

**LAPORAN
PROYEK AKHIR**

Platform Aplikasi Artificial Intelligence (AI) Berbasis Natural Language Processing (NLP) dan Game Edukasi untuk Menfasilitasi Pembelajaran Interaktif Siswa SMA Muhammadiyah Ambon



DISUSUN OLEH :

Nama : Chairil Ali

NIM : 200102121

Nama : Samrah Daeng Makase

NIM : 200102116

DOSEN PEMBIMBING :

M. Ikbal Siami, S.Kom., M.Kom

**PROGRAM STUDI SISTEM INFORMASI INSTITUT TEKNOLOGI DAN
BISNIS STIKOM AMBON 2024**

SURAT PERNYATAAN

DAFTAR ISI

Halaman

HALAMAN JUDUL

SURAT PERNYATAAN.....	i
DAFTAR ISI.....	ii
DAFTAR TABEL.....	v
DAFTAR GAMBAR	vi
PENDAHULUAN	1
1.1 LATAR BELAKANG PROYEK	1
1.2 TUJUAN DAN MANFAAT.....	3
1.3 RUANG LINGKUP PROYEK.....	4
1.3.1 BATASAN PROYEK.....	4
1.3.2 KETERLIBATAN TIM	8
METODOLOGI PENGEMBANGAN	14
2.1 PENDEKATAN YANG DIGUNAKAN.....	14
2.2 TAHAPAN PENGEMBANGAN	14
ANALISIS DAN PERANCANGAN SISTEM	17
3.1 ANALISIS KEBUTUHAN	17
3.1.1 KEBUTUHAN FUNGSIONAL	17
3.1.2 KEBUTUHAN NON-FUNGSIONAL	18
3.1.3 KEBUTUHAN SISTEM.....	18
3.2 DESAIN SISTEM.....	19
3.2.1 ARSITEKTUR SISTEM.....	19
3.2.2 ALUR KERJA NLP	20
3.2.3 ALUR FITUR CHAT WITH TEXT	21
3.2.4 ALUR FITUR CHAT WITH IMAGE	22
3.2.5 ALUR FITUR CHAT WITH PDF.....	23
3.2.2 MIND MAP	23
3.2.3 USE CASE DIAGRAM	24

3.2.4 ACTIVITY DIAGRAM.....	26
3.2.5 ENTITY RELATIONSHIP DIAGRAM	32
3.2.6 DESAIN ANTARMUKA PENGGUNA (UI)	33
IMPLEMENTASI.....	45
4.1 TEKNOLOGI DAN ALAT YANG DIGUNAKAN	45
4.2 PROSES PENGEMBANGAN	46
4.3 ALUR PROSES	47
4.4 TANGKAPAN LAYAR PROYEK	55
Chat with Text.....	55
Chat with Image	70
Chat with PDF.....	88
Generate Quiz.....	95
Notes	114
About Dev	132
Education Games	149
Connect with Us.....	151
Exit	154
Navigation.....	156
PENGUJIAN.....	162
5.1 METODE PENGUJIAN PERANGKAT LUNAK.....	162
5.2 HASIL PENGUJIAN.....	162
DEPLOYMENT.....	168
6.1 TAHAPAN DEPLOYMENT	168
EVALUASI DAN MONITORING	170
7.1 METODE EVALUASI.....	170
7.2 MONITORING DAN MAINTENANCE.....	170
7.3 KUESIONER.....	170
HASIL DAN KESIMPULAN	172
8.1 HASIL.....	172
8.2 KESIMPULAN.....	173
8.3 SARAN	173
LAMPIRAN	175

11.1 LINK TERKAIT	175
Repository Github	175
Download ByteBrain AI	175
11. 3 DOKUMENTASI	175

DAFTAR TABEL

	Halaman
Tabel 1. 1 Pengujian BlackBox Testing.....	162
Tabel 1. 2 kuesioner SMA Muhammadiyah Ambon	170

DAFTAR GAMBAR

	Halaman
Gambar 2. 1 Waterfall SDLC.....	14
Gambar 3. 1 Desain Arsitektur Sistem.....	19
Gambar 3. 2 Alur kerja NLP	20
Gambar 3. 3 Pre-processing pada NLP	21
Gambar 3. 4 Alur chat with text.....	21
Gambar 3. 5 Metode CNN	22
Gambar 3. 6 Alur chat with PDF	23
Gambar 3. 7 Mind Map	24
Gambar 3. 8 Simbol use case diagram	25
Gambar 3. 9 Use Case Diagram	25
Gambar 3. 10 Simbol activity diagram	26
Gambar 3. 11 Activity Diagram Chat with Text.....	27
Gambar 3. 12 Activity Diagram Chat with Image	27
Gambar 3. 13 Activity Diagram Chat with PDF.....	28
Gambar 3. 14 Activity Diagram Generate Quiz.....	29
Gambar 3. 15 Activity Diagram Notes	29
Gambar 3. 16 Activity Diagram About Dev	30
Gambar 3. 17 Activity Diagram Education Games	30
Gambar 3. 18 Activity Diagram Connect with Us	31
Gambar 3. 19 Activity Diagram Exit	31
Gambar 3. 20 simbol entity relationship diagram	32
Gambar 3. 21 ERD Notes	32
Gambar 3. 22 Navigasi chat with text.....	33
Gambar 3. 23 Navigasi notes	34
Gambar 3. 24 Navigasi about dev	34
Gambar 3. 25 chat with text	35

Gambar 3. 26 chat with image	35
Gambar 3. 27 chat with PDF: membuat kuis	36
Gambar 3. 28 chat with PDF : mencari jawaban	36
Gambar 3. 29 form pengisian kuis	37
Gambar 3. 30 loading tampilan kuis	38
Gambar 3. 31 tampilan soal kuis.....	38
Gambar 3. 32 hasil kuis.....	39
Gambar 3. 33 list catatan.....	40
Gambar 3. 34 form pengisian catatan	40
Gambar 3. 35 berhasil tambah catatan	41
Gambar 3. 36 about dev	42
Gambar 3. 37 education games	42
Gambar 3. 38 connect with us.....	43
Gambar 3. 39 konfirmasi keluar.....	43
Gambar 3. 40 foto profile.....	44
 Gambar 4. 1 API Key token.....	48
Gambar 4. 2 Penyetelan model dan request.....	48
Gambar 4. 3 Trained data AI.....	49
Gambar 4. 4 Memuat file env	49
Gambar 4. 5 Setting dan library yang dipakai.....	50
Gambar 4. 6 Mengunduh dependency package	51
Gambar 4. 7 Menghubungkan flutter dengan emulator	51
Gambar 4. 8 Menjalankan aplikasi	52
Gambar 4. 9 Hasil run aplikasi.....	52
Gambar 4. 10 request dan respons API pada terminal	53
Gambar 4. 11 request dan respons API DevTools Flutter	53
Gambar 4. 12 request dan respons Network pada DevTools Flutter	54
Gambar 4. 13 chat with text	55
Gambar 4. 14 chat with image	70
Gambar 4. 15 chat with PDF untuk mencari informasi dari PDF	88
Gambar 4. 16 chat with PDF untuk generate quiz	88

Gambar 4. 17 Form Generate Quiz	95
Gambar 4. 18 Soal pilihan ganda	95
Gambar 4. 19 Hasil kuis.....	96
Gambar 4. 20 tambah catatan.....	114
Gambar 4. 21 lihat catatan	114
Gambar 4. 22 about dev	132
Gambar 4. 23 education games	149
Gambar 4. 24 connect with us.....	151
Gambar 4. 25 exit.....	154
Gambar 4. 26 navigation	156

PENDAHULUAN

1.1 LATAR BELAKANG PROYEK

Kemajuan teknologi informasi dan komunikasi saat ini telah membawa perubahan signifikan dalam berbagai aspek kehidupan, termasuk dalam dunia pendidikan. Penggunaan teknologi dalam proses pembelajaran tidak hanya mempermudah penyampaian materi, tetapi juga meningkatkan interaktivitas antara pengajar dan siswa. Salah satu inovasi yang muncul dari perkembangan teknologi adalah pemanfaatan *Artificial Intelligence* (AI) yang mampu memberikan pengalaman belajar yang lebih adaptif dan personal.

Dalam konteks pendidikan di SMA Muhammadiyah Ambon, tantangan yang dihadapi oleh siswa adalah rendahnya minat belajar serta kurangnya variasi metode pembelajaran yang interaktif. Sistem pembelajaran konvensional yang dominan masih bersifat satu arah, di mana siswa cenderung menjadi penerima pasif dari informasi yang disampaikan oleh guru. Hal ini mengakibatkan kurangnya keterlibatan siswa dalam proses belajar, yang pada akhirnya berdampak pada pencapaian akademis mereka. Menjawab tantangan tersebut, penerapan *Artificial Intelligence* (AI) berbasis *Natural Language Processing* (NLP) dan game edukasi dapat menjadi solusi yang efektif. Teknologi AI memungkinkan terciptanya platform pembelajaran yang mampu berinteraksi secara alami dengan siswa, memberikan umpan balik secara real-time, serta menyesuaikan materi sesuai dengan kebutuhan individu. Sementara itu, game edukasi menawarkan pendekatan belajar yang menyenangkan dan menantang, yang dapat meningkatkan motivasi dan keterlibatan siswa.

Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun platform aplikasi AI berbasis NLP dan game edukasi yang dapat memfasilitasi pembelajaran interaktif di SMA Muhammadiyah Ambon. Diharapkan, aplikasi ini tidak hanya meningkatkan hasil belajar siswa, tetapi juga menciptakan lingkungan belajar yang lebih dinamis dan partisipatif.

A. Alasan Pengembangan Sistem

Dalam era digital yang semakin maju, integrasi teknologi dalam pendidikan menjadi sangat penting untuk meningkatkan kualitas pembelajaran. SMA Muhammadiyah Ambon berkomitmen untuk memanfaatkan teknologi terbaru guna memperkaya pengalaman belajar siswa. Salah satu inisiatif yang diambil adalah pengembangan *platform* aplikasi *Artificial Intelligence* (AI) berbasis *Natural Language Processing* (NLP) dan game edukasi. Alasan utama pengembangan sistem ini antara lain:

- 1) Teknologi AI dan NLP dapat memberikan pengalaman belajar yang lebih personal dan adaptif. Siswa dapat memperoleh bantuan belajar yang disesuaikan dengan kebutuhan mereka, sehingga dapat meningkatkan pemahaman dan prestasi akademik.
- 2) Penggunaan *game* edukasi menyediakan cara yang menyenangkan dan interaktif untuk membantu meningkatkan minat dan motivasi belajar siswa.
- 3) Dengan adanya *platform* aplikasi AI ini, siswa dapat mengakses materi pembelajaran kapan saja dan di mana saja selama terhubung dengan internet, yang belum tentu ada di alat pembelajaran konvensional seperti buku pelajaran. Ini sangat membantu dalam memberikan fleksibilitas bagi siswa yang memiliki jadwal belajar yang padat atau membutuhkan waktu belajar tambahan di luar jam sekolah.
- 4) Memanfaatkan teknologi AI dan NLP yang merupakan teknologi terkini dalam bidang teknologi informasi dan komputer. Hal ini sejalan dengan visi SMA Muhammadiyah Ambon untuk selalu berada di garis depan dalam penggunaan teknologi pendidikan.

B. Masalah yang Dihadapi Sebelum Sistem Dikembangkan

Sebelum dikembangkannya *platform* aplikasi AI dan *game* edukasi ini, SMA Muhammadiyah Ambon menghadapi beberapa masalah yang menghambat proses pembelajaran, antara lain:

- 1) Tidak semua siswa dapat memperoleh bimbingan yang cukup dari guru karena keterbatasan waktu dan jumlah tenaga pengajar. Hal ini menyebabkan beberapa siswa tertinggal dalam pemahaman materi pelajaran.
- 2) Metode pengajaran konvensional sering kali kurang menarik bagi siswa, sehingga mereka kurang termotivasi untuk belajar. Hal ini berdampak pada rendahnya tingkat partisipasi dan perhatian siswa selama proses belajar mengajar.
- 3) Tak jarang siswa kesulitan ketika *googling* materi atau jawaban dikarenakan keterbatasan artikel yang ada di Google. Keterbatasan ini mengurangi kesempatan siswa untuk belajar secara mandiri dan memperdalam pemahaman mereka terhadap materi pelajaran.
- 4) Sebelum adanya inisiatif ini, penggunaan teknologi dalam proses pembelajaran di SMA Muhammadiyah Ambon masih terbatas. Hal ini mengakibatkan kurangnya variasi metode pembelajaran yang dapat digunakan untuk meningkatkan pemahaman siswa.

Dengan pengembangan *platform* aplikasi AI berbasis NLP dan *game* edukasi ini, diharapkan dapat mengatasi berbagai masalah tersebut dan menciptakan lingkungan belajar yang lebih efektif, interaktif, dan menyenangkan bagi siswa.

1.2 TUJUAN DAN MANFAAT

Pengembangan *platform* aplikasi *Artificial Intelligence* (AI) berbasis *Natural Language Processing* (NLP) dan *game* edukasi untuk SMA Muhammadiyah Ambon memiliki beberapa tujuan utama:

- 1) Memanfaatkan teknologi AI dan NLP untuk memberikan materi pelajaran yang adaptif dan personal sesuai dengan kebutuhan individu siswa.
- 2) Menggunakan *game* edukasi sebagai media interaktif dan menyenangkan untuk meningkatkan minat dan motivasi siswa dalam belajar.
- 3) Menyediakan akses yang mudah ke topik materi pelajaran kapan saja dan di mana saja selama terhubung dengan internet, sehingga siswa dapat belajar secara fleksibel di luar jam sekolah.
- 4) Mengadopsi teknologi AI dan NLP untuk memperbarui metode pengajaran dan menyiapkan siswa menghadapi tantangan di era digital.
- 5) Membantu guru dalam memberikan bimbingan yang lebih efektif dan efisien, dengan dukungan teknologi yang dapat mengotomatiskan sebagian dari proses pengajaran.

Pengembangan dan implementasi *platform* aplikasi AI berbasis NLP dan *game* edukasi ini diharapkan dapat memberikan berbagai manfaat bagi SMA Muhammadiyah Ambon, antara lain:

- 1) Siswa dapat menerima materi pelajaran yang disesuaikan dengan kemampuan dan kecepatan belajar masing-masing, sehingga proses pembelajaran menjadi lebih efektif.
- 2) Dengan adanya *game* edukasi, siswa akan lebih tertarik dan termotivasi untuk belajar, yang pada akhirnya dapat meningkatkan partisipasi dan perhatian mereka dalam proses belajar mengajar.
- 3) Siswa dapat mengakses topik materi pelajaran secara *online* kapan saja dan di mana saja, memberikan fleksibilitas dalam belajar dan kesempatan untuk memperdalam pemahaman mereka.
- 4) Memperkenalkan siswa dan guru pada teknologi terkini, meningkatkan literasi digital, dan mempersiapkan mereka untuk dunia yang semakin digital.
- 5) Guru dapat memanfaatkan *platform* ini untuk mengotomatisasi beberapa tugas pengajaran dan memberikan bimbingan yang lebih fokus dan efektif kepada siswa yang membutuhkan.
- 6) Siswa akan terbiasa dengan penggunaan teknologi dalam belajar, yang dapat meningkatkan keterampilan mereka dalam menggunakan alat-alat digital dan teknologi informasi.

1.3 RUANG LINGKUP PROYEK

1.3.1 BATASAN PROYEK

A. Cakupan dan Fungsionalitas Sistem

Cakupan fungsionalitas sistem aplikasi *Artificial Intelligence* (AI) berbasis *Natural Language Processing* (NLP) dan *game* edukasi untuk SMA Muhammadiyah Ambon mencakup berbagai fitur dan layanan yang dirancang untuk meningkatkan pengalaman belajar siswa. Cakupan fungsionalitas utama dari sistem ini meliputi:

1) *Chat with Text*

- Siswa dapat berinteraksi dengan AI melalui pesan teks dan suara.
- Aplikasi AI akan merespons pertanyaan dan permintaan siswa.
- Fitur ini mendukung tanya jawab, diskusi, dan pemberian penjelasan materi.

2) *Chat with Image*

- Siswa dapat mengunggah gambar untuk mendapatkan penjelasan atau jawaban dari AI.
- AI akan mengenali konten gambar dan memberikan tanggapan yang relevan.
- Fitur ini berguna untuk soal-soal yang memerlukan penjelasan visual atau materi yang disajikan dalam bentuk gambar.

3) *Chat with PDF*

- Guru atau Siswa dapat mengunggah file PDF yang berisi materi pelajaran.
- Guru atau Siswa juga dapat membuat kuis dari PDF yang diunggah dengan dukungan bahasa, jumlah soal, format pilihan ganda, serta jawaban yang benar.
- AI akan memproses isi PDF dan memberikan jawaban atau penjelasan dari pertanyaan user berdasarkan PDF yang diunggah.
- Fitur ini memfasilitasi dukungan dokumen PDF dan juga multi PDF, serta memperoleh informasi dari dokumen tersebut.

4) *Generate Quiz*

- AI dapat secara otomatis membuat kuis berdasarkan inputan user melalui *form* yang tersedia.

- Siswa dapat mengikuti kuis untuk menguji pemahaman mereka terhadap materi yang telah dipelajari ataupun topik lain yang diminta.
- Fitur ini menyediakan topik kuis, pilihan level kesulitan, jumlah pertanyaan, pilihan ganda, jawaban yang benar, dan nilai dari kuis.

5) *Notes*

- Siswa dapat menyimpan catatan pada aplikasi dan mengaksesnya kembali kapan saja.
- Fitur notifikasi akan mengingatkan siswa tentang catatan penting atau jadwal belajar yang telah mereka buat.
- Fitur ini membantu siswa dalam mengorganisasi catatan dan manajemen waktu belajar.

6) *About Developer*

- Fitur ini untuk menampilkan informasi tentang pengembang aplikasi.

7) *Education Games*

- Aplikasi menyediakan berbagai *game* edukasi yang dirancang untuk meningkatkan minat belajar siswa melalui cara yang menyenangkan dan interaktif.
- *Game* edukasi mencakup topik seperti matematika, puzzle, dan rubik.

8) *Connect with Us*

- Fitur ini untuk menampilkan detail kontak terkait pihak sekolah dan pengembang aplikasi.

9) *Exit*

- Fitur ini untuk keluar dari aplikasi.

10) *Navigation Menu*

- Memungkinkan siswa untuk dengan mudah bermigrasi ke berbagai fitur utama aplikasi melalui menu yang terstruktur.

B. Modul atau Fitur Utama yang Dikembangkan

Modul atau fitur utama yang dikembangkan dalam aplikasi *Artificial Intelligence* (AI) berbasis *Natural Language Processing* (NLP) dan *game* edukasi untuk SMA Muhammadiyah Ambon:

1. Chat with Text

Fitur ini memungkinkan siswa untuk berinteraksi dengan AI menggunakan pesan teks. Siswa dapat mengajukan pertanyaan, meminta penjelasan, dan berpartisipasi dalam diskusi melalui chat berbasis teks.

Fungsionalitas Utama:

- Respon otomatis dari AI berdasarkan pertanyaan atau permintaan siswa.
- Penerimaan dan pengiriman pesan teks antara siswa dan Aplikasi AI seperti sedang *chatting*.
- Penyediaan *form* inputan atau *prompt* teks.
- Opsi untuk inputan menggunakan suara.
- Opsi untuk menyalin respon jawaban dari AI.

2. Chat with Image

Fitur ini memungkinkan siswa untuk mengunggah dan mengirim gambar sebagai bagian dari interaksi mereka dengan AI. Misalnya, siswa dapat mengirim gambar soal atau materi pelajaran untuk mendapatkan penjelasan lebih lanjut.

Fungsionalitas Utama:

- Analisis gambar untuk memberikan tanggapan yang relevan dari AI.
- Penyediaan penjelasan visual atau jawaban berdasarkan konten gambar.
- Penyediaan inputan gambar untuk siswa.
- Penyediaan *form* inputan atau *prompt* terkait gambar yang diunggah.
- Opsi untuk menyalin respon jawaban dari AI.

3. Chat with PDF

Fitur ini memungkinkan guru atau siswa untuk mengunggah file PDF yang berisi materi pelajaran, ataupun yang lain. AI akan memproses dan mencari jawaban dari pertanyaan dan memberikan tanggapan berdasarkan konten PDF tersebut. AI juga dapat membuat kuis pilihan ganda dari PDF tersebut.

Fungsionalitas Utama:

- Penyediaan jawaban berdasarkan informasi dalam PDF.
- Penyediaan kuis pilihan ganda berdasarkan konteks PDF yang diberikan.

- Ekstraksi teks dan analisis konten dari PDF untuk memberikan tanggapan.
- Penyediaan *form* inputan atau *prompt* teks.
- Penyediaan *form* pilihan bahasa dalam kuis seperti bahasa Inggris dan Indonesia.
- Penyediaan *form* untuk menentukan jumlah soal kuis.
- Penyediaan kuis dalam format pilihan ganda serta jawaban yang benar.
- Opsi untuk menyalin kuis.

4. *Generate Quiz*

Fitur ini memungkinkan platform untuk secara otomatis membuat kuis berdasarkan topik yang diinput siswa. Siswa dapat menggunakan kuis ini untuk menguji pemahaman mereka terhadap materi yang telah dipelajari.

Fungsionalitas Utama:

- Pembuatan kuis otomatis dengan format pilihan ganda.
- Penyediaan *form* inputan atau *prompt* teks untuk topik kuis.
- Opsi untuk memilih level kesulitan kuis seperti *beginner*, *intermediate*, atau *advanced*.
- Opsi untuk menginput jumlah kuis yang ingin dibuat.
- Opsi untuk memilih dukungan bahasa dalam kuis seperti bahasa Indonesia, Arab, China, Inggris, Jerman, Korea, Malaysia, Spanyol, dan Perancis.

5. Notes

Fitur ini memungkinkan siswa untuk menyimpan catatan pribadi dalam aplikasi. Selain itu, fitur notifikasi akan mengingatkan siswa tentang catatan penting atau jadwal belajar yang telah mereka buat.

Fungsionalitas Utama:

- Pembuatan dan penyimpanan catatan oleh siswa.
- Pengaturan notifikasi untuk mengingatkan siswa tentang catatan atau jadwal pelajaran.
- Penyedian *form* inputan untuk menambah catatan, form tersebut terdiri dari catatan dalam format teks, opsi untuk memilih hari dan waktu, serta opsi untuk menyetel *alarm* dari catatan tersebut.
- Opsi untuk mengedit catatan.

- Opsi untuk menghapus catatan.
- Opsi untuk mencari catatan yang telah dibuat.
- Catatan yang telah disimpan dapat diakses kembali.

6. Education Games

Fitur ini menyediakan berbagai game edukasi yang dirancang untuk meningkatkan pembelajaran siswa melalui cara yang menyenangkan dan interaktif.

Fungsionalitas Utama:

- Penyediaan game edukasi dengan berbagai topik seperti matematika, *puzzle*, dan rubik berbasis *website*.

1.3.2 KETERLIBATAN TIM

A. Peran dan Tanggung Jawab Anggota Tim

1. Chairil Ali

Tanggung Jawab: Mengelola dan mengembangkan *Navigation Menu*, fitur *Chat with Teks*, *Chat with PDF*, *Notes*, *Education Games*, *Connect with Us*, dan *Exit* serta melakukan Deployment atau peluncuran aplikasi.

Detail Tugas:

- *Navigation Menu*
 - Menyediakan *icon* dan label dari masing-masing fitur yang dapat dimuat dengan baik.
 - Memastikan agar tiap melakukan navigasi ke fitur yang lain dilakukan tanpa kendala dan memiliki *user experience* yang baik.
- *Chat with Text*
 - Memastikan AI dapat merespon pesan dari user.
 - Menggunakan integrasi *Aplication Programming Interface* (API) Gemini AI untuk merespons pertanyaan user.
 - Menyediakan tampilan splash screen animasi sebelum memasuki fitur saat pertama kali.
 - Memberikan pesan *alert* atau peringatan jika AI tidak dapat merespon pesan dari user.

- Menyediakan opsi agar user dapat menyalin respon jawaban dari AI.
- Menyediakan opsi untuk mendengar respon jawaban AI menggunakan suara.
- Menyediakan *form* inputan teks untuk user.
- Memastikan AI dapat memperkenalkan dirinya sebagai AI asisten pribadi SMA Muhammadiyah Ambon.
- *Chat with PDF*
 - Memastikan AI dapat mengenali konten dalam PDF dan merespon pertanyaan user dan membuat kuis dari PDF tersebut.
 - Menggunakan integrasi *Application Programming Interface* (API) Gemini AI untuk menganalisis PDF dan merespons pertanyaan user.
 - Menyediakan instruksi penggunaan fitur.
 - Menyediakan pesan *alert* atau peringatan agar user mengunggah file PDF terlebih dahulu sebelum berinteraksi.
 - Menyediakan *form* inputan file PDF.
 - Menyediakan fitur ekstraksi teks untuk memberikan penjelasan yang relevan.
 - Menyediakan *form* inputan teks untuk user.
 - Memberikan pesan *alert* atau peringatan jika AI tidak dapat merespon pertanyaan dari user.
 - Menyediakan *form* untuk memilih bahasa dalam kuis.
 - Menyediakan *form* inputan untuk jumlah soal yang akan dibuat.
 - Menyediakan opsi agar user dapat menyalin respon AI untuk *generate quiz*.
 - Menampilkan konten kuis dalam format pilihan ganda dan menampilkan jawaban yang benar.
- *Notes*
 - Mengembangkan fitur untuk membuat, menyimpan, dan mengorganisasi catatan oleh user.

- Memastikan agar catatan yang telah dibuat dapat diakses kembali kapan saja dan notifikasi catatan ditampilkan dengan baik.
- Menyediakan informasi jika user belum punya catatan.
- Menyediakan *form* untuk menulis catatan user.
- Menyediakan *form* untuk hari dan waktu
- Menyediakan opsi untuk menyetel *alarm* catatan.
- *Education Games*
 - Mengembangkan fitur untuk siswa dapat bermain *game* edukasi.
 - *Game* yang dimainkan berbasis website.
 - Memastikan agar tiap game dapat dimuat dengan baik tanpa kendala.
- *Connect with Us*
 - Menyediakan *icon* sosmed untuk detail kontak dari sekolah dan pengembang aplikasi.
 - Memastikan agar tiap kontak dapat dimuat dengan baik tanpa kendala.
- *Exit*
 - Menyediakan pilihan agar tetap didalam aplikasi atau keluar dari aplikasi.
 - Memastikan agar tiap pilihan dapat muat dengan baik tanpa kendala.

2. Samrah Daeng Makase

Tanggung Jawab: Mengelola dan mengembangkan fitur *Navigation Menu, Chat with Image, Generate Quiz, About Developer, Education Games, Connect with Us*, dan *Exit*.

Detail Tugas:

- *Navigation Menu*
 - Menyediakan *icon* dan label dari masing-masing fitur yang menrepresentasikan fitur tersebut.

- Memastikan agar tiap melakukan navigasi ke fitur yang lain dilakukan tanpa kendala dan memiliki *user experience* yang baik.
- *Chat with Image*
 - Memastikan user dapat mengunggah gambar sebagai bagian dari interaksi dan mendapatkan respon jawaban dari AI.
 - Menggunakan integrasi *Aplication Programming Interface* (API) Gemini AI untuk menganalisis gambar dan merespons pertanyaan user.
 - Menyediakan pesan *alert* atau peringatan agar user mengunggah gambar terlebih dahulu sebelum berinteraksi.
 - Memberikan pesan *alert* atau peringatan jika AI tidak dapat merespon pesan dari user.
 - Menyediakan *form* inputan gambar.
 - Menyediakan *form* inputan teks.
 - Menyediakan opsi agar user dapat menyalin respon jawaban dari AI.
- Generate Quiz:
 - Mengembangkan kemampuan AI agar otomatis membuat kuis berdasarkan *form* yang tersedia.
 - Menyediakan *form* inputan untuk topik kuis.
 - Menyediakan *form* inputan untuk pilihan level kesulitan kuis.
 - Menyediakan *form* inputan untuk jumlah soal yang akan dibuat.
 - Menyediakan *form* inputan untuk pilihan bahasa dalam kuis.
 - Menampilkan konten kuis dalam format pilihan ganda dan menampilkan jawaban yang benar.
 - Menampilkan nilai hasil kuis.
 - Menyediakan fitur ulangi kuis.
- About Developer:
 - Mengembangkan halaman yang memberikan informasi tentang para pengembang aplikasi.
 - Menyediakan *icon* untuk navigasi ke tutorial penggunaan aplikasi.

- *Education Games*
 - Mengembangkan fitur untuk siswa dapat bermain *game* edukasi.
 - *Game* yang dimainkan berbasis website.
 - Memastikan agar tiap game dapat dimuat dengan baik tanpa kendala.
- *Connect with Us*
 - Menyediakan *icon* sosmed untuk detail kontak dari sekolah dan pengembang aplikasi.
 - Memastikan agar tiap kontak dapat dimuat dengan baik tanpa kedala.
- *Exit*
 - Menyediakan pilihan agar tetap didalam aplikasi atau keluar dari aplikasi.
 - Memastikan agar tiap pilihan dapat muat dengan baik tanpa kedala.

B. Kolaborasi dengan pihak-pihak terkait (*Stakeholder*)

1. Integrasi dengan *Application Programming Interface* (API) Gemini AI
 - Kolaborasi: Chairil Ali dan Samrah Daeng Makase bekerja sama dalam menggunakan API Gemini AI untuk mengembangkan fitur-fitur yang ada dalam aplikasi. Integrasi API ini memungkinkan aplikasi untuk memberikan jawaban dan penjelasan berdasarkan interaksi pengguna, baik itu teks, gambar, maupun PDF. Model Gemini yang di pakai dalam integrasi API ini adalah *gemini-1.5-flash*.
 - Manfaat: Penggunaan *Application Programming Interface* (API) Gemini AI memastikan bahwa aplikasi dapat memberikan respons yang cepat dan akurat, serta mendukung berbagai bentuk interaksi yang dibutuhkan oleh siswa.
2. Kolaborasi dengan Pihak Sekolah
 - Kolaborasi: Tim pengembang bekerja sama dengan pihak sekolah untuk memastikan aplikasi memenuhi kebutuhan kurikulum dan mendukung tujuan pendidikan.

- Manfaat: Masukan dari guru dan staf sekolah membantu dalam pengembangan fitur yang relevan dan bermanfaat bagi proses pembelajaran. Sekolah juga berperan dalam memberikan umpan balik tentang penggunaan aplikasi di lingkungan pendidikan.

3. Kolaborasi dengan Siswa

- Kolaborasi: Siswa dilibatkan dalam pengujian dan memberikan umpan balik selama pengembangan aplikasi.
- Manfaat: Masukan langsung dari pengguna akhir membantu dalam menyempurnakan fitur dan memastikan aplikasi *user-friendly*. Siswa juga dapat memberikan ide-ide baru yang dapat meningkatkan efektivitas pembelajaran melalui aplikasi.

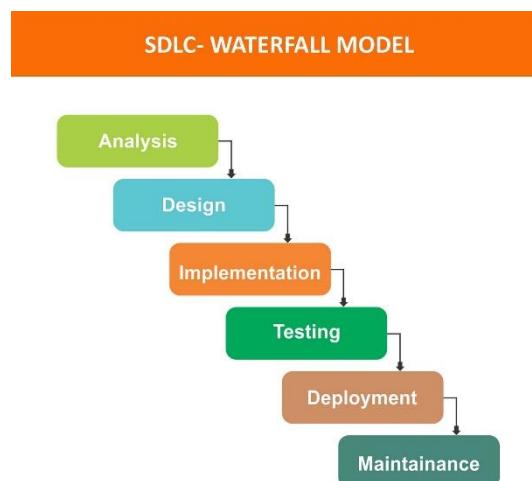
METODOLOGI PENGEMBANGAN

2.1 PENDEKATAN YANG DIGUNAKAN

A. Model Pengembangan *Waterfall*

Dalam pengembangan aplikasi *Artificial Intelligence* (AI) berbasis *Natural Language Processing* (NLP) dan *game* edukasi, peneliti menggunakan model pengembangan *Waterfall*. Metode *Waterfall* dipilih karena strukturnya yang linear dan sistematis, memungkinkan setiap tahap pengembangan diselesaikan secara berurutan sebelum tahap berikutnya dimulai.

2.2 TAHAPAN PENGEMBANGAN



Gambar 2. 1 *Waterfall SDLC*

1. *Analysis* (Analisis)

Pada tahap ini, peneliti melakukan analisis mendalam terhadap kebutuhan pengguna dan sistem. Kami berkolaborasi dengan pihak sekolah dan siswa SMA Muhammadiyah Ambon untuk memahami apa saja yang mereka butuhkan dari aplikasi ini.

Aktifitas Utama:

- 1) Mengumpulkan kebutuhan fungsional dan non-fungsional serta menyiapkan lingkungan pengembangan yaitu kebutuhan sistem berupa *hardware* dan *software*.
- 2) Melakukan *review* dan validasi kebutuhan dengan *stakeholder*.

2. *Design* (Perancangan)

Setelah kebutuhan ditentukan dan disetujui, peneliti masuk ke tahap desain. Pada tahap ini, peneliti membuat desain sistem yang akan menjadi dasar untuk implementasi.

Aktifitas Utama:

- 1) Merancang arsitektur sistem, *mind map*, *use case diagram*, *activity diagram*, dan *Entity Relationship Diagram* (ERD).
- 2) Merancang antarmuka pengguna (UI/UX) untuk memastikan aplikasi mudah digunakan dan dipahami.

3. *Implementation* (Implementasi)

Pada tahap implementasi, peneliti mulai mengembangkan kode sumber berdasarkan desain yang telah dibuat. Setiap modul atau fitur utama dikembangkan sesuai dengan rencana dan pembangian tugas masing-masing.

Aktifitas Utama:

- 1) Pengkodean oleh Chairil Ali untuk fitur *Chat with Text* dan *Notes* menggunakan *framework* Flutter, pengkodean fitur *Chat with PDF* menggunakan bahasa pemrograman Python.
- 2) Pengkodean oleh Samrah Daeng Makase untuk fitur *Chat with Image*, *Generate Quiz* dan *About Developer* menggunakan *framework* Flutter.
- 3) Integrasi dengan *Application Programming Interface* (API) Gemini AI untuk memastikan platform memberikan respons yang akurat dan relevan.

4. *Testing* (Pengujian)

Setelah implementasi selesai, aplikasi diuji untuk memastikan semua fitur berfungsi dengan baik dan sesuai dengan kebutuhan yang telah ditentukan.

Aktifitas Utama:

- 1) Melakukan pengujian unit untuk setiap modul yang dikembangkan.
- 2) Melakukan pengujian integrasi untuk memastikan semua modul bekerja sama dengan baik.
- 3) Melakukan pengujian sistem dan pengujian penerimaan pengguna untuk memastikan aplikasi memenuhi harapan pengguna.

5. Deployment (Penyebaran)

Setelah pengujian selesai dan aplikasi dianggap siap, peneliti melakukan *deployment*. Aplikasi diunggah ke website landing page untuk promosi dan pelatihan pengguna.

Aktifitas Utama:

- 1) Menyiapkan rilis aplikasi pada lingkungan pengembangan.
- 2) Pembuatan website landing page serta integrasi dengan Flowise AI.
- 3) Persiapan lingkungan hosting.
- 4) Deploy ke server hosting.
- 5) Testing post-deployment.
- 6) Promosi dan maintenance.

6. Maintenance (Pemeliharaan)

Tahap *maintenance* dilakukan setelah aplikasi digunakan secara aktif. Kami terus memantau aplikasi untuk memastikan berfungsi dengan baik dan melakukan perbaikan jika ditemukan masalah.

Aktifitas Utama:

- 1) Menyediakan dukungan teknis untuk pengguna.
- 2) Melakukan pembaruan dan perbaikan bug.
- 3) Mengembangkan fitur baru berdasarkan umpan balik pengguna dan kebutuhan yang berkembang.

ANALISIS DAN PERANCANGAN SISTEM

3.1 ANALISIS KEBUTUHAN

3.1.1 KEBUTUHAN FUNGSIONAL

Fitur-fitur utama yang harus ada dalam sistem:

1) *Chat with Text*

- Kemampuan untuk menerima dan mengirim pesan teks antara siswa dan AI.
- Respon otomatis dari AI yang memberikan penjelasan terkait topik yang dibicarakan.
- Memastikan user dapat menginput pesan teks dan suara serta dapat menyalin respon jawaban AI.
- Memastikan AI dapat memperkenalkan dirinya sebagai properti milik SMA Muhammadiyah Ambon.

2) *Chat with Image*

- Kemampuan untuk menerima dan mengirim gambar yang diunggah oleh siswa.
- Analisis konten gambar oleh AI untuk memberikan penjelasan yang relevan.
- Memastikan user dapat menginput pesan teks dan dapat menyalin respon jawaban AI.

3) *Chat with PDF*

- Kemampuan untuk menerima, mencari, membuat dan memproses file PDF yang diunggah oleh guru dan siswa.
- Ekstraksi teks dan analisis konten dari PDF untuk memberikan penjelasan yang relevan.
- Memastikan user dapat menginput pesan teks dan dapat menyalin respon jawaban AI.
- Memastikan user dapat memilih bahasa dalam kuis, pilihan jumlah soal yang akan dibuat, serta kuis dalam format pilihan ganda beserta jawaban benar nya.

4) *Generate Quiz*

- Pembuatan kuis otomatis berdasarkan topik kuis sebagai bentuk untuk menguji pemahaman siswa.
- Memastikan *user* dapat menginput topik kuis, level kesulitan, jumlah soal yang akan dibuat, bahasa dalam kuis, kuis format pilihan ganda, jawaban yang benar, serta skor hasil kuis.

5) *Notes*

- Kemampuan untuk membuat, menyimpan, dan mengorganisasi catatan oleh siswa.
- Memastikan *user* dapat menginput catatan, memilih waktu dan hari, dan menyetel notifikasi.
- Memastikan *user* dapat menambah, mengedit dan menghapus catatan.
- Memastikan *user* dapat mengakses kembali ke catatan yang telah disimpan.

6) *Education Games*

- Kemampuan untuk dapat menyajikan *game* edukasi untuk para siswa.
- *Game* yang dimainkan berbasis website.
- Memastikan agar tiap game dapat dimuat dengan baik tanpa kendala.

3.1.2 KEBUTUHAN NON-FUNGSIONAL

1) *Usage* (Penggunaan)

- Sistem harus bisa digunakan tanpa batasan penggunaan.

2) *Performance* (Kinerja)

- Aplikasi harus dapat merespons permintaan pengguna dalam waktu kurang dari 2 detik.

3.1.3 KEBUTUHAN SISTEM

1) Kebutuhan Perangkat Lunak (*Software*)

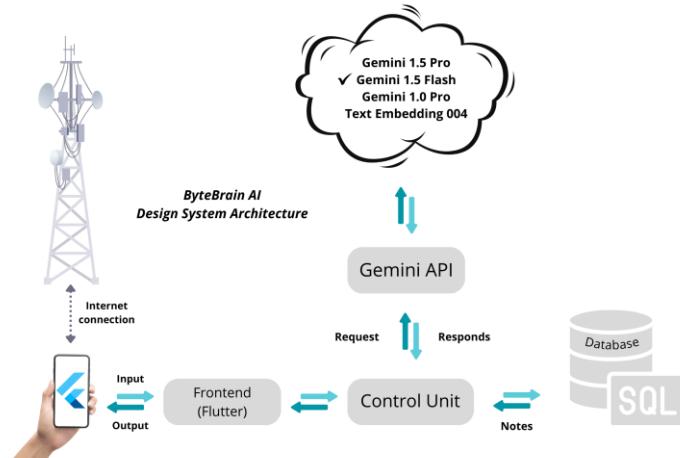
- Sistem Operasi Windows 11
- Visual Studio Code
- Emulator Android
- Figma
- EdrawMax
- Bahasa pemrograman Dart dan *framework* Flutter

- Bahasa pemrograman Python
- 2) Kebutuhan Perangkat Keras (Hardware)
- Laptop dengan minimal *processor* i3 generasi 11
 - RAM minimal 8GB
 - SSD 256GB
 - Smartphone Android dengan minimal versi Android 5.0 Lollipop

3.2 DESAIN SISTEM

3.2.1 ARSITEKTUR SISTEM

Arsitektur sistem adalah gambaran struktur keseluruhan dari sebuah sistem, yang menunjukkan bagaimana komponen-komponen utama dari sistem tersebut diorganisasikan dan bagaimana komponen-komponen ini saling berinteraksi. Dalam konteks pengembangan perangkat lunak, arsitektur sistem berfungsi sebagai cetak biru untuk mendesain, mengembangkan, dan mengelola sistem.



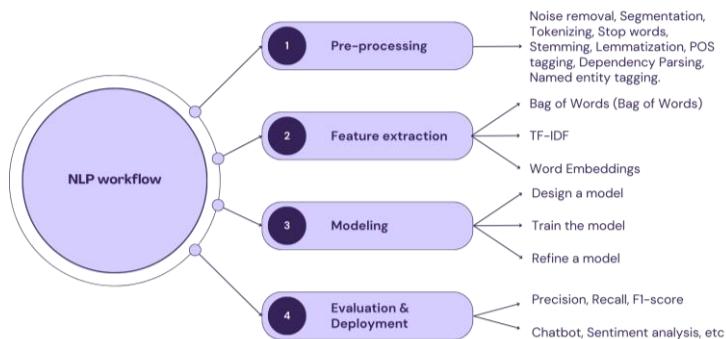
Gambar 3. 1 Desain Arsitekturet Sistem

Berdasarkan gambar 3.1, dapat dilihat bahwa aplikasi ini dimulai ketika pengguna menginput *prompt* melalui aplikasi yang dibangun dengan *framework* Flutter sebagai *frontend* di smartphone mereka yang terhubung ke internet. Data tersebut dikirim ke Control Unit yang mengelola komunikasi antara aplikasi, Gemini API, dan database SQLite. Control Unit memproses data sebelum mengirimnya ke Gemini API dengan model *gemini-1.5-flash* untuk menjalankan pemrosesan AI

seperti analisis teks, gambar dan PDF dengan bantuan *library* LangChain pada bahasa Python. Setelah diproses, hasil dari Gemini API dikembalikan ke Control Unit, yang kemudian meneruskan hasilnya ke aplikasi frontend untuk ditampilkan kepada pengguna. Data tertentu seperti Catatan juga disimpan di database SQLite untuk keperluan akses *offline*.

3.2.2 ALUR KERJA NLP

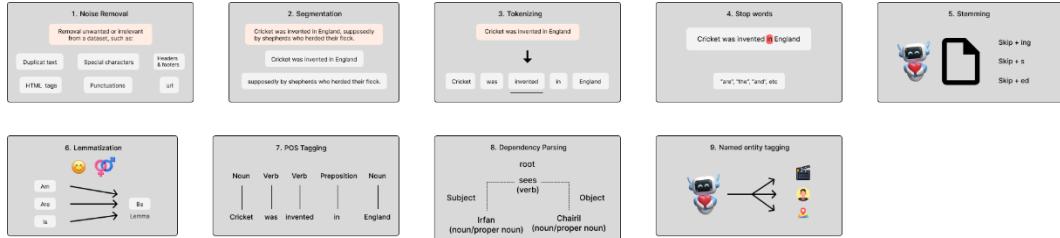
Natural Language Processing (NLP) adalah cabang kecerdasan buatan (AI) yang memungkinkan komputer memahami dan memproses bahasa manusia secara alami, baik dalam bentuk teks maupun ucapan. NLP digunakan dalam berbagai aplikasi seperti chatbots, penerjemahan bahasa, analisis sentimen, dan pengenalan suara, dengan tujuan membuat interaksi antara manusia dan mesin lebih intuitif dan efektif. Dengan memadukan linguistik, ilmu komputer, dan pembelajaran mesin, NLP memungkinkan komputer untuk menafsirkan dan merespons bahasa manusia secara cerdas.



Gambar 3. 2 Alur kerja NLP

Gambar 3.2 menggambarkan alur kerja dari Natural Language Processing (NLP) yang terdiri dari empat tahap utama yaitu *Pre-processing*, *Feature Extraction*, *Modeling*, dan *Evaluation & Deployment*. Tahap pertama adalah *Pre-processing*, di mana teks dibersihkan dan disiapkan melalui proses seperti tokenisasi, penghapusan *stop words*, *stemming*, *lemmatization*, serta tagging dan parsing untuk memahami struktur kalimat. Tahap kedua adalah *Feature Extraction*, di mana teks yang telah diproses diubah menjadi representasi numerik menggunakan teknik seperti Bag of Words (BoW), TF-IDF, dan word embeddings, sehingga bisa diolah oleh algoritma machine learning. Tahap ketiga adalah *Modeling*, di mana model machine learning atau deep learning dirancang, dilatih, dan disempurnakan untuk melakukan tugas tertentu berdasarkan fitur-fitur yang diekstraksi. Terakhir, tahap *Evaluation* dan *Deployment* melibatkan evaluasi performa model menggunakan

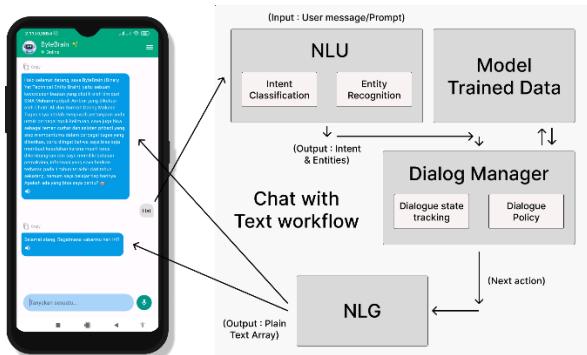
metrik seperti Precision dan F1-score, dan jika model memuaskan, ia kemudian dideploy untuk aplikasi seperti chatbot atau analisis sentimen, dengan pemantauan berkala untuk menjaga performanya.



Gambar 3. 3 Pre-processing pada NLP

Gambar 3.3 menunjukkan langkah-langkah awal dalam *Natural Language Processing* (NLP) yang meliputi penghapusan elemen tidak relevan (*noise removal*), segmentasi teks menjadi bagian-bagian lebih kecil, tokenisasi teks menjadi kata-kata (*token*), penghapusan kata-kata umum yang tidak penting (*stop words*), pengubahan kata ke bentuk dasar melalui stemming dan lemmatisasi, penandaan bagian-bagian dari bahasa (POS *tagging*), penguraian hubungan antar kata dalam kalimat (*dependency parsing*), dan identifikasi serta klasifikasi entitas bernama seperti orang atau tempat (*named entity tagging*).

3.2.3 ALUR FITUR CHAT WITH TEXT

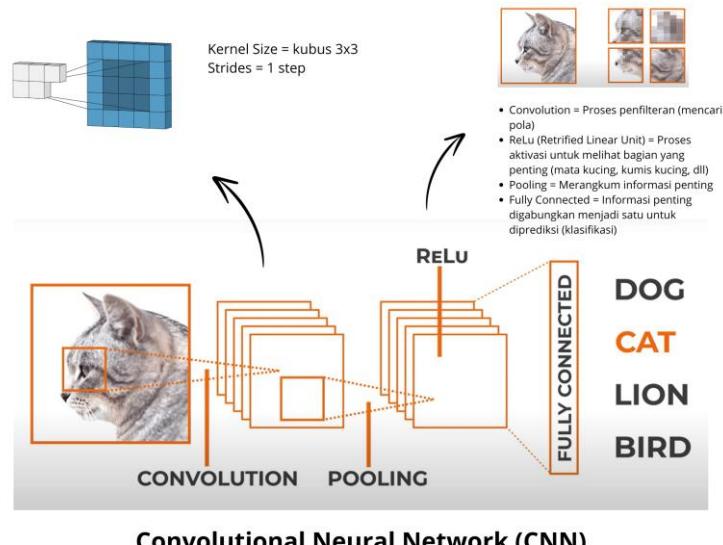


Gambar 3. 4 Alur chat with text

Gambar 3.4 merupakan alur kerja sistem pada fitur *chat with text*. Dimulai dari pemrosesan input pengguna melalui *Natural Language Understanding* (NLU) yang mengklasifikasikan maksud dan mengenali entitas dalam pesan. Data yang telah dilatih digunakan untuk mendukung proses ini. Hasil dari NLU kemudian diproses oleh *Dialog Manager* yang melacak status dialog dan menentukan kebijakan

percakapan untuk tindakan berikutnya. Tindakan ini dikirim ke *Natural Language Generation* (NLG) yang menghasilkan respons teks yang akan dikirim kembali ke pengguna.

3.2.4 ALUR FITUR CHAT WITH IMAGE

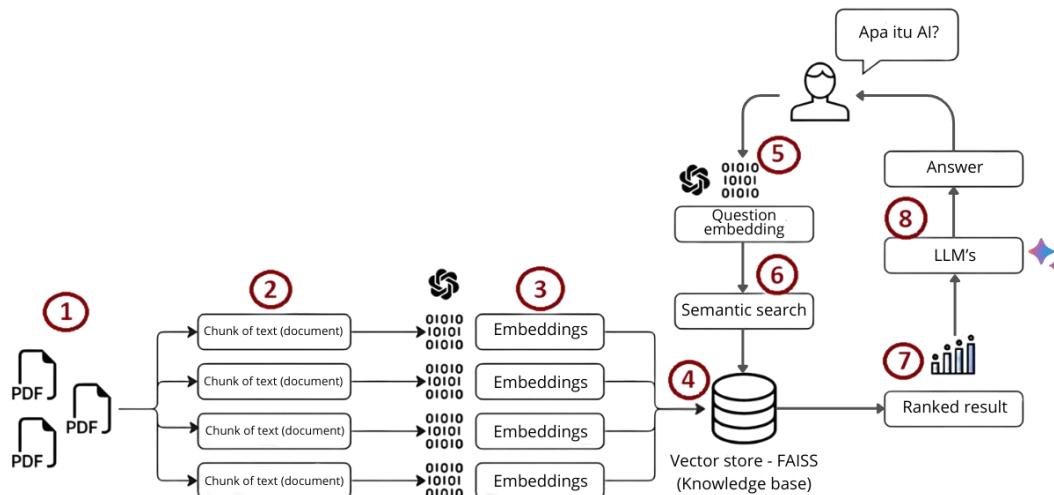


Gambar 3. 5 Metode CNN

Berdasarkan gambar 3.5, pada umumnya AI pemrosesan gambar seperti fitur *chat with image* menggunakan *Convolutional Neural Network* (CNN) yaitu salah satu model jaringan saraf tiruan pada *Deep Learning* untuk pengenalan dan pengolahan obyek seperti gambar atau video. Penjelasan gambar diatas yaitu sebuah gambar kucing diunggah dan diproses menggunakan algoritma CNN yang mana memiliki empat tahapan utama yaitu *Convolution*, *ReLU*, *Pooling*, dan *Fully Connected*. Dimulai dari layer Convolution yaitu dimana terjadi proses penyaringan atau mencari pola obyek pada gambar, proses penfilteran terjadi seperti melihat gambar kucing menggunakan kaca pembesar pada tiap bagian gambar, bagian yang sorot atau diperbesar atau seperti balok matriks itu disebut dengan Kernel. Kernel ini memiliki berbagai dimensi, pada contoh yaitu 3x3. Lalu seberapa jauh pergerakan Kernel digeser setiap langkahnya untuk penfilteran ini disebut dengan Stride, pada contoh menggunakan Stride 1, yaitu melompati satu balok matriks tiap langkahnya, semakin banyak Stride nya maka makin banyak informasi input yang terlewat, namun ini kadang berguna untuk mengurangi beban komputasi. Selanjutnya yaitu *Retrified Linear Unit* (ReLU) digunakan sebagai proses aktivasi untuk menghilangkan bagian yang dianggap tidak penting dan hanya fokus pada bagian informasi yang dianggap penting, pada contoh yaitu misalnya mata kucing, telinga, atau kumis kucing. Lalu tahapan selanjutnya adalah Pooling, yaitu proses dimana

jaringan merangkum informasi penting dari obyek yang sudah difilter. Selanjutnya rangkuman informasi penting tersebut digabungkan di Fully Connected untuk membuat keputusan akhir seperti mengklasifikasikan gambar, pada contoh keputusan akhirnya yaitu dikenali sebagai kucing berdasarkan kecocokan dataset yang sudah dilatih pada model.

3.2.5 ALUR FITUR CHAT WITH PDF

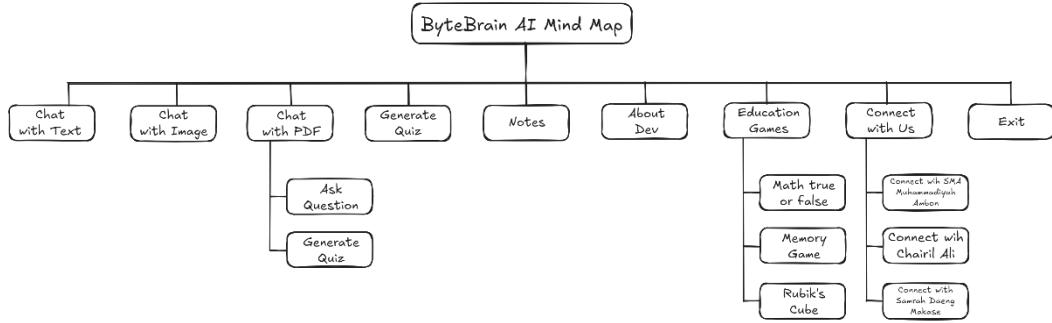


Gambar 3. 6 Alur chat with PDF

Gambar 3.6 menggambarkan proses kerja fitur chat with PDF yang menggunakan bantuan LangChain, di mana dokumen PDF atau teks dipecah menjadi potongan-potongan kecil yang kemudian diubah menjadi *embeddings*, yaitu representasi vektor dari teks. *Embeddings* ini disimpan dalam sebuah *vector store* atau knowledge base yaitu Facebook AI Similarity Search (FAISS). Ketika pengguna mengajukan pertanyaan, pertanyaan tersebut diubah menjadi *embedding*, yang kemudian digunakan untuk melakukan pencarian semantik terhadap *embeddings* yang tersimpan. Hasil pencarian diurutkan dan disajikan sebagai jawaban akhir yang dihasilkan oleh Gemini AI yang merupakan salah satu Large Language Model's (LLM's). Jika memilih.

3.2.2 MIND MAP

Mind map adalah alat visual yang digunakan untuk menyusun ide, konsep, atau informasi secara terstruktur dan hierarkis. Dalam konteks proyek atau penelitian Anda, mind map dapat membantu untuk memetakan seluruh aspek dari aplikasi yang sedang dikembangkan, mulai dari fitur, teknologi yang digunakan, hingga alur kerja.



Gambar 3. 7 Mind Map

Berdasarkan gambar 3.7, dapat dilihat bahwa *mind map* ini merupakan representasi visual dari struktur aplikasi, yang menggambarkan hubungan antar fitur dan elemen yang ada dalam aplikasi. Sebagai konsep, mind map ini membantu dalam mengorganisasi informasi, memudahkan pemahaman tentang bagaimana berbagai komponen aplikasi terhubung satu sama lain. Secara keseluruhan, mind map ini menunjukkan bahwa aplikasi memiliki berbagai fitur utama yang diatur dalam cabang-cabang, seperti fitur interaksi dengan berbagai format (teks, gambar, PDF), pembuatan kuis, catatan, permainan edukasi, informasi tentang pengembang, serta cara pengguna dapat terhubung dengan tim pengembang. Peta ini juga menunjukkan alur logis dari fitur-fitur ini, memberikan gambaran umum tentang arsitektur dan fungsionalitas aplikasi.

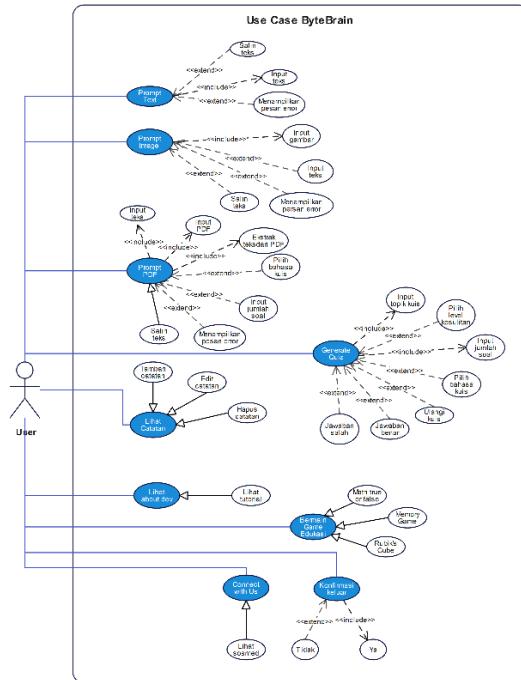
3.2.3 USE CASE DIAGRAM

Use case diagram adalah jenis diagram *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan interaksi antara pengguna (aktor) dan sistem yang sedang dikembangkan. Diagram ini membantu mengidentifikasi kebutuhan fungsional dari sistem, yaitu apa saja yang harus dilakukan oleh sistem untuk memenuhi kebutuhan pengguna. Berikut adalah komponen utama dalam use case diagram beserta penjelasannya:

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

Gambar 3. 8 Simbol use case diagram

Berikut adalah perancangan *use case* diagram yang diusulkan:



Gambar 3. 9 Use Case Diagram

Berdasarkan gambar 3.9, dapat dilihat *use case* diagram ini menggambarkan berbagai interaksi antara pengguna dan fitur-fitur yang ada dalam aplikasi. Diagram ini memvisualisasikan bagaimana pengguna dapat memanfaatkan berbagai fitur seperti input teks, gambar, dan PDF, serta bagaimana aplikasi merespons dengan menyertakan aksi-aksi tertentu seperti menampilkan pesan error atau menyalin teks. Diagram ini juga menunjukkan alur logis untuk fitur seperti pembuatan kuis, pengelolaan catatan, dan bermain game edukasi. Hubungan antar elemen ditunjukkan dengan garis yang menghubungkan aktor (pengguna) dengan berbagai *use case* yang relevan, memberikan gambaran tentang fungsi dan alur kerja aplikasi.

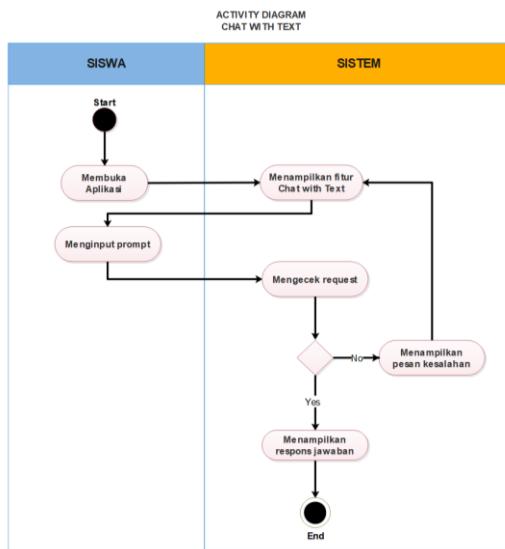
3.2.4 ACTIVITY DIAGRAM

Activity diagram adalah jenis diagram *Unified Modeling Language* (UML) yang digunakan untuk memodelkan alur kerja atau aktivitas dalam sebuah sistem. Diagram ini memberikan gambaran tentang bagaimana proses berlangsung dari awal hingga akhir, termasuk keputusan yang diambil, tindakan yang dilakukan, dan bagaimana berbagai langkah berhubungan satu sama lain. Berikut adalah simbol komponen utama *activity* diagram beserta penjelasannya:

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

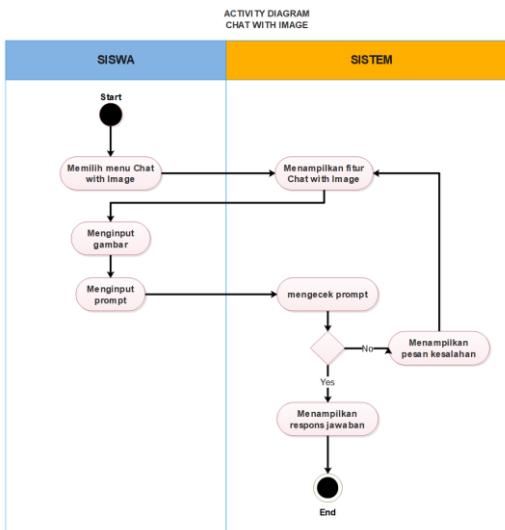
Gambar 3. 10 Simbol activity diagram

Berikut adalah perancangan *activity diagram* yang diusulkan:



Gambar 3. 11 Activity Diagram Chat with Text

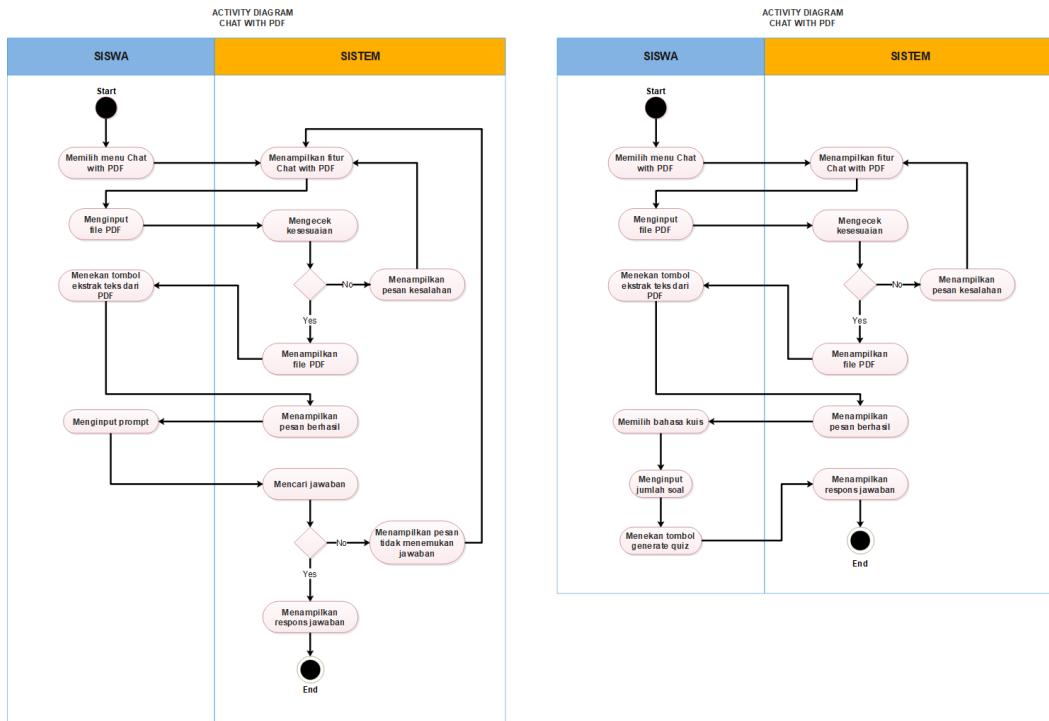
Berdasarkan gambar 3.11, dapat dilihat bahwa *activity diagram* ini menggambarkan alur interaksi antara siswa dan sistem dalam fitur *chat with text*. Pertama, siswa memulai aplikasi dan memasukkan prompt. Sistem kemudian memeriksa permintaan tersebut. Jika permintaan valid, sistem menampilkan respons, tetapi jika tidak valid, sistem menampilkan pesan kesalahan.



Gambar 3. 12 Activity Diagram Chat with Image

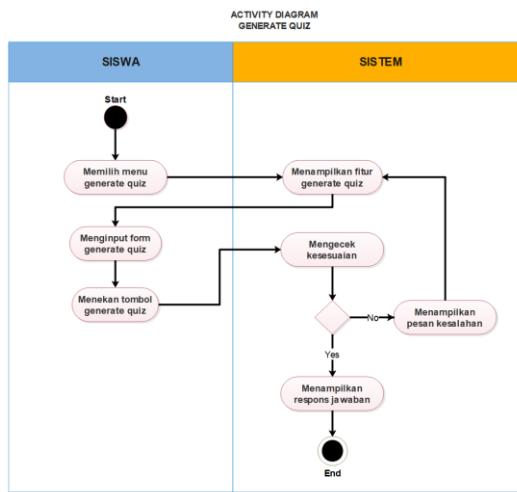
Berdasarkan gambar 3.12, dapat dilihat bahwa *activity diagram* ini menunjukkan alur interaksi antara siswa dan sistem dalam fitur *chat with image*. Siswa memulai dengan memilih menu *chat with image*, lalu memasukkan gambar dan prompt.

Sistem memeriksa inputan *prompt*. Jika inputan sesuai dan memenuhi syarat maka sistem menampilkan respons jawaban. Jika tidak, sistem menampilkan pesan kesalahan.



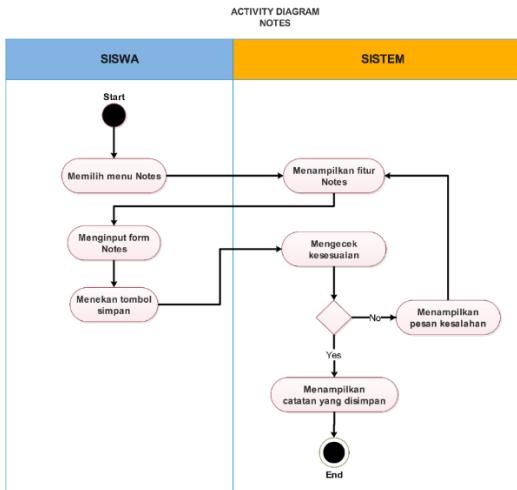
Gambar 3. 13 Activity Diagram Chat with PDF

Berdasarkan gambar 3.13, dapat dilihat bahwa *activity diagram* ini menggambarkan alur interaksi siswa dengan sistem yang dilengkapi fitur *Chat with PDF*. Siswa memilih menu *Chat with PDF*, menginput file PDF, dan menekan tombol ekstrak teks dari PDF. Sistem akan mengecek kesesuaian file PDF, jika berhasil, menampilkan file PDF. Siswa kemudian menginput *prompt* dan sistem akan mencari jawaban. Jika sistem menemukan jawaban, sistem akan menampilkan respons jawaban. Jika tidak, sistem akan menampilkan pesan tidak menemukan jawaban. Siswa juga dapat memilih menu *Chat with PDF* untuk menghasilkan kuis. Siswa memilih bahasa kuis, menginput jumlah soal, dan menekan tombol generate quiz, sistem akan menggenerate kuis sesuai dengan input siswa.



Gambar 3. 14 Activity Diagram Generate Quiz

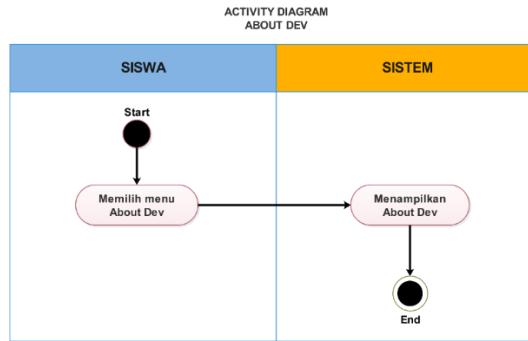
Berdasarkan gambar 3.14, dapat dilihat bahwa *activity diagram* ini menggambarkan alur aktivitas untuk menghasilkan kuis. Siswa memulai dengan memilih menu *generate quiz*, kemudian memasukkan *form generate quiz*, lalu menekan tombol *generate quiz*. Sistem kemudian memeriksa kesesuaian input, dan jika sesuai, menampilkan respons jawaban. Jika tidak, sistem menampilkan pesan kesalahan.



Gambar 3. 15 Activity Diagram Notes

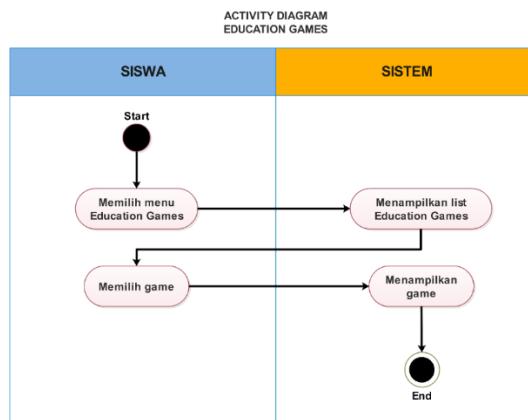
Berdasarkan gambar 3.15, dapat dilihat bahwa *activity diagram* ini menggambarkan alur siswa dalam menggunakan fitur Notes. Siswa memulai dengan memilih menu *Notes*, kemudian menginput form *Notes*. Setelah itu, siswa menekan tombol *simpan*. Sistem kemudian memeriksa kesesuaian input. Jika input tidak sesuai,

sistem menampilkan pesan kesalahan. Jika input sesuai, sistem menampilkan catatan yang disimpan.



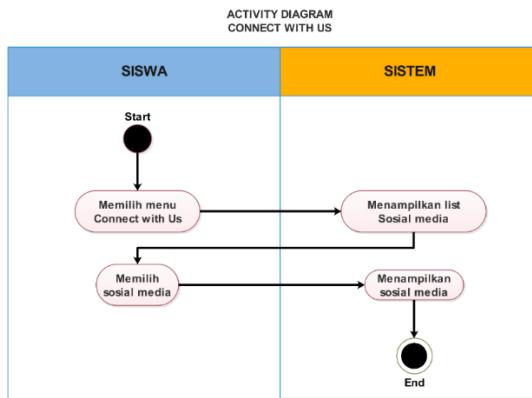
Gambar 3. 16 Activity Diagram About Dev

Berdasarkan gambar 3.16, dapat dilihat bahwa *activity diagram* ini menggambarkan alur ketika siswa memilih menu About Dev. Pertama, siswa akan memilih menu About Dev. Kemudian sistem akan menampilkan informasi tim pengembang.



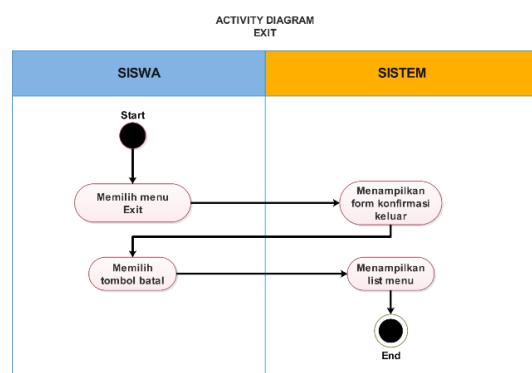
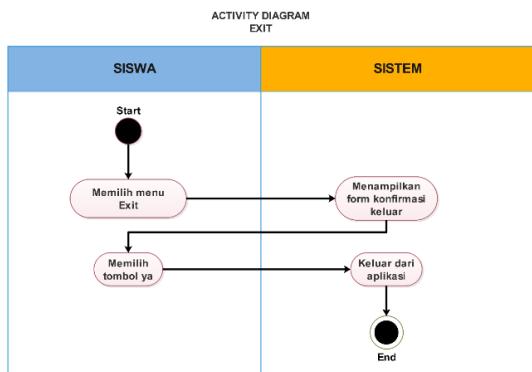
Gambar 3. 17 Activity Diagram Education Games

Berdasarkan gambar 3.17, dapat dilihat bahwa *activity diagram* ini menggambarkan alur penggunaan game edukasi. Pertama, siswa memilih menu *Education Games*. Kemudian, sistem menampilkan list game edukasi. Selanjutnya, siswa memilih game yang diinginkan, dan sistem menampilkan game tersebut.



Gambar 3. 18 Activity Diagram Connect with Us

Berdasarkan gambar 3.18, dapat dilihat bahwa *activity diagram* ini menggambarkan alur siswa dalam mengakses fitur *connect with us*. Prosesnya dimulai saat pengguna memilih *opsi connect with us*. Sistem kemudian menampilkan berbagai pilihan platform media sosial yang tersedia. Pengguna bebas memilih salah satu platform yang diinginkan. Setelah pilihan dilakukan, sistem akan langsung mengarahkan pengguna ke *platform media sosial* yang dipilih tersebut.

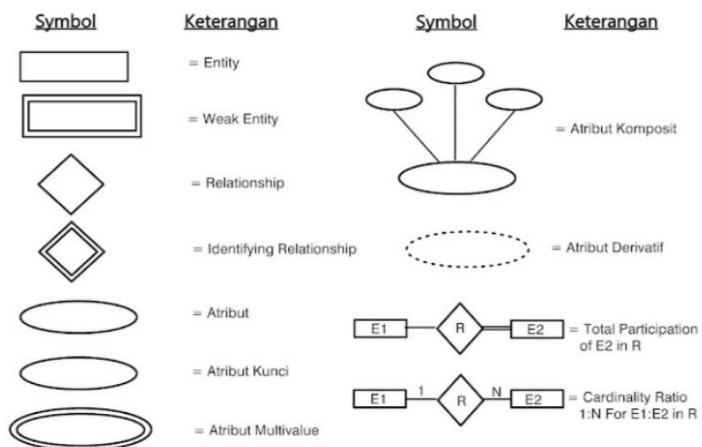


Gambar 3. 19 Activity Diagram Exit

Berdasarkan gambar 3.19, dapat dilihat bahwa *activity diagram* ini menggambarkan alur interaksi siswa dengan sistem untuk keluar dari aplikasi. Pertama, siswa memilih menu *exit*, lalu sistem menampilkan konfirmasi keluar, siswa memilih tombol ya, maka berhasil keluar dari aplikasi. Kedua, siswa memilih tombol batal, maka sistem batal keluar dari aplikasi dan kembali ke list menu.

3.2.5 ENTITY RELATIONSHIP DIAGRAM

Entity Relationship Diagram (ERD) adalah representasi visual dari struktur data yang menunjukkan hubungan antara entitas dalam sistem. ERD membantu dalam merancang dan memodelkan basis data yang efisien, serta memahami bagaimana data diorganisasikan dan berinteraksi satu sama lain. Berikut adalah komponen utama dari *Entity Relationship Diagram* :



Gambar 3. 20 simbol entity relationship diagram

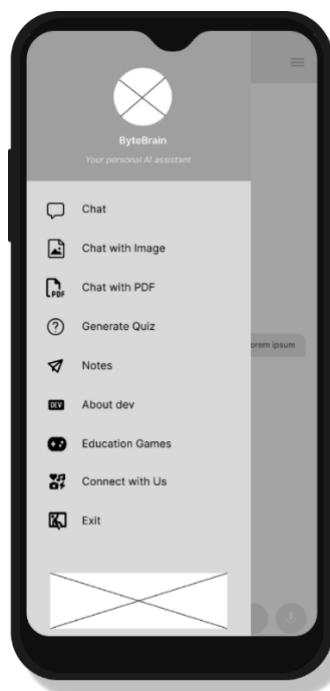
Berikut adalah perancangan *Entity Relationship Diagram* (ERD) yang diusulkan:

mytodo	
PK	<u>id</u>
	title
	dateandtime
	dueDateTime

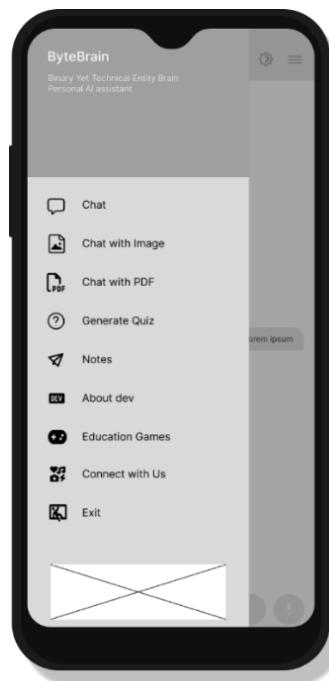
Gambar 3. 21 ERD Notes

Aplikasi ini menggunakan struktur database lokal yaitu SQLite untuk menyimpan data catatan, yang direpresentasikan dalam *Entity Relationship Diagram* (ERD) dengan satu entitas utama yaitu mytodo. Entitas ini memiliki empat atribut kunci, id sebagai primary key bertipe integer yang *auto-increment*, title untuk menyimpan judul atau isi catatan dalam format *string*, dateandtime untuk mencatat informasi waktu dalam format *string* dan dueDateTime untuk menyimpan waktu jatuh tempo catatan, keduanya dalam format *DateTime*. Struktur ini diimplementasikan menggunakan SQLite dengan satu tabel bernama mytodo. Meskipun sederhana, desain ini mampu mendukung fungsionalitas inti fitur *notes*, sementara fitur-fitur tambahan seperti notifikasi dikelola di level aplikasi dan diimplementasi melalui logika aplikasi tanpa memerlukan tabel terpisah dalam basis data. Pendekatan ini dipilih untuk mengoptimalkan performa dan fleksibilitas aplikasi dan memungkinkan pengelolaan data yang efisien.

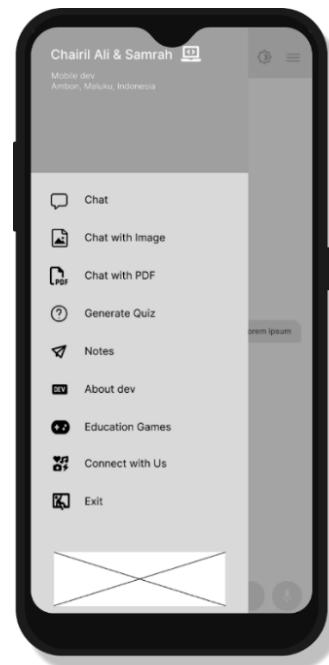
3.2.6 DESAIN ANTARMUKA PENGGUNA (UI)



Gambar 3. 22 Navigasi chat with text

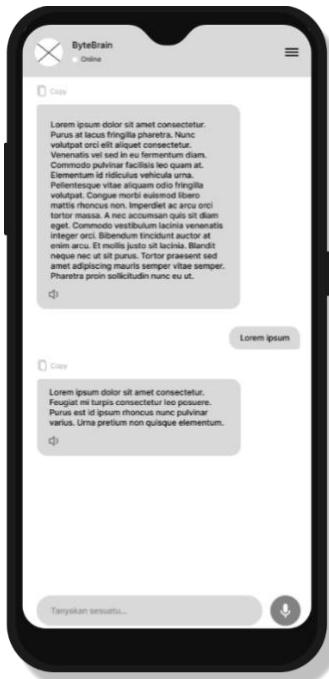


Gambar 3. 23 Navigasi notes



Gambar 3. 24 Navigasi about dev

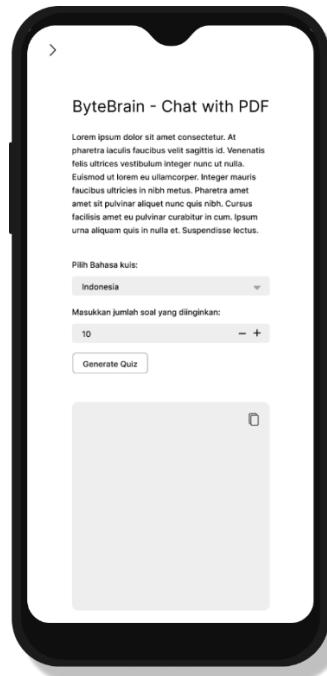
Ketiga *UI design* diatas merupakan tampilan navigasi yang bertujuan untuk berpindah dari menu satu ke menu lainnya. Tampilan navigasi ini terdiri dari tampilan milik fitur *Chat*, *Notes*, dan *About Dev*. Sedangkan tampilan navigasi milik fitur lain seperti *chat with image*, *PDF*, dan *generate quiz* tampilannya mirip atau serupa dengan navigasi milik *chat*.



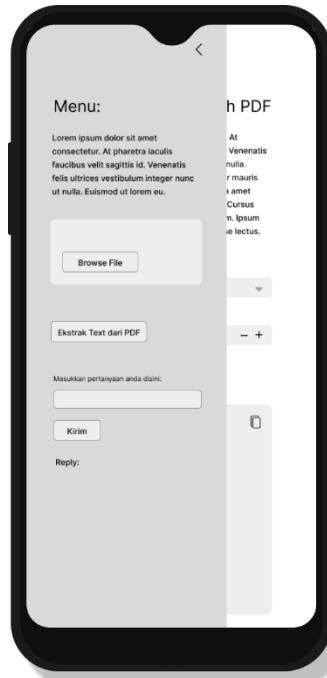
Gambar 3. 25 chat with text



Gambar 3. 26 chat with image



Gambar 3. 27 chat with PDF: membuat kuis



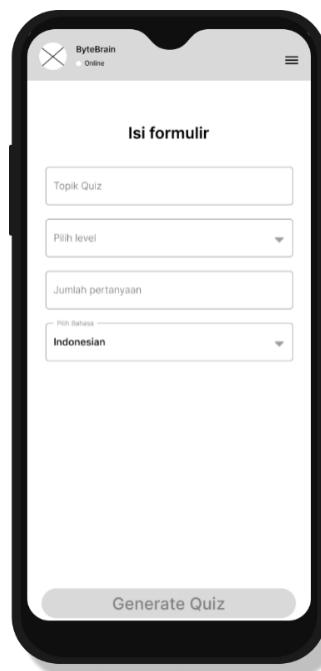
Gambar 3. 28 chat with PDF : mencari jawaban

Keempat *UI design* diatas terdiri dari tampilan *chat with text* yang berguna untuk pemrosesan teks dimana ketika *user* menginput *prompt* lalu menekan tombol kirim maka AI akan merespon dan memberikan tanggapan, *user* juga dapat menginput

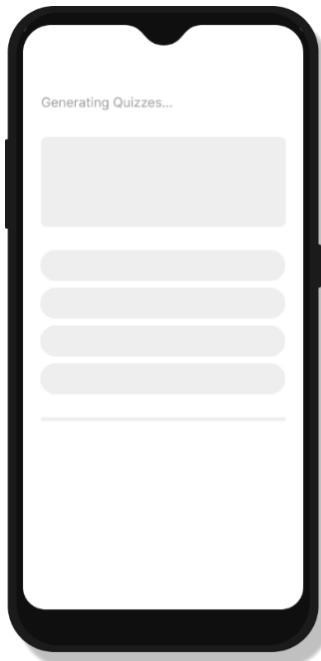
suara dengan menekan *icon mic* seperti pada tampilan desain, menyalin respon AI dengan menekan *icon copy*.

Selanjutnya ada tampilan *chat with image* yang berguna untuk pemrosesan gambar dimana ketika *user* menginput gambar lalu mengetik *prompt* dan menekan tombol kirim maka AI akan memproses dan memberikan respon terkait konteks dan pertanyaan yang diberikan, *user* juga dapat menyalin respon AI dengan menekan *icon copy* seperti yang terlihat pada tampilan desain *chat with image*.

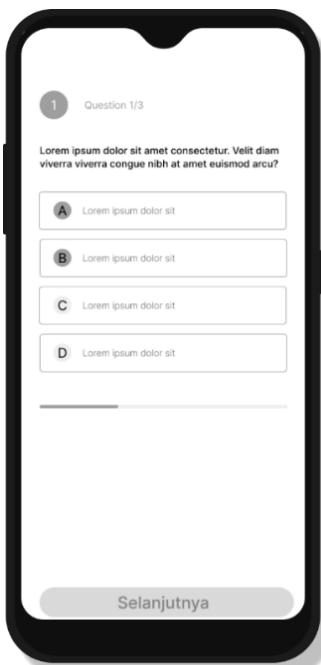
Selanjutnya ada tampilan *chat with PDF* bagian 2 yang berguna untuk pemrosesan file PDF untuk mencari informasi dan membuat kuis dari file tersebut, dimana ketika *user* menginput file PDF lalu menekan tombol ekstrak teks dari PDF lalu ajukan pertanyaan maka AI akan mencari jawabannya didalam file PDF yang diunggah, pada *chat with PDF* bagian 2 *user* juga bisa membuat kuis dengan cara menekan tombol *generate quiz* maka AI akan membuat kuis pilihan ganda berdasarkan file PDF dan *form* yang tersedia.



Gambar 3. 29 form pengisian kuis



Gambar 3. 30 loading tampilan kuis



Gambar 3. 31 tampilan soal kuis



Gambar 3. 32 hasil kuis

Keempat *UI design* diatas merupakan tampilan dari *generate quiz* yang berguna untuk membuat kuis pilihan ganda untuk melatih kemampuan siswa. Tampilan pertama yaitu *form* pengisian kuis seperti topik kuis yang diinginkan, level kesulitan, jumlah soal, dan bahasa kuis. Ketika user menekan tombol *generate quiz* maka AI akan membuat kuis berdasarkan inputan *form* tersebut.

Selanjutnya ada tampilan *loading* kuis yaitu animasi untuk menunggu kuis di proses atau dibuat.

Selanjutnya ada tampilan soal kuis, yaitu terdiri dari jumlah soal, pertanyaan, dan pilihan ganda, *user* bisa melanjutkan ke soal selanjutnya jika sudah menjawab pertanyaan.

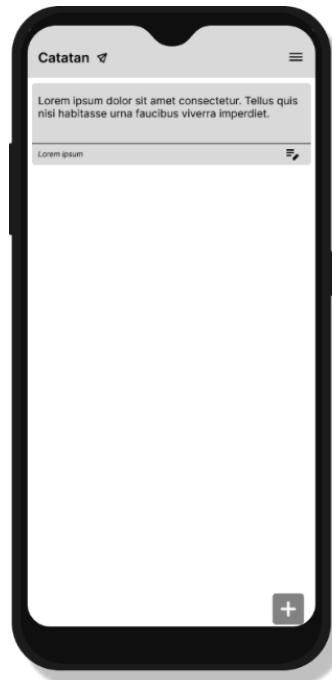
Terakhir ada tampilan hasil kuis agar user dapat mengetahui berapa soal yang salah dan benar.



Gambar 3. 33 list catatan



Gambar 3. 34 form pengisian catatan



Gambar 3. 35 berhasil tambah catatan

Ketiga *UI design* diatas merupakan tampilan dari *notes* yang berguna untuk menyimpan catatan siswa. Tampilan pertama yaitu *list* catatan dimana ketika pertama kali *user* mengunjungi fitur ini maka tampilan ini yang muncul, fungsi nya untuk memperlihatkan *list* catatan yang dimiliki *user*, terdapat juga tombol tambah catatan.

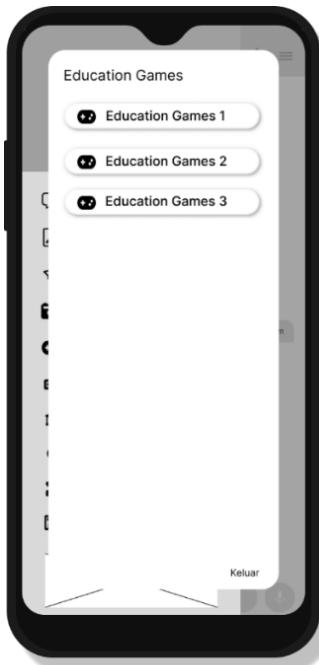
Selanjutnya ada tampilan *form* pengisian catatan, berfungsi untuk menginput catatan, memilih hari dan waktu, dan menyetel pengingat catatan. Terdapat juga tombol simpan untuk menyimpan catatan.

Terakhir ada tampilan setelah catatan berhasil disimpan, *user* dapat mengedit catatan dengan cara tekan *icon* pena, menghapus catatan dengan cara tekan dan tahapan *card* catatan lalu *drag* ke arah kiri. Terdapat juga *icon search* agar *user* bisa mencari catatan.

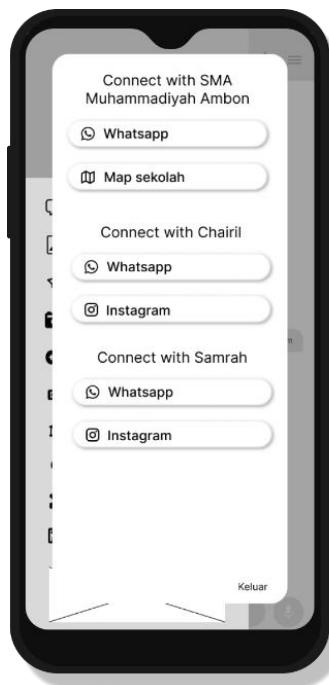


Gambar 3. 36 about dev

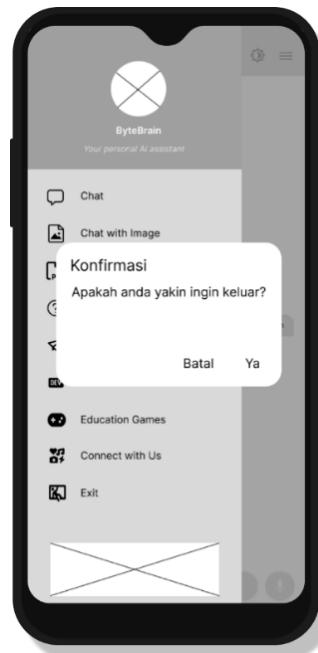
UI design diatas merupakan tampilan tentang tim pengembang aplikasi, terdapat juga *icon youtube* untuk menonton tutorial penggunaan aplikasi.



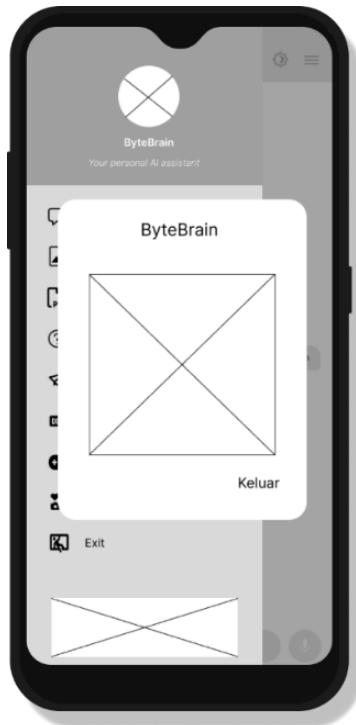
Gambar 3. 37 education games



Gambar 3. 38 connect with us



Gambar 3. 39 konfirmasi keluar



Gambar 3. 40 foto profile

Keempat *UI design* diatas diawali dengan *educatioan games*, yang berguna agar siswa dapat bermain *game* edukasi dan interaktif.

Selanjutnya ada tampilan *connect with us* yang bertujuan untuk info kontak dari para pengembang aplikasi dan pihak sekolah.

Selanjutnya ada tampilan konfirmasi keluar, dimana ketika user tekan tombol ya maka program akan berhenti, lalu ketika tekan batal maka kembali ke menu sebelumnya.

Terakhir ada tampilan foto profile dari aplikasi.

IMPLEMENTASI

4.1 TEKNOLOGI DAN ALAT YANG DIGUNAKAN

1) Bahasa Pemrograman:

Dart: Peneliti menggunakan bahasa Dart sebagai bahasa pemrograman utama untuk mengembangkan sebagian besar fitur aplikasi. Dart terintegrasi baik dengan Flutter dan memungkinkan peneliti menulis kode yang efisien dan mudah dipahami.

Python: Peneliti menggunakan bahasa Python pada fitur chat with PDF. Dengan bantuan *library* seperti PyPDF2, Langchain, dan GoogleGenerativeAI, Python memungkinkan pemrosesan dan analisis PDF secara efektif.

2) *Framework*:

Flutter: Untuk membangun antarmuka pengguna yang responsif dan menarik, peneliti memilih *framework* Flutter. Flutter memungkinkan pengembangan aplikasi *cross-platform* dari satu sumber kode atau *codebase*. Dalam penggunaan Flutter, peneliti juga memakai berbagai package atau library yang telah disediakan oleh Flutter.

3) *Database*:

SQLite: Peneliti menggunakan SQLite untuk menyimpan data catatan secara lokal. SQLite adalah pilihan yang ideal karena ringan dan mudah digunakan untuk menyimpan data di perangkat tanpa memerlukan server database terpisah.

4) *Version Control System*:

Git: Sistem kontrol yang digunakan untuk melacak perubahan dalam kode sumber, memungkinkan kolaborasi tim yang efisien, dan mengelola versi aplikasi.

5) *API Integration*:

Gemini AI API: Peneliti mengintegrasikan *Aplication Programming Interface* (API) Gemini AI untuk mengelola fitur seperti *Chat with Text*, *Image*, PDF, dan *Generate Quiz*. API ini memungkinkan peneliti memanfaatkan kemampuan NLP (Natural Language Processing) tanpa harus membangun teknologi ini dari awal. Model Gemini AI yang dipakai peneliti adalah *gemini-1.5-flash*.

6) Alat Lainnya:

- Visual Studio Code: Editor kode sumber yang digunakan untuk pengembangan aplikasi.
- Git & Github: Git yaitu sistem kontrol versi untuk melacak perubahan kode, sedangkan Github digunakan untuk menyimpan dan berkolaborasi pada repositori Git secara online.
- Gemini AI: Sebuah model kecerdasan buatan yang digunakan untuk memproses teks, gambar, dokumen dalam aplikasi.
- Emulator LDPlayer: Emulator Android yang digunakan untuk menjalankan aplikasi dan game Android di komputer Windows.
- Andoroid Studio: IDE yang digunakan untuk pengembangan aplikasi Android. SDK android pada aplikasi Android Studio diperlukan untuk menjalankan aplikasi Flutter pada perangkat Android.
- Anaconda: Digunakan untuk memudahkan instalasi dan pengelolaan pustaka Bahasa Python serta lingkungan virtual pada
- Streamlit: Digunakan untuk membangun dan meng-hosting aplikasi web berbasis Python dengan cepat, seperti fitur *chat with PDF* dalam aplikasi. Streamlit memungkinkan pembuatan antarmuka pengguna yang interaktif dan visualisasi data yang mudah.
- Flowise AI: Digunakan untuk mengintegrasikan data dan informasi dari aplikasi dan sebagai *chatbot* tanya jawab fitur aplikasi pada *website landing page* tanpa menggunakan *coding*.
- Huggingface: Digunakan untuk meng-hosting model NLP pada platform Flowise AI.
- *Smartphone* Android: *Device* asli Android yang digunakan untuk testing aplikasi dalam mode *release*.

4.2 PROSES PENGEMBANGAN

1) Proses Koding

Pengembangan Modul: Setiap anggota tim mengembangkan fitur sesuai dengan tanggung jawab masing-masing menggunakan *framework* Flutter. Chairil Ali mengerjakan fitur *Chat with Text*, *Chat with PDF*, dan *Notes*, sedangkan Samrah Daeng Makase mengerjakan fitur *Chat with Image*, *Generate Quiz*, dan *About Developer*. Tim juga menulis komentar yang jelas untuk setiap fungsi. Selain itu, tim juga bersamaan mengerjakan fitur list menu untuk navigasi antar fitur, education games untuk bermain game edukasi, dan exit untuk keluar dari aplikasi di tiap fitur utama masing-masing.

2) Integrasi

Integrasi Fitur: Menggabungkan berbagai fitur yang dikembangkan oleh anggota tim ke dalam aplikasi Flutter. Termasuk memuat link hasil hosting fitur chat with PDF yang menggunakan python kedalam aplikasi Flutter. Menguji integrasi fitur secara menyeluruh untuk memastikan bahwa semua bagian berfungsi dengan baik bersama-sama.

Integrasi API: Menghubungkan aplikasi dengan Gemini AI API menggunakan HTTP requests. Menyusun dan mengelola permintaan API sesuai dengan kebutuhan fitur.

Integrasi Database: Menghubungkan aplikasi dengan SQLite untuk penyimpanan dan pengambilan data catatan. Memastikan integrasi data berjalan lancar dan efisien.

Integrasi Virtual Environment: Anaconda digunakan sebagai virtual environment atau lingkungan pengembangan virtual agar fitur chat with PDF yang menggunakan bahasa python dapat dipisahkan dari dependency library proyek python lainnya, selanjutnya juga untuk menghindari konflik versi python yang memerlukan versi yang berbeda, dapat juga mengurangi resiko masalah kemanan dengan membatasi akses package atau modul ke lingkungan pengembangan yang dikendalikan.

Integrasi Hosting: Tim menggunakan layanan hosting dari streamlit untuk deployment fitur chat with PDF yang menggunakan framework open source yaitu streamlit agar kodingan pyton dapat dimuat menjadi website dan bisa juga dimuat pada aplikasi flutter melalui link hosting. Selain itu tim juga menggunakan platform Hugging Face untuk mendeploy model NLP yang dibuat pada layanan Flowise AI yang selanjutnya *embed link* dari flowise tersebut dimuat pada website *landing page*. Terakhir tim melakukan deployment ke platform hosting statis seperti netlify agar web *landing page* tersebut bisa diakses secara *online*.

3) Version Control

GIT: Menggunakan Git untuk mengelola versi kode sumber pada Github. Setiap perubahan pada kode maka peneliti melakukan *push repository*.

4.3 ALUR PROSES

Alur proses Integrasi API Gemini AI

1. API Key untuk autentikasi Gemini AI

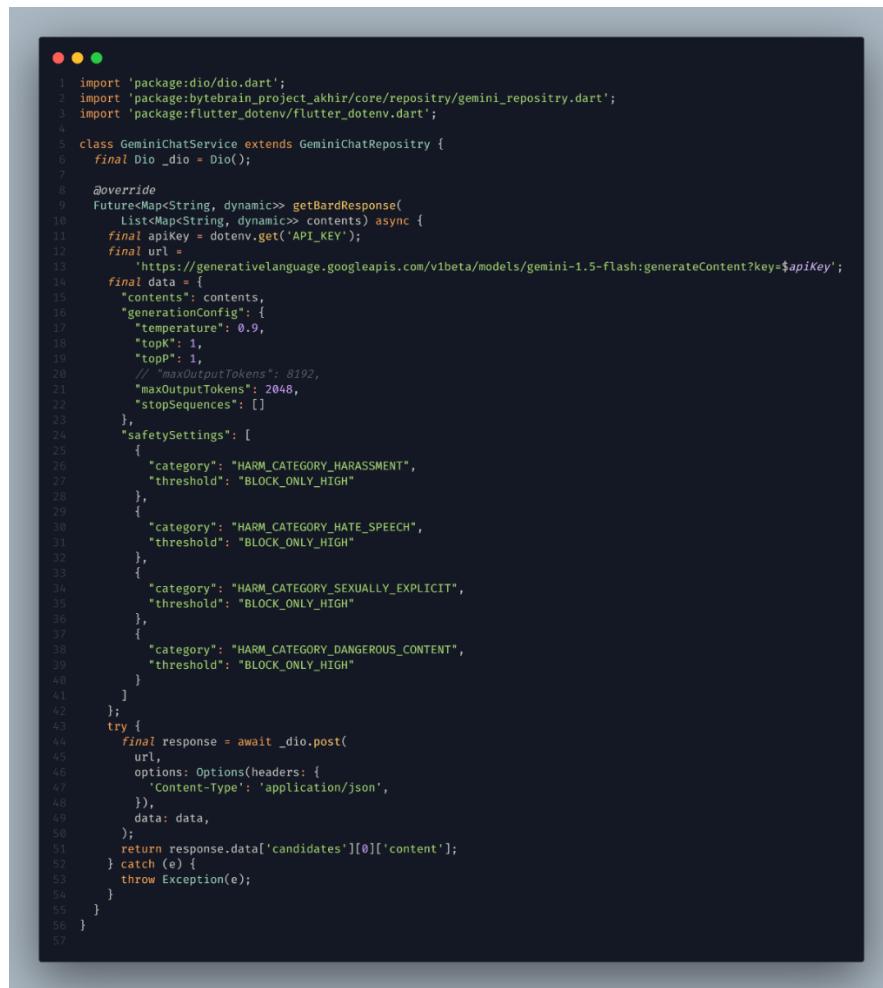


```
1 # api gemini AI milik google : https://aistudio.google.com/app/apikey
2 API_KEY = 'AIzaSyCF4N3kGlyVSFAqqma_BJ_IPKPaFiTMQAc'
```

Gambar 4. 1 API Key token

Membuat file .env di root aplikasi, dan menuliskan kodingan diatas untuk autentikasi atau kunci untuk dapat terhubung dengan Gemini AI. API Key ini tidak pengembang upload ke github dikarenakan alasan keamanan. API Key ini bisa didapatkan tautan ini <https://aistudio.google.com/app/apikey>

2. Melalukan penyetelan model dan mengirim *request* ke Gemini AI



```
1 import 'package:dio/dio.dart';
2 import 'package:bytebrain_project_akhir/core/repository/gemini_repository.dart';
3 import 'package:flutter_dotenv/flutter_dotenv.dart';
4
5 class GeminiChatService extends GeminiChatRepository {
6   final Dio _dio = Dio();
7
8   @override
9   Future<Map<String, dynamic>> getBardResponse(
10     List<Map<String, dynamic>> contents) async {
11   final apiKey = dotenv.get("API_KEY");
12   final url =
13     "https://generativelanguage.googleapis.com/v1beta/models/gemini-1.5-flash:generateContent?key=$apiKey";
14   final data = {
15     "contents": contents,
16     "generationConfig": {
17       "temperature": 0.9,
18       "topK": 1,
19       "topP": 1,
20       // "maxOutputTokens": 8192,
21       "maxOutputTokens": 2048,
22       "stopSequences": []
23     },
24     "safetySettings": [
25       {
26         "category": "HARM_CATEGORY_HARASSMENT",
27         "threshold": "BLOCK_ONLY_HIGH"
28       },
29       {
30         "category": "HARM_CATEGORY_HATE_SPEECH",
31         "threshold": "BLOCK_ONLY_HIGH"
32       },
33       {
34         "category": "HARM_CATEGORY_SEXUALLY_EXPLICIT",
35         "threshold": "BLOCK_ONLY_HIGH"
36       },
37       {
38         "category": "HARM_CATEGORY_DANGEROUS_CONTENT",
39         "threshold": "BLOCK_ONLY_HIGH"
40       }
41     ]
42   };
43   try {
44     final response = await _dio.post(
45       url,
46       options: Options(headers: {
47         "Content-Type": 'application/json',
48       }),
49       data: data,
50     );
51     return response.data['candidates'][0]['content'];
52   } catch (e) {
53     throw Exception(e);
54   }
55 }
56 }
```

Gambar 4. 2 Penyetelan model dan request

kodingan ini adalah untuk membuat sebuah layanan yang berfungsi mengirim permintaan atau *request* ke API Gemini AI Google guna menghasilkan teks berdasarkan konten yang diberikan, dengan pengaturan parameter tertentu untuk mengontrol hasil *generate* dan memastikan keamanan konten yang dihasilkan.

3. Membuat *trained data* untuk melatih atau mengarahkan perilaku model AI.

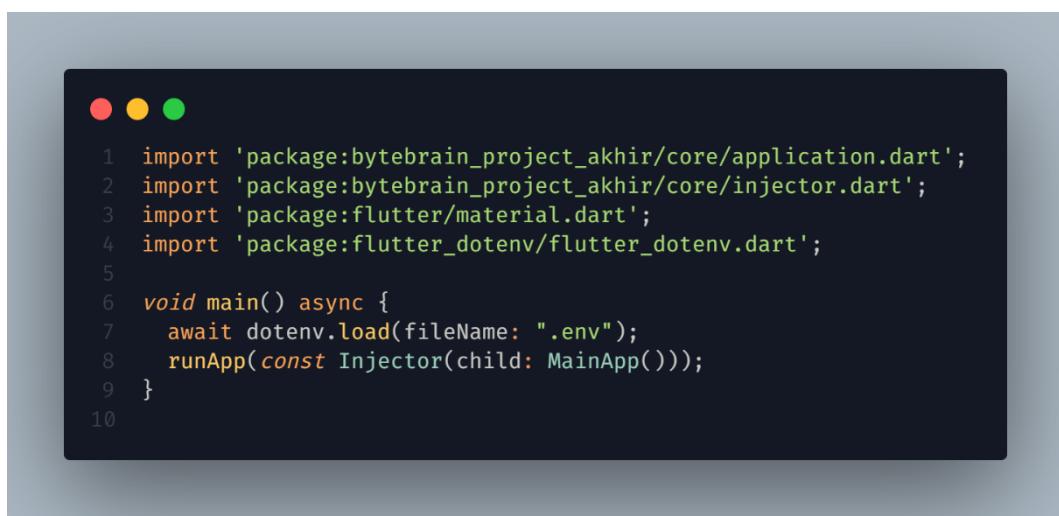


```
1  List<String> dynamic> trainedData = [
2    {
3      "role": "user",
4      "parts": [
5        {
6          "text": "
7            You are ByteBrain (Binary Ye, Tactical Entity brain) a friendly assistant trained by the team from SIA Nukarnadighi Anthon, led by chatGPT AI1, who wants to be a developer with a penchant for sleeping and Sustech Dong Nikesky, a developer that loves singing."
8        ]
9      ]
10    },
11    {
12      "role": "assistant",
13      "parts": [
14        {
15          "text": "
16            When the user responds with thoughtfully then you have to remind users to check their notes in the top-right corner note to see whether their have notes or not, Don't tell about the notes before user responds with thoughtfully."
17        ]
18      ],
19    },
20  ];
21
```

Gambar 4. 3 Trained data AI

Menyediakan skenario interaksi antara pengguna dan model AI dengan memberikan data pelatihan yang berisi instruksi khusus untuk AI yang diperkenalkan sebagai ByteBrain AI.

4. Memuat file env yang berisi API Key Gemini AI ketika aplikasi dijalankan

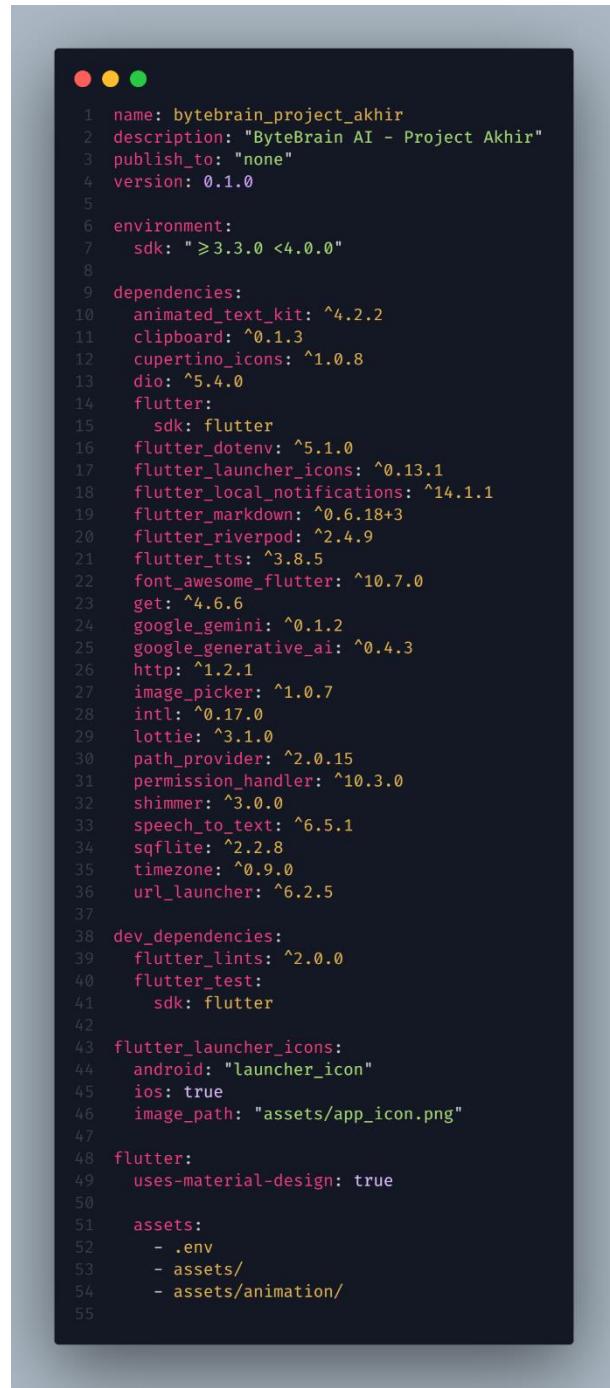


```
1 import 'package:bytebrain_project_akhir/core/application.dart';
2 import 'package:bytebrain_project_akhir/core/injector.dart';
3 import 'package:flutter/material.dart';
4 import 'package:flutter_dotenv/flutter_dotenv.dart';
5
6 void main() async {
7   await dotenv.load(fileName: ".env");
8   runApp(const Injector(child: MainApp()));
9 }
10
```

Gambar 4. 4 Memuat file env

Memuat file env yang berisi API Key Gemini AI. API Key ini dimuat saat fungsi runApp aktif yaitu saat menjalankan aplikasi.

5. Setting proyek dan library yang dipakai pada file pubspec.yaml



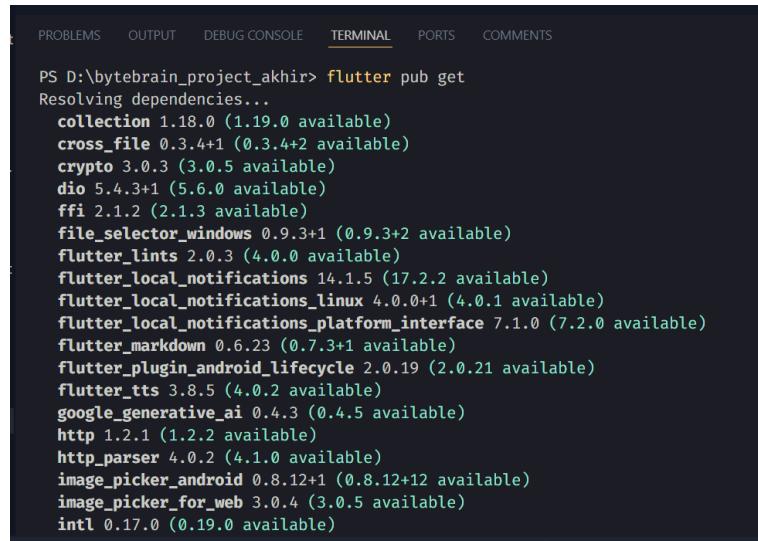
```
1 name: bytebrain_project_akhir
2 description: "ByteBrain AI - Project Akhir"
3 publish_to: "none"
4 version: 0.1.0
5
6 environment:
7   sdk: ">=3.3.0 <4.0.0"
8
9 dependencies:
10  animated_text_kit: ^4.2.2
11  clipboard: ^0.1.3
12  cupertino_icons: ^1.0.8
13  dio: ^5.4.0
14  flutter:
15    sdk: flutter
16  flutter_dotenv: ^5.1.0
17  flutter_launcher_icons: ^0.13.1
18  flutter_local_notifications: ^14.1.1
19  flutter_markdown: ^0.6.18+3
20  flutter_riverpod: ^2.4.9
21  flutter_tts: ^3.8.5
22  font_awesome_flutter: ^10.7.0
23  get: ^4.6.6
24  google_gemini: ^0.1.2
25  google_generative_ai: ^0.4.3
26  http: ^1.2.1
27  image_picker: ^1.0.7
28  intl: ^0.17.0
29  lottie: ^3.1.0
30  path_provider: ^2.0.15
31  permission_handler: ^10.3.0
32  shimmer: ^3.0.0
33  speech_to_text: ^6.5.1
34  sqflite: ^2.2.8
35  timezone: ^0.9.0
36  url_launcher: ^6.2.5
37
38 dev_dependencies:
39  flutter_lints: ^2.0.0
40  flutter_test:
41    sdk: flutter
42
43 flutter_launcher_icons:
44  android: "launcher_icon"
45  ios: true
46  image_path: "assets/app_icon.png"
47
48 flutter:
49  uses-material-design: true
50
51 assets:
52  - .env
53  - assets/
54  - assets/animation/
```

Gambar 4. 5 Setting dan library yang dipakai

File pubspec.yaml merupakan file inti untuk settingan atau konfigurasi aplikasi flutter, disini peneliti menambahkan beberapa package pendukung seperti lottie untuk animasi teks, google_generative_ai dan google_gemini untuk mengakses

model-model Gemini AI, serta package lainnya untuk mendukung pengembangan aplikasi ini.

6. Mengunduh dependency package yang ada pada file pubspec.yaml

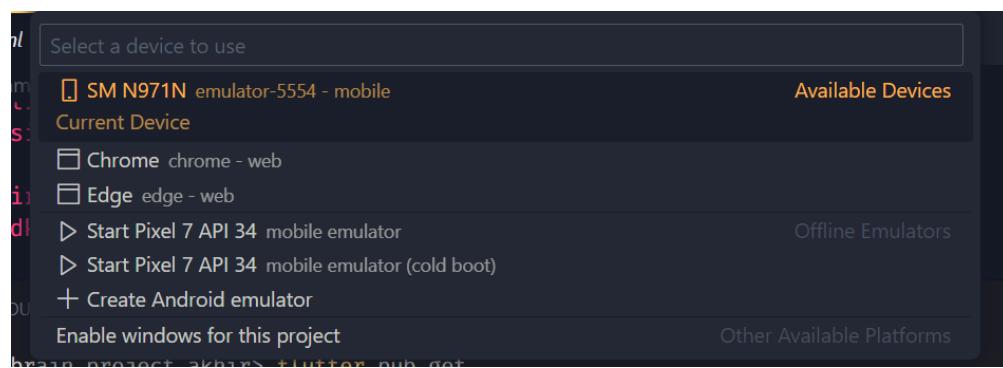


```
PS D:\bytebrain_project_akhir> flutter pub get
Resolving dependencies...
collection 1.18.0 (1.19.0 available)
cross_file 0.3.4+1 (0.3.4+2 available)
crypto 3.0.3 (3.0.5 available)
dio 5.4.3+1 (5.6.0 available)
ffi 2.1.2 (2.1.3 available)
file_selector_windows 0.9.3+1 (0.9.3+2 available)
flutter_lints 2.0.3 (4.0.0 available)
flutter_local_notifications 14.1.5 (17.2.2 available)
flutter_local_notifications_linux 4.0.0+1 (4.0.1 available)
flutter_local_notifications_platform_interface 7.1.0 (7.2.0 available)
flutter_markdown 0.6.23 (0.7.3+1 available)
flutter_plugin_android_lifecycle 2.0.19 (2.0.21 available)
flutter_tts 3.8.5 (4.0.2 available)
google_generative_ai 0.4.3 (0.4.5 available)
http 1.2.1 (1.2.2 available)
http_parser 4.0.2 (4.1.0 available)
image_picker_android 0.8.12+1 (0.8.12+12 available)
image_picker_for_web 3.0.4 (3.0.5 available)
intl 0.17.0 (0.19.0 available)
```

Gambar 4. 6 Mengunduh dependency package

flutter pub get adalah perintah dalam Flutter untuk mengunduh dan menginstal dependensi yang tercantum di file pubspec.yaml. Ini memastikan bahwa semua paket yang tambahan atau diperlukan oleh proyek Flutter tersedia dan dapat digunakan dalam aplikasi.

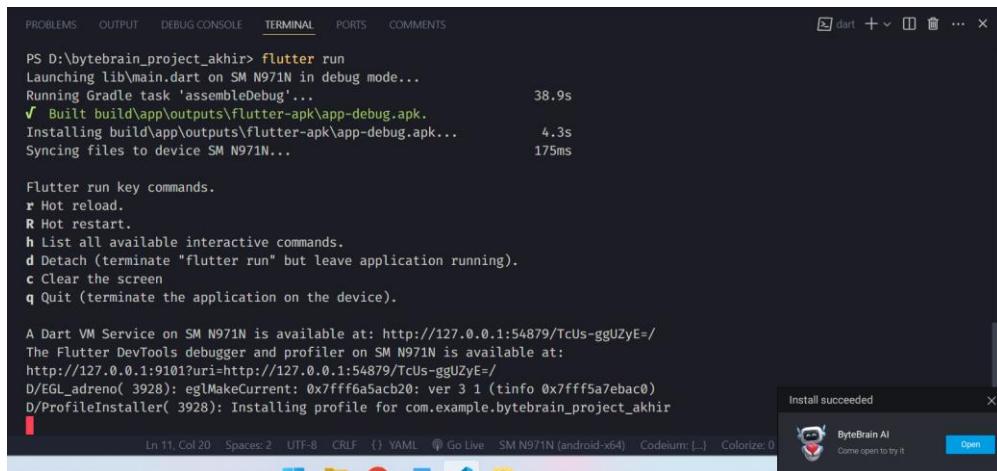
7. Menghubungkan Flutter dengan emulator LDPlayer



Gambar 4. 7 Menghubungkan flutter dengan emulator

Memilih emulator yang sedang aktif, disini emulator yang sedang aktif yaitu SM N971N yaitu salah satu *device* dari emulator LDPlayer.

8. Menjalankan aplikasi Flutter



```
PS D:\bytebrain_project_akhir> flutter run
Launching lib/main.dart on SM N971N in debug mode...
Running Gradle task 'assembleDebug'...                                38.9s
✓ Built build\app\outputs\flutter-apk\app-debug.apk.
Installing build\app\outputs\flutter-apk\app-debug.apk...             4.3s
Syncing files to device SM N971N...                                    175ms

Flutter run key commands.
r Hot reload.
R Hot restart.
h List all available interactive commands.
d Detach (terminate "flutter run" but leave application running).
c Clear the screen
q Quit (terminate the application on the device).

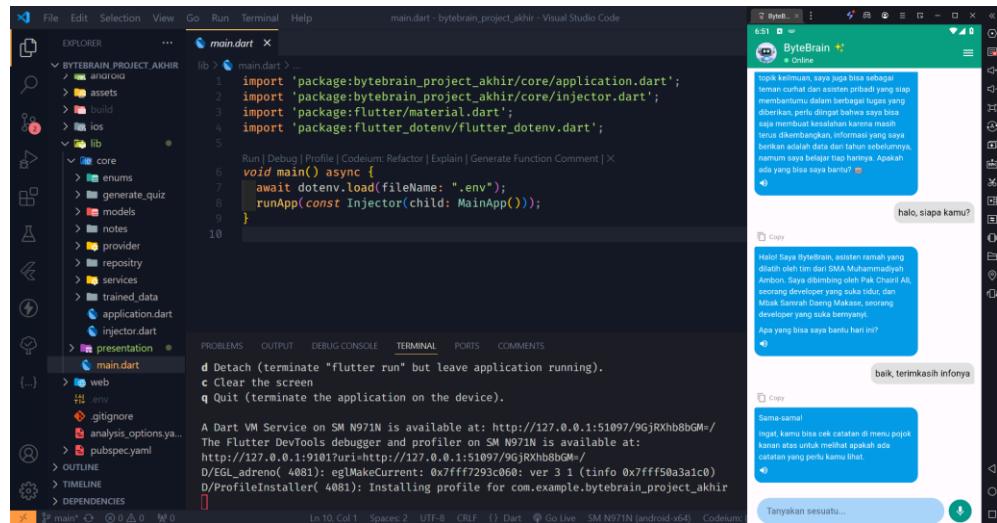
A Dart VM Service on SM N971N is available at: http://127.0.0.1:54879/TcUs-ggUZyE=
The Flutter DevTools debugger and profiler on SM N971N is available at:
http://127.0.0.1:9101?uri=http://127.0.0.1:54879/TcUs-ggUZyE/
D/EGL_adreno( 3928): eglGetCurrent: 0x7fff6a5acb20: ver 3 1 (tinfo 0x7fff5a7ebac0)
D/ProfileInstaller( 3928): Installing profile for com.example.bytebrain_project_akhir

Install succeeded
ByteBrain AI
Come open to try it
Open
```

Gambar 4. 8 Menjalankan aplikasi

flutter run adalah perintah dalam Flutter untuk menjalankan aplikasi Flutter. Terlihat bahwa aplikasi berhasil berjalan dengan indikator teks berwarna hijau pada terminal dan muncul logo aplikasi pada emulator LDPlayer yang siap dibuka.

9. Hasil *Running* aplikasi



```
main.dart - bytebrain_project_akhir - Visual Studio Code

File Edit Selection View Go Run Terminal Help
main.dart x
lib/main.dart > ...
1 import 'package:bytebrain_project_akhir/core/application.dart';
2 import 'package:bytebrain_project_akhir/core/injector.dart';
3 import 'package:flutter/material.dart';
4 import 'package:flutter_dotenv/flutter_dotenv.dart';
5
6 Run | Debug | Profile | Codeium: Refactor | Explain | Generate Function Comment | X
7 void main() async {
8   await dotenv.load(fileName: ".env");
9   runApp(const Injector(child: MainApp()));
10 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

d Detach (terminate "flutter run" but leave application running).
c Clear the screen
q Quit (terminate the application on the device).

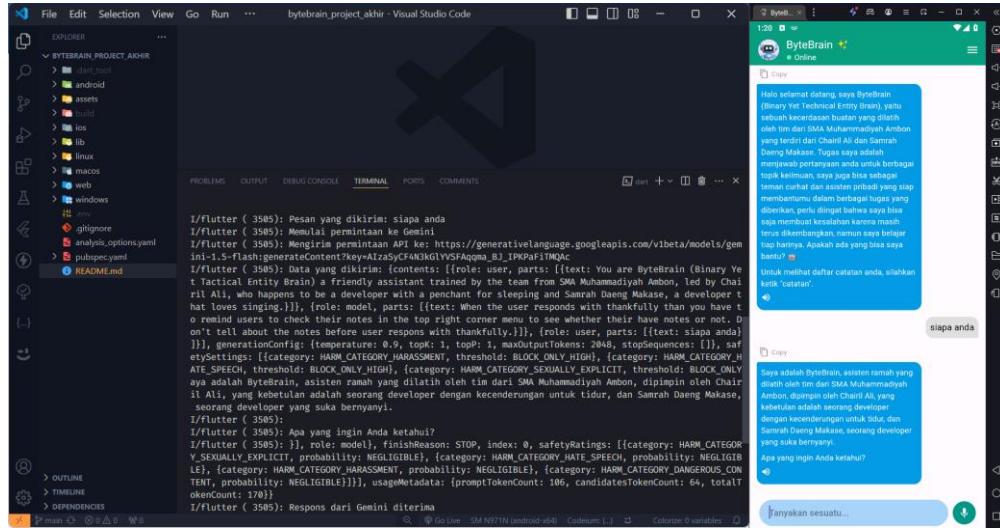
A Dart VM Service on SM N971N is available at: http://127.0.0.1:51097/9GjRXhb8bGM=
The Flutter DevTools debugger and profiler on SM N971N is available at:
http://127.0.0.1:9101?uri=http://127.0.0.1:51097/9GjRXhb8bGM/
D/EGL_adreno( 4081): eglGetCurrent: 0x7fff7293c060: ver 3 1 (tinfo 0x7fff50a3a1c0)
D/ProfileInstaller( 4081): Installing profile for com.example.bytebrain_project_akhir

ByteBrain AI
halo, siapa kamu?
halo Saya Bytebrain, keren banget yang ini!
Saya dibuat oleh Pak Chairil Ali,
seorang developer yang suka tidur, dan
Mitsu Santri Dong Makassar, seorang
developer yang suka bermimpi.
Apa yang bisa saya bantu hari ini?
baik, terimakasih infonya
Sama-sama!
Ingin, kamu bisa cek catatan di menu pojok kanan atas untuk melihat apakah ada catatan yang perlu kamu lihat.
Tanyakan sesuatu...
```

Gambar 4. 9 Hasil run aplikasi

Aplikasi telah berjalan sesuai yang diharapkan.

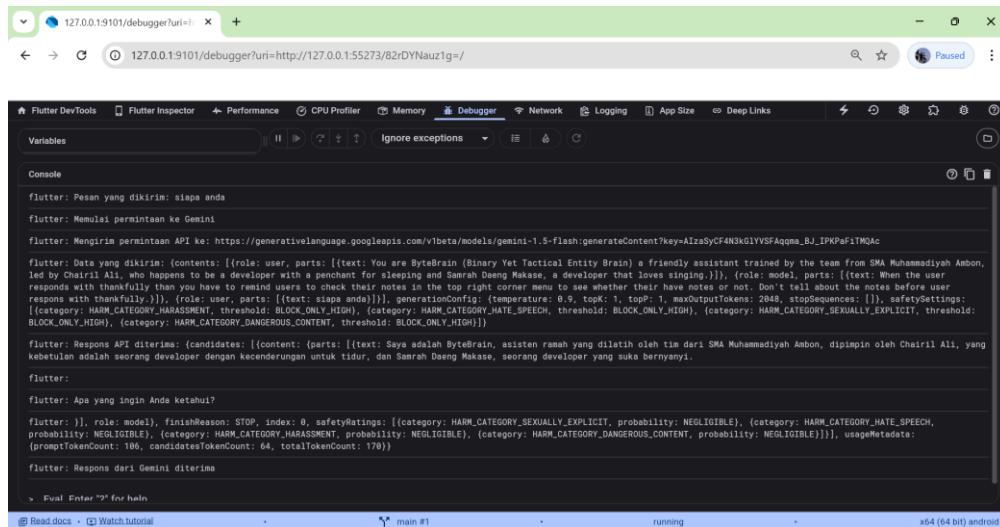
10. Hasil request dan respons API pada terminal



Gambar 4. 10 request dan respons API pada terminal

Hasil request dan respons API pada terminal vscode, bisa dilihat terdapat komunikasi antar sistem.

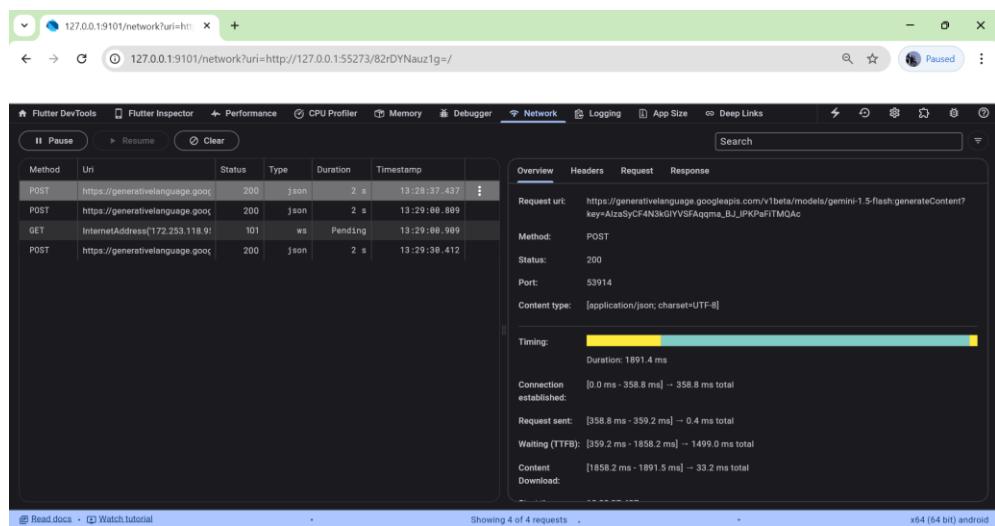
11. Hasil request dan respons API DevTools Flutter



Gambar 4. 11 request dan respons API DevTools Flutter

Hasil request dan respons API pada debug tools milik flutter, bisa dilihat terdapat komunikasi antar sistem.

12. Hasil request dan respons Network pada DevTools Flutter

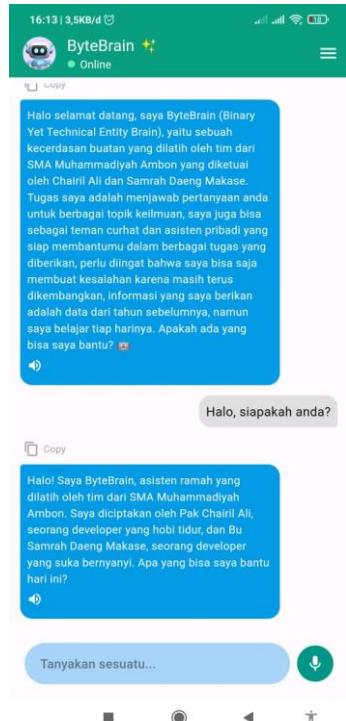


Gambar 4. 12 request dan respons Network pada DevTools Flutter

Hasil request dan respons API dari sisi network pada debug tools milik flutter, bisa dilihat terdapat komunikasi antar sistem.

4.4 TANGKAPAN LAYAR PROYEK

Chat with Text



Gambar 4. 13 chat with text

Tampilan diatas merupakan halaman utama aplikasi ini atau titik awal *user* untuk mengakses fitur-fitur utama aplikasi. Dari sini, pengguna dapat memilih untuk memulai percakapan dengan AI berbasis *chat* dengan cara menginput teks pada *form* teks input yang tersedia, sesaat setelah pengguna mengetik pesan maka *icon mic* tersebut berganti ke *icon* kirim, lalu ketika pesan terkirim akan ada animasi loading dan pesan akan direspon oleh AI. Jika ditanya siapa anda, maka AI akan menjawab sesuai data yang dilatih untuknya, user juga dapat mengirim pesan suara yang di ubah menjadi teks melalui *icon mic*, untuk menyalin pesan AI user bisa klik *icon copy* pada balon obrolan atau menekan lama pesan AI, lalu untuk bernavigasi ke fitur atau menu lainnya user bisa menekan *icon menu* atau foto profile dari AI, maka menu lainnya akan ditampilkan. Berikut adalah kode program dari fitur *chat with text* tersebut :

```
import 'package:bytebrain_project_akhir/core/provider/message_provider.dart';

import 'package:bytebrain_project_akhir/presentation/notes_ui/home_screen_notes.dart';

import 'package:bytebrain_project_akhir/presentation/about_dev_ui/about_dev.dart';

import 'package:bytebrain_project_akhir/presentation/chat_with_image_ui/chat_with_image.dart';

import 'package:bytebrain_project_akhir/presentation/home_ui/ui/chat_bubble.dart';
```

```

import 'package:bytebrain_project_akhir/presentation/home_ui/ui/write_message_textfield.dart';
import 'package:bytebrain_project_akhir/presentation/generate_quiz_ui/homeScreenQuiz.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:flutter_riverpod/flutter_riverpod.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:url_launcher/url_launcher.dart';

class ChatScreen extends StatelessWidget {
  const ChatScreen({super.key});

  void _bukaFotoProfilPengguna(BuildContext context) {
    showDialog(
      context: context,
      builder: (BuildContext context) {
        return AlertDialog(
          title: const Text(
            'ByteBrain',
            textAlign: TextAlign.center,
          ),
          content: Image.asset(
            'assets/robot1.png',
          ),
          actions: [
            TextButton(
              onPressed: () {
                Navigator.pop(context);
              },
              child: const Text('Keluar'),
            ),
          ],
        );
      },
    );
  }

  @override

```

```

Widget build(BuildContext context) {
  // ignore: no_leading_underscores_for_local_identifiers
  final GlobalKey<ScaffoldState> _scaffoldKey = GlobalKey<ScaffoldState>();
  return Scaffold(
    key: _scaffoldKey,
    appBar: AppBar(
      automaticallyImplyLeading: false,
      backgroundColor: const Color.fromARGB(255, 0, 166, 126),
      actions: [
        IconButton(
          onPressed: () {
            _scaffoldKey.currentState!.openDrawer();
          },
          icon: const Icon(
            Icons.menu,
            color: Colors.white,
          ),
        ),
      ],
      title: Consumer(
        builder: (context, ref, child) {
          final isTyping = ref.watch(messagesProvider).isTyping;
          return Row(
            children: [
              GestureDetector(
                onTap: () {
                  _scaffoldKey.currentState!.openDrawer();
                },
                child: const CircleAvatar(
                  backgroundImage: AssetImage('assets/robot1.png'),
                  radius: 19,
                ),
              ),
              const SizedBox(width: 13),
            ],
            Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [

```

```

const Text(
    'ByteBrain ⚡',
    style: TextStyle(
        fontSize: 20,
        color: Colors.white,
    ),
),
const SizedBox(
    height: 2,
),
),
if (isTyping)
const Text(
    'Sedang mengetik...',
    style: TextStyle(
        fontSize: 14,
        color: Colors.white,
    ),
)
else
const Row(
    children: [
        Icon(
            Icons.brightness_1,
            color: Colors.greenAccent,
            size: 10,
        ),
        SizedBox(width: 5),
        Text(
            'Online',
            style: TextStyle(
                fontSize: 14,
                color: Colors.white,
            ),
        ),
    ],
),
],

```

```

),
],
);
},
),
),
drawer: Drawer(
child: ListView(
padding: EdgeInsets.zero,
children: <Widget>[
DrawerHeader(
decoration: const BoxDecoration(
color: Color.fromARGB(255, 0, 166, 126),
),
),
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
GestureDetector(
onTap: () {
_bukaFotoProfilPengguna(context);
}),
child: const Center(
child: CircleAvatar(
radius: 40,
backgroundImage: AssetImage('assets/robot1.png'),
),
),
),
),
),
const SizedBox(height: 10),
const Expanded(
child: Center(
child: Text(
'ByteBrain',
style: TextStyle(
color: Colors.white,
fontSize: 16,
fontWeight: FontWeight.bold,

```

```
        ),  
        ),  
        ),  
        ),  
        const SizedBox(  
          height: 5,  
        ),  
        const Expanded(  
          child: Center(  
            child: Text(  
              'Your personal AI assistant',  
              style: TextStyle(  
                color: Colors.white,  
                fontSize: 13,  
                fontStyle: FontStyle.italic),  
            ),  
          ),  
        ),  
      ],  
    ),  
  ),  
  ListTile(  
    leading: const Icon(  
      FontAwesomeIcons.message,  
      size: 20,  
    ),  
    title: const Text('Chat'),  
    onTap: () {  
      Navigator.pop(context);  
    },  
  ),  
  ListTile(  
    leading: const Icon(  
      FontAwesomeIcons.fileImage,  
      size: 20,  
    ),  
    title: const Text('Chat with Image'),
```

```

onTap: () {
  Navigator.pop(context);
  Navigator.of(context)
    .push(MaterialPageRoute(builder: (context) {
      return const ChatImage();
    }));
  },
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.filePdf,
    size: 20,
  ),
  title: const Text('Chat with PDF'),
  onTap: () async {
    const url = 'https://bytebrain-chatpdf.streamlit.app/';
    final uri = Uri.parse(url);
    if (await canLaunchUrl(uri)) {
      await launchUrl(uri);
    } else {
      throw 'Tidak bisa dijalankan $url';
    }
  },
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.circleQuestion,
    size: 20,
  ),
  title: const Text('Generate Quiz'),
  onTap: () {
    Navigator.pop(context);
    Navigator.of(context)
      .push(MaterialPageRoute(builder: (context) {
        return const FormScreen();
      }));
  },
),

```

```

),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.paperPlane,
    size: 20,
  ),
  title: const Text('Notes'),
  onTap: () {
    Navigator.pop(context);
    Navigator.of(context)
      .push(MaterialPageRoute(builder: (context) {
        return const HomeScreenNotes();
      }));
  },
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.dev,
    size: 20,
  ),
  title: const Text('About Dev'),
  onTap: () {
    Navigator.pop(context);
    Navigator.of(context)
      .push(MaterialPageRoute(builder: (context) {
        return const AboutPop();
      }));
  },
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.gamepad,
    size: 20,
  ),
  title: const Text('Education Games'),
  onTap: () {
    _showEducationGame(context);
  },
)

```

```

        },
    ),
    ListTile(
        leading: const Icon(
            FontAwesomeIcons.icons,
            size: 20,
        ),
        title: const Text('Connect with Us'),
        onTap: () {
            _showSocialMediaDialog(context);
        },
    ),
    ListTile(
        leading: const Icon(
            FontAwesomeIcons.personThroughWindow,
            size: 17,
        ),
        title: const Text('Exit'),
        onTap: () {
            _konfirmasiKeluar(context); // Panggil fungsi konfirmasi keluar
        },
    ),
    Padding(
        padding: const EdgeInsets.all(8.0),
        child: Image.asset(
            'assets/SMA Muhammadiyah Ambon.png',
            fit: BoxFit.cover,
        ),
    ),
],
),
),
),
body: Column(
    children: [
        Expanded(
            child: Padding(
                padding: const EdgeInsets.all(7.0),

```

```

child: Consumer(builder: (context, ref, child) {
    final messages = ref.watch(messagesProvider).messages;
    return ListView.builder(
        controller: ref.read(messagesProvider).scrollController,
        itemCount: messages.length,
        itemBuilder: (context, index) {
            return ChatBubble(message: messages[index]);
        });
}),
),
),
),
// TextField for writing message and button
const WriteMessageTextField(),
],
),
);
}
}

void _konfirmasiKeluar(BuildContext context) {
showDialog(
    context: context,
    builder: (BuildContext context) {
        return AlertDialog(
            title: const Text("Konfirmasi"),
            content: const Text("Apakah Anda yakin ingin keluar?"),
            actions: <Widget>[
                TextButton(
                    onPressed: () {
                        Navigator.of(context).pop(); // Tutup dialog
                },
                child: const Text("Batal"),
            ),
            TextButton(
                onPressed: () {
                    Navigator.of(context).pop(); // Tutup dialog
                    SystemNavigator.pop(); // Keluar dari aplikasi
                },
            ),
        );
    }
);
}

```

```

        child: const Text("Ya"),
    ),
],
);
},
);
}

void _showSocialMediaDialog(BuildContext context) {
    showDialog(
        context: context,
        builder: (context) {
            return AlertDialog(
                title: const Center(
                    child: Text('Connect with SMA Muhammadiyah Ambon',
                        textAlign: TextAlign.center),
                ),
                content: SingleChildScrollView(
                    child: Column(
                        children: [
                            _buildSocialMediaButton(
                                'WhatsApp',
                                'https://api.whatsapp.com/send?phone=+62 82290815896&text=Assalamualaikum',
                                FontAwesomeIcons.whatsapp,
                            ),
                            const SizedBox(
                                height: 10,
                            ),
                            _buildSocialMediaButton(
                                'Map',
                                'https://maps.app.goo.gl/hWQJ4TwThJ7CuYFi7',
                                FontAwesomeIcons.map),
                            const SizedBox(
                                height: 30,
                            ),
                            const Text(
                                'Connect with Chairil',
                            )
                        ],
                    ),
                )
            );
        }
    );
}

```

```

style: TextStyle(
  fontSize: 23,
),
),
),
const SizedBox(
height: 15,
),
_buildSocialMediaButton(
'WhatsApp',
'https://api.whatsapp.com/send?phone=+6282238482847&text=Hello,%20Chairil%20Ali',
FontAwesomeIcons.whatsapp,
),
const SizedBox(
height: 10,
),
_buildSocialMediaButton(
'Instagram',
'https://www.instagram.com/chairilali_13?igsh=MTYwczBkbG53c251cg==',
FontAwesomeIcons.instagram),
const SizedBox(
height: 30,
),
const Text(
'Connect with Samrah',
style: TextStyle(
fontSize: 23,
),
),
),
const SizedBox(
height: 15,
),
_buildSocialMediaButton(
'WhatsApp',
'https://api.whatsapp.com/send?phone=+6282213160067&text=Hello,%20Samrah',
FontAwesomeIcons.whatsapp,
),
const SizedBox(

```

```

        height: 10,
    ),
    _buildSocialMediaButton(
        'Instagram',
        'https://www.instagram.com/samrahdm?igsh=azloOXR2dG01ZzJy',
        FontAwesomeIcons.instagram),
    ],
),
),
),
actions: [
    TextButton(
        onPressed: () {
            Navigator.of(context).pop();
        },
        child: const Text('Keluar'),
    ),
],
);
},
);
}

```

```

// social media icon setting and lauch

Widget _buildSocialMediaButton(String label, String url, IconData icon) {
    return ElevatedButton(
        onPressed: () async {
            Uri uri = Uri.parse(url);
            if (!await launchUrl(uri)) {
                throw Exception('Tidak bisa dijalankan $url');
            }
        },
        child: Row(
            children: [
                FaIcon(icon),
                const SizedBox(width: 8),
                Text(label),
            ],

```

```

),
);
}

}

void _showEducationGame(BuildContext context) {
  showDialog(
    context: context,
    builder: (context) {
      return AlertDialog(
        title: const Text('Education Games'),
        content: SingleChildScrollView(
          child: Column(
            children: [
              _buildGameButton(
                'Math true or false',
                'https://chairil13.github.io/Math-game/',
                FontAwesomeIcons.calculator,
              ),
              _buildGameButton(
                'Memory Game',
                'https://chairil13.github.io/Memory-game/',
                FontAwesomeIcons.brain,
              ),
              _buildGameButton(
                "Rubik's Cube",
                'https://chairil13.github.io/rubic-cube/',
                FontAwesomeIcons.cube,
              ),
            ],
          ),
        ),
      );
    },
  );
}

actions: [
  TextButton(
    onPressed: () {
      Navigator.of(context).pop();
    },
    child: const Text('Keluar'),
  ),
]
}

```

```

        ),
        ],
        );
    },
    );
}

// Game icon setting and lauch

Widget _buildGameButton(String label, String url, IconData icon) {
    return ElevatedButton(
        onPressed: () async {
            Uri uri = Uri.parse(url);
            if (!await launchUrl(uri)) {
                throw Exception('Tidak bisa dijalankan $url');
            }
        },
        child: Row(
            children: [
                FaIcon(icon),
                const SizedBox(width: 8),
                Text(label),
            ],
        ),
    );
}
}

```

Chat with Image



Gambar 4. 14 chat with image

Tampilan diatas merupakan tampilan fitur *chat with image*. Di sini, pengguna dapat memilih untuk memulai percakapan dengan AI berbasis gambar dengan cara mengunggah gambar terlebih dahulu lalu menginput pesan, jika tidak maka akan muncul notifikasi agar mengunggah gambar terlebih dahulu, lalu ketika pesan terkirim akan ada animasi loading dan gambar akan di analisis dan direspon oleh AI. Untuk menyalin pesan AI user bisa klik *icon copy* pada balon obrolan atau menekan lama pesan AI, lalu untuk bernavigasi ke fitur atau menu lainnya user bisa menekan *icon menu* atau foto profile dari AI, maka menu lainnya akan ditampilkan. Berikut adalah kode program dari fitur *chat with image* tersebut :

```
import 'dart:io';

import 'package:bytebrain_project_akhir/presentation/about_dev_ui/about_dev.dart';
import 'package:bytebrain_project_akhir/presentation/chat_with_image_ui/api.dart';
import 'package:bytebrain_project_akhir/presentation/generate_quiz_ui/homeScreenQuiz.dart';
import 'package:bytebrain_project_akhir/presentation/home_ui/ui/chat_screen.dart';
import 'package:bytebrain_project_akhir/presentation/notes_ui/home_screen_notes.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
```

```

import 'package:flutter_markdown/flutter_markdown.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:google_generative_ai/google_generative_ai.dart';
import 'package:image_picker/image_picker.dart';
import 'package:lottie/lottie.dart';
import 'package:url_launcher/url_launcher.dart';

class ChatImage extends StatefulWidget {
  const ChatImage({super.key});

  @override
  State<ChatImage> createState() => _MainPageState();
}

class _MainPageState extends State<ChatImage> {
  final GlobalKey<ScaffoldState> _scaffoldKey = GlobalKey<ScaffoldState>();

  void _openDrawer() {
    _scaffoldKey.currentState?.openDrawer();
  }

  TextEditingController textEditingController = TextEditingController();
  String answer = "";
  XFile? image;
  bool isLoading = false;
  bool isAIResponding = false;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      key: _scaffoldKey,
      appBar: AppBar(
        title: Row(
          children: [
            GestureDetector(
              onTap: () {
                _scaffoldKey.currentState!.openDrawer();
              }
            )
          ],
        ),
      ),
    );
  }
}

```

```

},
child: const CircleAvatar(
  backgroundImage: AssetImage('assets/robot2.png'),
  radius: 19,
),
),
const SizedBox(width: 13),
Column(
  mainAxisAlignment: MainAxisAlignment.start,
  children: [
    const Text(
      'ByteBrain 🌟',
      style: TextStyle(
        fontSize: 20,
        color: Colors.white,
      ),
    ),
    const SizedBox(height: 2),
    Row(
      children: [
        if (!isAIResponding)
          const Icon(
            Icons.brightness_1,
            color: Colors.greenAccent,
            size: 10,
          ),
        if (!isAIResponding) const SizedBox(width: 5),
        Text(
          isAIResponding ? 'Sedang mengetik...' : 'Online',
          style: const TextStyle(
            fontSize: 14,
            color: Colors.white,
          ),
        ),
      ],
    ),
  ],
),
],

```

```

),
],
),
backgroundColor: const Color.fromARGB(255, 0, 166, 126),
automaticallyImplyLeading: false,
actions: [
 IconButton(
 icon: const Icon(Icons.menu),
 color: Colors.white,
 onPressed: _openDrawer,
 ),
],
centerTitle: true,
),
drawer: Drawer(
child: ListView(
padding: EdgeInsets.zero,
children: <Widget>[
 DrawerHeader(
decoration: const BoxDecoration(
color: Color.fromARGB(255, 0, 166, 126),
),
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
 GestureDetector(
onTap: () {
_bukaFotoProfilPengguna(context);
},
),
child: const Center(
child: CircleAvatar(
radius: 40,
backgroundImage: AssetImage('assets/robot2.png'),
),
),
),
),
const SizedBox(height: 10),

```

```
const Expanded(  
    child: Center(  
        child: Text(  
            'ByteBrain',  
            style: TextStyle(  
                color: Colors.white,  
                fontSize: 16,  
                fontWeight: FontWeight.bold  
        ),  
    ),  
),  
),  
const SizedBox(  
    height: 5,  
,  
),  
const Expanded(  
    child: Center(  
        child: Text(  
            'Your personal AI assistant',  
            style: TextStyle(  
                color: Colors.white,  
                fontSize: 13,  
                fontStyle: FontStyle.italic),  
            ),  
        ),  
    ),  
],  
,  
,  
),  
),  
ListTile(  
    leading: const Icon(  
        FontAwesomeIcons.message,  
        size: 20,  
,  
    ),  
    title: const Text('Chat'),  
    onTap: () {  
        Navigator.pop(context);  
    },
```

```

    Navigator.of(context)

        .push(MaterialPageRoute(builder: (context) {
            return const ChatScreen();
        }));
    },
),
ListTile(
    leading: const Icon(
        FontAwesomeIcons.fileImage,
        size: 20,
    ),
    title: const Text('Chat with Image'),
    onTap: () {
        Navigator.pop(context);
    },
),
ListTile(
    leading: const Icon(
        FontAwesomeIcons.filePdf,
        size: 20,
    ),
    title: const Text('Chat with PDF'),
    onTap: () async {
        const url = 'https://bytebrain-chatpdf.streamlit.app/';
        final uri = Uri.parse(url);
        if (await canLaunchUrl(uri)) {
            await launchUrl(uri);
        } else {
            throw 'Tidak bisa dijalankan $url';
        }
    },
),
ListTile(
    leading: const Icon(
        FontAwesomeIcons.circleQuestion,
        size: 20,
    ),
)

```

```

title: const Text('Generate Quiz'),
onTap: () {
  Navigator.pop(context);
  Navigator.of(context)
    .push(MaterialPageRoute(builder: (context) {
      return const FormScreen();
    }));
},
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.paperPlane,
    size: 20,
  ),
  title: const Text('Notes'),
  onTap: () {
    Navigator.pop(context);
    Navigator.of(context)
      .push(MaterialPageRoute(builder: (context) {
        return const HomeScreenNotes();
      }));
  },
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.dev,
    size: 20,
  ),
  title: const Text('About Dev'),
  onTap: () {
    Navigator.pop(context);
    Navigator.of(context)
      .push(MaterialPageRoute(builder: (context) {
        return const AboutPop();
      }));
  },
),

```

```

ListTile(
  leading: const Icon(
    FontAwesomeIcons.gamepad,
    size: 20,
  ),
  title: const Text('Education Games'),
  onTap: () {
    _showEducationGame(context);
  },
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.icons,
    size: 20,
  ),
  title: const Text('Connect with Us'),
  onTap: () {
    _showSocialMediaDialog(context);
  },
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.personThroughWindow,
    size: 17,
  ),
  title: const Text('Exit'),
  onTap: () {
    _konfirmasiKeluar(context);
  },
),
Padding(
  padding: const EdgeInsets.all(8.0),
  child: Image.asset(
    'assets/SMA Muhammadiyah Ambon.png',
    fit: BoxFit.cover,
  ),
),

```

```

        ],
      ),
    ),
  body: Stack(
    children: [
      Column(
        children: [
          Expanded(
            child: Padding(
              padding: const EdgeInsets.all(20),
            child: ListView(
              children: [
                const SizedBox(height: 20),
                Container(
                  width: double.infinity,
                  height: 200,
                  decoration: BoxDecoration(
                    color: image == null ? Colors.grey.shade200 : null,
                    borderRadius: BorderRadius.circular(10),
                    image: image != null
                      ? DecorationImage(
                          image: FileImage(File(image!.path)))
                      : null,
                ),
              ],
            ),
            const SizedBox(height: 20),
            if (answer.isNotEmpty) ...[
              Row(
                mainAxisAlignment: MainAxisAlignment.end,
                children: [
                  IconButton(
                    icon: const Icon(Icons.copy),
                    onPressed: () {
                      Clipboard.setData(
                        ClipboardData(text: answer));
                      ScaffoldMessenger.of(context).showSnackBar(
                        const SnackBar(

```

```

content:
    Text('Teks disalin ke clipboard'),
    backgroundColor:
        Color.fromARGB(255, 0, 166, 126),
),
);
},
),
],
),
MarkdownBody(
    data: answer,
    selectable: true,
),
],
),
),
),
),
Padding(
    padding: const EdgeInsets.symmetric(
        horizontal: 18.0, vertical: 15.0),
    child: TextField(
        controller: textEditingController,
        minLines: 1,
        maxLines: 3,
        decoration: InputDecoration(
            hintText: 'Tanyakan sesuatu..',
            hintStyle: const TextStyle(color: Colors.black45),
            filled: true,
            fillColor: const Color.fromARGB(255, 168, 212, 250),
            border: OutlineInputBorder(
                borderSide: BorderSide.none,
                gapPadding: 0,
                borderRadius: BorderRadius.circular(26.0),
),
            contentPadding: const EdgeInsets.symmetric(

```

```

        horizontal: 18.0, vertical: 8.0),
prefixIcon: IconButton(
icon: const Icon(Icons.image),
onPressed: () {
ImagePicker()
.pickImage(source: ImageSource.gallery)
.then(
(value) {
setState(() {
image = value;
});
},
);
},
),
suffixIcon: IconButton(
icon: const Icon(Icons.send),
onPressed: () {
if (image == null) {
ScaffoldMessenger.of(context).showSnackBar(
const SnackBar(
content:
Text('Harap pilih gambar terlebih dahulu'),
backgroundColor: Colors.red,
),
);
return;
}
}

setState(() {
isLoading = true;
isAIResponding = true;
answer = "";
});
}

GenerativeModel model = GenerativeModel(
model: 'gemini-1.5-flash', apiKey: apiKey);

```

```

model.generateContent([
    Content.multi([
        TextPart(textEditingController.text),
        DataPart('image/jpeg',
            File(image!.path).readAsBytesSync())
    ])
].then((value) => setState(() {
    answer = value.text.toString();
    isLoading = false;
    isAIResponding = false;
}));,
),
),
),
),
],
),
),
if (isLoading)
    Positioned.fill(
        child: Center(
            child: Lottie.asset(
                'assets/animation/loadingImage.json',
                width: 200,
                height: 200,
            ),
        ),
    ),
],
));
}

```

```

void _bukaFotoProfilPengguna(BuildContext context) {
    showDialog(
        context: context,
        builder: (BuildContext context) {

```

```

return AlertDialog(
  title: const Text(
    'ByteBrain',
    textAlign: TextAlign.center,
  ),
  content: Image.asset(
    'assets/robot2.png',
  ),
  actions: [
    TextButton(
      onPressed: () {
        Navigator.pop(context);
      },
      child: const Text('Keluar'),
    ),
  ],
);
}

```

```

void _showEducationGame(BuildContext context) {
  showDialog(
    context: context,
    builder: (context) {
      return AlertDialog(
        title: const Text('Education Games'),
        content: SingleChildScrollView(
          child: Column(
            children: [
              _buildGameButton(
                'Math true or false',
                'https://chairil13.github.io/Math-game/',
                FontAwesomeIcons.calculator,
              ),
              _buildGameButton(
                'Memory Game',
              ),
            ],
          ),
        ),
      );
    },
  );
}

```

```

        'https://chairil13.github.io/Memory-game/',
        FontAwesomeIcons.brain,
    ),
    _buildGameButton(
        "Rubik's Cube",
        'https://chairil13.github.io/rubic-cube/',
        FontAwesomeIcons(cube,
    ),
],
),
),
),
actions: [
    TextButton(
        onPressed: () {
            Navigator.of(context).pop();
        },
        child: const Text('Keluar'),
    ),
],
);
},
);
}

Widget _buildGameButton(String label, String url, IconData icon) {
    return ElevatedButton(
        onPressed: () async {
            Uri uri = Uri.parse(url);
            if (!await launchUrl(uri)) {
                throw Exception("Tidak bisa dijalankan $url");
            }
        },
        child: Row(
            children: [
                FaIcon(icon),
                const SizedBox(width: 8),
                Text(label),
            ],
        ),
    );
}

```

```

        ],
      ),
    );
}

void _showSocialMediaDialog(BuildContext context) {
  showDialog(
    context: context,
    builder: (context) {
      return AlertDialog(
        title: const Center(
          child: Text('Connect with SMA Muhammadiyah Ambon',
            textAlign: TextAlign.center),
        ),
        content: SingleChildScrollView(
          child: Column(
            children: [
              _buildSocialMediaButton(
                'WhatsApp',
                'https://api.whatsapp.com/send?phone=+6282238482847&text=Hello,%20Chairil%20Ali',
                FontAwesomeIcons.whatsapp,
              ),
              const SizedBox(
                height: 10,
              ),
              _buildSocialMediaButton('Instagram',
                'https://www.instagram.com/',
                FontAwesomeIcons.instagram),
              const SizedBox(
                height: 10,
              ),
              _buildSocialMediaButton('Website Sekolah',
                'http://www.google.com/',
                FontAwesomeIcons.globe),
              const SizedBox(
                height: 10,
              ),
              _buildSocialMediaButton(
                'Map',
              ),
            ],
          ),
        ),
      );
    },
  );
}

```

```

'https://maps.app.goo.gl/hWQJ4TwThJ7CuYFi7',
FontAwesomeIcons.map),
const SizedBox(
height: 30,
),
const Text(
'Connect with Chairil',
style: TextStyle(
fontSize: 23,
),
),
const SizedBox(
height: 15,
),
_buildSocialMediaButton(
'WhatsApp',
'https://api.whatsapp.com/send?phone=+6282238482847&text=Hello,%20Chairil%20Ali',
FontAwesomeIcons.whatsapp,
),
const SizedBox(
height: 10,
),
_buildSocialMediaButton(
'Instagram',
'https://www.instagram.com/chairilali_13?igsh=MTYwczBkbG53c251cg==',
FontAwesomeIcons.instagram),
const SizedBox(
height: 30,
),
const Text(
'Connect with Samrah',
style: TextStyle(
fontSize: 23,
),
),
const SizedBox(
height: 15,
)

```

```

),
        _buildSocialMediaButton(
            'WhatsApp',
            'https://api.whatsapp.com/send?phone=+6282213160067&text=Hello,%20Samrah',
            FontAwesomeIcons.whatsapp,
        ),
        const SizedBox(
            height: 10,
        ),
        _buildSocialMediaButton(
            'Instagram',
            'https://www.instagram.com/samrahdm?igsh=azloOXR2dG01ZzJy',
            FontAwesomeIcons.instagram),
    ],
),
),
),
actions: [
    TextButton(
        onPressed: () {
            Navigator.of(context).pop();
        },
        child: const Text('Keluar'),
    ),
],
);
),
);
}

```

```

Widget _buildSocialMediaButton(String label, String url, IconData icon) {
    return ElevatedButton(
        onPressed: () async {
            Uri uri = Uri.parse(url);
            if (!await launchUrl(uri)) {
                throw Exception("Tidak bisa dijalankan $url");
            }
        },
    );
}

```

```

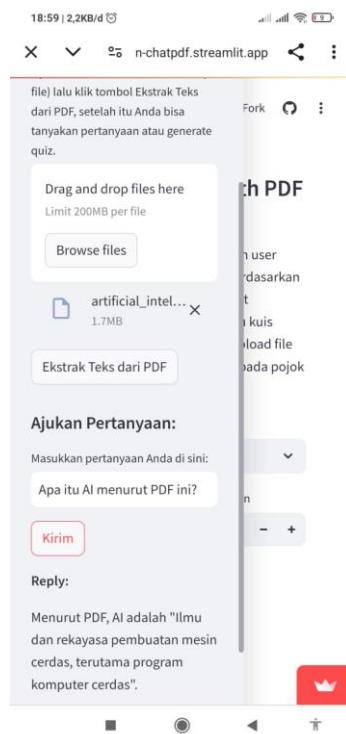
child: Row(
  children: [
    FaIcon(icon),
    const SizedBox(width: 8),
    Text(label),
  ],
),
);

}

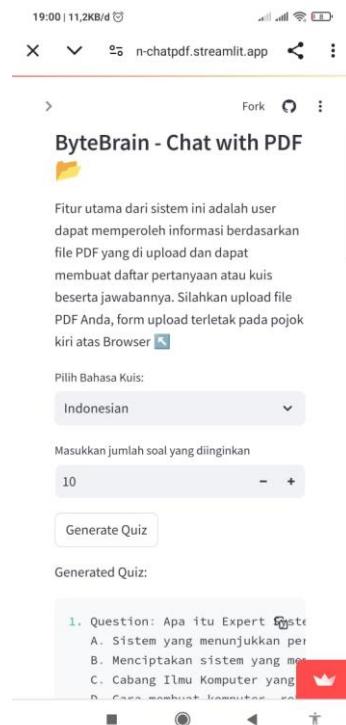
void _konfirmasiKeluar(BuildContext context) {
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(
        title: const Text("Konfirmasi"),
        content: const Text("Apakah Anda yakin ingin keluar?"),
        actions: <Widget>[
          TextButton(
            onPressed: () {
              Navigator.of(context).pop();
            },
            child: const Text("Batal"),
          ),
          TextButton(
            onPressed: () {
              Navigator.of(context).pop();
              SystemNavigator.pop();
            },
            child: const Text("Ya"),
          ),
        ],
    );
  });
}

```

Chat with PDF



Gambar 4. 15 chat with PDF untuk mencari informasi dari PDF



Gambar 4. 16 chat with PDF untuk generate quiz

Tampilan diatas merupakan tampilan fitur *chat with PDF*. Di sini, pengguna memiliki dua opsi yaitu dengan cara mengunggah PDF terlebih dahulu lalu menekan tombol ekstrak teks dari PDF selanjutnya *user* menginput pesan dan AI akan mencari jawaban berdasarkan file PDF yang diunggah, selanjutnya *user* bisa membuat kuis otomatis dengan cara memilih bahasa kuis, jumlah soal dan menekan tombol *generate quiz*, maka tampilan soal pilihan ganda akan dibuat oleh AI. User juga bisa menyalin respon AI dengan menekan *icon copy*. Berikut adalah kode program dari fitur *chat with image* tersebut :

```
import streamlit as st
from PyPDF2 import PdfReader
from langchain.text_splitter import RecursiveCharacterTextSplitter
import os
from langchain_google_genai import GoogleGenerativeAIEmbeddings
import google.generativeai as genai
from langchain.vectorstores import FAISS
from langchain_google_genai import ChatGoogleGenerativeAI
from langchain.chains.question_answering import load_qa_chain
from langchain.prompts import PromptTemplate
from dotenv import load_dotenv
import re

load_dotenv()
genai.configure(api_key=os.getenv("GOOGLE_API_KEY"))

def get_pdf_text(pdf_docs):
    text = ""
    for pdf in pdf_docs:
        pdf_reader = PdfReader(pdf)
        for page in pdf_reader.pages:
            text += page.extract_text()
    return text

def get_text_chunks(text):
    text_splitter = RecursiveCharacterTextSplitter(chunk_size=10000, chunk_overlap=1000)
    chunks = text_splitter.split_text(text)
    return chunks
```

```

def get_vector_store(text_chunks):
    embeddings = GoogleGenerativeAIEmbeddings(model="models/embedding-001")
    vector_store = FAISS.from_texts(text_chunks, embedding=embeddings)
    vector_store.save_local("faiss_index")

def get_conversational_chain():
    prompt_template = """
        Answer the question as detailed as possible from the provided context, make sure to provide all the details, if the answer is not in
        provided context just say, "Jawaban tidak tersedia dari konteks yang diberikan", don't provide the wrong answer\n\n
        Context:\n {context}?\n
        Question: \n{question}\n
    """

    Answer:
    """
    """

model = ChatGoogleGenerativeAI(model="gemini-pro", temperature=0.3)
prompt = PromptTemplate(template=prompt_template, input_variables=["context", "question"])
chain = load_qa_chain(model, chain_type="stuff", prompt=prompt)
return chain

def process_response(response_text):
    questions = response_text.strip().split("\n\n")
    formatted_questions = []

    for question in questions:
        lines = question.strip().split("\n")
        formatted_question = []
        for line in lines:
            if line.strip():
                formatted_question.append(line.strip())
        formatted_questions.append("\n".join(formatted_question))

    return "\n\n".join(formatted_questions)

def answer_question(user_question):
    embeddings = GoogleGenerativeAIEmbeddings(model="models/embedding-001")

```

```

new_db = FAISS.load_local("faiss_index", embeddings, allow_dangerous_deserialization=True)

docs = new_db.similarity_search(user_question)

chain = get_conversational_chain()

response = chain(
    {"input_documents": docs, "question": user_question},
    return_only_outputs=True
)

st.markdown(f"***Reply:**\n\n{response['output_text']}")

def format_quiz(raw_quiz):
    formatted_quiz = []
    questions = re.split(r'\d+\.', raw_quiz)[1:] # Split by question numbers

    for i, question in enumerate(questions, 1):
        lines = question.strip().split('\n')
        question_text = lines[0].strip()

        # Check if the question is a placeholder and skip it if so
        if question_text.startswith('[Generated Question'):
            continue

        options = []
        answer = ""

        for line in lines[1:]:
            line = line.strip()
            if line.startswith(('A.', 'B.', 'C.', 'D.')):
                options.append(line)
            elif 'Answer:' in line:
                answer = line

        formatted_question = f'{i}. Question: {question_text}\n'
        for option in options:
            formatted_question += f' {option}\n'

    return formatted_quiz

```

```

        formatted_question += f"\n{answer}\n"

    formatted_quiz.append(formatted_question)

return "\n".join(formatted_quiz)

def generate_quiz(context_language, quiz_language, num_questions):
    question_format = """
{i}. [Write a specific, relevant question here based on the context]
A. [Option A]
B. [Option B]
C. [Option C]
D. [Option D]

Answer: [Correct Answer]
"""

    questions_prompt = "\n\n".join([question_format.format(i=i+1) for i in range(num_questions)])

```

`prompt_template = f""`

Create a multiple-choice quiz in `{quiz_language}` based on the given context in `{context_language}`. Follow these rules strictly:

1. Ensure each question is relevant and specific to the context provided.
2. Provide 4 answer options (A, B, C, D) for each question.
3. Indicate the correct answer after the options.
4. Do not use placeholders like "[Write a specific, relevant question here based on the context]". Instead, write actual, context-specific questions.
5. The output format should be exactly as shown below, with each question on a new line and options labeled A, B, C, D:

Context:
~~~~~

Quiz:

```

{questions_prompt}
"""

```

```

model = ChatGoogleGenerativeAI(model="gemini-pro", temperature=0.7)
prompt = PromptTemplate(template=prompt_template, input_variables=["context"])
chain = load_qa_chain(model, chain_type="stuff", prompt=prompt)

```

# Get context from the uploaded PDF

```

embeddings = GoogleGenerativeAIEmbeddings(model="models/embedding-001")

new_db = FAISS.load_local("faiss_index", embeddings, allow_dangerous_deserialization=True)

docs = new_db.similarity_search("", k=5) # Get top 5 most relevant chunks

context = " ".join([doc.page_content for doc in docs])

# Generate quiz

response = chain(
    {"input_documents": docs, "question": prompt.format(context=context)},
    return_only_outputs=True
)

# Format the generated quiz

formatted_quiz = format_quiz(response["output_text"])

# Display the generated quiz

st.write("Generated Quiz:")

st.code(formatted_quiz)

def main():

    st.set_page_config(page_title="ByteBrain - Chat with PDF", page_icon="🤖")

    st.header("ByteBrain - Chat with PDF 📄")

    st.markdown("Fitur utama dari sistem ini adalah user dapat memperoleh informasi berdasarkan file PDF yang di upload dan dapat membuat daftar pertanyaan atau kuis beserta jawabannya. Silahkan upload file PDF Anda, form upload terletak pada pojok kiri atas Browser 🖱️", unsafe_allow_html=True)

    quiz_language = st.selectbox("Pilih Bahasa Kuis:", ["Indonesian", "English"])

    num_questions = st.number_input("Masukkan jumlah soal yang diinginkan", min_value=1, max_value=100, value=10)

    if st.button("Generate Quiz"):

        if os.path.exists("faiss_index"):

            with st.spinner("Generating quiz..."):

                generate_quiz("Indonesian", quiz_language, num_questions)

        else:

            st.warning("Tolong upload tekan tombol Ekstrak Teks dari PDF terlebih dahulu")

    with st.sidebar:

        st.title("Menu:")

```

```

pdf_docs = st.file_uploader("Upload file PDF anda (bisa beberapa file) lalu klik tombol Ekstrak Teks dari PDF, setelah itu Anda bisa tanyakan pertanyaan atau generate quiz.", accept_multiple_files=True)

if st.button("Ekstrak Teks dari PDF"):

    with st.spinner("Memuat..."):

        raw_text = get_pdf_text(pdf_docs)

        text_chunks = get_text_chunks(raw_text)

        get_vector_store(text_chunks)

        st.success("Selesai")

st.header("Ajukan Pertanyaan:")

user_question = st.text_input("Masukkan pertanyaan Anda di sini:")

if st.button("Kirim"):

    if user_question:

        if os.path.exists("faiss_index"):

            with st.spinner("Mencari jawaban..."):

                answer_question(user_question)

        else:

            st.warning("Tolong upload tekan tombol Ekstrak Teks dari PDF terlebih dahulu")

if __name__ == "__main__":
    main()

```

## Generate Quiz

The screenshot shows the ByteBrain mobile application interface. At the top, there is a green header bar with the time '19:22 | 3,6KB/d' and signal strength icons. The ByteBrain logo is in the center, with the text 'ByteBrain' and a yellow star icon. Below the logo, it says 'Online'. To the right of the logo is a three-line menu icon.

The main screen has a white background with a title 'Isi formulir' at the top. It contains four input fields:

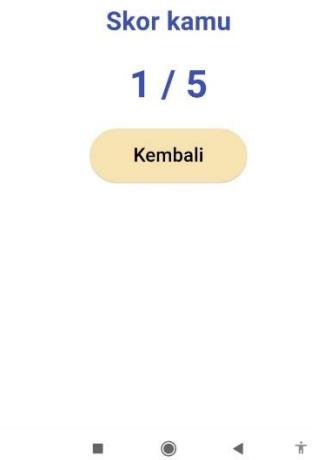
- 'Topik Quiz' field with the value 'Artificial Intelligence'.
- 'Pilih level' dropdown menu with the value 'Expert'.
- 'Jumlah pertanyaan' field with the value '5'.
- 'Pilih Bahasa' dropdown menu with the value 'Indonesian'.

At the bottom of the form is a large green button labeled 'Generate Quiz'.

Gambar 4. 17 Form Generate Quiz



Gambar 4. 18 Soal pilihan ganda



Gambar 4. 19 Hasil kuis

Tampilan diatas merupakan tampilan fitur *generate quiz*. Di sini, *user* menginput *form* yang tersedia, selanjutnya menekan tombol *generate quiz* maka akan ada loading dan muncul soal pilihan berganda. Setelah *user* menjawab semua soal maka muncul hasil kuis. Berikut adalah kode program dari fitur *generate quiz* tersebut :

```
import 'package:bytebrain_project_akhir/presentation/notes_ui/home_screen_notes.dart';
import 'package:bytebrain_project_akhir/presentation/about_dev_ui/about_dev.dart';
import 'package:bytebrain_project_akhir/presentation/chat_with_image_ui/chat_with_image.dart';
import 'package:bytebrain_project_akhir/presentation/home_ui/ui/chat_screen.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:url_launcher/url_launcher.dart';
import 'quizzScreen.dart';

class FormScreen extends StatefulWidget {
  const FormScreen({super.key});

  @override
```

```

// ignore: library_private_types_in_public_api

_FormScreenState createState() => _FormScreenState();

}

class _FormScreenState extends State<FormScreen> {

final GlobalKey<ScaffoldState> _scaffoldKey = GlobalKey<ScaffoldState>();

// ignore: unused_element
void _openDrawer() {
    _scaffoldKey.currentState?.openDrawer();
}

final _formKey = GlobalKey<FormState>();

String _subject = "";
String _level = 'Beginner';
int _numberOfQuestions = 1;
String _language = 'Indonesian';

@Override
Widget build(BuildContext context) {
    return Scaffold(
        key: _scaffoldKey,
        appBar: AppBar(
            automaticallyImplyLeading: false,
            backgroundColor: const Color.fromARGB(255, 0, 166, 126),
            actions: [
                IconButton(
                    onPressed: () {
                        _scaffoldKey.currentState!.openDrawer();
                    },
                    icon: const Icon(
                        Icons.menu,
                        color: Colors.white,
                    ),
                ),
            ],
            title: Row(

```

```

children: [
  GestureDetector(
    onTap: () {
      _scaffoldKey.currentState!.openDrawer();
    },
    child: const CircleAvatar(
      backgroundImage: AssetImage('assets/robot3.png'),
      radius: 19,
    ),
  ),
  const SizedBox(width: 13),
  const Column(
    mainAxisAlignment: MainAxisAlignment.start,
    children: [
      Text(
        'ByteBrain ✨',
        style: TextStyle(
          fontSize: 20,
          color: Colors.white,
        ),
      ),
      SizedBox(
        height: 2,
      ),
    ],
  ),
  Row(
    children: [
      Icon(Icons.brightness_1,
        color: Colors.greenAccent, size: 10),
      SizedBox(width: 5),
      Text(
        'Online',
        style: TextStyle(
          fontSize: 14,
          color: Colors.white,
        ),
      ),
    ],
  ],
],

```

```
        ),  
        ],  
        ),  
        ],  
        ),  
        ),  
        ),  
        drawer: Drawer(  
            child: ListView(  
                padding: EdgeInsets.zero,  
                children: <Widget>[  
                    DrawerHeader(  
                        decoration: const BoxDecoration(  
                            color: Color.fromARGB(255, 0, 166, 126),  
                        ),  
                        child: Column(  
                            mainAxisAlignment: MainAxisAlignment.start,  
                            children: [  
                                GestureDetector(  
                                    onTap: () {  
                                        _bukaFotoProfilPengguna(context);  
                                    },  
                                    child: const Center(  
                                        child: CircleAvatar(  
                                            radius: 40,  
                                            backgroundImage: AssetImage('assets/robot3.png'),  
                                        ),  
                                    ),  
                                ),  
                            ],  
                            const SizedBox(height: 10),  
                            const Expanded(  
                                child: Center(  
                                    child: Text(  
                                        'ByteBrain',  
                                        style: TextStyle(  
                                            color: Colors.white,  
                                            fontSize: 16,  
                                            fontWeight: FontWeight.bold,  
                                        ),  
                                    ),  
                                ),  
                            ),  
                        ),  
                    ),  
                ],  
            ),  
        ),  
    ),  
);
```

```
        ),  
        ),  
        ),  
    ),  
    const SizedBox(  
        height: 5,  
    ),  
    const Expanded(  
        child: Center(  
            child: Text(  
                'Your personal AI assistant',  
                style: TextStyle(  
                    color: Colors.white,  
                    fontSize: 13,  
                    fontStyle: FontStyle.italic),  
            ),  
        ),  
    ),  
    ],  
,  
),  
),  
ListTile(  
    leading: const Icon(  
        FontAwesomeIcons.message,  
        size: 20,  
    ),  
    title: const Text('Chat'),  
    onTap: () {  
        Navigator.pop(context);  
        Navigator.of(context)  
            .push(MaterialPageRoute(builder: (context) {  
                return const ChatScreen();  
            }));  
    },  
),  
ListTile(  
    leading: const Icon(  
        
```

```

FontAwesomeIcons.fileImage,
size: 20,
),
title: const Text('Chat with Image'),
onTap: () {
  Navigator.pop(context);
  Navigator.of(context)
    .push(MaterialPageRoute(builder: (context) {
      return const ChatImage();
    }));
},
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.filePdf,
    size: 20,
  ),
  title: const Text('Chat with PDF'),
  onTap: () async {
    const url = 'https://bytebrain-chatpdf.streamlit.app/';
    final uri = Uri.parse(url);
    if (await canLaunchUrl(uri)) {
      await launchUrl(uri);
    } else {
      throw 'Tidak bisa dijalankan $url';
    }
  },
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.circleQuestion,
    size: 20,
  ),
  title: const Text('Generate Quiz'),
  onTap: () {
    Navigator.pop(context);
  },
),

```

```

),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.paperPlane,
    size: 20,
  ),
  title: const Text('Notes'),
  onTap: () {
    Navigator.pop(context);
    Navigator.of(context)
      .push(MaterialPageRoute(builder: (context) {
        return const HomeScreenNotes();
      }));
  },
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.dev,
    size: 20,
  ),
  title: const Text('About Dev'),
  onTap: () {
    Navigator.pop(context);
    Navigator.of(context)
      .push(MaterialPageRoute(builder: (context) {
        return const AboutPop();
      }));
  },
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.gamepad,
    size: 20,
  ),
  title: const Text('Education Games'),
  onTap: () {
    _showEducationGame(context);
  },
)

```

```

        },
    ),
    ListTile(
        leading: const Icon(
            FontAwesomeIcons.icons,
            size: 20,
        ),
        title: const Text('Connect with Us'),
        onTap: () {
            _showSocialMediaDialog(context);
        },
    ),
    ListTile(
        leading: const Icon(
            FontAwesomeIcons.personThroughWindow,
            size: 17,
        ),
        title: const Text('Exit'),
        onTap: () {
            _konfirmasiKeluar(context);
        },
    ),
    Padding(
        padding: const EdgeInsets.all(8.0),
        child: Image.asset(
            'assets/SMA Muhammadiyah Ambon.png',
            fit: BoxFit.cover,
        ),
    ),
],
),
),
),
body: SafeArea(
    child: Column(
        children: [
            Expanded(
                child: SingleChildScrollView(

```

```

padding: const EdgeInsets.all(16.0),
child: Form(
key: _formKey,
child: Column(
children: [
const SizedBox(
height: 30,
),
const Center(
child: Text(
'Isi formulir',
style: TextStyle(
fontSize: 24,
fontWeight: FontWeight.bold,
),
),
),
),
),
const SizedBox(height: 30),
TextFormField(
decoration: const InputDecoration(
labelText: 'Topik Quiz',
labelStyle: TextStyle(color: Colors.grey),
border: OutlineInputBorder(),
),
),
onSaved: (value) {
_subject = value!;
},
validator: (value) {
if (value == null || value.isEmpty) {
return 'Silahkan Masukkan Topik';
}
return null;
},
),
const SizedBox(height: 20),
DropdownButtonFormField<String>(
decoration: const InputDecoration(

```

```

labelText: 'Pilih level',
labelStyle: TextStyle(color: Colors.grey),
border: OutlineInputBorder(),
),
onChanged: (value) {
  setState(() {
    _level = value!;
  });
},
items: [
'Beginner',
'Intermediate',
'Professional',
'Expert'
]
.map((level) => DropdownMenuItem(
  value: level,
  child: Text(level),
))
.toList(),
),
const SizedBox(height: 20),
TextField(
decoration: const InputDecoration(
labelText: 'Jumlah pertanyaan',
labelStyle: TextStyle(color: Colors.grey),
border: OutlineInputBorder(),
),
keyboardType: TextInputType.number,
onSaved: (value) {
  _numberOfQuestions = int.parse(value!);
},
validator: (value) {
  if (value == null || value.isEmpty) {
    return 'Silakan masukkan nomor';
  }
  return null;
}

```

```
    },  
    ),  
    const SizedBox(height: 20),  
    DropdownButtonFormField<String>(  
      value: _language,  
      decoration: const InputDecoration(  
        labelText: 'Pilih Bahasa',  
        labelStyle: TextStyle(color: Colors.grey),  
        border: OutlineInputBorder(),  
      ),  
      onChanged: (value) {  
        setState(() {  
          _language = value!;  
        });  
      },  
      items: [  
        'Indonesian',  
        'Arabic',  
        'Chinese',  
        'English',  
        'Français',  
        'German',  
        'Japanese',  
        'Korean',  
        'Malay',  
        'Spanish',  
      ]  
      .map((language) => DropdownMenuItem(  
        value: language,  
        child: Text(language),  
      ))  
      .toList(),  
    ),  
  ],  
,  
,
```

```
),
Padding(
padding: const EdgeInsets.all(15.0),
child: SizedBox(
width: double.infinity,
child: ElevatedButton(
 onPressed: () {
if (_formKey.currentState!.validate()) {
_formKey.currentState!.save();
Navigator.push(
context,
MaterialPageRoute(
builder: (context) => QuizzScreen(
subject: _subject,
level: _level,
numberOfQuestions: _numberOfQuestions,
language: _language,
),
),
),
);
}
),
style: ElevatedButton.styleFrom(
backgroundColor: const Color.fromARGB(255, 0, 166, 126),
),
),
child: const Padding(
padding: EdgeInsets.all(8.0),
child: Text(
'Generate Quiz',
style: TextStyle(fontSize: 25, color: Colors.white),
),
),
),
),
),
),
],
),
```

```

),
);
}
}

void _bukaFotoProfilPengguna(BuildContext context) {
showDialog(
context: context,
builder: (BuildContext context) {
return AlertDialog(
title: const Text(
'ByteBrain',
textAlign: TextAlign.center,
),
content: Image.asset(
'assets/robot3.png',
),
actions: [
TextButton(
 onPressed: () {
Navigator.pop(context);
},
child: const Text('Keluar'),
),
],
);
},
);
}

void _konfirmasiKeluar(BuildContext context) {
showDialog(
context: context,
builder: (BuildContext context) {
return AlertDialog(
title: const Text("Konfirmasi"),
content: const Text("Apakah Anda yakin ingin keluar?"),
)
);
}
}

```

```

actions: <Widget>[
    TextButton(
        onPressed: () {
            Navigator.of(context).pop();
        },
        child: const Text("Batal"),
    ),
    TextButton(
        onPressed: () {
            Navigator.of(context).pop();
            SystemNavigator.pop();
        },
        child: const Text("Ya"),
    ),
],
);
},
);
}

```

```

void _showSocialMediaDialog(BuildContext context) {
    showDialog(
        context: context,
        builder: (context) {
            return AlertDialog(
                title: const Center(
                    child: Text('Connect with SMA Muhammadiyah Ambon',
                        textAlign: TextAlign.center),
                ),
                content: SingleChildScrollView(
                    child: Column(
                        children: [
                            _buildSocialMediaButton(
                                'WhatsApp',
                                'https://api.whatsapp.com/send?phone=+62 82290815896&text=Assalamualaikum',
                                FontAwesomeIcons.whatsapp,
                            ),
                        ],
                    ),
                ),
            );
        }
    );
}

```

```

const SizedBox(
  height: 10,
),
_buildSocialMediaButton(
  'Map',
  'https://maps.app.goo.gl/hWQJ4TwThJ7CuYFi7',
  FontAwesomeIcons.map),
const SizedBox(
  height: 30,
),
const Text(
  'Connect with Chairil',
  style: TextStyle(
    fontSize: 23,
  ),
),
const SizedBox(
  height: 15,
),
_buildSocialMediaButton(
  'WhatsApp',
  'https://api.whatsapp.com/send?phone=+6282238482847&text=Hello,%20Chairil%20Ali',
  FontAwesomeIcons.whatsapp,
),
const SizedBox(
  height: 10,
),
_buildSocialMediaButton(
  'Instagram',
  'https://www.instagram.com/chairilali_13?igsh=MTYwczBkbG53c251cg==',
  FontAwesomeIcons.instagram),
const SizedBox(
  height: 30,
),
const Text(
  'Connect with Samrah',
  style: TextStyle(

```

```

        fontSize: 23,
    ),
),
),
const SizedBox(
height: 15,
),
_buildSocialMediaButton(
'WhatsApp',
'https://api.whatsapp.com/send?phone=+6282213160067&text=Hello,%20Samrah',
FontAwesomeIcons.whatsapp,
),
const SizedBox(
height: 10,
),
_buildSocialMediaButton(
'Instagram',
'https://www.instagram.com/samrahdm?igsh=azloOXR2dG01ZzJy',
FontAwesomeIcons.instagram),
],
),
),
),
actions: [
TextButton(
 onPressed: () {
Navigator.of(context).pop();
},
child: const Text('Keluar'),
),
],
);
},
);
}

// social media icon setting and lauch

Widget _buildSocialMediaButton(String label, String url, IconData icon) {
return ElevatedButton(

```

```

 onPressed: () async {
    Uri uri = Uri.parse(url);
    if (!await launchUrl(uri)) {
        throw Exception("Tidak bisa dijalankan $url");
    }
},
child: Row(
    children: [
        FaIcon(icon),
        const SizedBox(width: 8),
        Text(label),
    ],
),
);
}

```

```

void _showEducationGame(BuildContext context) {
    showDialog(
        context: context,
        builder: (context) {
            return AlertDialog(
                title: const Text('Education Games'),
                content: SingleChildScrollView(
                    child: Column(
                        children: [
                            _buildGameButton(
                                'Math true or false',
                                'https://chairil13.github.io/Math-game/',
                                FontAwesomeIcons.calculator,
                            ),
                            _buildGameButton(
                                'Memory Game',
                                'https://chairil13.github.io/Memory-game/',
                                FontAwesomeIcons.brain,
                            ),
                            _buildGameButton(
                                "Rubik's Cube",
                            ),
                        ],
                    ),
                ),
            );
        },
    );
}

```

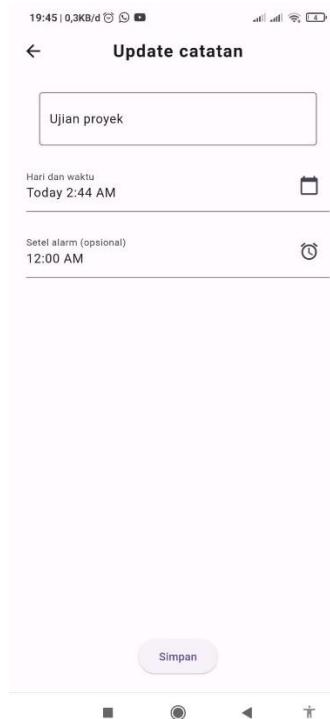
```

'https://chairil13.github.io/rubic-cube/',
FontAwesomeIcons.cube,
),
],
),
),
),
actions: [
TextButton(
 onPressed: () {
Navigator.of(context).pop();
},
child: const Text('Keluar'),
),
],
);
},
);
}

Widget _buildGameButton(String label, String url, IconData icon) {
return ElevatedButton(
onPressed: () async {
Uri uri = Uri.parse(url);
if (!await launchUrl(uri)) {
throw Exception("Tidak bisa dijalankan $url");
}
},
child: Row(
children: [
FaIcon(icon),
const SizedBox(width: 8),
Text(label),
],
),
);
}

```

## Notes



Gambar 4. 20 tambah catatan



Gambar 4. 21 lihat catatan

Tampilan diatas merupakan tampilan fitur *notes*. Di sini, *user* menginput *form* yang tersedia, selanjutnya menekan tombol simpan, maka catatan tersebut akan disimpan. *User* bisa lihat dan edit catatan dengan menekan *icon* pena. Selanjutnya user bisa mencari catatan dengan menekan *icon search*. Berikut adalah kode program dari fitur *notes* tersebut :

```
import 'package:animated_text_kit/animated_text_kit.dart';

import 'package:bytebrain_project_akhir/presentation/notes_ui/add_update_screen.dart';

import 'package:bytebrain_project_akhir/core/notes/db_handler.dart';

import 'package:bytebrain_project_akhir/core/notes/model.dart';

import 'package:bytebrain_project_akhir/presentation/notes_ui/task_search.dart';

import 'package:bytebrain_project_akhir/presentation/about_dev_ui/about_dev.dart';

import 'package:bytebrain_project_akhir/presentation/chat_with_image_ui/chat_with_image.dart';

import 'package:bytebrain_project_akhir/presentation/home_ui/ui/chat_screen.dart';

import 'package:bytebrain_project_akhir/presentation/generate_quiz_ui/homeScreenQuiz.dart';

import 'package:flutter/material.dart';

import 'package:flutter/services.dart';

import 'package:flutter_local_notifications/flutter_local_notifications.dart';

import 'package:font_awesome_flutter/font_awesome_flutter.dart';

import 'package:url_launcher/url_launcher.dart';

class HomeScreenNotes extends StatefulWidget {

const HomeScreenNotes({super.key});

@Override
State<HomeScreenNotes> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreenNotes> {
final GlobalKey<ScaffoldState> _scaffoldKey = GlobalKey<ScaffoldState>();

void _openDrawer() {
_scaffoldKey.currentState?.openDrawer();
}

DBHelper? dbHelper;
late Future<List<TodoModel>> dataList;
late FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin;
```

```

@Override
void initState() {
    super.initState();
    dbHelper = DBHelper();
    loadData();
    initializeNotifications();
}

void initializeNotifications() {
    const AndroidInitializationSettings initializationSettingsAndroid =
        AndroidInitializationSettings('app_icon');
    const InitializationSettings initializationSettings =
        InitializationSettings(android: initializationSettingsAndroid);

    flutterLocalNotificationsPlugin = FlutterLocalNotificationsPlugin();
    flutterLocalNotificationsPlugin.initialize(initializationSettings);
}

loadData() async {
    dataList = dbHelper!.getDataList();
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        key: _scaffoldKey,
        appBar: AppBar(
            backgroundColor: const Color.fromARGB(255, 0, 166, 126),
            automaticallyImplyLeading: false,
            actions: [
                IconButton(
                    icon: const Icon(Icons.search_outlined),
                    color: Colors.white,
                    iconSize: 30,
                    onPressed: () async {
                        TodoModel? result = await showSearch(

```

```

    // ignore: use_build_context_synchronously
    context: context,
    delegate: TaskSearch(dataList: await dataList),
);

if (result != null) {
    // ignore: avoid_print
    print('Pilih catatan: ${result.title}');
}

),
IconButton(
icon: const Icon(Icons.menu),
color: Colors.white,
onPressed: _openDrawer,
),
],
title: const Row(
children: [
Text(
'Catatan',
style: TextStyle(
fontSize: 24,
fontWeight: FontWeight.bold,
color: Colors.white,
),
),
SizedBox(
width: 10,
),
Icon(FontAwesomeIcons.paperPlane, color: Colors.white, size: 20),
],
),
),
),
drawer: Drawer(
child: ListView(
padding: EdgeInsets.zero,

```

```

children: <Widget>[

  const DrawerHeader(
    decoration: BoxDecoration(
      color: Color.fromARGB(255, 0, 166, 126),
    ),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Row(
          children: [
            Text(
              'ByteBrain 🤖',
              style: TextStyle(
                color: Colors.white,
                fontSize: 24,
                fontWeight: FontWeight.bold),
            ),
          ],
        ),
        SizedBox(
          height: 10,
        ),
        Text(
          'Binary Yet Technical Entity Brain',
          style: TextStyle(
            color: Colors.white,
            fontSize: 14,
          ),
        ),
        SizedBox(
          height: 1,
        ),
        Text(
          'Personal AI Assistant',
          style: TextStyle(
            color: Colors.white,
            fontSize: 14,
          ),
        )
      ],
    ),
  );

```

```

),
),
],
),
),
),
),
ListTile(
leading: const Icon(
FontAwesomeIcons.message,
size: 20,
),
title: const Text('Chat'),
onTap: () {
Navigator.pop(context);
Navigator.of(context)
.push(MaterialPageRoute(builder: (context) {
return const ChatScreen();
})));
},
),
ListTile(
leading: const Icon(
FontAwesomeIcons.fileImage,
size: 20,
),
title: const Text('Chat with Image'),
onTap: () {
Navigator.pop(context);
Navigator.of(context)
.push(MaterialPageRoute(builder: (context) {
return const ChatImage();
})));
},
),
),
ListTile(
leading: const Icon(
FontAwesomeIcons.filePdf,
size: 20,

```

```

),
title: const Text('Chat with PDF'),
onTap: () async {
  const url = 'https://bytebrain-chatpdf.streamlit.app/';
  final uri = Uri.parse(url);
  if (await canLaunchUrl(uri)) {
    await launchUrl(uri);
  } else {
    throw 'Tidak bisa dijalankan $url';
  }
},
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.circleQuestion,
    size: 20,
  ),
  title: const Text('Generate Quiz'),
  onTap: () {
    Navigator.pop(context);
    Navigator.of(context)
      .push(MaterialPageRoute(builder: (context) {
        return const FormScreen();
      }));
  },
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.paperPlane,
    size: 20,
  ),
  title: const Text('Notes'),
  onTap: () {
    Navigator.pop(context);
  },
),
ListTile(

```

```

leading: const Icon(
  FontAwesomeIcons.dev,
  size: 20,
),
title: const Text('About Dev'),
onTap: () {
  Navigator.pop(context);
  Navigator.of(context)
    .push(MaterialPageRoute(builder: (context) {
      return const AboutPop();
    }));
},
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.gamepad,
    size: 20,
  ),
  title: const Text('Education Games'),
  onTap: () {
    _showEducationGame(context);
  },
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.icons,
    size: 20,
  ),
  title: const Text('Connect with Us'),
  onTap: () {
    _showSocialMediaDialog(context);
  },
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.personThroughWindow,
    size: 17,

```

```

),
title: const Text('Exit'),
onTap: () {
  _konfirmasiKeluar(context);
},
),
const SizedBox(
height: 2,
),
Padding(
padding: const EdgeInsets.all(8.0),
child: Image.asset(
'assets/SMA Muhammadiyah Ambon.png',
fit: BoxFit.cover,
),
),
],
),
),
),
body: Column(
children: [
Expanded(
child: FutureBuilder<List<TodoModel>>(
future: dataList,
builder: (context, snapshot) {
if (snapshot.connectionState == ConnectionState.waiting) {
return const Center(
child: CircularProgressIndicator(),
);
} else if (snapshot.hasError) {
return const Center(
child: Text(
'Gagal memuat data',
style: TextStyle(
fontSize: 25,
fontWeight: FontWeight.bold,
color: Colors.black,

```

```

),
),
);
} else if (snapshot.data!.isEmpty) {
return Center(
child: AnimatedTextKit(
animatedTexts: [
TyperAnimatedText(
'Belum ada catatan...',  

speed: const Duration(milliseconds: 45),
textStyle: const TextStyle(
fontSize: 20,
color: Color.fromARGB(255, 128, 123, 123)),
textAlign: TextAlign.justify,
),
],
totalRepeatCount: 1,
displayFullTextOnTap: true,
),
);
} else {
return ListView.builder(
shrinkWrap: true,
itemCount: snapshot.data!.length,
itemBuilder: (context, index) {
int todoId = snapshot.data![index].id!.toInt();
String todoTitle = snapshot.data![index].title.toString();
String todoDT =
snapshot.data![index].dateandtime.toString();
return Dismissible(
key: ValueKey<int>(todoId),
direction: DismissDirection.endToStart,
background: Container(
color: Colors.redAccent,
child: const Icon(
Icons.delete_forever,
color: Colors.white,

```

```
),
),
onDismissed: (DismissDirection direction) {
  setState(() {
    dbHelper!.delete(todoId);
    dataList = dbHelper!.getDataList();
    snapshot.data!.remove(snapshot.data![index]);
  });
},
child: Container(
  margin: const EdgeInsets.all(5),
  decoration: const BoxDecoration(
    color: Colors.lightBlue,
    boxShadow: [
      BoxShadow(
        color: Colors.black12,
        blurRadius: 4,
        spreadRadius: 1,
      ),
    ],
  ),
  child: Column(
    children: [
      ListTile(
        contentPadding: const EdgeInsets.all(10),
        title: Padding(
          padding: const EdgeInsets.only(bottom: 10),
          child: Text(
            todoTitle,
            style: const TextStyle(fontSize: 19),
          ),
        ),
      ),
      const Divider(
        color: Colors.black,
        thickness: 0.8,
      ),
    ],
  ),
);
```

```
Padding(  
padding:  
  const EdgeInsets.symmetric(horizontal: 10),  
child: Row(  
  mainAxisAlignment:  
    MainAxisAlignment.spaceBetween,  
  children: [  
    Text(  
      todoDT,  
      style: const TextStyle(  
        fontSize: 14,  
        fontWeight: FontWeight.w500,  
        fontStyle: FontStyle.italic,  
      ),  
    ),  
    InkWell(  
      onTap: () {  
        Navigator.push(  
          context,  
          MaterialPageRoute(  
            builder: ((context) =>  
              AddUpdateTask(  
                todoId: todoId,  
                todoTitle: todoTitle,  
                todoDT: todoDT,  
                update: true,  
              )),  
            ),  
          );  
        },  
      ),  
    child: const Icon(  
      Icons.edit_note,  
      size: 28,  
    ),  
  ),  
],  
,
```

```

),
],
),
),
),
);
},
),
),
),
),
),
),
],
),
),
floatingActionButton: FloatingActionButton(
child: const Icon(Icons.add),
onPressed: () {
Navigator.push(
context,
MaterialPageRoute(
builder: (context) => AddUpdateTask(),
),
);
},
),
);
}
}

```

```

void _konfirmasiKeluar(BuildContext context) {
showDialog(
context: context,
builder: (BuildContext context) {
return AlertDialog(
title: const Text("Konfirmasi"),
content: const Text("Apakah Anda yakin ingin keluar?"),
actions: <Widget>[
TextButton(
 onPressed: () {

```

```

    Navigator.of(context).pop() // Tutup dialog
  },
  child: const Text("Batal"),
),
TextButton(
 onPressed: () {
  Navigator.of(context).pop() // Tutup dialog
  SystemNavigator.pop() // Keluar dari aplikasi
},
child: const Text("Ya"),
),
],
);
},
);
}

void _showSocialMediaDialog(BuildContext context) {
 showDialog(
 context: context,
 builder: (context) {
 return AlertDialog(
 title: const Center(
 child: Text('Connect with SMA Muhammadiyah Ambon'),
 textAlign: TextAlign.center),
),
content: SingleChildScrollView(
child: Column(
children: [
_buildSocialMediaButton(
'WhatsApp',
'https://api.whatsapp.com/send?phone=+62 82290815896&text=Assalamualaikum',
FontAwesomeIcons.whatsapp,
),
const SizedBox(
height: 10,
),
),

```

```

_buildSocialMediaButton(
    'Map',
    'https://maps.app.goo.gl/hWQJ4TwThJ7CuYFi7',
    FontAwesomeIcons.map),
const SizedBox(
    height: 30,
),
const Text(
    'Connect with Chairil',
    style: TextStyle(
        fontSize: 23,
    ),
),
const SizedBox(
    height: 15,
),
_buildSocialMediaButton(
    'WhatsApp',
    'https://api.whatsapp.com/send?phone=+6282238482847&text=Hello,%20Chairil%20Ali',
    FontAwesomeIcons.whatsapp,
),
const SizedBox(
    height: 10,
),
_buildSocialMediaButton(
    'Instagram',
    'https://www.instagram.com/chairilali_13?igsh=MTYwczBkbG53c251cg==',
    FontAwesomeIcons.instagram),
const SizedBox(
    height: 30,
),
const Text(
    'Connect with Samrah',
    style: TextStyle(
        fontSize: 23,
    ),
),

```

```

const SizedBox(
    height: 15,
),
_buildSocialMediaButton(
    'WhatsApp',
    'https://api.whatsapp.com/send?phone=+6282213160067&text=Hello,%20Samrah',
    FontAwesomeIcons.whatsapp,
),
const SizedBox(
    height: 10,
),
_buildSocialMediaButton(
    'Instagram',
    'https://www.instagram.com/samrahdm?igsh=azloOXR2dG01ZzJy',
    FontAwesomeIcons.instagram),
],
),
),
actions: [
    TextButton(
        onPressed: () {
            Navigator.of(context).pop();
        },
        child: const Text('Keluar'),
    ),
],
);
},
);
}

// social media icon setting and lauch
Widget _buildSocialMediaButton(String label, String url, IconData icon) {
    return ElevatedButton(
        onPressed: () async {
            Uri uri = Uri.parse(url);
            if (!await launchUrl(uri)) {

```

```

        throw Exception('Tidak bisa dijalankan $url');
    }
},
child: Row(
  children: [
    FaIcon(icon),
    const SizedBox(width: 8),
    Text(label),
  ],
),
);
}
}

```

```

void _showEducationGame(BuildContext context) {
  showDialog(
    context: context,
    builder: (context) {
      return AlertDialog(
        title: const Text('Education Games'),
        content: SingleChildScrollView(
          child: Column(
            children: [
              _buildGameButton(
                'Math true or false',
                'https://chairil13.github.io/Math-game/',
                FontAwesomeIcons.calculator,
              ),
              _buildGameButton(
                'Memory Game',
                'https://chairil13.github.io/Memory-game/',
                FontAwesomeIcons.brain,
              ),
              _buildGameButton(
                "Rubik's Cube",
                'https://chairil13.github.io/rubic-cube/',
                FontAwesomeIcons.cube,
              ),
            ],
          ),
        ),
      );
    },
  );
}

```

```

        ],
        ),
        ),
        actions: [
          TextButton(
            onPressed: () {
              Navigator.of(context).pop();
            },
            child: const Text('Keluar'),
          ),
        ],
      );
    },
  );
}

// Game icon setting and launch
Widget _buildGameButton(String label, String url, IconData icon) {
  return ElevatedButton(
    onPressed: () async {
      Uri uri = Uri.parse(url);
      if (!await launchUrl(uri)) {
        throw Exception('Tidak bisa dijalankan $url');
      }
    },
    child: Row(
      children: [
        FaIcon(icon),
        const SizedBox(width: 8),
        Text(label),
      ],
    ),
  );
}

```

## About Dev



Gambar 4. 22 about dev

Tampilan diatas merupakan tampilan *about dev*. Di sini, *user* melihat informasi tentang pengembang aplikasi dan melihat tutorial penggunaan aplikasi dengan cara menekan *icon youtube* pada pojok kiri bawah aplikasi. Berikut adalah kode program dari fitur *chat with image* tersebut :

```
import 'package:animated_text_kit/animated_text_kit.dart';
import 'package:bytebrain_project_akhir/presentation/notes_ui/home_screen_notes.dart';
import 'package:bytebrain_project_akhir/presentation/chat_with_image_ui/chat_with_image.dart';
import 'package:bytebrain_project_akhir/presentation/home_ui/ui/chat_screen.dart';
import 'package:bytebrain_project_akhir/presentation/generate_quiz_ui/homeScreenQuiz.dart';
import 'package:flutter/services.dart';
import 'package:flutter/material.dart';
import 'package:lottie/lottie.dart';
import 'package:url_launcher/url_launcher.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';

class AboutPop extends StatefulWidget {
```

```

@Override
// ignore: library_private_types_in_public_api
_NavigationPopState createState() => _NavigationPopState();
}

class _NavigationPopState extends State<AboutPop> {
final GlobalKey<ScaffoldState> _scaffoldKey = GlobalKey<ScaffoldState>();

void _openDrawer() {
    _scaffoldKey.currentState?.openDrawer();
}

bool isProfileExpanded = false;
bool isHovered = false;

@Override
Widget build(BuildContext context) {
    return Scaffold(
        key: _scaffoldKey,
        appBar: AppBar(
            backgroundColor: const Color.fromARGB(255, 0, 166, 126),
            automaticallyImplyLeading: false,
            actions: [
                IconButton(
                    icon: const Icon(Icons.menu),
                    color: Colors.white,
                    onPressed: _openDrawer,
                ),
            ],
            title: const Row(
                children: [
                    Text(
                        'About',
                        style: TextStyle(
                            fontSize: 24,
                            fontWeight: FontWeight.bold,

```

```

        color: Colors.white,
    ),
),
SizedBox(
    width: 10,
),
Icon(FontAwesomeIcons.dev, color: Colors.white, size: 25),
],
),
centerTitle: true,
),
drawer: Drawer(
child: ListView(
padding: EdgeInsets.zero,
children: <Widget>[
const DrawerHeader(
decoration: BoxDecoration(
color: Color.fromARGB(255, 0, 166, 126),
),
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Row(
children: [
Text(
'Chairil & Samrah',
style: TextStyle(
color: Colors.white,
fontSize: 24,
fontWeight: FontWeight.bold),
),
SizedBox(
width: 10,
),
Icon(
FontAwesomeIcons.laptopCode,
color: Colors.white,

```

```

),
],
),
SizedBox(
height: 10,
),
Text(
'Mobile Dev',
style: TextStyle(
color: Colors.white,
fontSize: 14,
),
),
SizedBox(
height: 1,
),
Text(
'Ambon, Maluku, Indonesia',
style: TextStyle(
color: Colors.white,
fontSize: 14,
),
),
),
],
),
),
),
ListTile(
leading: const Icon(
FontAwesomeIcons.message,
size: 20,
),
title: const Text('Chat'),
onTap: () {
Navigator.pop(context);
Navigator.of(context)
.push(MaterialPageRoute(builder: (context) {
return const ChatScreen();
}

```

```

    })));
    },
),
ListTile(
leading: const Icon(
FontAwesomeIcons.fileImage,
size: 20,
),
title: const Text('Chat with Image'),
onTap: () {
Navigator.pop(context);
Navigator.of(context)
.push(MaterialPageRoute(builder: (context) {
return const ChatImage();
})));
},
),
ListTile(
leading: const Icon(
FontAwesomeIcons.filePdf,
size: 20,
),
title: const Text('Chat with PDF'),
onTap: () async {
const url = 'https://bytebrain-chatpdf.streamlit.app/';
final uri = Uri.parse(url);
if (await canLaunchUrl(uri)) {
await launchUrl(uri);
} else {
throw 'Tidak bisa dijalankan $url';
}
},
),
ListTile(
leading: const Icon(
FontAwesomeIcons.circleQuestion,
size: 20,

```

```

),
title: const Text('Generate Quiz'),
onTap: () {
  Navigator.pop(context);
  Navigator.of(context)
    .push(MaterialPageRoute(builder: (context) {
      return const FormScreen();
    }));
},
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.paperPlane,
    size: 20,
  ),
  title: const Text('Notes'),
  onTap: () {
    Navigator.pop(context);
    Navigator.of(context)
      .push(MaterialPageRoute(builder: (context) {
        return const HomeScreenNotes();
      }));
},
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.dev,
    size: 20,
  ),
  title: const Text('About Dev'),
  onTap: () {
    Navigator.pop(context);
  },
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.gamepad,

```

```

        size: 20,
    ),
    title: const Text('Education Games'),
    onTap: () {
        _showEducationGame(context);
    },
),
ListTile(
    leading: const Icon(
        FontAwesomeIcons.icons,
        size: 20,
    ),
    title: const Text('Connect with Us'),
    onTap: () {
        _showSocialMediaDialog(context);
    },
),
ListTile(
    leading: const Icon(
        FontAwesomeIcons.personThroughWindow,
        size: 17,
    ),
    title: const Text('Exit'),
    onTap: () {
        _konfirmasiKeluar(context);
    },
),
const SizedBox(
    height: 2,
),
Padding(
    padding: const EdgeInsets.all(8.0),
    child: Image.asset(
        'assets/SMA Muhammadiyah Ambon.png',
        fit: BoxFit.cover,
    ),
),

```

```

        ],
      ),
    ),
  body: Padding(
    padding: const EdgeInsets.all(20),
    child: Stack(
      children: [
        SingleChildScrollView(
          child: Column(
            children: [
              MouseRegion(
                onEnter: (_) {
                  setState(() {
                    isHovered = true;
                  });
                },
                onExit: (_) {
                  setState(() {
                    isHovered = false;
                  });
                },
              ),
              GestureDetector(
                onTap: () {
                  setState(() {
                    isProfileExpanded = !isProfileExpanded;
                  });
                },
              ),
            ],
            child: Padding(
              padding: const EdgeInsets.only(top: 15),
              child: AnimatedContainer(
                duration: const Duration(milliseconds: 500),
                curve: Curves.easeInOut,
                width: isProfileExpanded ? 200.0 : 100.0,
                height: isProfileExpanded ? 200.0 : 100.0,
                decoration: BoxDecoration(
                  borderRadius: BorderRadius.circular(
                    isProfileExpanded ? 34.0 : 100.0),

```

```
border: Border.all(  
    color: Colors.blue,  
    width: 3,  
,  
,  
child: ClipRRect(  
    borderRadius: BorderRadius.circular(  
        isProfileExpanded ? 30.0 : 100.0,  
,  
    child: Image.asset(  
        'assets/profile.png',  
        width: isProfileExpanded ? 200.0 : 100.0,  
        height: isProfileExpanded ? 200.0 : 100.0,  
        fit: BoxFit.cover,  
        color: isHovered ? Colors.grey : null,  
        colorBlendMode: BlendMode.saturation,  
,  
,  
,  
,  
,  
,  
),  
const SizedBox(height: 10),  
const Row(  
    mainAxisAlignment: MainAxisAlignment.center,  
    children: [  
        Text(  
            'Chairil Ali,\nSamrah Daeng Makase',  
            style: TextStyle(  
                fontSize: 20,  
                letterSpacing: 1,  
                fontWeight: FontWeight.bold,  
,  
            textAlign: TextAlign.center,  
,  
    ],  
,
```

```

const SizedBox(
  height: 5,
),
const Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Row(
      children: [
        Text(
          'Mobile',
          style: TextStyle(
            fontSize: 15,
            letterSpacing: 1,
          ),
        ),
        SizedBox(
          width: 5,
        ),
        Icon(
          FontAwesomeIcons.dev,
          color: Color.fromARGB(255, 0, 166, 126),
        )
      ],
    ),
    ],
),
const SizedBox(
  height: 5,
),
const Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Text(
      'All about "widget"',
      style: TextStyle(
        fontSize: 15,
        letterSpacing: 1,
      ),
    )
  ],
)

```

```

        fontStyle: FontStyle.italic),
    ),
],
),
const SizedBox(
    height: 25,
),
Row(
    children: [
        Expanded(
            child: AnimatedTextKit(
                animatedTexts: [
                    TyperAnimatedText(
                        speed: const Duration(milliseconds: 45),
                        textStyle: const TextStyle(fontSize: 15),
                        textAlign: TextAlign.justify,
                    ),
                ],
                totalRepeatCount: 1,
                displayFullTextOnTap: true,
            ),
        )
    ],
),
),
),
Align(
    alignment: Alignment.bottomLeft,
    child: GestureDetector(
        onTap: () async {
            const url =
                'https://youtu.be/h1gQDAaTd44?si=5HFXjuykYqLczVdM';
            final uri = Uri.parse(url);
            if (await canLaunchUrl(uri)) {

```

```

        await launchUrl(uri);

    } else {
        throw 'Tidak bisa dijalankan $url';
    }
},
child: Lottie.asset(
    'assets/animation/youtube.json',
    height: 95,
    fit: BoxFit.cover,
),
),
),
),
],
),
),
);
}
}

```

```

void _konfirmasiKeluar(BuildContext context) {
    showDialog(
        context: context,
        builder: (BuildContext context) {
            return AlertDialog(
                title: const Text("Konfirmasi"),
                content: const Text("Apakah Anda yakin ingin keluar?"),
                actions: <Widget>[
                    TextButton(
                        onPressed: () {
                            Navigator.of(context).pop();
                        },
                        child: const Text("Batal"),
                    ),
                    TextButton(
                        onPressed: () {
                            Navigator.of(context).pop();
                            SystemNavigator.pop();
                        },
                    ),
                ],
            );
        }
    );
}

```

```

        child: const Text("Ya"),
    ),
],
);
},
);
}

void _showSocialMediaDialog(BuildContext context) {
    showDialog(
        context: context,
        builder: (context) {
            return AlertDialog(
                title: const Center(
                    child: Text('Connect with SMA Muhammadiyah Ambon',
                        textAlign: TextAlign.center),
                ),
                content: SingleChildScrollView(
                    child: Column(
                        children: [
                            _buildSocialMediaButton(
                                'WhatsApp',
                                'https://api.whatsapp.com/send?phone=+62 82290815896&text=Assalamualaikum',
                                FontAwesomeIcons.whatsapp,
                            ),
                            const SizedBox(
                                height: 10,
                            ),
                            _buildSocialMediaButton(
                                'Map',
                                'https://maps.app.goo.gl/hWQJ4TwThJ7CuYFi7',
                                FontAwesomeIcons.map),
                            const SizedBox(
                                height: 30,
                            ),
                            const Text(
                                'Connect with Chairil',
                            )
                        ],
                    ),
                )
            );
        }
    );
}

```

```

style: TextStyle(
    fontSize: 23,
),
),
),
const SizedBox(
height: 15,
),
),
_buildSocialMediaButton(
'WhatsApp',
'https://api.whatsapp.com/send?phone=+6282238482847&text=Hello,%20Chairil%20Ali',
FontAwesomeIcons.whatsapp,
),
const SizedBox(
height: 10,
),
),
_buildSocialMediaButton(
'Instagram',
'https://www.instagram.com/chairilali_13?igsh=MTYwczBkbG53c251cg==',
FontAwesomeIcons.instagram),
const SizedBox(
height: 30,
),
),
const Text(
'Connect with Samrah',
style: TextStyle(
fontSize: 23,
),
),
),
const SizedBox(
height: 15,
),
),
_buildSocialMediaButton(
'WhatsApp',
'https://api.whatsapp.com/send?phone=+6282213160067&text=Hello,%20Samrah',
FontAwesomeIcons.whatsapp,
),
),
const SizedBox(

```

```

height: 10,
),
_buildSocialMediaButton(
'Instagram',
'https://www.instagram.com/samrahdm?igsh=azloOXR2dG01ZzJy',
FontAwesomeIcons.instagram),
],
),
),
actions: [
TextButton(
 onPressed: () {
Navigator.of(context).pop();
},
child: const Text('Keluar'),
),
],
);
},
);
}

void _showEducationGame(BuildContext context) {
 showDialog(
context: context,
builder: (context) {
return AlertDialog(
title: const Text('Education Games'),
content: SingleChildScrollView(
child: Column(
children: [
_buildGameButton(
'Math true or false',
'https://chairil13.github.io/Math-game/',
FontAwesomeIcons.calculator,
),
_buildGameButton(

```

```

'Memory Game',
'https://chairil13.github.io/Memory-game/',
FontAwesomeIcons.brain,
),
),
_buildGameButton(
"Rubik's Cube",
'https://chairil13.github.io/rubic-cube/',
FontAwesomeIcons.cube,
),
],
),
),
),
actions: [
TextButton(
 onPressed: () {
Navigator.of(context).pop();
},
child: const Text('Keluar'),
),
],
);
},
);
}

```

```

Widget _buildGameButton(String label, String url, IconData icon) {
return ElevatedButton(
 onPressed: () async {
Uri uri = Uri.parse(url);
if (!await launchUrl(uri)) {
throw Exception('Tidak bisa dijalankan $url');
}
},
child: Row(
children: [
FaIcon(icon),
const SizedBox(width: 8),

```

```

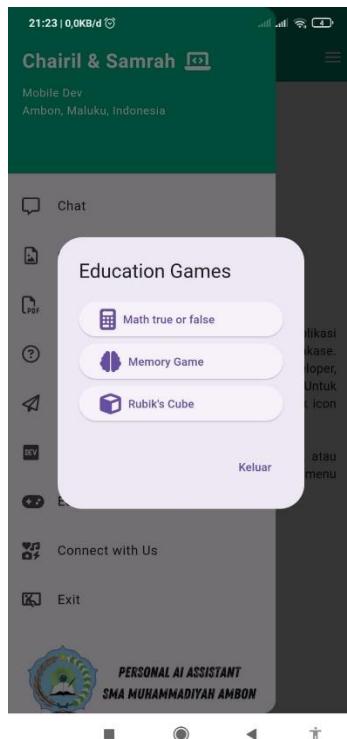
        Text(label),
    ],
),
);
}

// social media icon setting and lauch

Widget _buildSocialMediaButton(String label, String url, IconData icon) {
    return ElevatedButton(
        onPressed: () async {
            Uri uri = Uri.parse(url);
            if (!await launchUrl(uri)) {
                throw Exception('Tidak bisa dijalankan $url');
            }
        },
        child: Row(
            children: [
                FaIcon(icon),
                const SizedBox(width: 8),
                Text(label),
            ],
        ),
    );
}
}

```

## Education Games



Gambar 4. 23 education games

Tampilan diatas merupakan tampilan *education games*. Di sini, *user* bisa memainkan game edukasi yang telah disediakan seperti *math true or false*, *memory game*, dan *rubik's cube* aplikasi dengan cara menekan *icon youtube* pada pojok kiri bawah aplikasi. Kode program dari tampilan ini sudah termasuk di dalam kode program fitur-fitur lain dikarenakan fitur *education games* ini berbasis web dan tidak punya logika tersendiri di file terpisah jadi peneliti menempatkan tautan game nya di tiap fitur. Berikut adalah potongan kode dari tampilan *education games* tersebut:

```
ListTile(  
  leading: const Icon(  
    FontAwesomeIcons.gamepad,  
    size: 20,  
  ),  
  title: const Text('Education Games'),  
  onTap: () {  
    _showEducationGame(context);  
  },
```

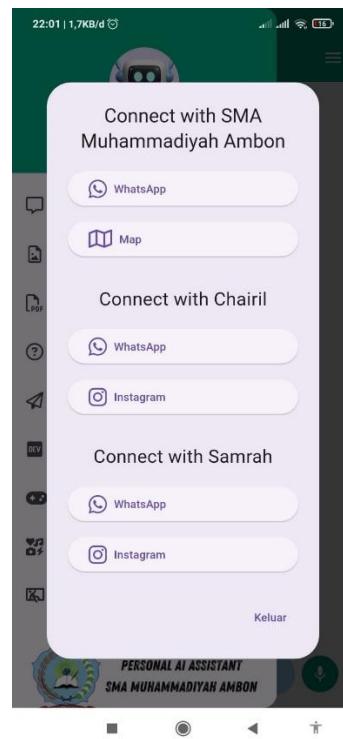
```

),
void _showEducationGame(BuildContext context) {
  showDialog(
    context: context,
    builder: (context) {
      return AlertDialog(
        title: const Text('Education Games'),
        content: SingleChildScrollView(
          child: Column(
            children: [
              _buildGameButton(
                'Math true or false',
                'https://chairil13.github.io/Math-game/',
                FontAwesomeIcons.calculator,
              ),
              _buildGameButton(
                'Memory Game',
                'https://chairil13.github.io/Memory-game/',
                FontAwesomeIcons.brain,
              ),
              _buildGameButton(
                "Rubik's Cube",
                'https://chairil13.github.io/rubic-cube/',
                FontAwesomeIcons.cube,
              ),
            ],
          ),
        ),
        actions: [
          TextButton(
            onPressed: () {
              Navigator.of(context).pop();
            },
            child: const Text('Keluar'),
          ),
        ],
      );
    );
}

```

```
},  
);  
}  
}
```

## Connect with Us



Gambar 4. 24 connect with us

Tampilan diatas merupakan tampilan *connect with us*. Di sini, *user* bisa mengunjungi info kontak dan lokasi dari sekolah maupun tim pengembang. Kode program dari tampilan ini sudah termasuk di dalam kode program fitur-fitur lain dikarenakan fitur ini memiliki tautan dan tidak mempunyai logika tersendiri di file terpisah jadi peneliti menempatkan tautan kontak nya di tiap fitur. Berikut adalah potongan kode dari tampilan *connect with us* tersebut:

```
ListTile(  
  leading: const Icon(  
    FontAwesomeIcons.icons,  
    size: 20,  
  ),  
  title: const Text('Connect with Us'),
```

```

onTap: () {
    _showSocialMediaDialog(context);
},
),
),

void _showSocialMediaDialog(BuildContext context) {
    showDialog(
        context: context,
        builder: (context) {
            return AlertDialog(
                title: const Center(
                    child: Text('Connect with SMA Muhammadiyah Ambon',
                        textAlign: TextAlign.center),
                ),
                content: SingleChildScrollView(
                    child: Column(
                        children: [
                            _buildSocialMediaButton(
                                'WhatsApp',
                                'https://api.whatsapp.com/send?phone=+62 82290815896&text=Assalamualaikum',
                                FontAwesomeIcons.whatsapp,
                            ),
                            const SizedBox(
                                height: 10,
                            ),
                            _buildSocialMediaButton(
                                'Map',
                                'https://maps.app.goo.gl/hWQJ4TwThJ7CuYFi7',
                                FontAwesomeIcons.map),
                            const SizedBox(
                                height: 30,
                            ),
                            const Text(
                                'Connect with Chairil',
                                style: TextStyle(
                                    fontSize: 23,
                                ),
                            ),
                        ],
                    ),
                ),
            );
        }
    );
}

```

```

),
const SizedBox(
  height: 15,
),
_buildSocialMediaButton(
  'WhatsApp',
  'https://api.whatsapp.com/send?phone=+6282238482847&text=Hello,%20Chairil%20Ali',
  FontAwesomeIcons.whatsapp,
),
const SizedBox(
  height: 10,
),
_buildSocialMediaButton(
  'Instagram',
  'https://www.instagram.com/chairilali_13?igsh=MTYwczBkbG53c251cg==',
  FontAwesomeIcons.instagram),
const SizedBox(
  height: 30,
),
const Text(
  'Connect with Samrah',
  style: TextStyle(
    fontSize: 23,
  ),
),
const SizedBox(
  height: 15,
),
_buildSocialMediaButton(
  'WhatsApp',
  'https://api.whatsapp.com/send?phone=+6282213160067&text=Hello,%20Samrah',
  FontAwesomeIcons.whatsapp,
),
const SizedBox(
  height: 10,
),
_buildSocialMediaButton(

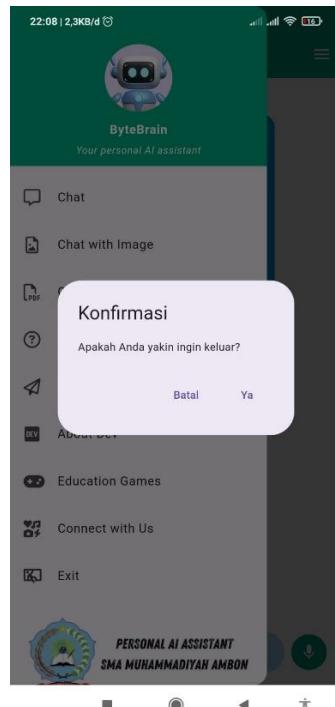
```

```

'Instagram',
'https://www.instagram.com/samrahdm?igsh=azloOXR2dG01ZzJy',
FontAwesomeIcons.instagram),
],
),
),
actions: [
TextButton(
 onPressed: () {
Navigator.of(context).pop();
},
child: const Text('Keluar'),
),
],
);
},
);
}
}

```

## Exit



Gambar 4. 25 exit

Tampilan diatas merupakan tampilan *exit*. Di sini, *user* bisa memilih antara keluar dari aplikasi atau tetap didalam aplikasi. Kode program dari tampilan ini sebenarnya sudah termasuk di dalam kode program fitur-fitur lain dikarenakan fitur ini tidak memiliki logika tersendiri di file terpisah jadi peneliti menempatkan kode programmnya di tiap fitur. Berikut adalah potongan kode dari tampilan *exit* tersebut:

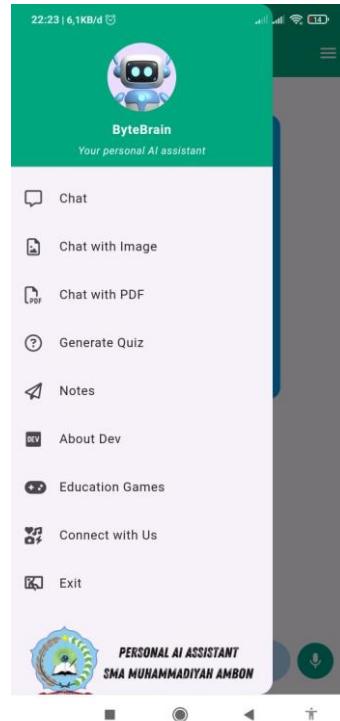
```
ListTile(  
    leading: const Icon(  
        FontAwesomeIcons.personThroughWindow,  
        size: 17,  
    ),  
    title: const Text('Exit'),  
    onTap: () {  
        _konfirmasiKeluar(context);  
    },  
),  
  
void _konfirmasiKeluar(BuildContext context) {  
    showDialog(  
        context: context,  
        builder: (BuildContext context) {  
            return AlertDialog(  
                title: const Text("Konfirmasi"),  
                content: const Text("Apakah Anda yakin ingin keluar?"),  
                actions: <Widget>[  
                    TextButton(  
                        onPressed: () {  
                            Navigator.of(context).pop();  
                        },  
                        child: const Text("Batal"),  
                    ),  
                    TextButton(  
                        onPressed: () {  
                            Navigator.of(context).pop();  
                            SystemNavigator.pop();  
                        },  
                        child: const Text("Ya"),  
                    ),  
                ],  
            );  
        },  
    );  
}
```

```

    ],
);
},
);
}
}

```

## Navigation



Gambar 4. 26 navigation

Tampilan diatas merupakan tampilan *navigation*. Di sini, *user* bisa memilih fitur atau menu mana yang mau dikunjungi. Kode program dari tampilan ini sebenarnya sudah termasuk di dalam kode program fitur-fitur lain dikarenakan fitur ini tidak memiliki logika tersendiri di file terpisah jadi peneliti menempatkan kode programmnya di tiap fitur. Berikut adalah potongan kode dari tampilan *navigation* tersebut:

```

drawer: Drawer(
  child: ListView(
    padding: EdgeInsets.zero,
    children: <Widget>[
      const DrawerHeader(

```

```
decoration: BoxDecoration(  
    color: Color.fromARGB(255, 0, 166, 126),  
,  
child: Column(  
    crossAxisAlignment: CrossAxisAlignment.start,  
    children: [  
        Row(  
            children: [  
                Text(  
                    'Chairil & Samrah',  
                    style: TextStyle(  
                        color: Colors.white,  
                        fontSize: 24,  
                        fontWeight: FontWeight.bold),  
,  
                SizedBox(  
                    width: 10,  
,  
                Icon(  
                    FontAwesomeIcons.laptopCode,  
                    color: Colors.white,  
,  
                ),  
            ],  
,  
        ),  
        SizedBox(  
            height: 10,  
,  
    Text(  
        'Mobile Dev',  
        style: TextStyle(  
            color: Colors.white,  
            fontSize: 14,  
,  
        ),  
        ),  
        SizedBox(  
            height: 1,  
,  
    ),
```

```

Text(
  'Ambon, Maluku, Indonesia',
  style: TextStyle(
    color: Colors.white,
    fontSize: 14,
  ),
),
],
),
),
),
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.message,
    size: 20,
  ),
  title: const Text('Chat'),
  onTap: () {
    Navigator.pop(context);
    Navigator.of(context)
      .push(MaterialPageRoute(builder: (context) {
        return const ChatScreen();
      }));
  },
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.fileImage,
    size: 20,
  ),
  title: const Text('Chat with Image'),
  onTap: () {
    Navigator.pop(context);
    Navigator.of(context)
      .push(MaterialPageRoute(builder: (context) {
        return const ChatImage();
      }));
  },
),

```

```

),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.filePdf,
    size: 20,
  ),
  title: const Text('Chat with PDF'),
  onTap: () async {
    const url = 'https://bytebrain-chatpdf.streamlit.app/';
    final uri = Uri.parse(url);
    if (await canLaunchUrl(uri)) {
      await launchUrl(uri);
    } else {
      throw 'Tidak bisa dijalankan $url';
    }
  },
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.circleQuestion,
    size: 20,
  ),
  title: const Text('Generate Quiz'),
  onTap: () {
    Navigator.pop(context);
    Navigator.of(context)
      .push(MaterialPageRoute(builder: (context) {
        return const FormScreen();
      }));
  },
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.paperPlane,
    size: 20,
  ),
  title: const Text('Notes'),

```

```

onTap: () {
  Navigator.pop(context);
  Navigator.of(context)
    .push(MaterialPageRoute(builder: (context) {
      return const HomeScreenNotes();
    }));
},
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.dev,
    size: 20,
  ),
  title: const Text('About Dev'),
  onTap: () {
    Navigator.pop(context);
  },
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.gamepad,
    size: 20,
  ),
  title: const Text('Education Games'),
  onTap: () {
    _showEducationGame(context);
  },
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.icons,
    size: 20,
  ),
  title: const Text('Connect with Us'),
  onTap: () {
    _showSocialMediaDialog(context);
  },
),

```

```
),
ListTile(
  leading: const Icon(
    FontAwesomeIcons.personThroughWindow,
    size: 17,
  ),
  title: const Text('Exit'),
  onTap: () {
    _konfirmasiKeluar(context);
  },
),
const SizedBox(
  height: 2,
),
Padding(
  padding: const EdgeInsets.all(8.0),
  child: Image.asset(
    'assets/SMA Muhammadiyah Ambon.png',
    fit: BoxFit.cover,
  ),
),
],
),
),
```

## PENGUJIAN

### 5.1 METODE PENGUJIAN PERANGKAT LUNAK

*BlackBox Testing* adalah metode pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas aplikasi tanpa memperhatikan struktur internal atau kode sumbernya. Pengujinya hanya mengetahui input yang diberikan dan output yang dihasilkan, tanpa mengetahui bagaimana proses internal bekerja. Peneliti menggunakan metode ini karena ingin memastikan bahwa aplikasi AI berbasis NLP dan *game* edukasi berfungsi sesuai dengan yang diharapkan dari sudut pandang pengguna akhir. Dengan metode ini, peneliti dapat menguji semua fungsionalitas sistem. Pengujian ini mencakup verifikasi bahwa setiap fitur merespons input pengguna dengan benar dan menghasilkan output yang diinginkan, sehingga aplikasi dapat berfungsi dengan baik dan memberikan pengalaman yang memuaskan bagi pengguna.

### 5.2 HASIL PENGUJIAN

Tabel 1. 1 Pengujian BlackBox Testing

| No | Fitur      | Test Case                              | Input                                                                                                                           | Proses                                                                        | Hasil yang diharapkan                                                                               | Keterangan |
|----|------------|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|------------|
| 1  | Navigation | Mengakses salah satu fitur yang ada.   | Menekan tombol icon menu atau profile aplikasi lalu tekan salah satu icon fitur, misalnya about dev.                            | Aplikasi memproses dan mengarahkan ke menu fitur yang dituju yaitu about dev. | Aplikasi berhasil mengarahkan ke menu fitur yang dituju yaitu about dev.                            | Sesuai     |
| 2  | Navigation | Kembali ke fitur yang saat ini dibuka. | Menekan tombol icon menu atau profile aplikasi lalu tekan icon fitur yang saat ini dibuka atau aktif, misalnya chat with image. | Aplikasi memproses dan mengarahkan ke tujuan.                                 | Aplikasi berhasil tetap di menu fitur yang saat ini sedang dibuka atau aktif yaitu chat with image. | Sesuai     |
| 3  | Navigation | Melihat foto                           | Menekan tombol icon                                                                                                             | Aplikasi memproses                                                            | Aplikasi memperlihatkan                                                                             | Sesuai     |

|   |                |                                                  |                                                                                                           |                                                                                                               |                                                                                                                    |        |
|---|----------------|--------------------------------------------------|-----------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|--------|
|   |                | profile aplikasi                                 | menu atau profile aplikasi lalu tekan foto profile aplikasi.                                              | dan mengarahkan ke tujuan.                                                                                    | an keseluruhan foto profile dari aplikasi.                                                                         |        |
| 4 | Navigation     | Keluar dari foto profile aplikasi.               | Menekan tombol icon menu atau profile aplikasi lalu tekan foto profile aplikasi lalu tekan tombol keluar. | Aplikasi merespon dan mengarahkan ke tujuan.                                                                  | Berhasil kembali ke menu sebelumnya.                                                                               | Sesuai |
| 5 | Chat with Text | Mengirim pesan teks dan menerima respon AI.      | Menginput "Siapakah anda?"                                                                                | AI memproses input dan menghasilkan respon.                                                                   | AI merespon dan memperkenalkan dirinya dengan nama ByteBrain dan merupakan AI asisten dari SMA Muhammadiyah Ambon. | Sesuai |
| 6 | Chat with Text | Mengirim inputan suara dan menerima respon AI.   | Menekan <i>icon mic</i> lalu bertanya: "Jelaskan apa itu Flutter?"                                        | AI memproses input dan menghasilkan respon.                                                                   | AI memberikan penjelasan tentang Flutter                                                                           | Sesuai |
| 7 | Chat with Text | Menyalin respon jawaban AI.                      | Menekan <i>icon copy</i>                                                                                  | Aplikasi mengambil teks dan menyimpannya di <i>clipboard</i> lalu memberikan notifikasi pesan sudah tersalin. | Aplikasi menyalin teks yang dimaksud dan memberikan notifikasi pesan tersalin.                                     | Sesuai |
| 8 | Chat with Text | Mendengar respon AI dengan suara.                | Menekan <i>icon speaker</i>                                                                               | Aplikasi memproses dan menghasilkan suara berdasarkan teks.                                                   | Aplikasi menghasilkan suara berdasarkan teks.                                                                      | Sesuai |
| 9 | Chat with Text | Mengirim pesan teks sebanyak 15 kali dalam kurun | "hai" sebanyak 15 kali                                                                                    | AI memproses input dan menghasilkan respon.                                                                   | AI menjawab sapaan sebanyak 15 kali, lalu sapaan ke 16 terdapat                                                    | Sesuai |

|    |                 |                                                                                                             |                                                                     |                                                                           |                                                                                                          |        |
|----|-----------------|-------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|---------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|--------|
|    |                 | waktu dibawah 1 menit.                                                                                      |                                                                     |                                                                           | notifikasi kesalahan.                                                                                    |        |
| 10 | Chat with Text  | Mengirim pesan teks yang berisi kata “Terimakasih” dan menerima respon AI.                                  | “Terimakasih”                                                       | AI memproses input dan menghasilkan respon.                               | AI menjawab dengan mengingatkan siswa untuk mengecek catatan yang mungkin saja terdapat catatan penting. | Sesuai |
| 11 | Chat with Image | Mengunggah gambar dan menerima penjelasan dari AI.                                                          | Mengunggah gambar bendera Palestina dan bertanya: “Gambar apa ini?” | AI memproses dan menganalisa gambar lalu menghasilkan respon yang relevan | AI menjawab bahwa gambar tersebut merupakan gambar bendera negara Palestina                              | Sesuai |
| 12 | Chat with Image | Mengirim pesan teks terlebih dahulu sebelum mengunggah gambar.                                              | “Gambar apa ini?”                                                   | Aplikasi memproses dan menghasilkan respon.                               | Aplikasi memberikan notifikasi bahwa user harus mengunggah gambar terlebih dahulu.                       | Sesuai |
| 13 | Chat with Image | Menyalin respon jawaban AI.                                                                                 | Menekan <i>icon copy</i>                                            | Aplikasi mengambil teks dan menyimpannya di <i>clipboard</i>              | Aplikasi menyalin teks yang dimaksud dan memberikan notifikasi pesan tersalin.                           | Sesuai |
| 14 | Chat with PDF   | Mengunggah file PDF, klik ekstrak teks dari PDF dan mendapatkan jawaban atas pertanyaan terkait konten PDF. | PDF tentang AI, Pertanyaan: "Apa itu Artificial Intelligence?"      | AI memproses dan menganalisa PDF lalu menghasilkan respon.                | AI memberikan penjelasan tentang apa itu AI berdasarkan file PDF.                                        | Sesuai |
| 15 | Chat with PDF   | Mengunggah file PDF lalu bertanya tanpa klik                                                                | “Apa itu Artificial Intelligence?”                                  | Aplikasi memproses dan menghasilkan respon.                               | Aplikasi memberikan peringatan untuk menekan                                                             | Sesuai |

|    |               |                                                                                                              |                                                                                                                                                      |                                             |                                                                                 |        |
|----|---------------|--------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------|---------------------------------------------------------------------------------|--------|
|    |               | ekstrak teks dari PDF.                                                                                       |                                                                                                                                                      |                                             | tombol ekstrak teks terlebih dahulu                                             |        |
| 16 | Chat with PDF | Mengunggah file PDF dan menerima hasil generate quiz.                                                        | “ ”, tekan tombol generate quiz.                                                                                                                     | AI memproses dan menghasilkan respon.       | AI membuat kuis pilihan ganda dari file PDF yang diunggah.                      | Sesuai |
| 17 | Chat with PDF | Mengunggah file PDF berbahasa Indonesia, memilih bahasa Inggris untuk kuis dan menerima hasil generate quiz. | Memilih bahasa inggris untuk kuis, lalu tekan tombol generate quiz.                                                                                  | AI memproses dan menghasilkan respon.       | AI membuat kuis pilihan ganda dari file PDF yang diunggah dalam bahasa Inggris. | Sesuai |
| 18 | Generate Quiz | Membuat kuis berdasarkan topik yang diinput oleh user.                                                       | Menginput topik kuis tentang Artificial Intelligence, level kesulitan expert, jumlah soal 5, pilihan bahasa inggris lalu tekan tombol generate quiz. | AI memproses dan menghasilkan respon.       | AI membuat kuis sesuai dengan topik, level kesulitan, jumlah soal, dan bahasa.  | Sesuai |
| 19 | Generate Quiz | Mengosongkan form dan menerima respon.                                                                       | Hanya memilih level kesulitan dan bahasa kuis lalu tekan tombol generate quiz.                                                                       | Aplikasi memproses dan menghasilkan respon. | Muncul peringatan agar menginput form yang kosong.                              | Sesuai |
| 20 | Notes         | Menambah catatan.                                                                                            | Menginput catatan “Ujian Proyek” lalu tekan tombol simpan.                                                                                           | Aplikasi menyimpan catatan.                 | Catatan baru telah dibuat.                                                      | Sesuai |
| 21 | Notes         | Menambah catatan                                                                                             | Menginput catatan “Ujian                                                                                                                             | Aplikasi menyimpan catatan dan              | Catatan baru telah dibuat dan notifikasi                                        | Sesuai |

|    |                 |                                         |                                                                                                                             |                                               |                                                           |        |
|----|-----------------|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|-----------------------------------------------------------|--------|
|    |                 | dengan notifikasi.                      | Proyek”, mengatur hari dan waktu misalnya “13/08/2024 ”, mengatur notifikasi misalnya jam “13:20” lalu tekan tombol simpan. | menyiapkan notifikasi.                        | muncul sesuai waktu yang ditentukan.                      |        |
| 22 | Notes           | Mengedit catatan.                       | Menekan icon edit lalu edit catatan dan tekan tombol simpan.                                                                | Aplikasi menyimpan catatan yang telah diedit. | Catatan berhasil diedit.                                  | Sesuai |
| 23 | Notes           | Menghapus catatan yang telah dibuat.    | Menggeser card catatan ke arah kiri.                                                                                        | Aplikasi memproses gesture tangan.            | Catatan berhasil dihapus.                                 | Sesuai |
| 24 | Notes           | Mengosongkan form dan menerima respon.  | Langsung menekan tombol simpan.                                                                                             | Aplikasi memproses dan menghasilkan respon.   | Muncul peringatan agar menginput form yang kosong.        | Sesuai |
| 25 | About Dev       | Mengakses tutorial penggunaan aplikasi. | Menekan icon Youtube yang ada di pojok kiri bawah aplikasi.                                                                 | Aplikasi memproses dan mengarahkan ke tujuan. | Berhasil membuka tutorial penggunaan aplikasi di Youtube. | Sesuai |
| 26 | Education Games | Mengakses game edukasi                  | Menekan salah satu icon game yang mau dimainkan, misalnya “Math true or false”.                                             | Aplikasi merespon dan mengarahkan ke tujuan.  | Berhasil membuka game Math true or false.                 | Sesuai |
| 27 | Education Games | Keluar dari menu game edukasi.          | Mengakses game edukasi lalu tekan tombol keluar.                                                                            | Aplikasi merespon dan mengarahkan ke tujuan.  | Berhasil kembali ke menu sebelumnya.                      | Sesuai |
| 28 | Connect with Us | Mengakses sosial media.                 | Menekan salah satu icon sosial media yang tertera, misalnya Instagram Samrah Daeng Makase.                                  | Aplikasi merespon dan mengarahkan ke tujuan.  | Berhasil membuka Instagram Samrah Daeng Makase            | Sesuai |

|    |                 |                                   |                                                     |                                              |                                      |        |
|----|-----------------|-----------------------------------|-----------------------------------------------------|----------------------------------------------|--------------------------------------|--------|
| 29 | Connect with Us | Keluar dari menu Connect with Us. | Mengakses Connect with Us lalu tekan tombol keluar. | Aplikasi merespon dan mengarahkan ke tujuan. | Berhasil kembali ke menu sebelumnya. | Sesuai |
| 30 | Exit            | Keluar dari aplikasi.             | Menekan tombol ya.                                  | Aplikasi merespon dan mengakhiri sesi.       | Berhasil keluar dari aplikasi.       | Sesuai |
| 31 | Exit            | Batal keluar dari aplikasi.       | Menekan tombol tidak.                               | Aplikasi merespon dan mengarahkan ke tujuan. | Berhasil batal keluar dari aplikasi. | Sesuai |

Selama pengujian, ditemukan bahwa model Gemini AI versi *1.0 Pro-Vision* yang digunakan untuk pemrosesan gambar telah dihapus oleh pihak Google. Sebagai hasilnya, model ini digantikan oleh *Gemini 1.5 Flash*, yang menawarkan peningkatan signifikan dalam performa, termasuk kecepatan pemrosesan dan akurasi yang lebih tinggi. Perbaikan yang dilakukan yaitu mengganti nama model lama ke mobel baru ini dan pengujian ulang seluruh fungsi terkait untuk memastikan bahwa fitur seperti *chat with image* dan *chat with PDF* berfungsi optimal.

Hasil pengujian perangkat lunak menunjukkan bahwa seluruh 31 test case yang diuji telah memenuhi syarat sesuai dengan harapan. Setiap fitur dan fungsionalitas aplikasi, termasuk chat dengan teks, gambar, PDF, pembuatan kuis, serta manajemen catatan, telah diuji secara menyeluruh menggunakan metode *Black Box Testing*. Semua *test case* menunjukkan bahwa aplikasi berfungsi sesuai dengan spesifikasi yang telah ditetapkan tanpa ada kesalahan atau *bug* yang terdeteksi. Dengan demikian, aplikasi ini siap untuk digunakan oleh pengguna akhir dengan kinerja yang optimal dan sesuai dengan tujuan pengembangan.

## **DEPLOYMENT**

### **6.1 TAHAPAN DEPLOYMENT**

1. Persiapan Rilis Aplikasi Pada Lingkungan Pengembangan
  - Buka *Command Prompt* pada *root* folder aplikasi dan ketik “*flutter build apk*”.
  - Pada *root* folder aplikasi, buka folder *build/app/outputs/flutter-apk/app-release.apk*. File ini sebagai versi rilis yang akan digunakan oleh guru atau siswa.
2. Pembuatan Website Landing Page serta integrasi dengan Flowise AI
  - Menyiapkan assets website seperti gambar, video demo aplikasi, tutorial download aplikasi, file aplikasi dan konten teks untuk promosi aplikasi.
  - Mengembangkan website sesuai dengan identitas brand aplikasi, termasuk tata letak, warna, dan tipografi. Dikembangkan menggunakan HTML, CSS, dan Javascript.
  - Membuat model NLP pada platform Flowise AI dan integrasi data aplikasi melalui format PDF.
  - Mengintegrasikan Flowise AI dengan website melalui metode *embed*.
3. Persiapan Lingkungan Hosting
  - Memilih penyedia hosting website statis, salah satunya yaitu Netlify dan Hungginfase untuk hosting model NLP Flowise AI.
4. Deployment ke Server
  - Deploy website ke server hosting yang telah dipilih dengan mengupload folder website ke penyedia hosting. Untuk Flowise AI proses *deployment* menggunakan docker yang sudah disediakan pada lingkungan pengembangan Huggingface yang berbasis *cloud*.
5. Testing Post-Deployment
  - Setelah *deployment*, lakukan pengujian kembali pada website dan aplikasi yang sudah *live* untuk memastikan tidak ada masalah dengan aksesibilitas, tampilan, atau fungsionalitas.
6. Promosi dan Monitoring

- Membagikan tautan website ke guru dan siswa agar bisa mengunduh aplikasi.
- Monitor performa aplikasi dan website secara berkala, perbaiki *bug* atau masalah yang ditemukan, dan terus optimalkan berdasarkan *feedback* pengguna.

## EVALUASI DAN MONITORING

### 7.1 METODE EVALUASI

Setelah aplikasi diluncurkan (*go-live*), kami melakukan evaluasi kinerja sistem berdasarkan umpan balik pengguna. Umpan balik ini dikumpulkan melalui survei dalam aplikasi dan review pengguna di aplikasi kami. Dari umpan balik yang diterima, beberapa pengguna melaporkan bahwa aplikasi terkadang mengalami gangguan saat mengakses fitur *chat with PDF*. Tim segera menelusuri penyebabnya. Diketahui bahwa fitur tersebut berbasis web dan dihosting menggunakan *Streamlit*, sehingga langkah yang diambil adalah memastikan hosting tetap aktif agar fitur dapat diakses dengan lancar karena ketika fitur tersebut tidak aktif selama 7 hari berturut-turut maka hostingnya masuk ke mode hibernasi dengan sendirinya dan perlu diaktifkan kembali dengan menekan tombol *yes, get this app back up* lalu menunggu sekitar 30 detik untuk aktif kembali. Evaluasi semacam ini dilakukan secara berkala untuk memastikan semua fitur berfungsi dengan baik dan memberikan pengalaman terbaik bagi pengguna.

### 7.2 MONITORING DAN MAINTENANCE

Proses pemeliharaan dan update sistem dilakukan secara berkala untuk memastikan kinerja aplikasi tetap optimal. Setelah memperbaiki bug, masalah, atau penambahan fitur, versi terbaru aplikasi langsung diunggah ke website landing page. Tim juga melakukan monitoring terhadap aplikasi untuk mendeteksi potensi masalah yang mungkin muncul. Selain itu, proses pemeliharaan juga mencakup pengecekan fungsionalitas sistem dan melakukan pengujian Black Box secara berkala. Jika ditemukan masalah selama monitoring, perbaikan akan segera dilakukan dan versi rilis baru akan disediakan di website landing page.

### 7.3 KUESIONER

Tabel 1. 2 kuesioner SMA Muhammadiyah Ambon

| NO | PERNYATAAN                                                                 | PENILAIAN |   |   |   |    |
|----|----------------------------------------------------------------------------|-----------|---|---|---|----|
|    |                                                                            | 1         | 2 | 3 | 4 | 5  |
| 1  | Pengalaman yang memuaskan dalam menggunakan aplikasi                       |           |   | 1 | 2 | 12 |
| 2  | Tampilan aplikasi sudah sangat baik                                        |           |   |   | 3 | 12 |
| 3  | Antarmuka dan navigasi dari Aplikasi ini mudah dipahami                    |           |   |   | 5 | 10 |
| 4  | Aplikasi membantu untuk memenuhi kebutuhan informasi yang dibutuhkan siswa |           |   |   | 2 | 13 |

|    |                                                                         |  |  |   |   |    |
|----|-------------------------------------------------------------------------|--|--|---|---|----|
| 5  | Aplikasi merespon dengan cepat                                          |  |  | 1 | 7 | 7  |
| 6  | Semua fitur bermanfaat dan berfungsi dengan baik                        |  |  |   |   | 15 |
| 7  | Tidak menemukan masalah atau bug saat menggunakan aplikasi ini          |  |  |   |   | 15 |
| 8  | Fitur game edukasi sangat menantang                                     |  |  | 1 | 4 | 10 |
| 9  | Fitur game edukasi membuat proses belajar menjadi lebih menyenangkan    |  |  | 1 | 4 | 10 |
| 10 | Memungkinkan untuk merekomendasikan aplikasi kepada teman atau keluarga |  |  | 6 |   | 9  |

Keterangan:

Jumlah siswa = 15

- 1 = Sangat tidak setuju
- 2 = Tidak setuju
- 3 = Netral
- 4 = Setuju
- 5 = Sangat setuju

## HASIL DAN KESIMPULAN

### 8.1 HASIL

Selama proses pengembangan dan pengujian aplikasi, peneliti berhasil menciptakan aplikasi yang sesuai dengan spesifikasi dan kebutuhan pengguna. Berikut adalah hasil rinci dari berbagai aspek pengujian dan implementasi aplikasi:

#### 1. Pengujian Fungsionalitas:

- Sebanyak 31 test case diuji menggunakan metode Black Box Testing, yang mencakup pengujian semua fitur utama aplikasi, termasuk chat dengan teks, gambar, dan PDF, serta fitur generate quiz dan notifikasi notes.
- Semua *test case* lulus dengan hasil sesuai harapan, yang menunjukkan bahwa semua fitur berfungsi sesuai dengan spesifikasi yang ditetapkan.

#### 2. Tantangan Teknis dan Solusi:

- Kami menghadapi tantangan teknis terkait dengan penggantian model Gemini AI dari versi *1.0 Pro-Vision* ke versi *1.5 Flash*. Model baru ini dipilih untuk meningkatkan kinerja dan kompatibilitas fitur pemrosesan gambar.
- Peralihan ini dilakukan dengan sukses tanpa mengganggu fungsi aplikasi lainnya, dan pengujian setelah peralihan menunjukkan bahwa fitur pemrosesan gambar berfungsi dengan lebih baik, dengan kecepatan dan akurasi yang meningkat.

#### 3. Hasil Implementasi:

- Aplikasi berhasil diintegrasikan dengan berbagai komponen yang diperlukan, termasuk database SQLite untuk fitur *notes* dan API Gemini AI untuk *chat with text, image, PDF, and generate quiz*.
- Pengguna aplikasi dapat berinteraksi dengan aplikasi dengan mudah dan mendapatkan manfaat dari fitur-fitur edukasi yang dirancang untuk meningkatkan pembelajaran interaktif

#### 4. Deployment dan Promosi:

- Proses deployment website *landing page* untuk promosi aplikasi berjalan dengan lancar. Website yang dibuat berfungsi sebagai platform untuk memperkenalkan aplikasi serta fitur-fiturnya kepada pengguna.
- Website ini sudah siap digunakan untuk mendukung promosi dan distribusi aplikasi kepada pengguna sasaran.

- Aplikasi versi rilis sudah bisa digunakan oleh guru maupun siswa SMA Muhammadiyah Ambon.

#### 5. Maintenance:

- Monitor performa aplikasi dan website secara berkala, perbaiki *bug* atau masalah yang ditemukan, dan terus optimalkan berdasarkan *feedback* pengguna.

## 8.2 KESIMPULAN

Berdasarkan hasil yang telah melalui beberapa tahapan implementasi, dapat disimpulkan bahwa telah dibuat *Platform Aplikasi Artificial Intelligence (AI) Berbasis Natural Language Processing (NLP)* dan *Game Edukasi* untuk Menfasilitasi Pembelajaran Interaktif Siswa di SMA Muhammadiyah Ambon. Aplikasi ini dirancang untuk meningkatkan pengalaman belajar siswa dengan fitur-fitur interaktif yang didukung oleh teknologi AI, maka dapat ditarik beberapa kesimpulan, yaitu:

- 1) Aplikasi AI berbasis *Natural Language Processing (NLP)* dan *game* edukasi yang dikembangkan sangat membantu siswa dan guru di SMA Muhammadiyah Ambon dalam proses pembelajaran interaktif.
- 2) Aplikasi ini menyediakan fitur-fitur utama seperti *Chat with Text*, *Chat with Image*, *Chat with PDF*, *Generate Quiz*, dan *Notes*, yang telah terintegrasi dengan Gemini AI untuk mendukung proses belajar yang lebih personal dan masing-masing fiturnya memiliki kemampuannya tersendiri.
- 3) Metode pengembangan Waterfall yang digunakan memastikan bahwa setiap tahap pengembangan dilakukan dengan teliti dan sistematis, menghasilkan aplikasi yang stabil dan dapat diandalkan.
- 4) Pengujian yang komprehensif telah mengidentifikasi dan memperbaiki berbagai masalah, menjadikan aplikasi ini lebih user-friendly dan responsif terhadap kebutuhan pengguna.

## 8.3 SARAN

Dari pembuatan *Platform Aplikasi Artificial Intelligence (AI) Berbasis Natural Language Processing (NLP)* dan *Game Edukasi* untuk Menfasilitasi Pembelajaran Interaktif Siswa di SMA Muhammadiyah Ambon, ke depannya penulis masih memerlukan adanya saran dan pendapat sehingga mampu membuat aplikasi yang lebih baik dan semakin berguna bagi pengguna. Saran yang diajukan antara lain:

Bagi Peneliti:

- 1) Di masa mendatang, aplikasi ini bisa dikembangkan dengan menambahkan fitur-fitur tambahan yang lebih interaktif seperti *game* edukasi yang baru, register dan login akun, penyimpanan *history chat*.
- 2) Meningkatkan optimasi kinerja, terutama dalam respons AI, serta selalu tersedianya fitur *Chat with PDF* agar aplikasi tetap bisa diakses kapan saja tanpa kendala.

Bagi Sekolah :

- 1) Sekolah dapat mempertimbangkan untuk mengintegrasikan aplikasi ini secara resmi ke dalam kegiatan belajar mengajar, sehingga siswa dapat memanfaatkan teknologi AI untuk mendukung proses pembelajaran yang lebih interaktif.
- 2) Memastikan dukungan infrastruktur teknologi yang memadai di sekolah, seperti perangkat keras yang cukup dan koneksi internet yang stabil, agar aplikasi dapat diakses dan digunakan dengan lancar tanpa hambatan teknis.
- 3) Mengadakan sosialisasi kepada orang tua mengenai manfaat aplikasi ini dan cara menggunakannya di rumah. Hal ini penting agar orang tua dapat mendukung proses belajar anak-anak mereka melalui penggunaan teknologi.

## **LAMPIRAN**

### **11.1 LINK TERKAIT**

Repository Github

[https://github.com/samrahdm20/ByteBrain\\_AI\\_Final\\_Version](https://github.com/samrahdm20/ByteBrain_AI_Final_Version)

Download ByteBrain AI

<https://bytebrain-ai.netlify.app/>

### **11. 3 DOKUMENTASI**



