
Software Requirement Specification

Document

Project Name: BrainRowth

Author: Pradipa Yogananda

| | |
|----------------|---|
| Document ID | SOFTWARE REQUIREMENT SPECIFICATION-v0.1 |
| Version Number | 0.1 |
| Issue Date | December 01, 2025 |
| Classification | Public |

Copyright Notice

© COMPANYNAME, (original issue year – current issue year)

All Rights Reserved

The information contained in this document is the property of COMPANYNAME. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means; mechanical, photocopying, recording, or otherwise, without the prior written consent of COMPANYNAME. Under the law, copying includes translating into another language or format. Legal action will be taken against any infringement.

The information contained in this document is subject to change without notice and does not carry any contractual obligation for COMPANYNAME. COMPANYNAME reserves the right to make changes to any products or services described in this document at any time without notice. COMPANYNAME shall not be held responsible for the direct or indirect consequences of the use of the information contained in this document.

Revision History

| Date | Version | Description | Author (s) |
|------------|---------|---------------|------------|
| 01/12/2025 | 0.1 | Draft Version | Jane Doe |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

| Reviewed By (Customer) | Signature | Date |
|------------------------|-----------|------|
| | | |
| | | |
| | | |
| | | |
| | | |

The reviewer signoff shall signify the recommendation for acceptance of this document.

Sign Off

| Prepared By | Acknowledged By |
|-------------------|-------------------|
| | |
| <Name> | <Name> |
| Title: <Position> | Title: <Position> |
| COMPANYNAME | COMPANYNAME |
| Date: | Date: |

| Accepted By | Accepted By |
|--------------------------|--------------------------|
| | |
| <Name> | <Name> |
| Title: <Position> | Title: <Position> |
| <Customer Company Name > | <Customer Company Name > |
| Date: | Date: |

Daftar Isi

| | | |
|----------|---|-----------|
| 1 | Introduction..... | 8 |
| 1.1 | Purpose | 8 |
| 1.2 | Document Conventions | 8 |
| 1.3 | Intended Audience and Reading Suggestions..... | 9 |
| 1.4 | Project Scope | 9 |
| 1.5 | References | 10 |
| 1.6 | Acronyms and Abbreviations | 10 |
| 2 | Overall Description | 11 |
| 2.1 | Product Perspective | 11 |
| 2.2 | Product Features..... | 11 |
| 2.3 | User Classes and Characteristics..... | 12 |
| 2.4 | Operating Environment | 13 |
| 2.5 | Design and Implementation Constraints | 13 |
| 2.6 | User Documentation..... | 13 |
| 2.7 | Assumptions and Dependencies | 14 |
| 3 | System Features..... | 15 |
| 3.1 | SF-01 — Input Soal Matematika (Teks)..... | 15 |
| 3.1.1 | Description and Priority | 15 |
| 3.1.2 | Stimulus/Response Sequences..... | 15 |
| 3.1.3 | Functional Requirements | 15 |
| 3.2 | SF-02 — Input Soal Matematika dari Gambar (OCR) | 16 |
| 3.2.1 | Description and Priority | 16 |
| 3.2.2 | Stimulus/Response Sequences..... | 16 |
| 3.2.3 | Functional Requirements | 16 |
| 3.3 | SF-03 — Pemrosesan Soal oleh AI..... | 17 |
| 3.3.1 | Description and Priority | 17 |

| | | |
|----------|---|-----------|
| 3.3.2 | Stimulus/Response Sequences..... | 17 |
| 3.3.3 | Functional Requirements | 17 |
| 3.4 | SF-04 — Menampilkan Solusi dari AI | 18 |
| 3.4.1 | Description and Priority | 18 |
| 3.4.2 | Stimulus/Response Sequences..... | 18 |
| 3.4.3 | Functional Requirements | 18 |
| 3.5 | SF-05 — Manajemen Riwayat/History..... | 19 |
| 3.5.1 | Description and Priority | 19 |
| 3.5.2 | Stimulus/Response Sequences..... | 19 |
| 3.5.3 | Functional Requirements | 19 |
| 3.6 | SF-06 — Koneksi ke Server Lokal | 20 |
| 3.6.1 | Description and Priority | 20 |
| 3.6.2 | Stimulus/Response Sequences..... | 20 |
| 3.6.3 | Functional Requirements | 20 |
| 3.7 | SF-07 — UI Pengguna(UI/UX)..... | 21 |
| 3.7.1 | Description and Priority | 21 |
| 3.7.2 | Stimulus/Response Sequences..... | 21 |
| 3.7.3 | Functional Requirements | 21 |
| 4 | External Interface Requirements | 22 |
| 4.1 | User Interfaces | 22 |
| 4.2 | Hardware Interfaces | 24 |
| 4.3 | Software Interfaces | 24 |
| 4.4 | Communications Interfaces | 26 |
| 5 | Other Nonfunctional Requirements | 27 |
| 5.1 | Performance Requirements..... | 27 |
| 5.2 | Safety Requirements..... | 27 |
| 5.3 | Security Requirements..... | 27 |
| 5.4 | Software Quality Attributes | 28 |

| | | |
|----------|-----------------------------------|-----------|
| 6 | Other Requirements | 29 |
| | Appendix A: Glossary | 33 |

1 Introduction

1.1 Purpose

Dokumen Software Requirement Specification (SRS) ini dibuat untuk mendeskripsikan kebutuhan perangkat lunak dari aplikasi BrainRowth, sebuah aplikasi mobile Android yang digunakan untuk menyelesaikan soal matematika secara otomatis menggunakan model kecerdasan buatan (Large Language Model) yang dijalankan secara lokal di laptop pengembang.

Dokumen ini menjelaskan:

- Ruang lingkup sistem
- Fitur dan alur kerja
- Kebutuhan fungsional dan non-fungsional
- Interaksi antara aplikasi, backend, dan LLM lokal
- Batasan serta asumsi operasional

SRS ini akan menjadi acuan bagi pengembang, penguji, dan pihak penilai selama proses pembangunan dan presentasi aplikasi.

1.2 Document Conventions

- **SF-XX** - Identifier untuk System Features
- **FR-XX** - Identifier untuk Functional Requirements
- **NFR-XX** - Identifier untuk Non-Functional Requirements
- **PR-XX** - identifier untuk Performance Requirements
- **SR-XX** – Identifier untuk Safety Requirements
- **SEC-XX** – Identifier untuk Security Requirements
- **UQ-XX** – Identifier untuk Usability
- **RQ-XX** – Identifier untuk Reliability
- Istilah **“BrainRowth AI Engine”** digunakan untuk menyebut **LLM lokal** yang dipanggil melalui LM Studio (menggunakan model **Qwen3 4B Instruct 2507**).
- Kata **“harus/shall”** menunjukkan kebutuhan wajib, sedangkan **“sebaiknya/should”** menunjukkan preferensi atau rekomendasi.

1.3 *Intended Audience and Reading Suggestions*

- **Pengembang (Developers)** – untuk memahami alur kerja, integrasi, dan implementasi aplikasi.
- **Penguji (QA/Testers)** – sebagai dasar dalam menyusun test case dan skenario pengujian.
- **Dosen /Instructor /Reviewer** – untuk mengevaluasi kelayakan desain dan implementasi aplikasi.
- **Pengguna teknis lanjutan / pemelihara sistem** – sebagai panduan untuk pengembangan di masa depan.

Rekomendasi pembacaan:

1. Mulai dari **Introduction** dan **Overall Description (Section 2)** untuk memahami gambaran besar.
2. Pelajari **System Features (Section 3)** untuk mengetahui kebutuhan fungsional secara rinci.
3. Baca **External Interface Requirements (Section 4)** untuk memahami interaksi aplikasi dengan backend dan LLM lokal.
4. Lanjutkan ke **Other Requirements (Section 5)** untuk memahami batasan performa, keamanan, serta kualitas sistem.

1.4 *Project Scope*

BrainRowth adalah aplikasi mobile Android berbasis AI yang berfungsi untuk:

- Menyelesaikan soal matematika **berbasis teks**.
- Menyelesaikan soal matematika dari **gambar** menggunakan **OCR** dan di konversi ke text.
- Menampilkan solusi dalam bentuk:
 - Langkah-langkah penyelesaian (step-by-step JSON)
 - Penjelasan naratif (untuk soal cerita)
- Menyimpan riwayat penyelesaian soal secara **lokal di perangkat** (tanpa cloud).

Komponen utama proyek:

- **Aplikasi Android** (UI/UX menggunakan Jetpack Compose)

- **Backend Node.js/Express** sebagai perantara ke LLM
- **LLM Lokal (BrainRowth AI Engine)** yang berjalan di LM Studio
- **ML Kit OCR** untuk membaca soal dari gambar

Fitur di luar cakupan (tidak termasuk versi ini):

- Sistem akun online atau cloud sync
- Pembelajaran adaptif, rekomendasi materi, atau leaderboard
- Deployment ke iOS atau platform lain
- Penyimpanan history online
- Multi-server operasional

1.5 References

- Dokumentasi resmi **LM Studio** untuk penggunaan model LLM secara lokal.
- Dokumentasi **Android Developer** (Jetpack Compose, Retrofit, ViewModel, Navigation).
- Dokumentasi **ML Kit Text Recognition (OCR)**.
- Dokumentasi **Node.js dan Express** untuk pembuatan backend API.
- Dokumen UseCase, FlowChart, ActivityDiagram, ERD, dan KamusData.

1.6 Acronyms and Abbreviations

| | |
|-------------|------------------------------------|
| SRS | Software Requirement Specification |
| LLM | Large Language Model |
| OCR | Optical Character Recognition |
| API | Application Programming Interface |
| JSON | JavaScript Object Notation |
| UI | User Interface |
| UX | User Experience |
| AI | Artificial Intelligence |

2 Overall Description

2.1 Product Perspective

BrainRowth adalah aplikasi mobile Android yang berfungsi sebagai asisten matematika berbasis AI. Aplikasi ini memungkinkan pengguna memasukkan soal matematika berupa teks ataupun gambar. Soal tersebut kemudian diproses melalui **backend Node.js**, yang meneruskan pertanyaan tersebut ke **LLM lokal (BrainRowth AI Engine)** yang berjalan melalui LM Studio.

Sistem bekerja dalam tiga komponen utama:

- a. **Aplikasi Android (Frontend)**
 - Menyediakan **UI** pengguna.
 - Mengirim soal ke backend.
 - Menampilkan hasil (jawaban akhir, langkah-langkah, narasi).
- b. **Backend Node.js/Express**
 - Mengatur komunikasi antara aplikasi dan AI engine.
 - Membersihkan, memvalidasi, dan memformat hasil dari LLM.
 - Menyediakan endpoint seperti `/api/solve-text`.
- c. **LLM Lokal melalui LM Studio**
 - Model AI dijalankan secara local menggunakan Model **Qwen3 4B Instruct 2507**.
 - Mendapat input dari backend dan menghasilkan jawaban.
 - Tidak menggunakan layanan cloud, sehingga data aman dan local.

2.2 Product Features

- a. **Memasukkan Soal Matematika**
 - Input teks manual
 - Input gambar yang diproses dengan OCR lokal
- b. **Memproses Soal Menggunakan AI Lokal**
 - Mengirim soal ke backend
 - Backend meneruskan ke LLM

- LLM memproses dan menghasilkan solusi
- c. Menampilkan Solusi**
- Jawaban akhir
 - Langkah-langkah penyelesaian (step-by-step)
 - Narasi penjelasan untuk soal cerita
- d. Mengelola Riwayat (History) Lokal**
- Menyimpan soal yang pernah diselesaikan
 - Menampilkan daftar riwayat
 - Menghapus riwayat tertentu
- e. Mendukung Bahasa Indonesia**
- Solusi dan penjelasan AI ditampilkan dalam Bahasa Indonesia

2.3 *User Classes and Characteristics*

- a. Pelajar (SD–SMA)**
- Membutuhkan bantuan memahami langkah penyelesaian soal
 - Fokus pada penjelasan sederhana & mudah dimengerti
- b. Mahasiswa**
- Membutuhkan solusi matematis yang lebih kompleks
 - Membantu saat belajar mandiri
- c. Umum / Pengguna Dewasa**
- Mencari solusi cepat tanpa harus memahami seluruh teori
 - Cocok untuk pekerjaan yang membutuhkan perhitungan ringan

Kemampuan pengguna yang diharapkan:

- Mampu mengoperasikan aplikasi smartphone dasar
- Mampu memasukkan soal dalam bentuk teks atau foto
- Tidak perlu kemampuan teknis tentang AI

2.4 *Operating Environment*

BrainRowth berjalan pada lingkungan:

Perangkat Android :

- OS minimal Android 7.0
- Kamera berfungsi untuk OCR

Laptop Pengembang :

- Menjalankan LM Studio
- Menjalankan Backend Node.js

2.5 *Design and Implementation Constraints*

- Sistem hanya bekerja jika laptop pengembang aktif
- Jika WiFi tidak stabil, aplikasi tidak dapat mengirim soal
- OCR dapat gagal jika gambar buram
- Output LLM bisa bervariasi (perlu cleaning di backend)
- Tidak mendukung cloud mode

2.6 *User Documentation*

Dokumentasi yang Disediakan:

1. Panduan Penggunaan Aplikasi (User Guide)

- Tersedia dalam bentuk halaman bantuan (help) di dalam aplikasi.
- Berisi langkah-langkah:
 - Cara memasukkan soal matematika berbasis teks
 - Cara menggunakan fitur kamera/galeri untuk mengambil gambar soal
 - Cara membaca hasil solusi dan langkah-langkah AI
 - Cara mengakses dan menghapus riwayat (history)

2. Informasi Tambahan di Aplikasi

- Terdapat deskripsi singkat mengenai fungsi utama BrainRowth pada halaman pertama aplikasi.
- Penjelasan ringkas mengenai cara kerja AI lokal.

3. FAQ & Troubleshooting Ringan

- Bagian tanya jawab dasar seperti:
 - “Mengapa solusi tidak muncul?”
 - “Mengapa OCR gagal membaca teks?”
 - “Mengapa aplikasi gagal terhubung ke server AI lokal?”

Bentuk Dokumentasi

- Ditampilkan langsung di aplikasi (bagian **Help / Bantuan**)
- PDF opsional untuk pendamping laporan proyek

2.7 *Assumptions and Dependencies*

Asumsi

- Pengembang menyediakan laptop + server lokal saat demo
- Tester terhubung ke jaringan WiFi yang sama dengan pengembang
- LM Studio sudah load model sebelum aplikasi digunakan
- Backend berjalan tanpa error

Ketergantungan

- LM Studio
- Model AI yang kompatibel (**Qwen3 4B Instruct 2507**)
- Jaringan yang memadai
- Node.js backend
- Aplikasi Android di perangkat client

3 System Features

Setiap fitur diberi identifier **SF-XX**, dan setiap kebutuhan fungsional diberi **FR-XX**.

3.1 SF-01 — Input Soal Matematika (Teks)

3.1.1 Description and Priority

Fitur ini memungkinkan pengguna memasukkan soal matematika dalam bentuk teks melalui antarmuka aplikasi BrainRowth.

3.1.2 Stimulus/Response Sequences

Stimulus

1. Pengguna membuka halaman **Solve Text**.
2. Pengguna mengetikkan soal matematika.
3. Pengguna menekan tombol **“Solve”**.

Response

1. Sistem memvalidasi bahwa input tidak kosong.
2. Sistem mengirimkan soal ke backend.
3. Jika backend merespon, aplikasi melanjutkan ke proses menampilkan hasil.
4. Jika backend gagal dihubungi, sistem menampilkan pesan error.

3.1.3 Functional Requirements

FR-01 - Sistem harus menyediakan antarmuka untuk memasukkan soal matematika berbentuk teks.

FR-02 - Sistem harus memvalidasi bahwa input teks tidak kosong sebelum diproses.

FR-03 - Sistem harus mengirim soal teks ke backend melalui endpoint `/api/solve-text`.

FR-04 - Sistem harus menampilkan pesan error jika user mengirimkan input kosong.

FR-05 - Sistem harus menampilkan pesan error jika backend tidak dapat dihubungi.

3.2 SF-02 — Input Soal Matematika dari Gambar (OCR)

3.2.1 Description and Priority

Fitur ini memungkinkan pengguna mengambil foto atau memilih gambar soal, kemudian aplikasi melakukan OCR untuk mengekstrak teks.

3.2.2 Stimulus/Response Sequences

Stimulus

1. Pengguna membuka halaman **Solve from Image**.
2. Pengguna memilih kamera atau galeri.
3. Pengguna mengambil gambar atau memilih file gambar.
4. Pengguna menekan tombol **“Extract Text / Solve”**.

Response

1. Sistem menjalankan OCR untuk mengekstrak teks dari gambar.
2. Sistem menampilkan teks hasil OCR kepada pengguna.
3. Jika OCR gagal, sistem menampilkan pesan bahwa teks tidak terbaca.
4. Jika berhasil, teks dikirim ke backend untuk diproses.

3.2.3 Functional Requirements

FR-06 - Sistem harus menyediakan opsi mengambil gambar menggunakan kamera atau memilih dari galeri.

FR-07 - Sistem harus menggunakan OCR untuk mengekstrak teks dari gambar.

FR-08 - Sistem harus menampilkan teks hasil OCR kepada user sebelum mengirimkannya.

FR-09 - Jika OCR gagal membaca teks, aplikasi harus menampilkan peringatan.

FR-10 - Setelah konfirmasi, aplikasi harus mengirimkan teks hasil OCR ke backend.

3.3 SF-03 — Pemrosesan Soal oleh AI

3.3.1 Description and Priority

Backend menerima soal dari aplikasi dan meneruskannya ke LLM lokal yang dijalankan melalui LM Studio, kemudian mengembalikan solusi.

3.3.2 Stimulus/Response Sequences

Stimulus

1. Backend menerima request HTTP POST dari aplikasi.
2. Aplikasi mengirim teks soal untuk diproses.
3. LLM diminta memberikan jawaban sesuai format yang ditentukan.

Response

1. Backend membentuk prompt dan mengirimkannya ke LLM via LM Studio.
2. LLM menghasilkan jawaban (steps + final answer atau narasi).
3. Backend membersihkan output, mengekstrak JSON bila mungkin.
4. Backend mengirim hasil kembali ke aplikasi dalam format JSON konsisten.
5. Jika terjadi error (LLM mati, parsing error), backend mengirim fallback response.

3.3.3 Functional Requirements

- FR-11 - Backend harus menerima soal dari aplikasi melalui HTTP POST.
- FR-12 - Backend harus mengirim soal ke LLM lokal melalui interface API LM Studio.
- FR-13 - Backend harus memformat prompt agar LLM menjawab dalam Bahasa Indonesia.
- FR-14 - Backend harus membersihkan output dari token seperti <think> atau format tidak valid.
- FR-15 - Backend harus mengembalikan struktur JSON yang dapat diparsing aplikasi.
- FR-16 - Sistem harus tetap mengirim fallback jika JSON parsing gagal.

3.4 *SF-04 — Menampilkan Solusi dari AI*

3.4.1 Description and Priority

Aplikasi menampilkan hasil solusi dari backend berupa langkah-langkah dan jawaban akhir.

3.4.2 Stimulus/Response Sequences

Stimulus

1. Aplikasi menerima respon dari backend.
2. Backend mengirim hasil berupa langkah-langkah, jawaban akhir, atau penjelasan naratif.
3. Pengguna membuka halaman hasil.

Response

1. Sistem menampilkan langkah-langkah penyelesaian (jika tersedia).
2. Sistem menampilkan jawaban akhir secara jelas.
3. Sistem menampilkan narasi mentah jika format JSON tidak valid.
4. Sistem menampilkan error jika respon backend tidak dapat diproses.

3.4.3 Functional Requirements

FR-17 - Sistem harus menampilkan langkah-langkah penyelesaian jika tersedia.

FR-18 - Sistem harus menampilkan jawaban akhir dengan jelas.

FR-19 - Sistem harus menampilkan narasi jika model tidak memberikan format langkah.

FR-20 - Sistem harus menampilkan narasi jika model tidak memberikan format langkah.

3.5 *SF-05 — Manajemen Riwayat/History*

3.5.1 Description and Priority

Sistem menyimpan dan menampilkan riwayat soal serta solusi secara lokal di perangkat pengguna.

3.5.2 Stimulus/Response Sequences

Stimulus

1. Pengguna menekan tombol **"Save to History"** pada halaman hasil AI.
2. Pengguna membuka halaman **History**.
3. Pengguna menekan tombol **Delete** pada salah satu item history.

Response

1. Sistem menyimpan data soal + solusi ke penyimpanan lokal.
2. Sistem menampilkan daftar riwayat pada halaman history.
3. Sistem menghapus item history dari penyimpanan dan memperbarui tampilan list.

3.5.3 Functional Requirements

FR-21 - Sistem harus menyimpan riwayat di local storage (Room Database / SharedPreferences).

FR-22 - Sistem harus menampilkan daftar riwayat kepada pengguna.

FR-23 - Sistem harus memungkinkan pengguna menghapus item history.

FR-24 - Riwayat harus tetap tersimpan meskipun aplikasi ditutup.

3.6 SF-06 — Koneksi ke Server Lokal

3.6.1 Description and Priority

Aplikasi harus berfungsi melalui jaringan LAN WiFi, di mana backend dan LLM berjalan di laptop pengembang.

3.6.2 Stimulus/Response Sequences

Stimulus

1. Aplikasi mencoba mengirim request ke backend (misal /api/solve-text).
2. Pengguna menekan tombol "Solve".
3. Jaringan berubah (misal WiFi off atau berpindah jaringan).

Response

1. Sistem mencoba menghubungi backend berdasarkan IP server lokal.
2. Jika koneksi gagal, sistem menampilkan pesan error seperti:
 - “Tidak dapat terhubung ke server AI.”
 - “Pastikan perangkat Anda berada dalam jaringan yang sama.”
3. Jika berhasil, sistem menerima respon JSON dari backend dan meneruskan ke layer solusi.

3.6.3 Functional Requirements

FR-25 - Sistem harus menghubungkan aplikasi ke IP lokal laptop pengembang (contoh: <http://192.168.x.x:3000>).

FR-26 - Sistem harus menampilkan error jika perangkat tidak berada pada jaringan WiFi yang sama.

FR-27 - Sistem harus secara otomatis menangani timeout atau kegagalan koneksi.

FR-28 - Sistem harus mengizinkan konfigurasi URL backend bila diperlukan (settings).

3.7 SF-07 — UI Pengguna(UI/UX)

3.7.1 Description and Priority

Fitur yang mengatur tampilan antarmuka, navigasi, dan elemen visual aplikasi BrainRowth.

3.7.2 Stimulus/Response Sequences

Stimulus

1. Pengguna membuka aplikasi BrainRowth.
2. Pengguna berpindah halaman (Solve Text, Solve Image, History).
3. Aplikasi sedang menunggu respon AI.

Response

1. Sistem menampilkan layout utama dan navigasi.
2. Sistem menampilkan halaman tujuan sesuai menu yang dipilih pengguna.
3. Sistem menampilkan loading indicator selama komunikasi dengan backend.
4. Sistem menampilkan hasil AI dalam format yang mudah dibaca.

3.7.3 Functional Requirements

FR-29 - Sistem harus menyediakan navigasi sederhana untuk menu utama, text solver, image solver, dan history.

FR-30 - Sistem harus menampilkan loading indicator saat menunggu respon backend.

FR-31 - UI harus tampil responsif di berbagai ukuran layar ponsel.

4 External Interface Requirements

4.1 User Interfaces

4.1.1 Halaman Beranda (Main Menu)

Elemen :

- Tombol Solve Text
- Tombol Solve from Image
- Tombol History
- Informasi singkat mengenai BrainRowth

Respon:

- Menavigasi ke halaman sesuai menu yang dipilih pengguna.

4.1.2 Halaman Solve Text

Elemen:

- TextField untuk memasukkan soal matematika
- Tombol Solve
- Loading indicator saat backend sedang memproses AI

Respon:

- Jika input kosong → tampil pesan error
- Jika backend dapat diakses → tampil halaman solusi
- Jika terjadi error koneksi → tampil pesan “Tidak dapat terhubung ke server AI”

4.1.3 Halaman Solve from Image

Elemen:

- Tombol kamera
- Tombol pilih gambar dari galeri
- Preview gambar
- Hasil OCR

- Tombol Solve (mengirim hasil OCR ke backend)

Respon:

- Jika OCR berhasil → tampil teks hasil ekstraksi
- Jika OCR gagal → tampil pesan “Teks tidak terbaca”

4.1.4 Halaman Hasil Solusi

Elemen:

- Soal
- Daftar langkah penyelesaian
- Jawaban akhir
- Tombol Save to History
- Tombol kembali

Respon:

- Jika langkah tersedia → ditampilkan dalam list
- Jika hanya narasi tersedia → tampil narasi
- Jika backend error → tampil “Solusi tidak dapat diproses”

4.1.5 Halaman History

Elemen:

- Daftar riwayat soal
- Setiap item memuat ringkasan dan tanggal
- Tombol delete untuk menghapus history

Respon:

- Menampilkan seluruh data history lokal
- Menghapus item bila tombol delete ditekan

4.2 *Hardware Interfaces*

4.2 Hardware Interface

BrainRowth membutuhkan hardware berikut:

Di Perangkat Android (Client / Pengguna)

- Kamera belakang untuk mengambil foto soal
- Memori internal untuk menyimpan history
- Koneksi WiFi agar dapat mengakses backend lokal
- CPU/GPU standar smartphone untuk menjalankan aplikasi Android

Di Laptop Pengembang (Server Lokal)

- CPU yang mendukung inferensi model AI
- GPU yang mendukung untuk load AI nya
- RAM minimal 16 GB untuk menjalankan Software pendukung
- VRAM minimal 6 GB untuk load AI model **Qwen3 4B**.
- Koneksi WiFi untuk membuat server dapat diakses oleh perangkat Android

4.3 *Software Interfaces*

4.3.1 LM Studio (Local LLM Engine)

- Digunakan untuk menjalankan model AI secara lokal
- Harus mendukung OpenAI-Compatible API

Endpoint standar LM Studio:

POST `http://127.0.0.1:1234/v1/chat/completions`

Backend berinteraksi dengan LM Studio menggunakan format JSON.

4.3.2 Backend Node.js

- Dibangun dengan Express.js
- Menyediakan REST API kepada aplikasi Android

Endpoint utama:

POST /api/solve-text

Respons backend dalam format:

```
{  
  "question": "...",  
  "steps": [],  
  "final_answer": "...",  
  "raw_answer": "...",  
  "parse_error": null  
}
```

4.3.3 ML Kit OCR

- Digunakan untuk memproses teks dari gambar
- Berjalan lokal pada perangkat Android
- Tidak membutuhkan internet

4.3.4 Android OS & Jetpack Compose

- UI dibuat menggunakan Jetpack Compose

Komunikasi dilakukan via Retrofit/HTTPClient

4.4 *Communications Interfaces*

4.4.1 Komunikasi Android → Backend

- Menggunakan HTTP REST
- Menggunakan metode **POST**
- Berjalan melalui jaringan **LAN WiFi**
- Aplikasi harus mengganti URL backend sesuai IP laptop pengembang

Contoh:

`http://192.168.0.25:3000/api/solve-text`

4.4.2 Komunikasi Backend → LLM Lokal

- Format OpenAI-API
- JSON-based
- Contoh Port LM Studio: **1234**

4.4.3 Protokol Pendukung

- HTTP/1.1
- JSON sebagai format pertukaran data
- Koneksi harus berada dalam jaringan yang sama (mode LAN)

4.4.4 Error Communication Handling

Jika koneksi gagal:

- Aplikasi harus menampilkan pesan error
- Timeout maksimal 10–15 detik
- Sistem harus kembali ke halaman sebelumnya tanpa crash

5 Other Nonfunctional Requirements

5.1 Performance Requirements

- **PR-01** — Sistem harus dapat memproses satu permintaan penyelesaian soal matematika dalam waktu **maksimal 5–15 detik** pada kondisi normal, tergantung kemampuan perangkat yang menjalankan LLM lokal.
- **PR-02** — Aplikasi harus mampu menampilkan respons backend tanpa lag yang berarti (< 500ms setelah menerima respons).
- **PR-03** — Backend harus mampu menangani **minimal 3 permintaan berturut-turut** tanpa terjadi crash.
- **PR-04** — Proses OCR harus selesai dalam waktu **maksimal 3 detik** pada gambar beresolusi standar.
- **PR-05** — History lokal harus dapat disimpan dan ditampilkan **tanpa penurunan performa**, meskipun jumlah entri mencapai 200+.
- **PR-06** — Sistem harus tetap dapat merespons ketika WiFi mengalami latency, dengan retry atau error handling yang sesuai.

5.2 Safety Requirements

1. **SR-01** — Sistem harus memastikan bahwa data yang dikirim melalui LAN tidak menyebabkan perubahan atau kerusakan pada perangkat pengguna.
2. **SR-02** — Aplikasi harus menangani kondisi error AI (misal output tidak dapat diparsing) tanpa menyebabkan crash.
3. **SR-03** — Model AI lokal tidak boleh melakukan operasi file atau aksi yang mengubah sistem host di luar konteks inferensi.
4. **SR-04** — Aplikasi harus mencegah terjadinya loop permintaan yang tidak disengaja (misalnya pengguna menekan tombol Solve berulang kali saat koneksi lambat).
5. **SR-05** — Sistem harus menampilkan peringatan apabila koneksi LAN terputus ketika proses solving sedang berlangsung.

5.3 Security Requirements

1. **SEC-01** — Komunikasi antara aplikasi Android dan backend harus dilakukan dalam jaringan **LAN tertutup** (private network).
2. **SEC-02** — Backend tidak boleh menerima request dari luar LAN kecuali jika sengaja diperbolehkan (misalnya melalui ngrok untuk keperluan demo).

3. **SEC-03** — Sistem tidak menyimpan soal atau jawaban pada server; semua history bersifat **lokal pada device pengguna**.
4. **SEC-04** — Aplikasi tidak boleh meminta permission yang tidak relevan selain kamera dan storage.
5. **SEC-05** — Backend harus memvalidasi input untuk menghindari request kosong atau malformed yang berpotensi menyebabkan crash.
6. **SEC-06** — Endpoint backend tidak boleh diekspos untuk operasi selain inference.

5.4 *Software Quality Attributes*

5.4.1 Usability

- **UQ-01** — UI harus mudah dipahami oleh pelajar dan pengguna awam.
- **UQ-02** — Tombol-tombol utama harus terlihat jelas dan mudah diakses.
- **UQ-03** — Semua pesan error harus ditulis dalam Bahasa Indonesia yang sederhana.

5.4.3 Availability

- **AV-01** — Sistem harus tersedia selama laptop pengembang, backend, dan LM Studio aktif.
- **AV-02** — Aplikasi harus menampilkan pesan yang sesuai jika backend sedang tidak aktif.

5.4.4 Maintainability

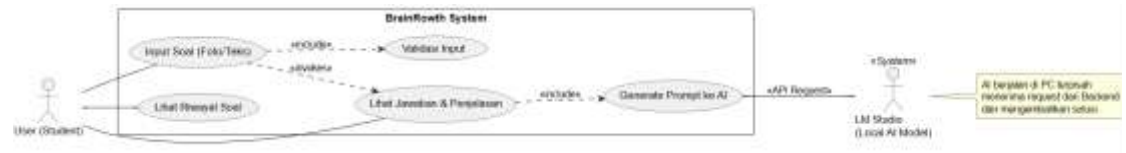
- **MT-01** — Backend harus menggunakan struktur kode modular sehingga prompt atau pemrosesan AI dapat diperbarui dengan mudah.
- **MT-02** — Model AI dapat diganti (misal DeepSeek → Qwen) tanpa perubahan besar di sisi aplikasi.
- **MT-03** — UI dibuat dengan Jetpack Compose agar mudah dikembangkan di versi selanjutnya.

5.4.5 Portability

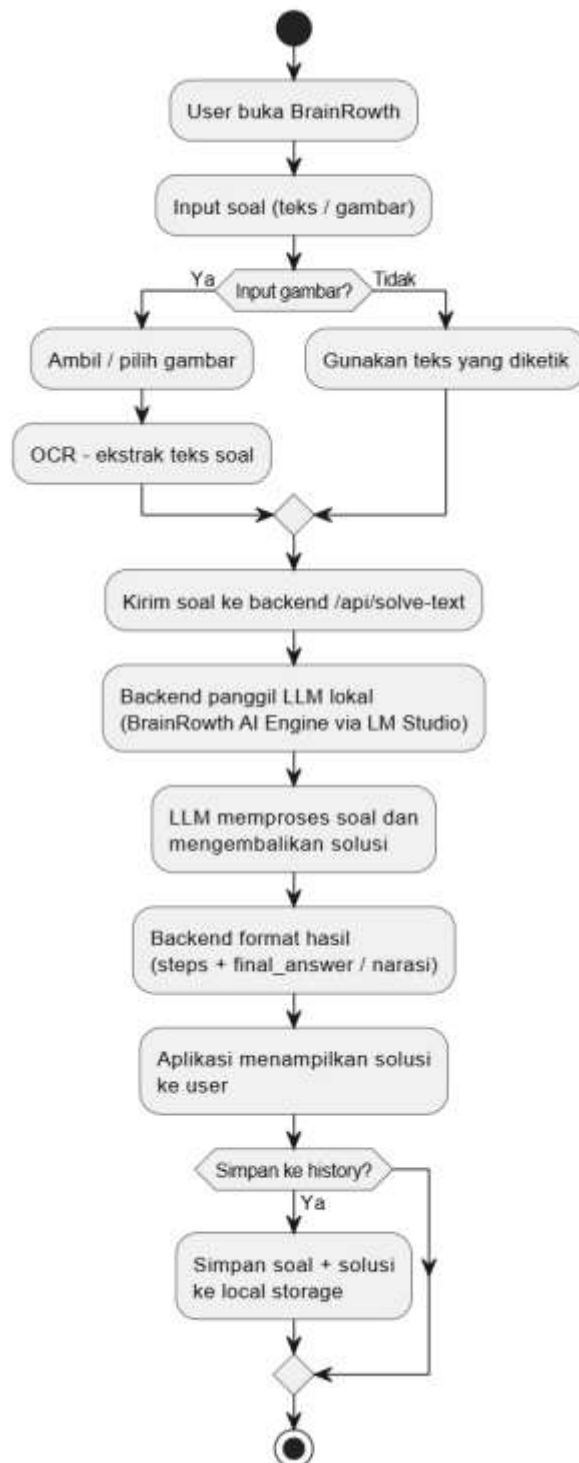
- **PT-01** — Aplikasi harus dapat dijalankan minimal pada Android 7.0 hingga versi terbaru.
- **PT-02** — Backend harus dapat dijalankan di Windows atau Linux.
- **PT-03** — Sistem dapat dipindahkan ke jaringan WiFi mana pun tanpa konfigurasi tambahan selain mengganti IP backend.

6 Other Requirements

USE CASE



FLOWCHART



Activity Diagram



Kamus Data

1. Data History

| Nama Atribut | Tipe Data | Deskripsi | Contoh Data |
|--------------|-----------|-------------------------------|-------------|
| id (PK) | String | ID unik entri history. Dibuat | "1" |

| Nama Atribut | Tipe Data | Deskripsi | Contoh Data |
|---------------------|------------------|---|---|
| | | otomatis oleh aplikasi. | |
| question | String | Soal matematika yang dimasukkan user (hasil teks atau OCR). | "2x + 5 = 11" |
| steps | Array<String> | Daftar langkah-langkah solusi yang dikirim backend. | ["Kurangi 5 dari kedua sisi", "Bagi 2"] |
| final_answer | String | Jawaban akhir dari proses AI. | "x = 3" |
| created_at | Long (Timestamp) | Waktu penyimpanan entri ke history (Epoch ms). | 1709235000000 |

Appendix A: Glossary

BrainRowth - Nama aplikasi mobile yang digunakan untuk membantu menyelesaikan soal matematika dengan bantuan AI lokal.

BrainRowth AI Engine - Sebutan untuk model Large Language Model (LLM) yang dijalankan secara lokal melalui LM Studio dan digunakan untuk memproses soal matematika.

LLM (Large Language Model) - Model kecerdasan buatan berskala besar yang mampu memahami dan menghasilkan teks, dalam konteks ini digunakan untuk menyelesaikan dan menjelaskan soal matematika.

LM Studio - Aplikasi desktop yang digunakan untuk menjalankan model LLM secara lokal, menyediakan API yang kompatibel dengan OpenAI sehingga dapat diakses oleh backend BrainRowth.

Backend - Komponen server (Node.js/Express) yang berjalan di laptop pengembang, bertugas menerima soal dari aplikasi Android, meneruskannya ke LLM, lalu mengembalikan hasil ke aplikasi.

Frontend / Aplikasi Android - Aplikasi mobile yang digunakan oleh pengguna (misalnya instruktur) untuk memasukkan soal, melihat hasil solusi, dan mengelola history.

History - Riwayat soal dan solusi yang telah diproses oleh BrainRowth dan disimpan secara lokal di perangkat Android pengguna.

OCR (Optical Character Recognition) - Teknologi yang digunakan untuk mengenali teks dari gambar (misalnya foto soal matematika) dan mengubahnya menjadi teks yang dapat diproses.

ML Kit - Library dari Google yang digunakan di sisi Android untuk melakukan OCR secara lokal tanpa koneksi internet.

JSON (JavaScript Object Notation) - Format data yang digunakan untuk bertukar informasi antara backend dan aplikasi Android, misalnya untuk mengirim langkah-langkah penyelesaian dan jawaban akhir.