

SQAPLAN-
SE4302-01

DOKUMEN RENCANA
PENJAMINAN KUALITAS
PERANGKAT LUNAK

I Hear

Dipersiapkan oleh:


Muhammad	(1201222041)
Evi Fitriya	(1201222005)
Farhan Nugraha Sasongko Putra	(1201220449)
Muhammad Adib Firmansyah	(1201220012)
Shim Hyen Jin	(1201220442)
Farrel Gilang N. M.	(1201220022)

Program Studi S1 Rekayasa Perangkat Lunak –

Fakultas Informatika

Universitas Telkom Surabaya

2022

 UNIVERSITAS Telkom	Program Studi S1 RPL - Fakultas Informatika	Nomor Dokumen		Halaman
		SQAPLAN-xxx		<halaman>
		Revisi	<nomor revisi>	Tgl: <isi tanggal>

Daftar Perubahan

Tanggal	Versi	Deskripsi	Direvisi oleh	Diperiksa oleh	Disetujui oleh
<dd/mm/yy>	<x.x>	<details>	<name>	Muhammad, Evi Fitriya, Shim Hyen Jin, Muhammad Adib Firmansyah, Farhan Nugraha S. P, Farrel Gilang N. M.	

Daftar Isi

Daftar Perubahan.....	1
Daftar Isi	2
1. Pendahuluan	3
1.1 Tujuan	3
1.2 Cakupan	Error! Bookmark not defined.
1.3 Referensi	3
1.4 Gambaran Umum	Error! Bookmark not defined.
2. Quality Goals	4
3. Manajemen	5
3.1 Organisasi.....	5
3.2 Tugas dan Tanggung Jawab	Error! Bookmark not defined.
4. Metric	6
4.1 Product Metric.....	6
4.2 Project Metric.....	Error! Bookmark not defined.
4.3 Process Metric.....	Error! Bookmark not defined.
5. Rencana Aktivitas Review	9
5.1 Requirements Document Review	9
5.2 Design inspection, dst.	9
6. Rencana Pengujian Perangkat Lunak.....	9

1. Pendahuluan

Dokumen Software Quality Assurance Plan (SQA Plan) ini dirancang untuk memberikan panduan menyeluruh tentang metodologi dan prosedur yang akan digunakan dalam menjamin kualitas perangkat lunak iHear. Perangkat lunak iHear merupakan aplikasi inovatif yang bertujuan untuk memfasilitasi komunikasi antara teman tuli dan teman dengar melalui teknologi translasi bahasa isyarat ke teks dan fitur-fitur lainnya. Rencana ini berfokus pada proses penjaminan kualitas yang mencakup berbagai aktivitas, seperti pengujian, validasi, dan pemantauan pengembangan aplikasi iHear. Dengan pendekatan ini, SQA Plan memastikan bahwa perangkat lunak memenuhi standar kualitas yang telah ditetapkan serta ekspektasi pengguna, baik dari segi fungsionalitas, keandalan, maupun keamanan. Dokumen ini juga mencakup langkah-langkah mitigasi risiko yang berkaitan dengan kebutuhan perangkat keras dan lunak, seperti penggunaan kamera untuk translasi bahasa isyarat, pemrosesan data melalui protokol API, serta penyimpanan data yang efisien dan aman.

Dengan demikian, tim pengembang memiliki acuan jelas untuk menjalankan proses pengembangan secara konsisten dan terarah. Melalui penerapan rencana ini, diharapkan aplikasi iHear tidak hanya menjadi solusi inovatif bagi kesenjangan komunikasi, tetapi juga menjadi perangkat lunak yang dapat diandalkan dalam berbagai situasi penggunaan.

1.1 Tujuan

Dokumen ini bertujuan memastikan kualitas perangkat lunak iHear melalui berbagai proses seperti pengujian, validasi, dan pemantauan pengembangan. Fokusnya adalah menjamin perangkat lunak memenuhi standar kualitas, keandalan, fungsionalitas, dan keamanan yang ditetapkan, sekaligus mengatasi risiko teknis yang terkait dengan kebutuhan perangkat keras dan lunak, seperti penggunaan kamera, pemrosesan data melalui API, dan penyimpanan data yang aman.

Selain itu, dokumen ini dirancang untuk memberikan panduan terarah bagi tim pengembang dalam menjalankan proses pengembangan secara konsisten dan efisien. Melalui penerapan SQA Plan ini, aplikasi iHear diharapkan tidak hanya menjadi solusi inovatif untuk mendukung komunikasi antara teman tuli dan teman dengar, tetapi juga menjadi perangkat lunak yang andal dan efektif dalam berbagai skenario penggunaan.

1.2 Ruang Lingkup

Cakupan ruang lingkup dokumen ini termasuk meliputi kualitas SRS dan SDD berdasarkan hasil temuan dalam review SQA, termasuk major dan minor issue.

1.3 Referensi

- Dokumen SRS Ihear
- Dokumen SDD Ihear
- Verification Ceklist SRS Ihear
- Verification Ceklist SDD Ihear
- Laporan Review Ihear

1.4 Sistematika Dokumen

Penjelasan tentang bagaimana dokumen ini disusun terdiri dari

1. Pendahuluan

Bagian ini menjelaskan tujuan utama dari dokumen penjaminan kualitas perangkat lunak (SQA Plan), ruang lingkup proyek atau modul perangkat lunak yang dicakup, dokumen referensi yang digunakan sebagai standar acuan seperti ISO 9001 atau IEEE serta dokumen proyek seperti SRS, SDD dll.

2. Quality Goals

Bagian ini menetapkan tujuan kualitas yang ingin dicapai pengembangan perangkat lunak, baik dari aspek fungsional maupun non-fungsional pengukuran kuantitatif digunakan untuk mengukur sejauh mana perangkat lunak memenuhi standar kualitas yang telah ditentukan.

3. Manajemen

Menjelaskan struktur tim yang bertanggung jawab atas penjaminan kualitas, serta pembagian tugas dan tanggung jawab masing – masing anggota tim. Hal ini bertujuan untuk memastikan setiap bagian dalam pengembangan perangkat lunak terkelola dengan baik dan sesuai dengan standar yang ditetapkan.

4. Metrik

Bagian ini membahas metrik yang akan digunakan untuk mengukur dan memonitor kualitas perangkat lunak selama pengembangan. Metrik ini meliputi metrik produk (seperti jumlah bug atau pengembangan pengguna), metrik proyek (kemajuan waktu dan biaya), dan metrik proses (kecepatan pengembangan dan ditetapkan).

5. Rencana aktivitas review

Menjelaskan kegiatan review yang akan dilakukan pada setiap tahap pengembangan perangkat lunak, termasuk review dokumen spesifikasi, desain perangkat lunak, serta inspeksi kode. Proses ini bertujuan untuk mengidentifikasi dan memperbaiki kesalahan sejak dini

6. Rencana pengujian perangkat lunak

Bagian ini merinci rencana pengujian yang mencakup unit testing, integration testing dan system testing. Setiap jenis pengujiannya tujuan dan metode yang berbeda, dengan jadwal pelaksanaan dan penanggung jawaban yang jelas untuk memastikan perangkat lunak bebas dari cacat

2. Quality Goals {PRODUCT}

Quality goals merepresentasikan *quality requirement* yang harus dimiliki perangkat lunak yang akan dikembangkan. Pengukuran kuantitatif biasanya digunakan daripada pengukuran kualitatif pada saat menentukan *quality goal* karena dapat memberikan penilaian yang lebih objektif kepada tim pengembang atas kinerja perangkat lunak selama proses pengembangan dan pengujian sistem. Perlu dicatat bahwa satu *quality goal* tidak sepenuhnya sama dengan yang lain.

Qualitative Requirement	Quantitative Quality Goal yang Berhubungan
<i>Availability</i>	<i>Perangkat Lunak harus terus dapat beroperasi 7 hari perminggu, 24 jam per hari tanpa gagal</i>
<i>Reliability</i>	<i>Kegagalan dalam proses tranlansi memiliki toleransi 1 kata per sesi</i>
<i>Efficiency</i>	<i>Tampilan antarmuka praktis dan mudah digunakan</i>
<i>Portability</i>	<i>Perangkat Lunak dapat dipakai di Android dan IOS</i>

<i>Prodi SI RPL - Universitas Telkom</i>	<i>SQAPlan-xxx</i>	<i>Halaman 4 dari 15</i>
<i>Dokumen ini dan informasi yang ada di dalamnya adalah milik Prodi SI Rekayasa Perangkat Lunak- Universitas Telkom dan bersifat rahasia. Dilarang untuk mereproduksi dokumen ini tanpa diketahui oleh Program Studi SI Rekayasa Perangkat Lunak, Universitas Telkom</i>		

<i>Memory</i>	<i>menggunakan memori yang efisien agar tetap berjalan normal</i>
<i>Response Time</i>	<i>Perangkat Lunak mampu mentralasikan bahasa secara langsung, Perangkat Lunak mampu melakukan pemberitahuan jika ada suara dalam latensi maksimal 5 detik</i>
<i>Safety</i>	<i>Otorisasi dan Otentikasi, Pencegahan Serangan DDOS, Logging dan Monitoring, dan Pembaruan Keamanan.</i>
<i>Security</i>	<i>Perangkat lunak menggunakan standar enkripsi HTTPS</i>

3. Manajemen

3.1 Organisasi

Struktur organisasi SQA untuk proyek IHeard mencakup berbagai komponen utama yang bertanggung jawab dalam memastikan kualitas perangkat lunak. Setiap komponen memiliki peran penting dalam menjamin bahwa standar kualitas perangkat lunak tercapai.

- SQA Unit

Merupakan tim yang dibentuk secara khusus dengan tanggung jawab utama untuk menjalankan proses penjaminan kualitas perangkat lunak. Tim ini berperan memastikan bahwa semua standar kualitas terpenuhi selama seluruh siklus pengembangan perangkat lunak, termasuk pengujian dan validasi.

- Development Team

Adalah kelompok profesional yang secara langsung terlibat dalam proses pengembangan perangkat lunak. Mereka bertugas merancang, mengimplementasikan, dan memelihara perangkat lunak sesuai dengan kebutuhan yang telah ditentukan, dengan tetap mempertimbangkan standar kualitas yang ditetapkan oleh organisasi.

- SQA Committee

Komite ini terdiri dari para profesional berpengalaman dan praktisi yang memiliki keahlian serta minat khusus dalam bidang kualitas perangkat lunak. Mereka bertanggung jawab memberikan arahan strategis, menetapkan kebijakan, serta mengevaluasi efektivitas proses penjaminan kualitas untuk memastikan bahwa perangkat lunak yang dihasilkan memenuhi ekspektasi pengguna dan standar industri.

3.2 Tugas dan Tanggung Jawab

SQA Unit

Tugas:

- Mengembangkan dan mengimplementasikan rencana penjaminan kualitas perangkat lunak (Software Quality Assurance Plan).
- Melakukan pemantauan dan evaluasi terhadap proses pengembangan perangkat lunak untuk memastikan standar kualitas terpenuhi.
- Mengidentifikasi potensi risiko yang dapat memengaruhi kualitas perangkat lunak dan mengusulkan langkah mitigasi.
- Melakukan audit kualitas secara berkala untuk memvalidasi kepatuhan terhadap prosedur dan standar.

Tanggung Jawab:

- Memastikan seluruh aktivitas pengembangan perangkat lunak mengikuti standar kualitas yang telah ditetapkan.
- Memberikan pelaporan dan rekomendasi perbaikan kepada tim pengembang dan manajemen.

Development Team**Tugas:**

- Mengembangkan perangkat lunak sesuai dengan spesifikasi kebutuhan yang telah ditentukan.
- Melakukan pengujian internal pada setiap modul atau fitur yang dikembangkan untuk memastikan fungsionalitasnya.
- Berkolaborasi dengan SQA Unit untuk memastikan bahwa perangkat lunak memenuhi kriteria kualitas.
- Melakukan perbaikan terhadap cacat atau bug yang ditemukan selama pengujian.

Tanggung Jawab:

- Menghasilkan perangkat lunak yang andal, efisien, dan sesuai dengan kebutuhan pengguna.
- Menjaga konsistensi kode dan dokumentasi selama proses pengembangan.

SQA Committee**Tugas:**

- Menyusun kebijakan dan pedoman penjaminan kualitas yang akan diterapkan pada proyek pengembangan perangkat lunak.
- Memberikan konsultasi dan masukan strategis terkait kualitas perangkat lunak kepada organisasi.
- Melakukan review terhadap proses dan hasil pengembangan perangkat lunak untuk memastikan kepatuhan terhadap standar kualitas.
- Mengevaluasi efektivitas metode dan alat penjaminan kualitas yang digunakan.

Tanggung Jawab:

- Memastikan bahwa kebijakan kualitas diimplementasikan dengan baik di seluruh tahap pengembangan perangkat lunak.
- Mendukung organisasi dalam mencapai tujuan kualitas perangkat lunak, termasuk kepuasan pengguna dan kepatuhan terhadap regulasi industri.

4. Metric

4.1 Product Metric

Product metric adalah metrik yang digunakan untuk mengukur kualitas dan karakteristik produk perangkat lunak, baik dari segi fungsionalitas maupun non-fungsionalitas. Fokusnya adalah pada hasil akhir perangkat lunak.

4.1.1 Reliability

1. *Mean Time to Failure (MTTF)*
Mean Time to Failure (MTTF) adalah waktu rata-rata yang diharapkan sebelum suatu sistem atau produk yang tidak bisa diperbaiki mengalami kegagalan pertama.
2. *Mean Time to Repair (MTTR)*

Prodi S1 RPL - Universitas Telkom	SQAPlan-xxx	Halaman 6 dari 15
Dokumen ini dan informasi yang ada di dalamnya adalah milik Prodi S1 Rekayasa Perangkat Lunak-Universitas Telkom dan bersifat rahasia. Dilarang untuk mereproduksi dokumen ini tanpa diketahui oleh Program Studi S1 Rekayasa Perangkat Lunak, Universitas Telkom		

Mean Time to Repair (MTTR) adalah waktu rata-rata yang diperlukan untuk memperbaiki sistem atau produk setelah terjadi kerusakan. Ini menunjukkan seberapa cepat kerusakan biasanya diperbaiki.

4.1.2 Usability

1. Task Success Rate

$$TSR = (\text{Jumlah tugas yang berhasil diselesaikan} / \text{Total jumlah tugas yang diberikan}) \times 100\%$$
2. Time on Task
3. Error Rate

$$\text{Rumus} = \text{Jumlah Kesalahan} / \text{Total Interaksi atau tindakan pengguna}$$
4. Sistem Usability Scale
5. Customer Satisfaction Score (CSAT)

$$CSAT = (\text{Jumlah pengguna yang puas} / \text{Total jumlah pengguna yang disurvei}) \times 100\%$$
6. Retention Rate

$$\text{Rumus} = (\text{Jumlah pengguna yang kembali} / \text{Jumlah Penggunaan Awal}) \times 100\%$$
7. Navigation Efficiency

$$\text{Rumus} = (\text{Jumlah langkah optimal} / \text{Jumlah langkah aktual}) \times 100\%$$

4.1.3 Correctness

1. Mean Time to Failure (MTTF)

$$MTTF = \text{Total Waktu Operasi} / \text{jumlah kejadian kegagalan}$$
2. Task Success Rate

$$TSR = (\text{Jumlah tugas yang berhasil diselesaikan} / \text{Total tugas yang diberikan}) \times 100\%$$

4.1.4 Security

1. Vulnerability Denisty

$$\text{Vulnerability Denisty} = (\text{Jumlah kerentanan} / \text{KLOCK})$$
2. MTTD

$$MTTD = (\text{Total waktu deteksi} / \text{Jumlah insiden})$$
3. MTTR

$$MTTR = (\text{Total waktu perbaikan} / \text{jumlah insiden})$$
4. Patch management efficiency

$$\text{Patch management efficiency} = (\text{Jumlah keterangan yang dipatch} / \text{total kerentanan}) \times 100\%$$

4.1.5 Efisiensi

1. Energy performance index EPI

$$EPI = (\text{Energi yang digunakan} / \text{Output atau performa})$$
2. Specific energy consumption SEC

$$SEC = (\text{Energi total yang digunakan} / \text{jumlah output})$$

4.2 Proses Metric

Process metric adalah metrik yang digunakan untuk mengukur efisiensi dan efektivitas proses pengembangan perangkat lunak. Fokusnya adalah pada langkah-langkah dan metode yang digunakan selama pengembangan.

Prodi S1 RPL - Universitas Telkom	SQAPlan-xxx	Halaman 7 dari 15
Dokumen ini dan informasi yang ada di dalamnya adalah milik Prodi S1 Rekayasa Perangkat Lunak- Universitas Telkom dan bersifat rahasia. Dilarang untuk mereproduksi dokumen ini tanpa diketahui oleh Program Studi S1 Rekayasa Perangkat Lunak, Universitas Telkom		

Metric	Definisi	Tujuan
Defect Density	Jumlah cacat tiap 1000 baris kode	Mengukur kualitas kode
Defect Removal Efficiency	$(\text{Cacat yang ditemukan selama pengujian} \div \text{Total cacat}) \times 100$	Mengukur efektifitas pengujian
Mean Time Between Failures	Rata-rata waktu antara kegagalan berturut-turut	Mengukur reliabilitas perangkat lunak

4.3 Project Metric

Project metric adalah metrik yang digunakan untuk memantau dan mengontrol kinerja proyek secara keseluruhan. Fokusnya adalah pada pengelolaan proyek, seperti waktu, biaya, dan sumber daya.

Metric	Definisi	RUMUS	Tujuan
Scheduler Performance Index (SPI)	Mengukur efisiensi proyek dalam menyelesaikan tugas sesuai jadwal.	$\text{Earned Value} / \text{Planned Value}$	Memastikan proyek berjalan sesuai rencana waktu.
Cost Performance Index (CPI)	Mengukur efisiensi penggunaan anggaran proyek	$\text{Earned value} / \text{Actual Cost}$	Memastikan proyek sesuai dengan alokasi biaya yang direncanakan
Effort Variance (EVa)	Mengukur perbedaan antara usaha yang direncanakan (planned effort) dengan usaha aktual (actual effort)	$(\text{Planned Effort} - \text{Actual Effort}) / \text{Planned Effort}$	Mengevaluasi efisiensi penggunaan waktu tim
Requirements Stability	Mengukur jumlah perubahan pada kebutuhan sistem selama siklus hidup proyek	$\text{Kebutuhan yang Tidak Berubah} / \text{Total Kebutuhan}$	Menilai stabilitas kebutuhan dan dampaknya pada jadwal dan biaya.
Milestone Adherence	Persentase penyelesaian milestone sesuai dengan jadwal yang telah ditentukan.	$\text{Milestone yang Dicapai Tepat Waktu} / \text{Total Milestone}$	Memastikan proyek mencapai tonggak penting tepat waktu.
Team Productivity	Jumlah output yang dihasilkan oleh tim dalam satuan waktu tertentu (misalnya, baris kode per jam, modul yang diselesaikan per sprint).	$\text{Jumlah Output} / \text{Total Waktu yang Dhabiskan}$	Mengukur produktivitas tim dalam proyek.
Risk Resolution Efficiency	Mengukur efisiensi tim dalam mengidentifikasi dan menyelesaikan risiko selama proyek	$\text{Risiko yang Terselesaikan} / \text{Total Risiko yang Diidentifikasi}$	Mengurangi dampak risiko pada jadwal dan biaya proyek

5. Rencana Aktivitas Review {STATIC TESTING}

Static Testing adalah metode pengujian perangkat lunak yang dilakukan tanpa menjalankan kode program. Pengujian ini berfokus pada pemeriksaan dokumen, kode sumber, atau artefak lainnya untuk mendeteksi kesalahan.

5.1 Requirements Document Review

Aktivitas review contoh : focus pada kebutuhan fungsional dan non fungsional

- **Review Kebutuhan Sistem (SRS)**
 - **Fokus :** Verifikasi kelengkapan fitur, evaluasi kejelasan alur proses dan validasi kebutuhan keamanan
 - **Tanggung Jawab :** Sistem Analis dan tim QA
- **Review Desain Sistem (SDD)**
 - **Fokus :** Pengujian kesesuaian desain arsitektur dengan kebutuhan, optimasi struktur data dan evaluasi antarmuka pengguna.
 - **Tanggung Jawab :** Tim developer dan desainer UX yang bekerja sama dengan tim QA

5.2 Design review SDD.

Memeriksa desain teknis yang dijelaskan dalam dokumen Software Design Description (SDD). Peninjauan ini bertujuan untuk memastikan desain arsitektur, data, dan modul sesuai dengan kebutuhan di SRS.

Fokus utamanya adalah:

- Arsitektur sistem: Memastikan modularitas, skalabilitas, dan efisiensi.
- Arsitektur Perangkat Lunak : Memastikan setiap komponen arsitektur software dapat ditelusuri kembali ke kebutuhan di SRSs serta memastikan desain mencakup langkah-langkah untuk melindungi data dari ancaman keamanan.
- Traceability : Memastikan bahwa semua kebutuhan dalam SRS telah dipenuhi secara menyeluruh oleh desain.
- Desain data: Memastikan konsistensi format data dan aliran data.
- Desain antarmuka pengguna (UI/UX): Memastikan desain sesuai dengan prinsip usability dan kompatibilitas.

6. Rencana Pengujian Perangkat Lunak {DYNAMIC TESTING}

Dynamic Testing adalah metode pengujian perangkat lunak yang dilakukan dengan menjalankan kode program. Tujuannya adalah untuk memeriksa apakah perangkat lunak berfungsi sesuai dengan spesifikasi.

Prodi S1 RPL - Universitas Telkom	SQAPlan-xxx	Halaman 9 dari 15
Dokumen ini dan informasi yang ada di dalamnya adalah milik Prodi S1 Rekayasa Perangkat Lunak-Universitas Telkom dan bersifat rahasia. Dilarang untuk mereproduksi dokumen ini tanpa diketahui oleh Program Studi S1 Rekayasa Perangkat Lunak, Universitas Telkom		

System Testing

Black Box Testing adalah metode pengujian perangkat lunak yang berfokus pada validasi fungsionalitas perangkat lunak tanpa memperhatikan struktur internal, logika kode, atau implementasi program.

Fungsi	Jenis / tipe pengujian	Jadwal	Tools	Prosedur khusus	Pengujian	Pemeriksaan hasil uji
Login	Black Box Testing	Minggu 1	Manual/Automated	Memeriksa apakah input valid dapat login dan input invalid ditolak	Tim QA	Lead QA
Bahasa Isyarat ke Teks	Black Box Testing	Minggu 1	Manual/Automated	Memastikan bahasa isyarat diterjemahkan menjadi teks yang benar	Tim QA	Lead QA
Kosa Kata	Black Box Testing	Minggu 2	Manual/Automated	Memastikan pengguna dapat menambahkan kosa kata baru	Tim QA	Lead QA
Ucapan ke Teks	Black Box Testing	Minggu 2	Manual/Automated	Memastikan ucapan pengguna dapat diubah menjadi teks yang benar	Tim QA	Lead QA
Komunitas	Black Box Testing	Minggu 3	Manual/Automated	Memastikan fitur komunitas dapat digunakan sesuai spesifikasi	Tim QA	Lead QA
Profil	Black Box Testing	Minggu 3	Manual/Automated	Memastikan data profil yang ditampilkan	Tim QA	Lead QA

				sesuai dengan data yang didaftarkan		
Menambahkan Kosakata	Black Box Testing	Minggu 4	Manual/Automated	Memastikan pengguna dapat menambahkan kosakata baru ke dalam aplikasi	Tim QA	Lead QA

Integration Testing

Integration Testing adalah metode pengujian perangkat lunak yang memverifikasi interaksi antara dua atau lebih modul atau komponen yang telah diintegrasikan untuk memastikan bahwa mereka bekerja bersama dengan benar.

Fungsi	Jenis / tipe pengujian	Jadwal	Tools	Prosedur khusus	Pengujian	Pemeriksaan hasil uji
Login	Integration Testing	Minggu ke-3 hingga ke-4	Postman	Memeriksa aliran data dari antarmuka pengguna ke API Firebase untuk autentikasi.	Tim QA	Lead QA
Bahasa Isyarat ke Teks	Integration Testing	Minggu ke-3 hingga ke-4	Postman	Memverifikasi pengiriman data video dari modul kamera ke sistem translasi dan hasilnya ditampilkan.	Tim QA	Lead QA
Kosa Kata	Integration Testing	Minggu ke-3 hingga ke-4	Postman	Memastikan data kosa kata baru	Tim QA	Lead QA

				tersimpan dengan benar di database dan dapat diakses oleh modul lain.		
Ucapan ke Teks	Integration Testing	Minggu ke-3 hingga ke-4	Postman	Memverifikasi pengiriman data suara dari modul mikrofon ke sistem konversi teks dan hasilnya ditampilkan .	Tim QA	Lead QA
Komunitas	Integration Testing	Minggu ke-3 hingga ke-4	Postman	Memeriksa pengiriman data posting dan komentar dari modul komunitas ke database SQL.	Tim QA	Lead QA
Profil	Integration Testing	Minggu ke-3 hingga ke-4	Postman	Memastikan modul profil berkomunikasi dengan database untuk pembaruan informasi pengguna.	Tim QA	Lead QA

Menambahkan Kosakata	Integration Testing	Minggu ke-3 hingga ke-4	Postman	Memastikan bahwa modul penambahan kosakata dapat menyimpan data ke database dengan benar.	Tim QA	Lead QA
----------------------	---------------------	-------------------------	---------	---	--------	---------

Unit Testing

Unit testing adalah proses pengujian perangkat lunak yang dilakukan pada bagian terkecil dari kode program, yaitu unit.

Bagian yang diuji	Jenis/tipe pengujian	Jadwal	Tools	Prosedur khusus	Penguji	Pemeriksa Hasil Uji
Fungsi Login	Unit Testing	Minggu 1	Flutter unit test	Verifikasi input valid/invalid	Tim QA	Lead QA
Fungsi Bahasa isyarat ke teks	Unit Testing	Minggu 1	Flutter unit test	Memastikan bahasa isyarat dapat ditranslate menjadi teks	Tim QA	Lead QA
Fungsi kosa kata	Unit Testing	Minggu 2	Flutter unit test	Dapat menambahkan kosa kata baru pada aplikasi	Tim QA	Lead QA
Fungsi Ucapan ke teks	Unit Testing	Minggu 2	Flutter unit test	Memastikan bahasa yang diucapkan berubah menjadi teks	Tim QA	Lead QA
Fungsi Komunitas	Unit Testing	Minggu 3	Flutter unit test	Memastikan komunitas dapat bekerja dengan seharusnya	Tim QA	Lead QA

Fungsi Profil	Unit Testing	Minggu 3	Flutter unit test	Menampilkan data yang sesuai dengan yang didaftarkan	Tim QA	Lead QA
Fungsi Menambahkan kosakata	Unit Testing	Minggu 4	Flutter unit test	Dapat menambahkan kosakata baru kedalam aplikasi	Tim QA	Lead QA

7. Lampiran

Muhammad : Bertanggung jawab bagian Software Desain dan Data Desain pada SDD, Consistent dan Ranked for importance and/or stability pada SRS

Muhammad Adib Firmansyah : Bertanggung jawab bagian SDD, user Interface Design dan Traceability, untuk bagian SRS Correct

Evi Fitriya : Bertanggung jawab bagian System Architecture dan Software Design pada SDD, dan bagian correctness beserta unambiguous pada SRS.

Farrel Gilang N. M. : Bertanggung jawab bagian User Interface Design dan Data Design pada SDD, dan bagian completeness beserta consistent pada SRS.

Farhan Nugraha S. P. : Bertanggung jawab bagian Software Design pada SDD, dan bagian completeness beserta unambiguous pada SRS.

Shim Hyen Jin : Bertanggung jawab bagian SDD Data Design, untuk bagian SRS Modifiable dan Traceable