumType user={session.user} />
</>
)}
</div>
);
}

```

8. Profil Pengguna

```

import dayjs from 'dayjs';
import type { Metadata } from 'next';
import { redirect } from 'next/navigation';

import { formatLongDateTime } from '@/lib/datetime';
import { createInitial, getDiceBearAvatar } from '@/lib/utils';
import { getServerAuthSession } from '@/server/auth';
import { api } from '@/trpc/server';

import { AvatarWithBadge } from '@/components/avatar-with-badge';
import { z } from 'zod';
import { UserStat } from './user-stat';
import { UserTabs } from './user-tabs';

export async function generateMetadata({
    params,
}: {
    params: { username: string };
}): Promise<Metadata> {
    const { username } = params;
    const user = await api.user.findUserByUsername.query(username);

    if (user) {
        return {
            title: `${user.name} (@${user.username}) - Yukitanya`,
            description: `Bergabung sejak ${formatLongDateTime(user.createdAt)}`,
            openGraph: {
                title: `${user.name} (@${user.username}) - Yukitanya`,
                description: `Bergabung sejak

```

```

    ${formatLongDateTime(user.createdAt)}`,
    images: [
      {
        url: user.image ?? getDiceBearAvatar(user.username),
        width: 600,
        height: 600,
      },
    ],
  },
};

return {};
}
const tabScheme = z
  .enum(['Pertanyaan', 'Jawaban', 'Favorit'])
  .default('Pertanyaan');

export default async function UserPage({
  params,
  searchParams,
}: {
  params: { username: string };
  searchParams?: { tab: string };
}) {
  const session = await getServerAuthSession();
  const user = await
api.user.findUserByUsername.query(params.username);
  if (!user) return redirect('/home');

  const activeTab = tabScheme.safeParse(searchParams?.tab);
  const membership = user.memberships.find((membership) =>
    dayjs().isBefore(membership.expiresAt),
  );

  return (
    <main>
      <div className='mx-4 mt-4 flex space-x-6 text-[#696984] md:m-8 md:space-x-8'>
        <AvatarWithBadge
          user={{
            ...user,
            membership,
            initial: createInitial(user.name ?? undefined),
          }}
          classNames='h-20 w-20 md:h-28 md:w-28'
          badgeSize='xl'
        />
        <div className='w-full space-y-2 text-[#696984]'>

```

```

        <p className='line-clamp-1 hidden font-medium md:block
md:text-lg'>
            {user.name}
        </p>
        <UserStat username={user.username} />
        <p className='hidden text-sm md:block md:text-base'>
            Bergabung sejak{' '}
            <span title={formatLongDateTime(user.createdAt)}>
                {dayjs(user.createdAt).format('D MMMM YYYY')}
            </span>
        </p>
        </div>
    </div>
    <div className='m-4 mb-6 space-y-1 text-[#696984] md:hidden'>
        <p className='line-clamp-1 font-medium
md:text-lg'>{user.name}</p>
        <p className='text-sm'>
            Bergabung sejak{' '}
            <span title={formatLongDateTime(user.createdAt)}>
                {dayjs(user.createdAt).format('D MMMM YYYY')}
            </span>
        </p>
        </div>
        <div className='w-full'>
            <UserTabs
                session={session ?? undefined}
                user={{
                    id: user.id,
                    name: user.name ?? user.username,
                    username: user.username,
                }}
                activeTab={activeTab.success ? activeTab.data :
'Pertanyaan'}
            />
        </div>
    </main>
);
}

```

Source code selengkapnya dilampirkan pada lampiran 7.

5.1.2 Implementasi Sistem

Implementasi sistem adalah tahap di mana sistem yang telah disetujui mulai diterapkan dan dioperasikan, termasuk program yang telah dikembangkan selama tahap perancangan. Pada tahap ini, semua komponen yang telah dirancang sebelumnya akan dipasang dan diintegrasikan untuk memastikan sistem berfungsi

sebagaimana mestinya. Proses ini mencakup penyiapan infrastruktur, konfigurasi perangkat lunak, dan pengujian untuk memastikan bahwa sistem berjalan dengan lancar. Berikut adalah rincian mengenai waktu dan lokasi penerapan sistem yang telah direncanakan dan disiapkan untuk pelaksanaan.

- a. Tempat : PT Forest Interactive Indonesia
- b. Alamat : Jl. Suryanagara No.27, RT.06/RW.05, Cibaduyut Wetan, Kec. Bojongloa Kidul, Kota Bandung, Jawa Barat 40238
- c. Waktu : Desember 2023

5.1.3 Spesifikasi Sistem

Spesifikasi sistem akan menjelaskan perangkat keras dan perangkat lunak yang diperlukan untuk menjalankan aplikasi forum tanya jawab tugas sekolah, termasuk informasi tentang komputer, *server*, dan program yang dibutuhkan. Tujuannya adalah memberikan gambaran jelas mengenai kebutuhan teknis aplikasi.

1. Spesifikasi perangkat keras

Di bawah ini merupakan spesifikasi perangkat keras yang diperlukan sebagai berikut:

Tabel 5.1 Spesifikasi Perangkat Keras

Processor	Dual-core processor (2 GHz atau lebih tinggi)
RAM	4 GB
SSD	10 GB <i>free space</i>

2. Spesifikasi perangkat lunak

Di bawah ini merupakan spesifikasi perangkat lunak yang diperlukan sebagai berikut:

Tabel 5.2 Spesifikasi Perangkat Lunak

Sistem Operasi	Windows 10, macOS 10.13+, atau Linux
Node.js	Versi 20 ke atas

PNPM	Versi 8 ke atas
PostgreSQL	Versi 15 ke atas
Git	Versi 2 ke atas
Web Browser	Chrome, Edge, Brave, Firefox, dll

5.1.4 Instalasi Sistem

Berikut adalah langkah-langkah untuk menginstall sistem program aplikasi forum tanya jawab tugas sekolah:

1. Pastikan semua *requirement* perangkat lunak telah diinstall.
2. Download *source code* dari *repository* github (<https://github.com/tfkhdtyukitanya>).
3. Buat salinan file konfigurasi “.env.example” dengan nama “.env”.
4. Masukan konfigurasi database dan lain lain ke dalam file .env tadi.
5. *Install dependencies* yang dibutuhkan dengan *command* “pnpm install”.
6. Migrasikan struktur *database* dengan *command* “pnpm db:push”.
7. *Build* aplikasi dengan *command* “pnpm build”.
8. Jalankan aplikasi dengan *command* “pnpm start”.
9. Aplikasi berhasil berjalan, kunjungi <http://localhost:3000> untuk membuka aplikasinya.

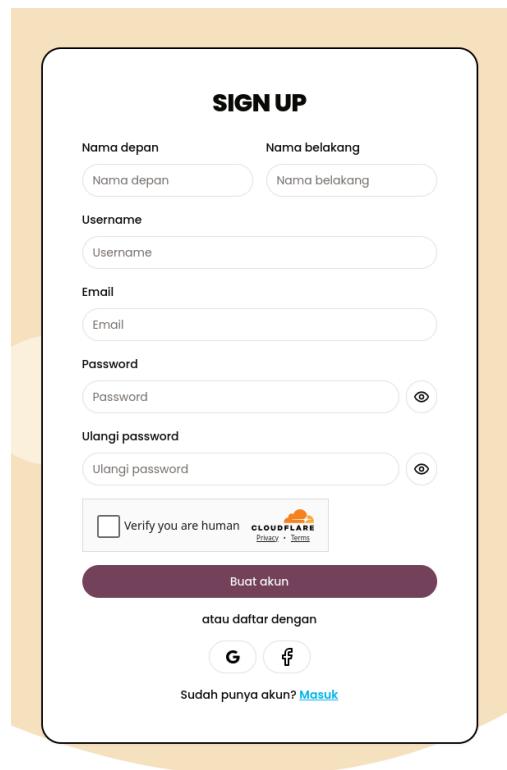
5.1.5 Menjalankan Sistem

Pada bagian ini akan dijelaskan bagaimana cara-cara menjalankan sistem aplikasi forum tanya jawab tugas sekolah.

1. Halaman Register

Sebelum pengguna dapat mengakses dan menggunakan aplikasi forum tanya jawab Yukitanya, pengguna diwajibkan untuk melakukan proses registrasi akun terlebih dahulu. Langkah ini melibatkan pengisian formulir registrasi yang mencakup berbagai informasi penting yang diperlukan untuk membuat akun pengguna. Proses registrasi ini bertujuan untuk memastikan bahwa setiap akun yang terdaftar memiliki identitas yang valid dan dapat dipercaya, sehingga pengguna dapat memanfaatkan semua fitur aplikasi dengan nyaman

dan aman. Setelah formulir diisi dan dikirimkan, pengguna kemudian akan menerima konfirmasi bahwa akun mereka telah aktif dan siap digunakan untuk berpartisipasi dalam forum, bertanya, menjawab, serta berinteraksi dengan sesama pengguna lainnya dalam konteks tugas sekolah mereka.



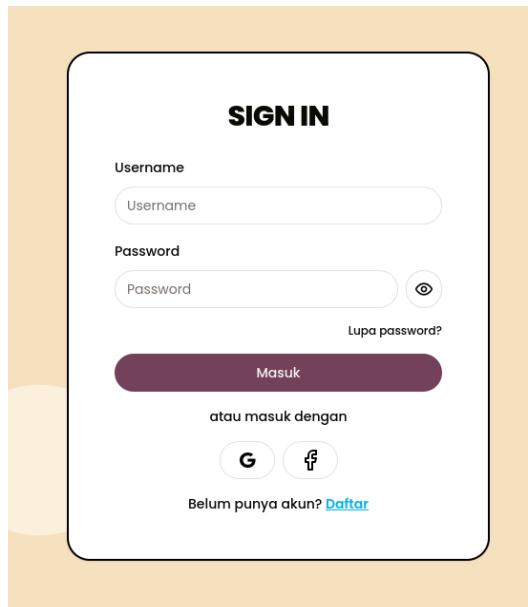
Gambar 5.1 Halaman Register

Pada halaman register, pengguna memiliki opsi untuk mendaftar dengan mengisi formulir, pengguna juga dapat mendaftar menggunakan akun Google dan Facebook.

2. Halaman Login

Sebelum dapat mengakses aplikasi forum tanya jawab tugas sekolah, pengguna diharuskan untuk melakukan proses login. Proses login ini dapat dilakukan dengan memasukkan *username* dan *password* yang telah terdaftar sebelumnya. Sebagai alternatif, pengguna juga memiliki opsi untuk login menggunakan akun Google atau Facebook mereka. Opsi login melalui media

sosial ini dirancang untuk memberikan kemudahan dan fleksibilitas tambahan bagi pengguna dalam mengakses aplikasi.



Gambar 5.2 Halaman Login

Pada halaman login, pengguna memiliki opsi untuk memasukkan *username* dan *password* yang telah terdaftar dalam sistem guna mengakses halaman beranda. Pengguna pun dapat menggunakan akun Google atau Facebook untuk login ke dalam aplikasi.

3. Halaman Beranda

Di halaman ini, akan terlihat daftar pertanyaan terbaru yang disusun dalam bentuk *timeline*. Di bagian kiri, terdapat *navigation bar* yang memudahkan perpindahan ke halaman-halaman lainnya. Sementara itu, di bagian kanan, terdapat *leaderboard* yang menampilkan pengguna-pengguna teraktif dalam 6 bulan terakhir serta pertanyaan-pertanyaan paling populer selama periode yang sama. Dengan demikian, perkembangan terbaru bisa dengan mudah diikuti dan pengguna yang paling aktif berkontribusi di platform ini dapat dilihat. Dan berikut adalah tampilan dari halaman beranda untuk mode desktop dan mobile:

The screenshot shows the desktop interface of Yukitanya. On the left, there's a sidebar with icons for Beranda, Cari, Notifikasi, Mata Pelajaran, Premium, Favorit, and a 'Buat pertanyaan' button. The main area is titled 'Beranda' and displays several user posts:

- IPA**: A post by a user with a brown profile picture asking about Newton's Law of Motion. It has 1 like, 0 comments, and 0 shares.
- Lainnya**: A post by 'Temlep' (@f4rlidanangs) asking what 'angka 1' means in history. It has 0 likes, 0 comments, and 0 shares.
- Lainnya**: A post by 'Donovan Solomeu' (@Donovan123) asking if 'Apakah itu Frutiger Aero'. It has 0 likes, 0 comments, and 0 shares.
- B. Indonesia**: A post by 'Alvitto Aprilian Gumay' (@alvittogenz09) asking about the Indonesian language. It has 1 like, 2 comments, and 0 shares.
- Lainnya**: A post by 'Azis Ramadhan' (@azrd) asking about chord progression. It has 0 likes, 1 comment, and 0 shares.
- Lainnya**: A post by 'Abdul Malik' (@up2dul) asking about prerequisites for calculus. It has 0 likes, 0 comments, and 0 shares.

On the right side, there are sections for 'PERTANYAAN TERPOPULER MUSIM INI' and 'PENGGUNA TERAKTIF MUSIM INI'.

Gambar 5.3 Halaman Beranda di Desktop

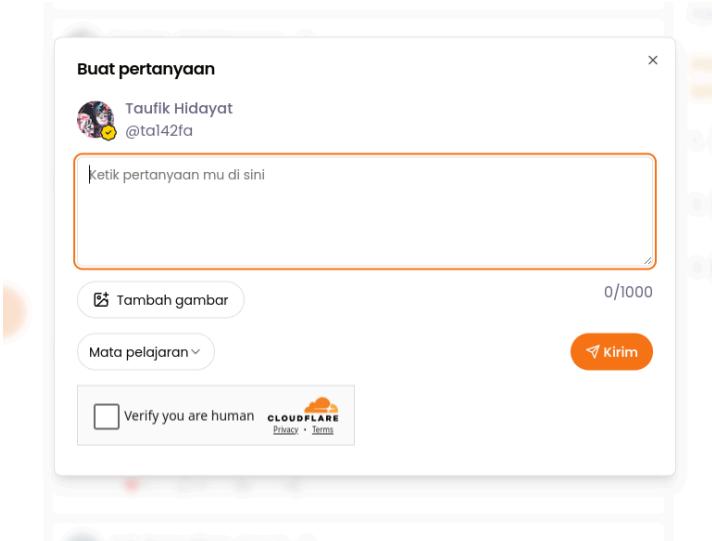
The screenshot shows the mobile version of the Yukitanya Beranda page. The layout is similar to the desktop version, displaying the same user posts and navigation elements at the bottom.

At the top, the status bar shows the time (8:46), battery level (9.08%), signal strength, and connectivity. The URL 'yukitanya.my.id/home' is visible in the browser header.

The main content area shows the same five posts from the desktop version, each with its author's profile picture, name, timestamp, and a brief description. The posts are identical to those shown in the desktop screenshot.

Gambar 5.4 Halaman Beranda di Mobile

4. Fitur Buat Pertanyaan



Gambar 5.5 Dialog Buat Pertanyaan

Untuk membuat pertanyaan, pengguna cukup menekan tombol “Buat Pertanyaan” di sebelah kiri, lalu memasukkan isi pertanyaan dan memilih mata pelajaran. Pengguna dengan membership dapat menambahkan gambar. CAPTCHA harus dicentang untuk mencegah spam, dan jika pertanyaan mengandung kata terlarang, notifikasi error akan muncul.



Gambar 5.6 Notifikasi Filtering Kata Terlarang

5. Halaman Detail Pertanyaan

Untuk melihat daftar jawaban, pengguna cukup menekan pertanyaan yang dipilih untuk berpindah ke halaman detail. Di halaman ini, pengguna dapat

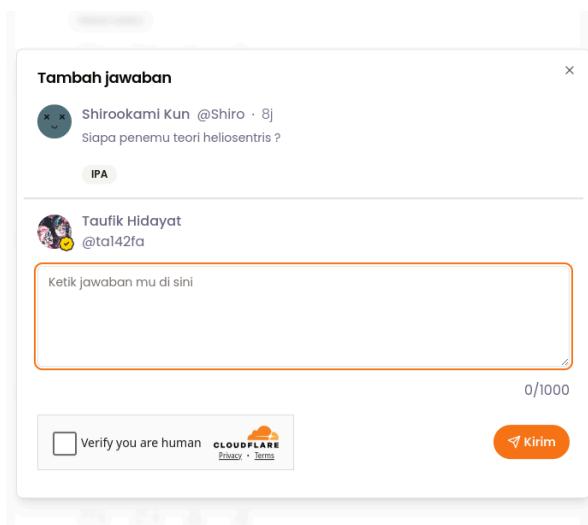
melihat jawaban yang ada dan memberikan nilai antara 1 hingga 5 bintang. Pemilik pertanyaan juga memiliki opsi untuk menandai jawaban sebagai jawaban terbaik. Dan berikut adalah tampilan dari halaman detail pertanyaan untuk mode desktop dan mobile:

Gambar 5.7 Halaman Detail Pertanyaan di Desktop

Gambar 5.8 Halaman Detail Pertanyaan di Mobile

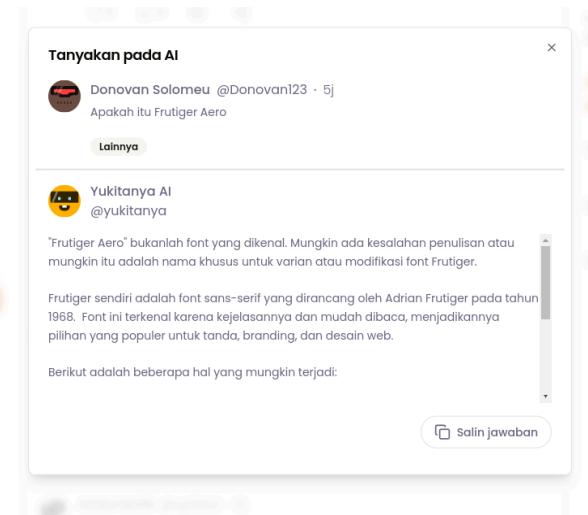
6. Fitur Buat Jawaban

Untuk menjawab pertanyaan, pengguna dapat mengklik tombol "Jawab" pada setiap pertanyaan. Setelah itu, jendela dialog akan muncul untuk memasukkan jawaban secara rinci. Sebelum mengirimkan jawaban, pengguna harus menyelesaikan CAPTCHA untuk mencegah spam dan memastikan jawaban berasal dari pengguna asli, bukan bot.



Gambar 5.9 Dialog Buat Jawaban

7. Fitur Tanyakan Pada AI

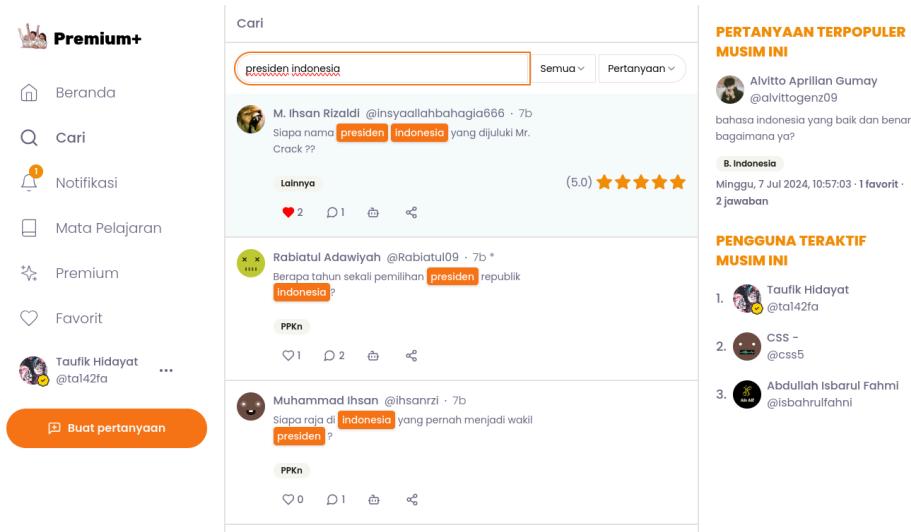


Gambar 5.10 Fitur Tanyakan Pada AI

Salah satu manfaat yang didapatkan pengguna yang menjadi membership premium adalah dapat menggunakan fitur Tanyakan pada AI. Dengan fitur ini pengguna dapat secara langsung menanyakan pertanyaan yang dia pilih kepada Google Gemini AI. Fitur ini memanfaatkan *SDK* dan *API Key* yang telah disediakan Google Gemini Studio. Cara kerjanya adalah ketika pengguna menekan tombol “Tanyakan pada AI”, maka sistem akan mengirimkan pertanyaan tersebut ke Google Gemini API. Jika berhasil maka Google Gemini API akan mengirimkan kembali *output* dari jawaban AI dan kemudian akan ditampilkan di aplikasi

8. Halaman Pencarian

Halaman ini berfungsi untuk mencari jawaban berdasarkan kata kunci dan mata pelajaran. Halaman ini juga bisa mencari pengguna lain berdasarkan *username* atau nama. Fitur ini mengintegrasikan *search engine* Typesense yang membuat proses pencarian menjadi lebih akurat dan cepat.



Gambar 5.11 Halaman Pencarian

9. Halaman Notifikasi

Di halaman ini, pengguna dapat melihat notifikasi terkait aktivitas mereka, termasuk pemberitahuan tentang jawaban baru pada pertanyaan mereka, rating baru untuk jawaban yang mereka berikan, dan jika jawaban mereka dipilih sebagai jawaban terbaik. Pengguna juga diberi tahu ketika pertanyaan

mereka menjadi favorit. Notifikasi ini membantu pengguna mengikuti perkembangan dan respons terhadap kontribusi mereka di platform.

Notifikasi

- Shirookami Kun menjadikan jawaban Anda menjadi yang terbaik · 3j
Penemu teori heliosentris adalah Nicolaus Copernicus
- Abdullah Isbarul Fahmi menyukai pertanyaan Anda · 8j
Explain the difference between the present perfect tense and the past perfect tens...
- Abdullah Isbarul Fahmi menjawab pertanyaan Anda · 8j
Present perfect tense digunakan untuk menunjukkan peristiwa atau kegiatan yang...
- Tri Ramdhani menjawab pertanyaan Anda · 8j
Beras jadi molah 🤤
- Tri Ramdhani menyukai pertanyaan Anda · 8j
Explain the difference between the present perfect tense and the past perfect tens...

PERTANYAAN TERPOPULER MUSIM INI

- Alvitto Aprilian Gumay @alvittogenz09 bahasa indonesia yang baik dan benar bagaimana ya?
- B. Indonesia Minggu, 7 Jul 2024, 10:57:03 · 1 favorit · 2 jawaban

PENGUNA TERAKTIF MUSIM INI

- Taufik Hidayat @ta142fa
- CSS - @css5
- Abdullah Isbarul Fahmi @isbahrulfahni

Gambar 5.12 Halaman Notifikasi

10. Halaman Mata Pelajaran

Mata Pelajaran	
IPA	IPS
Matematika	B. Indonesia
B. Inggris	PPKn
Lainnya	

MATA PELAJARAN TERPOPULER MUSIM INI

- Lainnya
- IPA
- Matematika

PERTANYAAN TERPOPULER MUSIM INI

- Alvitto Aprilian Gumay @alvittogenz09 bahasa indonesia yang baik dan benar bagaimana ya?
- B. Indonesia Minggu, 7 Jul 2024, 10:57:03 · 1 favorit · 2 jawaban

PENGUNA TERAKTIF MUSIM INI

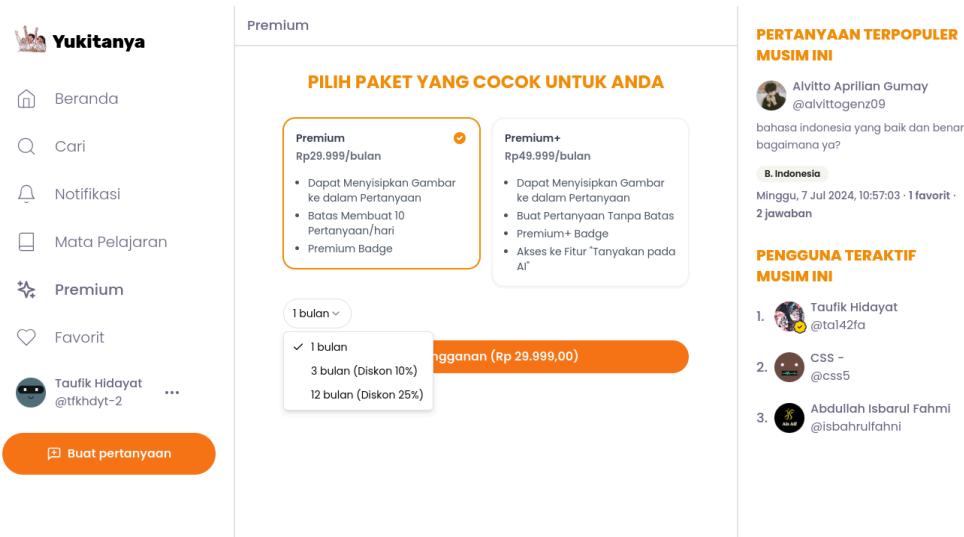
- Taufik Hidayat @ta142fa
- CSS - @css5
- Abdullah Isbarul Fahmi @isbahrulfahni

Gambar 5.13 Halaman Mata Pelajaran

Halaman ini berfungsi untuk melihat pertanyaan berdasarkan mata pelajaran tertentu.

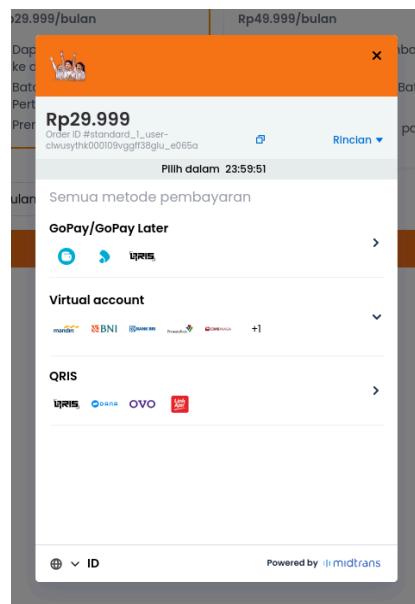
11. Halaman Premium

Halaman ini menampilkan daftar paket *membership* yang tersedia. Pengguna dapat memilih paket dan durasi (1 bulan hingga 1 tahun), lalu menekan tombol beli untuk melanjutkan pembayaran melalui dialog *payment gateway*.



Gambar 5.14 Halaman Premium

12. Dialog Pembayaran



Gambar 5.15 Dialog Pembayaran

Setelah user menekan tombol beli, maka akan muncul dialog pembayaran yang merupakan hasil integrasi dengan *payment gateway* Midtrans. Di halaman ini user dapat memilih metode pembayaran yang ingin dipilih.

13. Halaman Profil

Halaman ini menyajikan profil pengguna secara komprehensif, menampilkan berbagai informasi penting seperti foto profil, nama lengkap, username, jumlah favorit yang diterima, serta daftar lengkap pertanyaan yang diajukan dan jawaban yang diberikan oleh pengguna tersebut. Dengan adanya informasi ini, pengguna lain dapat memperoleh gambaran yang lebih mendalam tentang identitas pengguna, termasuk aktivitas dan kontribusi yang telah mereka berikan di platform. Hal ini tidak hanya memudahkan dalam memahami siapa pengguna tersebut, tetapi juga memberikan wawasan yang lebih jelas mengenai peran dan keterlibatan mereka dalam komunitas, serta bagaimana mereka berinteraksi dan berkontribusi dalam berbagai diskusi yang ada.

Gambar 5.16 Halaman Profil

5.2 Pengujian

Setelah tahap implementasi pengembangan aplikasi selesai, langkah berikutnya adalah melakukan pengujian. Penulis melakukan pengujian menggunakan metode *black box testing* untuk memeriksa fitur-fitur yang telah dikembangkan. Selain itu, *black box testing* juga dilakukan oleh 9 pengguna. Hasil dari penilaian pengujian tersebut adalah sebagai berikut, dan hasil pengujian *black box* lebih lengkap akan di tampilkan di lampiran 8:

Tabel 5.3 Hasil Uji oleh Pengguna

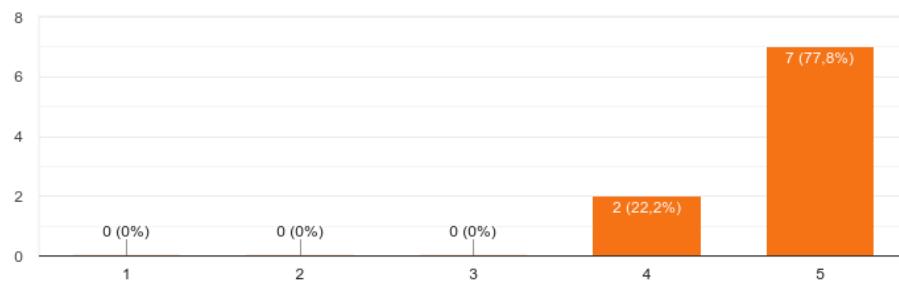
No	Jenis Pengujian	Presentase Capaian
1	Autentikasi (Register, Login, dan Logout)	86%
2	Halaman Beranda	100%
3	Fitur CAPTCHA	100%
4	Pengelolaan Pertanyaan	100%
5	Filtering Kata Terlarang	75%
6	Fitur Tanyakan pada AI	100%
7	Pengelolaan Jawaban	100%
8	Fitur Pencarian Pertanyaan dan Pengguna	89%
9	Fitur Notifikasi	89%
10	Halaman Profil	100%
Rata – rata seluruh aspek yang diukur		93.9%

Berikut adalah data kuesioner yang mencakup hasil pengujian validitas dan reliabilitas untuk memastikan data akurat dan dapat diandalkan. Hasil pengujian ini menampilkan persentase capaian pada tabel sebelumnya, memberikan gambaran performa kuesioner. Nilai akhir diperoleh dari rata-rata data di Google Sheets untuk hasil representatif. Jawaban atas berbagai pertanyaan dalam kuesioner ini memberikan wawasan mendalam tentang aspek yang diukur dan membantu memahami perspektif responden.

Apakah fitur registrasi menggunakan email, username, dan password mudah digunakan?

Salin

9 jawaban

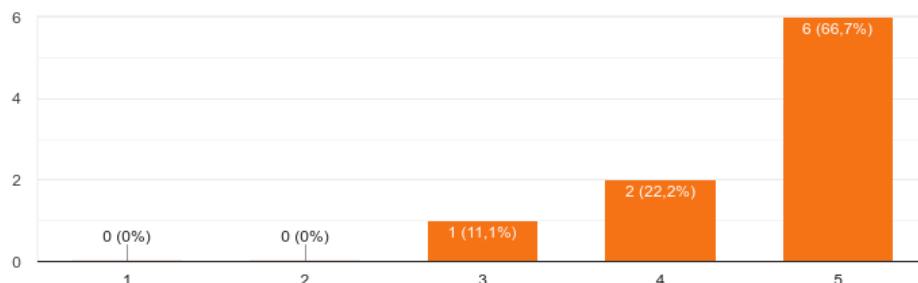


Gambar 5.17 Grafik pertanyaan kuisioner 1

Apakah Anda merasa nyaman menggunakan OAuth (Facebook atau Google) untuk registrasi?

Salin

9 jawaban

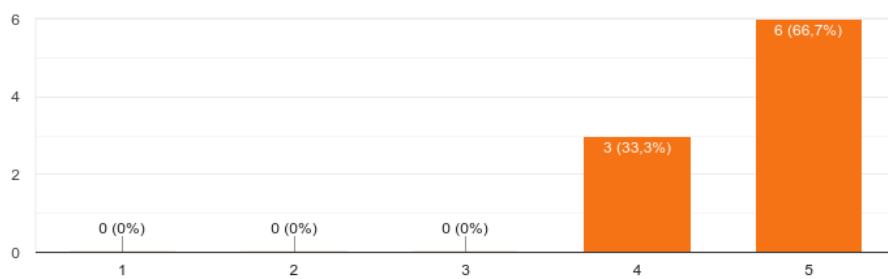


Gambar 5.18 Grafik pertanyaan kuisioner 2

Apakah fitur login menggunakan username dan password mudah diakses?

Salin

9 jawaban

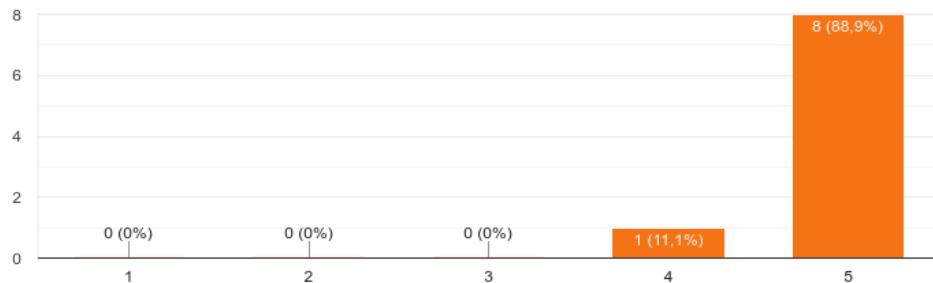


Gambar 5.19 Grafik pertanyaan kuisioner 3

Apakah fitur login menggunakan akun Facebook atau Google mudah diakses?

 Salin

9 jawaban

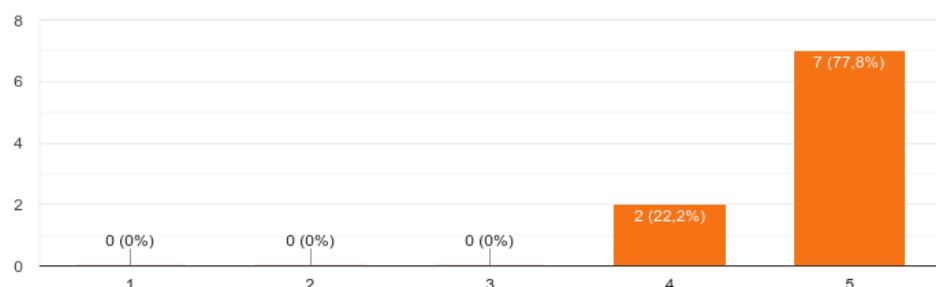


Gambar 5.20 Grafik pertanyaan kuisioner 4

Apakah tampilan pertanyaan di halaman beranda mudah dipahami dan menarik?

 Salin

9 jawaban

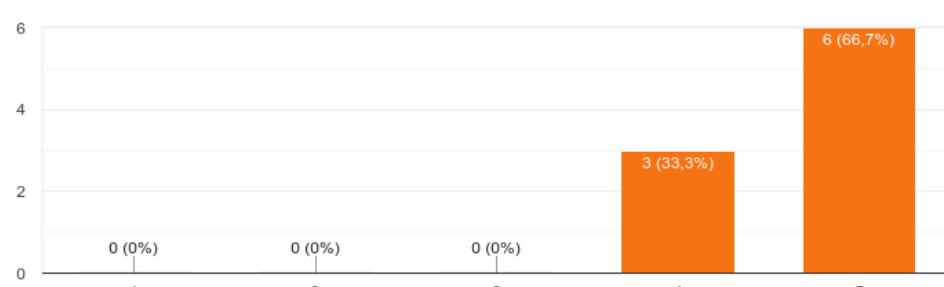


Gambar 5.21 Grafik pertanyaan kuisioner 5

Apakah proses pembuatan pertanyaan baru mudah dipahami?

 Salin

9 jawaban

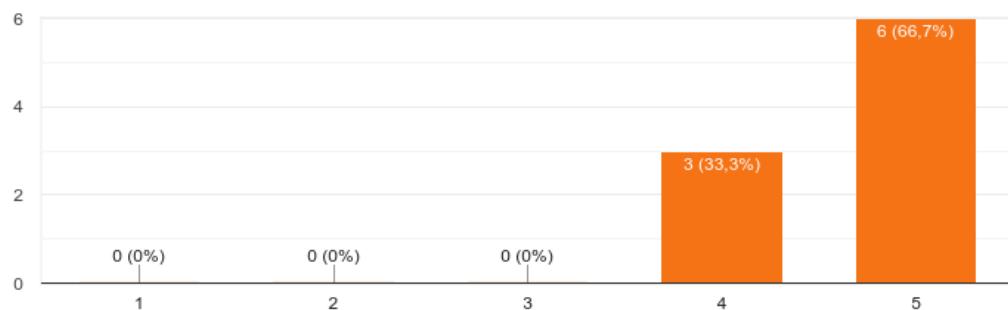


Gambar 5.22 Grafik pertanyaan kuisioner 6

Apakah fitur untuk mengedit pertanyaan mudah digunakan?

Salin

9 jawaban

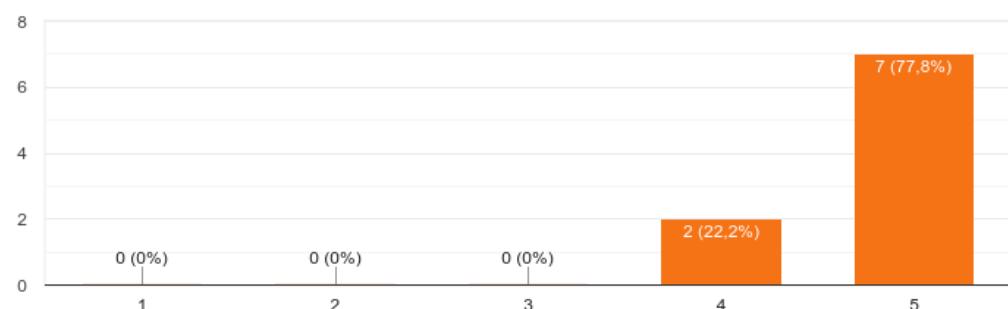


Gambar 5.23 Grafik pertanyaan kuisioner 7

Apakah fitur untuk menghapus pertanyaan mudah digunakan?

Salin

9 jawaban

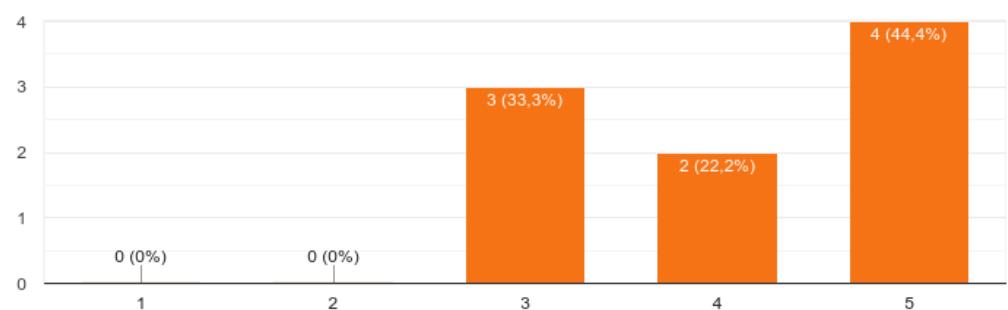


Gambar 5.24 Grafik pertanyaan kuisioner 8

Apakah menurut Anda fitur CAPTCHA efektif dalam mencegah spam?

Salin

9 jawaban

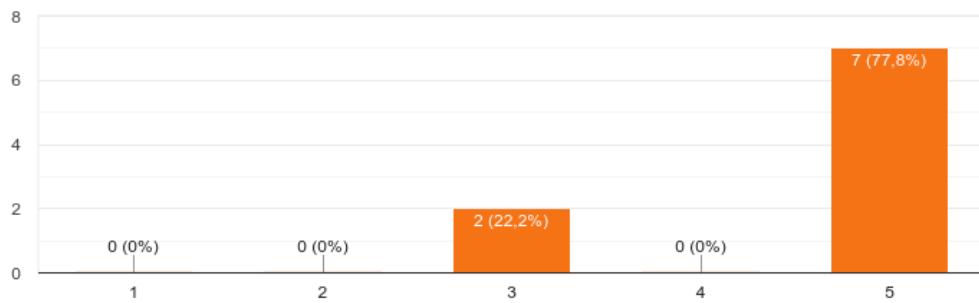


Gambar 5.25 Grafik pertanyaan kuisioner 9

Apakah CAPTCHA mudah diisi oleh pengguna?

 Salin

9 jawaban

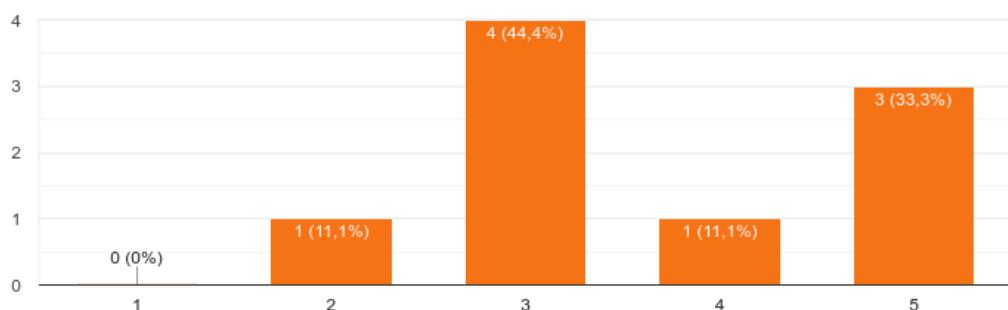


Gambar 5.26 Grafik pertanyaan kuisioner 10

Apakah fitur filtering kata efektif dalam mencegah penggunaan kata-kata terlarang?

 Salin

9 jawaban

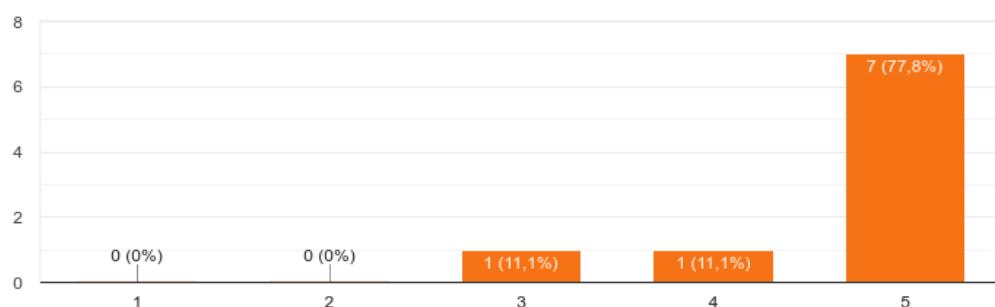


Gambar 5.27 Grafik pertanyaan kuisioner 11

Apakah tampilan halaman detail pertanyaan menarik dan mudah dipahami?

 Salin

9 jawaban

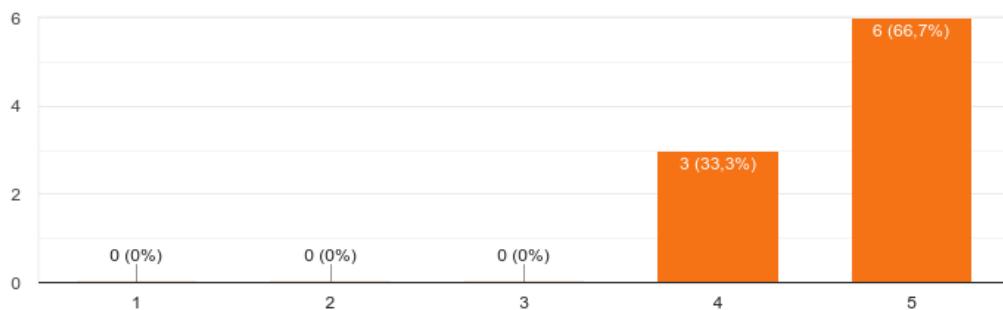


Gambar 5.28 Grafik pertanyaan kuisioner 12

Apakah fitur menandai pertanyaan sebagai favorit mudah digunakan?

 Salin

9 jawaban

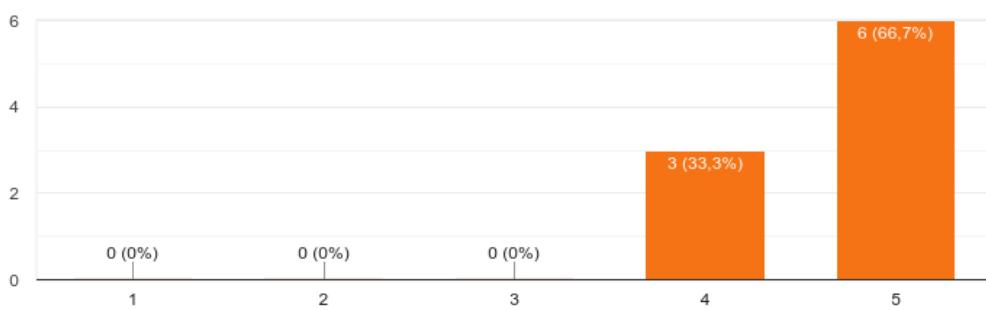


Gambar 5.29 Grafik pertanyaan kuisioner 13

Apakah proses pembuatan jawaban untuk pertanyaan mudah dipahami?

 Salin

9 jawaban

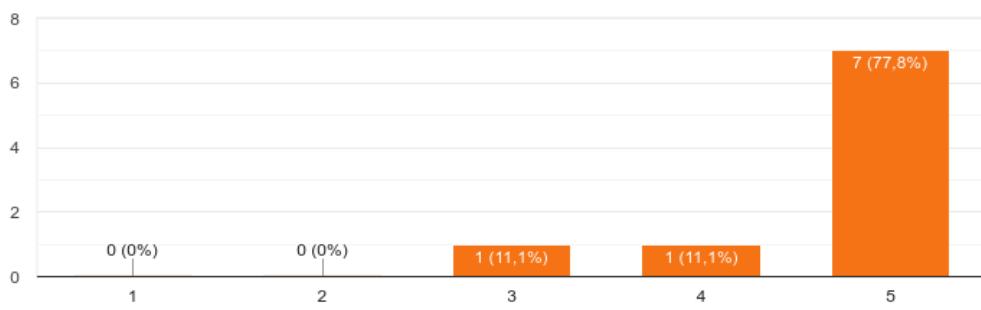


Gambar 5.30 Grafik pertanyaan kuisioner 14

Apakah fitur rating ini membantu Anda menilai kualitas jawaban?

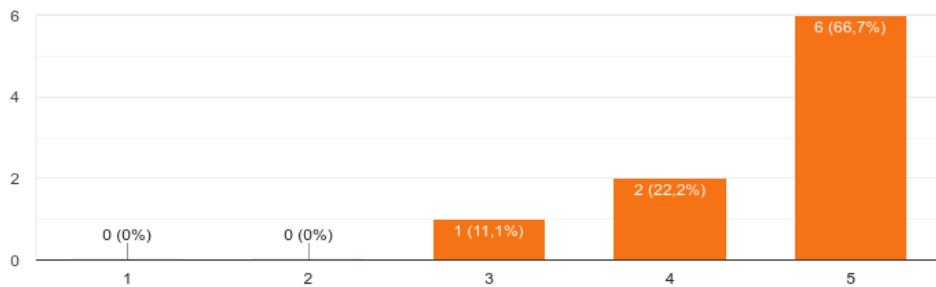
 Salin

9 jawaban



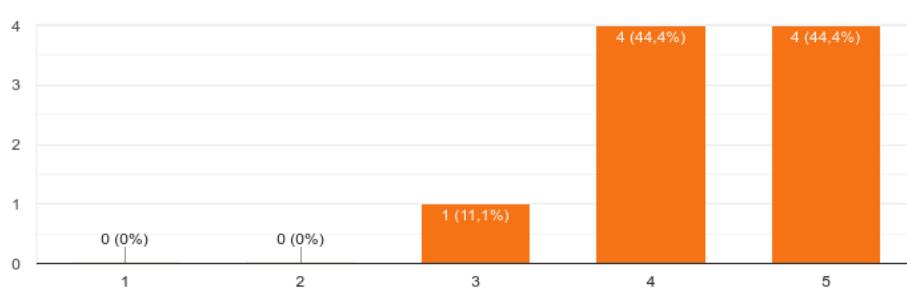
Gambar 5.31 Grafik pertanyaan kuisioner 15

Apakah fitur menandai jawaban sebagai “jawaban terbaik” mudah digunakan? Salin
9 jawaban



Gambar 5.32 Grafik pertanyaan kuisioner 16

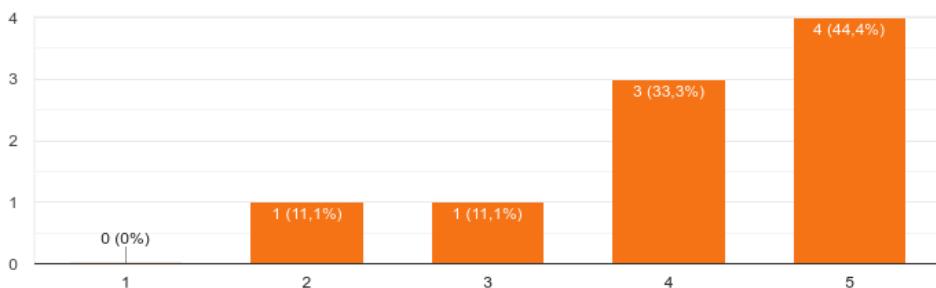
Apakah fitur share ini membantu Anda dalam membagikan pertanyaan yang menarik? Salin
9 jawaban



Gambar 5.33 Grafik pertanyaan kuisioner 17

Apakah fitur leaderboard ini membantu Anda untuk melihat pengguna teraktif dan pertanyaan terpopuler? Salin

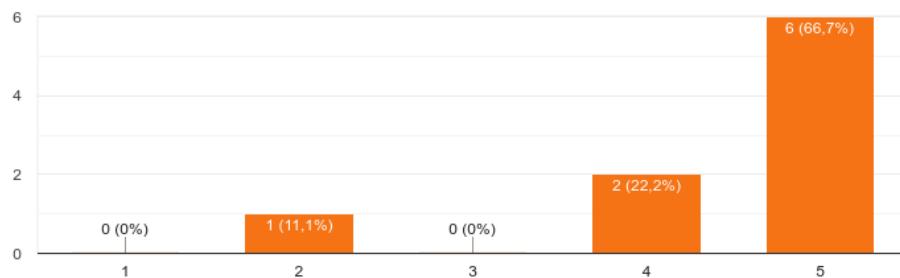
9 jawaban



Gambar 5.34 Grafik pertanyaan kuisioner 18

Apakah fitur pencarian membantu Anda menemukan pertanyaan atau profil pengguna dengan mudah? [□ Salin](#)

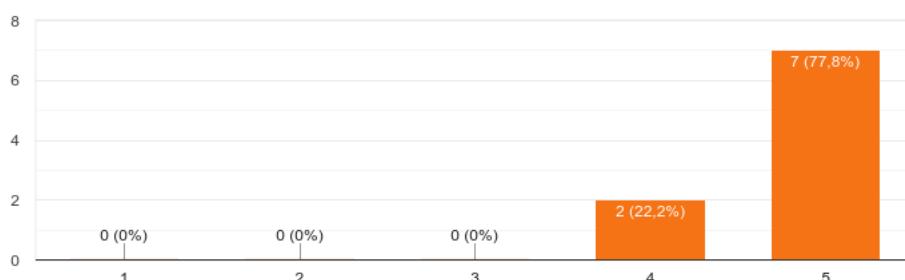
9 jawaban



Gambar 5.35 Grafik pertanyaan kuisioner 19

Apakah tampilan halaman mata pelajaran ini menarik dan mudah dipahami? [□ Salin](#)

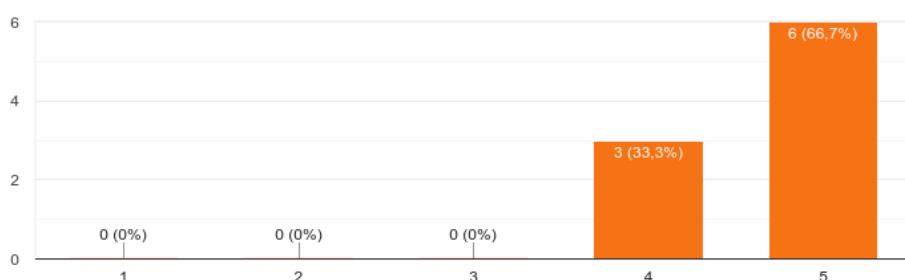
9 jawaban



Gambar 5.36 Grafik pertanyaan kuisioner 20

Apakah tampilan halaman profil menarik dan mudah dipahami? [□ Salin](#)

9 jawaban

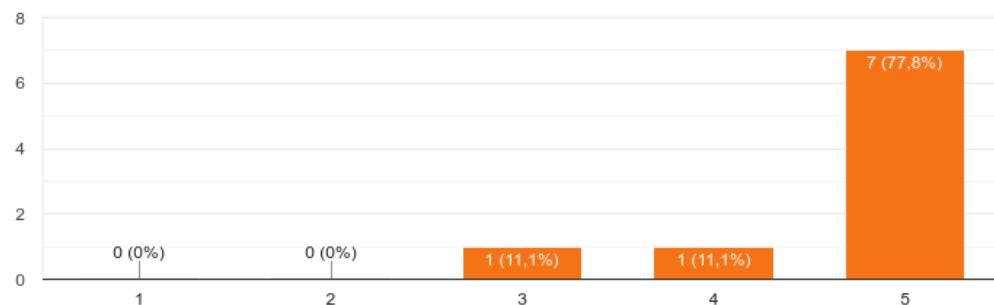


Gambar 5.37 Grafik pertanyaan kuisioner 21

Apakah fitur notifikasi membantu Anda untuk tetap update dengan kejadian yang terjadi?

 Salin

9 jawaban

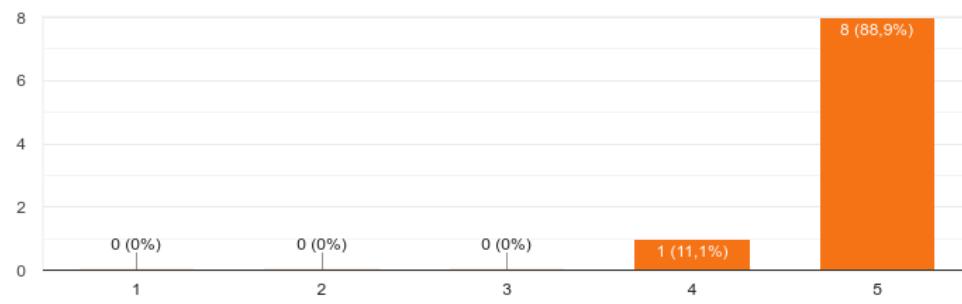


Gambar 5.38 Grafik pertanyaan kuisioner 22

Apakah fitur logout mudah untuk diakses?

 Salin

9 jawaban



Gambar 5.39 Grafik pertanyaan kuisioner 23

Hasil pengujian yang dilakukan oleh pengguna menunjukkan bahwa capaian untuk autentikasi adalah 86%, halaman beranda 100%, fitur CAPTCHA 100%, pengelolaan pertanyaan 100%, filtering kata terlarang 75%, fitur tanyakan pada AI 100%, pengelolaan jawaban 100%, fitur pencarian 89%, fitur notifikasi 89%, dan halaman profil 100% dengan rata-rata sebesar 93,9%. Secara keseluruhan, hasil pengujian menunjukkan bahwa aplikasi sudah berfungsi dengan baik, tetapi masih ada beberapa perbaikan yang diperlukan, terutama pada fitur filtering kata terlarang.

BAB VI

KESIMPULAN

6.1 Kesimpulan

Berdasarkan penelitian yang dilakukan penulis melalui beberapa tahapan yang dilakukan pada bab-bab sebelumnya, maka dapat disimpulkan bahwa:

1. Implementasi sistem filtering kata otomatis menggunakan algoritma Knuth-Morris-Prath telah berhasil mencegah munculnya kata-kata tidak sopan dalam pertanyaan dan jawaban di aplikasi forum tanya jawab Yukitanya. Sistem ini memfilter kata-kata tidak sopan secara *real-time* sebelum konten dipublikasikan.
2. Serangan spam di aplikasi forum tanya jawab Yukitanya dapat diatasi melalui mekanisme verifikasi pengguna menggunakan CAPTCHA. Algoritma Cloudflare Turnstile digunakan untuk mendeteksi dan memblokir aktivitas mencurigakan secara otomatis.
3. Meningkatkan minat orang dalam menggunakan aplikasi forum tanya jawab Yukitanya dicapai dengan menerapkan beberapa fitur baru, seperti fitur integrasi dengan Google Gemini AI, fitur leaderboard pengguna teraktif, tampilan antarmuka yang *user-friendly*, dan *SEO Optimization* agar website dapat terindeks oleh *Search Engine* seperti Google.
4. Hasil pengujian yang dilakukan oleh pengguna menunjukkan bahwa capaian untuk autentikasi adalah 86%, halaman beranda 100%, fitur CAPTCHA 100%, pengelolaan pertanyaan 100%, filtering kata terlarang 75%, fitur tanyakan pada AI 100%, pengelolaan jawaban 100%, fitur pencarian 89%, fitur notifikasi 89%, dan halaman profil 100% dengan rata-rata sebesar 93,9%. Secara keseluruhan, hasil pengujian menunjukkan bahwa aplikasi sudah berfungsi dengan baik, tetapi masih ada beberapa perbaikan yang diperlukan, terutama pada fitur *filtering* kata terlarang.

6.2 Saran

Berdasarkan hasil pengembangan dan implementasi aplikasi forum tanya jawab Yukitanya, terdapat beberapa rekomendasi untuk pengembangan lebih lanjut. Saran-saran berikut bertujuan untuk meningkatkan fungsionalitas, keamanan, dan daya tarik aplikasi, sehingga dapat memberikan manfaat yang lebih besar bagi penggunanya dalam konteks pendidikan.

a. Saran untuk Akademisi

1. Pengembangan dataset kata terlarang perlu ditingkatkan dengan memperluas dan memperbarui kumpulan kata secara berkala serta mempertimbangkan penggunaan teknik *machine learning*. Dengan ini, sistem dapat otomatis mendeteksi kata atau frasa tidak pantas, termasuk yang tersamarkan, sehingga *filtering* menjadi lebih dinamis dan efektif dalam menghadapi evolusi bahasa dan slang.
2. Diversifikasi sumber AI sangat direkomendasikan untuk meningkatkan akurasi dan keandalan jawaban di aplikasi Yukitanya. Saat ini, aplikasi bergantung pada model AI Google Gemini yang memiliki keterbatasan dan potensi bias. Mengintegrasikan beberapa model AI dapat mengurangi ketergantungan pada satu sumber dan meningkatkan akurasi melalui pendekatan konsensus. Sistem validasi silang dan pengembangan model AI khusus dengan dataset pendidikan juga dapat meningkatkan kesesuaian jawaban dengan konteks pembelajaran di Indonesia.

b. Saran untuk Perusahaan

1. Untuk meningkatkan eksposur dan daya tarik aplikasi Yukitanya, perusahaan sebaiknya mengadopsi strategi pemasaran yang efektif, termasuk kampanye media sosial dan kolaborasi dengan *influencer* di bidang terkait. Pendekatan ini dapat membantu menjangkau audiens yang lebih luas dan menarik pengguna baru. Selain itu, perusahaan dapat mempertimbangkan kerja sama dengan institusi pendidikan untuk mempromosikan aplikasi sebagai alat pembelajaran yang efektif.

DAFTAR PUSTAKA

- Achmad, Y. F., & Yulfitri, A. (2020). Pengujian sistem pendukung keputusan menggunakan black box testing studi kasus e-wisudawan di Institut Sains dan Teknologi Al-Kamal. *Jurnal Ilmu Komputer*, 5(1), 42.
- Anggara, M. B., & Muslimin, I. A. (2023). OPTIMASI PENGELOLAAN PERSEDIAAN BARANG MENGGUNAKAN METODE PERPETUAL PADA APLIKASI INVENTORY DI PT. VISI KARYA PRAKARSA. *COMPUTING| Jurnal Informatika*, 10(02), 77-81.
- Anggara, M. B., & Zaman, F. (2024). DESAIN ARSITEKTUR SERVER GOOGLE CLOUD UNTUK MENGOPTIMALKAN KINERJA APLIKASI DAILY CLOUD DALAM PEMANTAUAN KESEHATAN MENTAL. *COMPUTING| Jurnal Informatika*, 11(01), 14-21.
- Atmadja, A. R., & Pariyatin, Y. (2020). Rancang Bangun Aplikasi Forum Diskusi Usaha Mikro Kecil dan Menengah Berbasis Web. *Jurnal Algoritma*, 17(2), 168-175.
- Aziz, F. A., & Ikasari, I. H. (2023). Aplikasi Forum Diskusi Seputar Game Esport Berbasis Web Dengan Metode Extreme Programming. *Jurnal Informatika Multi*, 1(4), 421-429.
- Dwi Riyanto, N. R., Alfian, M., & Yuhana, U. L. (2023). Evaluasi Efisiensi Kinerja Object Relational Mapping pada Web API Point of Sale Menggunakan ISO/IEC 25010. *Jurnal Ilmu Komputer dan Agri-Informatika*, 10(1).
- Fowler, S., & Stanwick, V. (2004). *Web application design handbook: Best practices for web-based software*. Morgan Kaufmann.
- Hamida, S., & Putra, E. D. (2021). Peran Orang Tua dalam Meningkatkan Motivasi Belajar Siswa di Masa Pandemi COVID-19. *Mimbar Ilmu*, 26(2), 302-308.

- Handayani, H., Ayulya, A. M., Faizah, K. U., Wulan, D., Rozan, M. F., & Hamzah, M. L. (2023). Perancangan Sistem Informasi Inventory Barang Berbasis Web Menggunakan Metode Agile Software Development. *Jurnal Testing Dan Implementasi Sistem Informasi*, 1(1), 29-40.
- Herdiana, Y. (2023). Perancangan Aplikasi E-Arsip Menggunakan Algoritma Sequential Search Untuk Pengarsipan Surat Pada PT. Visi Karya Prakarsa. *COMPUTING| Jurnal Informatika*, 10(02), 72-76.
- Herdiana, Y., Rusdianto, D., & Geraldine, W. A. (2023). APLIKASI CV MATCHER UNTUK MELIHAT KECOCOKAN DAFTAR RIWAYAT HIDUP DENGAN LOWONGAN PEKERJAAN MENGGUNAKAN MACHINE LEARNING DAN METODE COSINE SIMILARITY BERBASIS WEB. *COMPUTING| Jurnal Informatika*, 10(01), 26-30.
- Herdiana, Y., & Kusumah, T. (2024). REKAYASA PERANGKAT LUNAK APLIKASI LIBRARY APP MENGGUNAKAN JSON UNTUK MEREKAM DATA BUKU DI PT RUANG RAYA INDONESIA. *COMPUTING| Jurnal Informatika*, 11(01), 1-6.
- Hidayat, T. (2024). PENGEMBANGAN APLIKASI BACK-END E-COMMERCE MENGGUNAKAN REST API GOLANG UNTUK OPTIMALISASI KINERJA SERVER. *COMPUTING| Jurnal Informatika*, 11(01), 7-13.
- Jefri, M., Anggraini, D., & Oktora, M. Z. (2022). Validitas dan Reliabilitas Kuesioner Penilaian Pembelajaran Skills Lab Secara Daring pada Mahasiswa Fakultas Kedokteran Universitas Baiturrahmah Padang. *Scientific Journal*, 1(1), 36-45.
- Ilham, M., & Mirza, A. H. (2020). Penerapan Algoritma Knuth Morris Pratt Dalam Fitur Pencarian Pengarsipan Dokumen Pada Sma Plus Negeri 17 Palembang. *Journal of Software Engineering Ampera*, 1(2), 110-121.
- Imamah, N. (2023). PENJADWALAN DAN PEMBERIAN PAKAN IKAN OTOMATIS BERBASIS INTERNET OF THING MENGGUNAKAN

- NODEMCU ESP8266 DAN APLIKASI BLYNK STUDI KASUS: TOKO FISH FRIENDLY. *COMPUTING| Jurnal Informatika*, 10(02).
- Kelirik, N. (2018). Penerapan Metode Diskusi Kelompok Untuk Meningkatkan Hasil Belajar Ipa Di Sekolah Dasar Negeri 1 Sukadana. *Jurnal Ika*, 16(1), 1-11.
- Muharam, Y., Anggara, M. B., & Hanafi, T. J. (2023). IMPLEMENTASI PETA 3 DIMENSI MENGGUNAKAN METODE IMSDD (INTERACTIVE MULTIMEDIA SYSTEM DESIGN AND DEVELOPMENT) DAN WEBGL API BERBASIS WEB. *COMPUTING| Jurnal Informatika*, 10(01), 37-42.
- Muharam, Y., & Rustiyana, D. N. RANCANG BANGUN APLIKASI PERSEDIAAN KAIN BERBASIS WEB DENGAN METODE REORDER POINT (ROP) DI DEPARTEMEN KREATIF PT. INDO PACIFIC.
- Muntaha, A. F., & Ardita, A. H. (2024). PERANCANGAN SISTEM INFORMASI SEKOLAH BERBASIS WEBSITE DI SMA BPPI BALEENDAH. *COMPUTING| Jurnal Informatika*, 11(01), 38-45.
- Mustafidah, H., Attafaqquf, A., & Pinandita, T. (2021, February). Pengembangan Sistem Informasi Manajemen Klinik Berbasis Relational Data Base Management System (RDBMS). In PROSIDING SEMINAR NASIONAL LPPM UMP (Vol. 2, pp. 488-494).
- Nistrina, K. (2023). PERANCANGAN DAN IMPLEMENTASI TEMPAT SAMPAH OTOMATIS BERBASIS ARDUINO UNO MENGGUNAKAN ALGORITMA FUZZY LOGIC. *COMPUTING| Jurnal Informatika*, 10(01), 21-25.
- Nistrina, K. (2024). PERANCANGAN APLIKASI PEMBUKUAN PERPUSTAKAAN BERBASIS WEB DI SMP KP 2 MAJALAYA. *COMPUTING| Jurnal Informatika*, 11(01), 46-52.
- Nugroho, R. A. (2021). Pengaruh penggunaan brainly terhadap hasil belajar siswa. *Jurnal Epistema*, 2(2), 76-92.

- Pendry, L. F., & Salvatore, J. (2015). Individual and social benefits of online discussion forums. *Computers in Human Behavior*, 50, 211-220.
- Salendra, D. (2024). PERANCANGAN USER INTERFACE FINANSIAL DAN INVESTASI MONEY MENTOR PRO BERBASIS WEB DENGAN METODE DESIGN THINKING. *COMPUTING| Jurnal Informatika*, 11(01), 22-27.
- Saputri, D., & Purnasari, P. D. (2023). Implementasi Kurikulum 2013 Terhadap Pentingnya Pengaruh Pendidikan Karakter (PPK) di Sekolah Dasar. *ALPEN: Jurnal Pendidikan Dasar*, 7(1), 63-76.
- Sarie, F., Sutaguna, I. N. T., Par, S. S., Par, M., Suiraoka, I. P., St, S., ... & Massenga, I. T. W. (2023). Metodelogi penelitian. Cendikia Mulia Mandiri.
- Satria, F., Chasanah, N., & Iskandar, D. (2020). Aplikasi Forum Diskusi Himpunan Mahasiswa Teknik Universitas Jenderal Soedirman Menggunakan Laravel. *Jurnal Teknik Informatika (JUTIF)*, 1(1), 1-6.
- Suharya, Y., & Imamah, N. (2023). PENGEMBANGAN APLIKASI PEMBELAJARAN JARINGAN KOMPUTER MENGGUNAKAN PNETLAB UNTUK SEKOLAH MENENGAH KEJURUAN JURUSAN TEKNIK KOMPUTER DAN JARINGAN DENGAN METODE WEB BASED LEARNING. *COMPUTING| Jurnal Informatika*, 10(01), 31-36.
- Suharya, Y., & Reygina, E. (2024). PERANCANGAN APLIKASI REKAP PENGUMPULAN TUGAS SISWA/I BERBASIS WEB DI SMA BPPI BALEENDAH. *COMPUTING| Jurnal Informatika*, 11(01), 28-37.
- Triawan, A., & Siboro, A. R. Y. (2021). Penerapan Application Programming Interface (API) Pada Push Notification Untuk Informasi Monitoring Stok Barang Minim. *TeknoIS: Jurnal Ilmiah Teknologi Informasi dan Sains*, 11(2), 107-114.
- Ulvi, H. (2023). ANALISIS PENGUJIAN KECEPATAN AKSES INTERNET PADA WIFI INDIHOME DENGAN APLIKASI SPEEDTEST DAN

SPEEDCHECK DI DINAS KEBUDAYAAN, PARIWISATA DAN EKONOMI KREATIF PROVINSI SUMATERA UTARA. COMPUTING| Jurnal Informatika, 10(02), 42-45.

Wijaya, E. P., Kosasi, S., & David, D. (2021). Implementasi Aplikasi Web Full Stack Pendataan Cloversy. id. Jurnal Sisfokom (Sistem Informasi dan Komputer), 10(3), 320-327.

Wisnuwardhana, A. P., Wolor, C. W., & Marsofiyati, M. (2024). Analisis Produktivitas Kerja Karyawan Pada CV Griya Alam Mulya. Jurnal Manajemen Riset Inovasi, 2(1), 13-27.

Zam, E. M. (2021). Peran Literasi Teknologi Informasi Dan Komunikasi Pada Pembelajaran Jarak Jauh Di Masa Pandemi Covid-19. *EDUTECH: Jurnal Inovasi Pendidikan Berbantuan Teknologi*, 1(1), 9-18.

LAMPIRAN

Lampiran 1: TOR

Sebelum melakukan penelitian skripsi penulis melakukan beberapa metode penelitian yaitu observasi dan studi pustaka. Setelah mengamati dan mempelajari lokasi penelitian yang telah ditentukan dan disetujui oleh instansi tempat penelitian yaitu di *Forest Interactive* yang berlokasi di Jl. Suryanagara No.27, RT.06/RW.05, Cibaduyut Wetan, Kec. Bojongloa Kidul, Kota Bandung, Jawa Barat. Pada semester 6, penulis telah melakukan Studi Independen di *Forest Interactive*. Di sana penulis telah membuat sebuah aplikasi forum tanya jawab tugas sekolah (Yukitanya) untuk menyelesaikan tugas sekolah secara bersama-sama. Namun masih ada beberapa masalah yang terjadi seperti belum ada fitur *filtering* kata pada pembuatan pertanyaan dan jawaban, masih rentan terhadap serangan *spam*, dan jumlah pengguna yang semakin menurun. Penelitian ini bertujuan untuk mengembangkan fitur *filtering* kata terlarang menggunakan algoritma Knuth-Morris-Pratt, mengimplementasikan metode CAPTCHA untuk mencegah *spam*, membuat fitur fitur menarik lain, dan melakukan optimasi SEO untuk menyelesaikan masalah yang telah disebutkan sebelumnya. Bahasa pemrograman yang digunakan adalah TypeScript, *framework* Next.js, dan *database* PostgreSQL. Untuk memastikan fokus penelitian adapun batasan masalah yang ditetapkan adalah sebagai berikut:

1. Aplikasi ini dibangun menggunakan bahasa pemrograman TypeScript, *framework* Next.js, dan *database* PostgreSQL.
2. Aplikasi ini menggunakan algoritma Knuth-Morris-Pratt untuk mendeteksi adanya kata-kata tidak sopan.
3. Aplikasi ini menggunakan metode CAPTCHA milik Cloudflare yaitu Turnstile untuk mencegah terjadinya serangan spam.
4. Pada aplikasi ini hanya mencakup aplikasi web saja tanpa adanya aplikasi mobile maupun desktop.
5. Aplikasi ini mengambil referensi dari Brainly.

Bandung, Mei 2024

Disetujui oleh:

Mahasiswa

Taufik Hidayat

Program Assistant Executive

Forest Interactive



Intan Prasanty Ardiningrum

Lampiran 2: Data Penurunan Jumlah Pengguna Aplikasi dari Database

1. Jumlah Pengguna Baru per Bulan

```
↳ Yukitanya - Jumlah Pengguna x +  
  
1 SELECT  
2   TO_CHAR(DATE_TRUNC('month', created_at), 'YYYY-MM') AS bulan,  
3   COUNT(*) AS jumlah_pengguna_baru  
4 FROM  
5   "user"  
6 GROUP BY  
7   TO_CHAR(DATE_TRUNC('month', created_at), 'YYYY-MM')  
8 ORDER BY  
9   bulan;
```


	bulan	jumlah_pengguna_baru
1	2023-12	81
2	2024-01	13
3	2024-02	3
4	2024-03	4
5	2024-04	6
6	2024-05	4

2. Jumlah Pertanyaan Baru per Bulan

```
↳ Yukitanya - Jumlah Pengguna x ↳ Yukitanya - Jumlah Pertanyaan x +  
  
1 SELECT  
2   TO_CHAR(DATE_TRUNC('month', created_at), 'YYYY-MM') AS bulan,  
3   COUNT(*) AS jumlah_pertanyaan_baru  
4 FROM  
5   question  
6 GROUP BY  
7   TO_CHAR(DATE_TRUNC('month', created_at), 'YYYY-MM')  
8 ORDER BY  
9   bulan;
```


	bulan	jumlah_pertanyaan_baru
1	2023-12	67
2	2024-01	1
3	2024-03	1

3. Jumlah Jawaban Baru per Bulan

```
↳ Yukitanya - Jumlah Pengguna    ↳ Yukitanya - Jumlah Pertanyaan    ↳ Yukitanya - Jumlah Jawaban × +  
1  SELECT  
2      TO_CHAR(DATE_TRUNC('month', created_at), 'YYYY-MM') AS bulan,  
3      COUNT(*) AS jumlah_jawaban_baru  
4  FROM  
5      answer  
6  GROUP BY  
7      TO_CHAR(DATE_TRUNC('month', created_at), 'YYYY-MM')  
8  ORDER BY  
9      bulan;
```

	bulan	jumlah_jawaban_baru
1	2023-12	88
2	2024-01	3
3	2024-03	2

Lampiran 3: Hasil Wawancara dengan Narasumber

Narasumber : Rangga Satrya
Jabatan/Posisi : Senior Web Developer & Mentor
Hari / Tanggal : Selasa, 23 April 2024
Instansi : Forest Interactive

Wawancara ini bertujuan untuk mengumpulkan data dalam penelitian yang berjudul “Pengembangan Aplikasi Forum Tanya Jawab Tugas Sekolah (Yukitanya) Menggunakan Framework Next.js dan PostgreSQL”. Berikut ini disajikan daftar pertanyaan wawancara beserta jawaban terkait.

No.	Pertanyaan	Jawaban
1	Apa saja masalah yang terjadi pada aplikasi Yukitanya?	Masih terdapat pertanyaan dan jawaban yang mengandung kata-kata yang tidak baik, aplikasi masih rentan terhadap serangan <i>spam</i> , jumlah pengguna yang semakin lama semakin menurun.
2	Apa dampak dari serangan <i>spam</i> yang terjadi pada aplikasi Yukitanya?	Munculnya banyak pertanyaan dan jawaban yang dibuat secara <i>auto-generated</i> setiap detik tanpa henti oleh <i>hacker</i> . Hal ini menyebabkan timeline halaman beranda dipenuhi oleh pertanyaan-pertanyaan sampah sehingga pengguna tidak dapat melihat pertanyaan sungguhan
3	Apakah serangan <i>spam</i> ini berpengaruh terhadap performa	Sangat berpengaruh, serangan ini menyebabkan performa aplikasi dan

	aplikasi Yukitanya?	database menjadi melambat karena adanya proses pembuatan pertanyaan otomatis setiap detik yang dilakukan oleh hacker.
4	Kira-kira apa yang menyebabkan menurunnya jumlah pengguna aplikasi Yukitanya?	Alasan pertama adalah karena Yukitanya masih belum memiliki fitur menarik lain yang dapat menarik perhatian orang untuk menggunakan aplikasi ini. Alasan kedua adalah karena aplikasi ini belum begitu dikenal oleh banyak orang.

Pewawancara

Taufik Hidayat

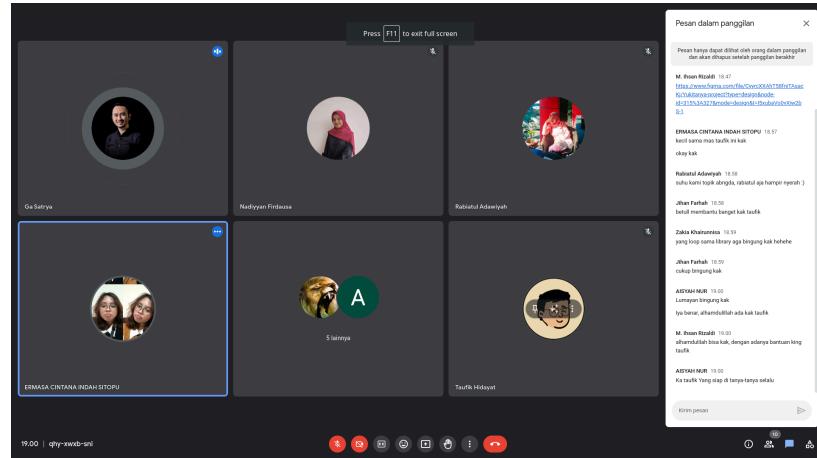
Narasumber



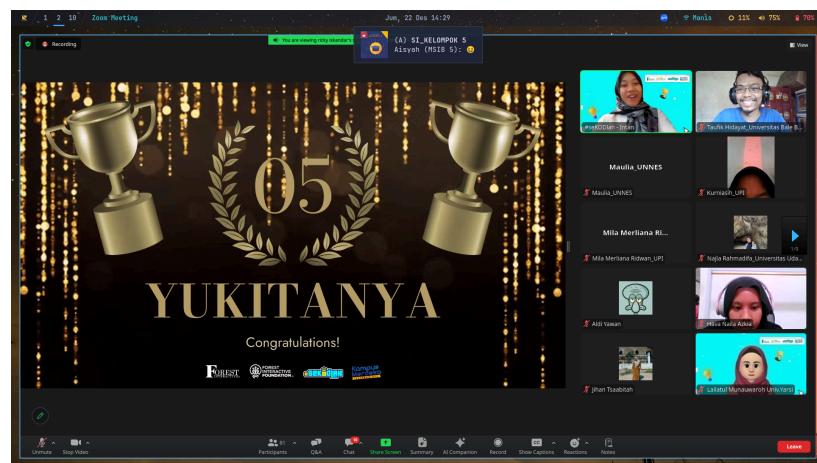
Rangga Satrya

Lampiran 4: Dokumentasi

1. Sesi mentoring antara Mentor dan anggota kelompok selama program Studi Independen di Forest Interactive melalui Google Meet.



2. Yukitanya terpilih sebagai Capstone Project terbaik.



3. Reward yang didapat untuk kelompok Capstone Project Terbaik.



4. Kunjungan ke Kantor Pusat Forest Interactive di Kuala Lumpur, Malaysia bersama mentor dan mahasiswa yang terpilih lainnya.



5. Yukitanya telah di-review oleh Pak Sandhika Galih, seorang dosen dari Universitas Pasundan melalui channel Youtube-nya (Web Programming UNPAS)

A screenshot of a YouTube video player. The video title is "(1B) Pamerin Project Kalian | 20 Juli 2024". The video content shows a person in a dark hoodie and headphones speaking into a microphone. The video interface includes a QR code at the bottom left, a progress bar, and a "187" views counter at the bottom right. The YouTube navigation bar at the top shows "Sat, 20 Jul 2024 22:29" and the URL "https://www.youtube.com/watch?v=Riveng3f6f04". The video is set against a background of a wall with framed certificates and a trophy display. The right side of the screen shows a "Chat teratas" (Top Chat) window with messages from viewers like "Belawan Kusuma", "Panda The Developer", "Ondes", "Gazam", "Ondes", "Taufik Hidayah", "VI", "Ramaedzah", "Henset B", and "M. Jadi". The bottom of the screen shows standard YouTube controls for "Semaia", "Dan Web Programming UNPAS", and "Terbaik".

Lampiran 5: Dataset Kata-kata Terlarang dalam format JSON

```
[  
    "ancuk",  
    "ancok",  
    "ajig",  
    "anjay",  
    "anjing",  
    "anying",  
    "anjir",  
    "asu",  
    "asyu",  
    "babangus",  
    "babi",  
    "bacol",  
    "bacot",  
    "bagong",  
    "bajingan",  
    "balegug",  
    "banci",  
    "bangke",  
    "bangsat",  
    "bedebah",  
    "bedegong",  
    "bego",  
    "belegug",  
    "beloon",  
    "bencong",  
    "bloon",  
    "blo'on",  
    "bodoh",  
    "boloho",  
    "buduk",  
    "budug",  
    "celeng",  
    "cibai",  
    "cibay",  
    "cocot",  
    "cocote",  
    "cok",  
    "cokil",  
    "colai",  
    "colay",  
    "coli",  
    "colmek",  
    "conge",  
    "congean",  
    "congek",  
    "congor",  
    "cuk",  
    "cukima",  
    "cukimai",  
    "cukimay",  
    "dancok",  
    "entot",  
]
```

"entotan",
"ewe",
"ewean",
"gelo",
"genjik",
"germo",
"gigolo",
"goblo",
"goblog",
"goblok",
"hencet",
"henceut",
"heunceut",
"homo",
"idiot",
"itil",
"jancuk",
"jancok",
"jablay",
"jalang",
"jembut",
"jiancok",
"jilmek",
"jurig",
"kacung",
"kampang",
"kampret",
"kampungan",
"kehed",
"kenthalu",
"kentot",
"kentu",
"keparat",
"kimak",
"kintil",
"kirik",
"kunyuk",
"kurap",
"konti",
"kontol",
"kopet",
"koplok",
"lacur",
"lebok",
"lonte",
"maho",
"meiki",
"memek",
"monyet",
"ndas",
"ndasmu",
"ngehe",
"ngentot",
"nggateli",
"nyepong",
"ngewe",

"ngocok",
"pante",
"pantek",
"patek",
"pathek",
"peju",
"pejuh",
"pecun",
"pecundang",
"pelacur",
"pelakor",
"peler",
"pepek",
"puki",
"pukima",
"pukimae",
"pukimak",
"pukimay",
"sampah",
"sepong",
"sial",
"sialan",
"silit",
"sinting",
"sontoloyo",
"tai",
"taik",
"tempek",
"tempik",
"tete",
"tetek",
"tiembokne",
"titit",
"toket",
"tolol",
"ublag",
"udik",
"wingkeng"

]

Lampiran 6: Data Pengujian dalam format CSV

Timestamp,Nama atau Inisial,Device yang anda gunakan,Web Brower yang anda gunakan dan versinya. (Contoh: Brave Brower - v1.67.123),"Apakah proses registrasi menggunakan email, username, dan password berjalan dengan lancar tanpa adanya error?", "Apakah fitur registrasi menggunakan email, username, dan password mudah digunakan?", Apakah fitur registrasi menggunakan akun Facebook atau Google berjalan dengan baik?, Apakah Anda merasa nyaman menggunakan OAuth (Facebook atau Google) untuk registrasi?, Apakah proses login menggunakan username dan password cepat dan tanpa kendala?, Apakah fitur login menggunakan username dan password mudah diakses?, Apakah fitur login menggunakan akun Facebook atau Google berjalan dengan lancar?, Apakah fitur login menggunakan akun Facebook atau Google mudah diakses?, Apakah pertanyaan-pertanyaan pada halaman beranda muncul dengan normal?, Apakah tampilan pertanyaan di halaman beranda mudah dipahami dan menarik?, Apakah fitur membuat pertanyaan berjalan dengan baik?, Apakah fitur untuk memilih mata pelajaran berjalan dengan baik?, Apakah proses pembuatan pertanyaan baru mudah dipahami?, Apakah fitur mengedit pertanyaan berjalan dengan lancar?, Apakah fitur untuk mengedit pertanyaan mudah digunakan?, Apakah fitur hapus pertanyaan berjalan dengan lancar?, Apakah fitur untuk menghapus pertanyaan mudah digunakan?, Apakah menurut Anda fitur CAPTCHA efektif dalam mencegah spam?, Apakah CAPTCHA mudah diisi oleh pengguna?, Apakah fitur filtering kata berjalan dengan baik?, Apakah fitur filtering kata efektif dalam mencegah penggunaan kata-kata terlarang?, Apakah Halaman Detail Pertanyaan menampilkan detail pertanyaan dan daftar jawaban dengan baik?, Apakah tampilan halaman detail pertanyaan menarik dan mudah dipahami?, Apakah fitur like/favorit ini berjalan dengan baik?, Apakah fitur menandai pertanyaan sebagai favorit mudah digunakan?, Apakah fitur pembuatan jawaban ini berjalan dengan lancar tanpa kendala?, Apakah proses pembuatan jawaban untuk pertanyaan mudah dipahami?, Apakah fitur rating (bintang 1-5) untuk jawaban berjalan dengan baik?, Apakah fitur rating ini membantu Anda menilai kualitas jawaban?, Apakah fitur jawaban terbaik ini berjalan dengan baik?, Apakah fitur menandai jawaban sebagai "jawaban terbaik" mudah digunakan?, Apakah fitur mengedit jawaban ini berjalan dengan baik?, Apakah fitur menghapus jawaban ini berjalan dengan baik?, Apakah fitur untuk membagikan pertanyaan ke sosial media lain berjalan dengan baik?, Apakah fitur share ini membantu Anda dalam membagikan pertanyaan yang menarik?, Apakah fitur leaderboard berjalan dengan baik?, Apakah fitur leaderboard ini membantu Anda untuk melihat pengguna teraktif dan pertanyaan terpopuler?, Apakah hasil pencarian relevan dengan kata kunci yang Anda masukkan?, Apakah fitur pencarian membantu

Anda menemukan pertanyaan atau profil pengguna dengan mudah?,Apakah halaman mata pelajaran menampilkan daftar pertanyaan berdasarkan mata pelajaran tertentu dengan baik?,Apakah tampilan halaman mata pelajaran ini menarik dan mudah dipahami?,Apakah halaman profil menampilkan informasi pengguna dengan lengkap dan jelas?,Apakah tampilan halaman profil menarik dan mudah dipahami?,Apakah fitur notifikasi berjalan dengan baik dan memberikan informasi yang relevan?,Apakah fitur notifikasi membantu Anda untuk tetap update dengan kejadian yang terjadi?,Apakah fitur logout berjalan dengan baik?,Apakah fitur logout mudah untuk diakses?

16/07/2024 16:58:35, Wandi, Mobile, Chrome

126.0.6478.122,Ya,5,Tidak,4,Ya,5,Tidak,5,Ya,5,Ya,Ya,4,Ya,4,Ya,5,5,5,Ya,3,Ya,5,Ya,4,Ya,4,Ya,5,Ya,5,Ya,Ya,Ya,3,Ya,3,Ya,5,Ya,5,Ya,5,Ya,5,Ya,5

16/07/2024 17:20:45, Shirookami, Mobile, Chrome

16/07/2024 18:50:09,alfian,Laptop/PC,Google Chrome Version 126.0.6478.127
(Official Build)

(64-bit), Ya, 5, Tidak, 3, Ya, 5, Tidak, 5, Ya, 5, Ya, Ya, 5, Ya, 5, Ya

a,5,Ya,5,Ya,4,Ya,Ya,Ya,4,Ya,5,Ya,4,Ya,5,Ya,5,Ya,5
16/07/2024 20:23:11,Alpa,Mobile,Google Chrome
Y,4,Y,5,Y,4,Y,5,Y,5,Y,4,Y,4,3,3,Y,3,Y,3,Y,4,Y,4,Y,3,Y,3

, Ya, 4, Ya, 5, Ya, 4, Ya, 5, Ya, 5, Ya, Ya, 4, Ya, 4, Ya, 4, 3, 3, Ya, Ya, Ya, 4, Ya, 2, Ti, 1, 1, 2, Ya, 5, Ya, 4, Ti, 1, 1, 2, Ya, 5,

126.0.64/8.127, Ya,5,Tidak,5,Ya,5,Ya,5,Ya,4,Ya,Ya,5,Ya,5,Ya,5,5,5,Ya,

a,5,Ya,5,Ya,5,Ya,5,Ya,5,Ya,5,Ya,5,Ya,5,Ya,5,Ya,5,Ya,5,Ya,5
17/07/2024 9:32:49,MF,Mobile,Samsung browser latest version on july 17

15·30·50 A.

Va 5 Va 5 Va 5 Va 4 Va 5 Va Va 5 Va 5 Va 5 5 5 Va 5

Lampiran 7: Listing Program

1. Proses *register* di database

```
await verifyCaptchaToken(input.token);

let user = await ctx.db.query.users.findFirst({
  where: eq(users.email, input.email),
});
if (user) {
  throw new Error('Email telah digunakan');
}

user = await ctx.db.query.users.findFirst({
  where: eq(users.username, input.username),
});
if (user) {
  throw new Error('Username telah digunakan');
}

const hashedPwd = await argon2.hash(input.password);

await ctx.db.insert(users).values({
  email: input.email,
  id: `user-${cuid()}`,
  name: `${input.firstName.trim()} ${{
    input.lastName?.trim() ?? ''
  }.trim()},`,
  password: hashedPwd,
  image: getDiceBearAvatar(input.username),
  username: input.username,
});
```

2. Proses buat pertanyaan di database

```
if (badwords.flag(payload.schema.content) as boolean) {
  throw new Error('Pertanyaan anda mengandung kata terlarang!');
}

await verifyCaptchaToken(payload.token);

const membership = await this.membershipRepo.findValidMembership(
  payload.schema.userId,
);
const todayQuestionCount = await
this.questionRepo.getTodayQuestionCount(
```

```

    payload.schema.userId,
);

match(membership?.type)
  .with('standard', () => {
    if (todayQuestionCount >= 10) {
      throw new Error(
        'Anda telah melewati batas pembuatan pertanyaan hari ini',
      );
    }
  })
  .with('plus', () => undefined)
  .otherwise(() => {
    if (todayQuestionCount >= 2) {
      throw new Error(
        'Anda telah melewati batas pembuatan pertanyaan hari ini',
      );
    }
  });
}

const question = await this.questionRepo.findQuestionBySlug(
  payload.schema.slug,
);
if (question) {
  throw new Error('Pertanyaan yang sama telah ada!');
}

const createdQuestion = await this.questionRepo.createQuestion(
  payload.schema,
);

if (payload.image && payload.image?.length > 0 && createdQuestion) {
  const imagesInput = payload.image.map((img) => ({
    ...img,
    questionId: createdQuestion.id,
  }));
  await this.questionImageRepo.addQuestionImage(...imagesInput);
}

await typesenseClient.collections('questions').documents().create({
  id: createdQuestion?.id,
  content: payload.schema.content,
  subjectId: payload.schema.subjectId,
});

```

3. Proses buat jawaban di *database*

```
if (badwords.flag(input.schema.content) as boolean) {
    throw new Error('Jawaban anda mengandung kata terlarang!');
}

await verifyCaptchaToken(input.token);

const question = await ctx.db.query.questions.findFirst({
    where: eq(questions.id, input.schema.questionId),
});

const answer = await ctx.db
    .insert(answers)
    .values(input.schema)
    .returning();

if (input.schema.userId !== question?.userId && answer[0] && question)
{
    await ctx.db.insert(notifications).values({
        id: `notification-${cuid()}`,
        questionId: input.schema.questionId,
        description: answer[0].content.slice(0, 100),
        receiverUserId: question.userId,
        transmitterUserId: input.schema.userId,
        type: 'new-answer',
    });
}
```

4. Proses Pencarian

```
const searchResult = await QuestionRepoTypesense.searchQuestion(
    query,
    subjectId,
);
const hitIds = searchResult.map((s) => s.id);

let data = await this.questionRepo.findAllQuestionsById(
    cursor,
    limit,
    ...hitIds,
);

data = data.map((dt) => ({
    ...dt,
    content:
```

```
    searchResult.find((snip) => snip.id === dt.id)?.snippet ??  
dt.content,  
}));  
  
let nextCursor: typeof cursor | undefined;  
if (data.length > limit) {  
  const nextItem = data.pop();  
  nextCursor = nextItem?.id;  
}  
  
return { data, nextCursor };
```

Lampiran 8: Hasil Pengujian Black Box

No.	Item Uji	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Registrasi akun menggunakan <i>email</i> dan <i>password</i>	Mengisi <i>form</i> <i>nama</i> , <i>username</i> , <i>email</i> , dan <i>password</i> , lalu menekan tombol daftar	Proses registrasi berjalan dengan baik, dan akun <i>user</i> berhasil dibuat	Sesuai harapan	Valid
2	Registrasi akun menggunakan OAuth (Akun Facebook atau Google)	Menekan tombol daftar dengan akun Facebook atau Google	Proses registrasi berjalan dengan baik, dan akun <i>user</i> berhasil dibuat	Sesuai harapan	Valid
3	<i>Login</i> menggunakan <i>username</i> dan <i>password</i>	Mengisi <i>form</i> <i>username</i> dan <i>password</i> lalu menekan tombol <i>login</i>	<i>User</i> dapat masuk ke halaman beranda dengan berhasil	Sesuai harapan	Valid
4	<i>Login</i> menggunakan OAuth (Akun Facebook atau Google)	Menekan tombol <i>login</i> dengan akun Facebook atau Google	<i>User</i> dapat masuk ke halaman beranda dengan berhasil	Sesuai harapan	Valid
5	Halaman Beranda	<i>User</i> masuk ke halaman beranda	Halaman beranda tampil dengan benar	Sesuai harapan	Valid
6	Buat Pertanyaan	Menekan tombol “Buat Pertanyaan”,	Pertanyaan berhasil dibuat	Sesuai harapan	Valid

		mengisi <i>form</i> , lalu <i>submit</i>			
7	Mengedit Pertanyaan	Menekan tombol edit pertanyaan, mengisi <i>form</i> , lalu <i>submit</i>	Pertanyaan berhasil diedit	Sesuai harapan	Valid
8	Menghapus Pertanyaan	Menekan tombol hapus pertanyaan, lalu konfirmasi	Pertanyaan berhasil dihapus	Sesuai harapan	Valid
9	<i>Filtering</i> Kata Terlarang	Membuat pertanyaan atau jawaban yang mengandung kata terlarang/kasar	Muncul pesan <i>error</i> jika pertanyaan atau jawaban mengandung kata terlarang/kasar	Sesuai harapan	Valid
10	Halaman Detail Pertanyaan	Menekan salah satu pertanyaan yang ada di halaman beranda	Detail pertanyaan dan daftar jawaban ditampilkan dengan baik	Sesuai harapan	Valid
11	Fitur <i>Like/Favorit</i>	Menekan tombol “Favorit” di salah satu pertanyaan	Jumlah favorit dari pertanyaan tersebut bertambah	Sesuai harapan	Valid
12	Buat Jawaban	Menekan tombol “jawab” pada salah satu pertanyaan, mengisi <i>form</i> , lalu <i>submit</i>	Jawaban berhasil ditambahkan	Sesuai harapan	Valid
13	<i>Rating</i>	Menekan	<i>Rating</i> berhasil	Sesuai	Valid

	Jawaban	tombol <i>rating</i> dan memilih antara bintang 1- 5 pada jawaban	ditambah, dan rata-rata <i>rating</i> berhasil dimunculkan	harapan	
14	Fitur Jawaban Terbaik	Pemilik pertanyaan menekan tombol “jawaban terbaik” pada salah satu jawaban.	Jawaban yang dipilih berhasil diatur sebagai jawaban terbaik dan berada di paling atas jawaban lainnya	Sesuai harapan	Valid
15	Mengedit Jawaban	Menekan tombol edit jawaban, mengisi <i>form</i> , lalu <i>submit</i>	Jawaban berhasil diedit	Sesuai harapan	Valid
16	Menghapus Jawaban	Menekan tombol hapus jawaban, lalu konfirmasi	Jawaban berhasil dihapus	Sesuai harapan	Valid
17	Fitur Bagikan Pertanyaan	Menekan tombol Bagikan, lalu pilih sosial media tujuan	Pertanyaan berhasil dibagikan ke <i>platform</i> sosial media yang dipilih	Sesuai harapan	Valid
18	<i>Leaderboard</i>	Membuka semua halaman, <i>leaderboard</i> akan muncul di sidebar kanan	<i>Leaderboard</i> akan menampilkan pertanyaan terpopuler dan <i>user</i> teraktif musim ini	Sesuai harapan	Valid
19	Fitur Pencarian	Membuka Halaman	Pertanyaan dan <i>user</i> yang dicari	Sesuai harapan	Valid

		Pencarian, memilih mode pencarian, lalu memasukkan <i>keyword</i>	muncul dengan akurat		
20	Halaman Mata Pelajaran	Membuka halaman pertanyaan dan memilih salah satu mata pelajaran	Pertanyaan yang sesuai dengan mata pelajaran yang dipilih berhasil ditampilkan	Sesuai harapan	Valid
21	Halaman Profil	Menekan foto profil, <i>username</i> , atau nama salah satu <i>user</i>	Halaman profil seorang <i>user</i> berhasil ditampilkan	Sesuai harapan	Valid
22	Halaman Notifikasi	Menekan tombol “Notifikasi” pada <i>Navbar</i>	Menampilkan semua notifikasi milik <i>user</i> tersebut.	Sesuai harapan	Valid
23	Fitur <i>Logout</i>	Menekan tombol <i>Logout</i> yang ada di <i>Navbar</i>	<i>User</i> berhasil keluar dari aplikasi dan kembali ke halaman <i>login</i>	Sesuai harapan	Valid

RIWAYAT HIDUP

Taufik Hidayat lahir di Kabupaten Majalengka, Provinsi Jawa Barat pada tanggal 02 April 2002. Penulis lahir dari pasangan Maman Rohman dan Iis Ismatu Sholihat. Merupakan anak sulung dari dua bersaudara, yakni Muhammad Fauzi Fathirohman.

Ketika tahun 2007 penulis masuk SDN Manggahang 1. Kemudian lulus pada tahun 2014. Selanjutnya menempuh pendidikan di SMPN 1 Baleendah dan lulus pada tahun 2017. Kemudian, masuk ke SMKN 7 Baleendah Jurusan Rekayasa Perangkat Lunak. Lalu, lulus pada tahun 2020. Di tahun 2020 penulis juga menjadi mahasiswa di Jurusan Teknik Informatika Fakultas Teknologi Informasi Universitas Bale Bandung.

Saat semester 6, penulis mengikuti program Studi Independen Bersertifikat di PT. Hacktivate Teknologi Indonesia dengan mengambil kelas Golang Backend Developer. Penulis pun kembali mengikuti program Studi Independen Bersertifikat saat semester 7 di PT. Forest Interactive dengan mengambil kelas JavaScript Developer. Selama Studi Independen di Forest Interactive penulis berhasil mendapatkan 2 prestasi, yang pertama yaitu menjadi *Best Academic Achievement* dengan hadiah mengikuti kunjungan industri ke *Headquarter* (Kantor Pusat) Forest Interactive di Kuala Lumpur, Malaysia. Dan yang kedua yaitu *capstone project* milik kelompok penulis berhasil menjadi *Best Capstone Project* dengan hadiah sebesar Rp10.000.000.

Selain berkuliah, sejak bulan Desember 2023 penulis pun bekerja secara *full-time remote* sebagai Full Stack Developer di PT. Netovas Eterna Teknologi.

