

PENGEMBANGAN APLIKASI BERLATIH MEMBACA CEPAT BERBAHASA INGGRIS BERBASIS *PROGRESSIVE WEB APP* DENGAN METODE *PROTOTYPING*

Muhammad Fajar Alfath^{*1}, Lutfi Fanani², Agi Putra Kharisma³

^{1,2,3}Universitas Brawijaya, Malang

Email: ¹mfalfath25@student.ub.ac.id, ²lutfifanani@ub.ac.id, ³agi@ub.ac.id

^{*}Penulis Korespondensi

(Naskah masuk: 09 November 2023, diterima untuk diterbitkan: 10 Oktober 2024)

Abstrak

Membaca merupakan proses memahami, menafsirkan, dan mengekstraksi informasi dari serangkaian kata yang memerlukan waktu. Melalui internet, akses membaca semakin luas dengan adanya konten seperti artikel, *e-book*, jurnal, dan lainnya yang mayoritas disajikan dalam bahasa Inggris. Melalui survei terhadap 30 mahasiswa mengenai kegiatan membaca cepat dan relevansinya terhadap bahasa Inggris, sebanyak 76.7% darinya tidak mengetahui kecepatan membaca mereka dan 56.7% mengaku sering membaca teks berbahasa Inggris. Ada baiknya jika kemampuan membaca cepat dilatih dengan penggunaan bahasa Inggris untuk membangun motivasi terhadap literasi digital yang luas. Melalui pengamatan tersebut, dikembangkan aplikasi berbasis *progressive web app* (PWA) menggunakan metode SDLC *prototyping*. Melalui proses *prototyping* sebanyak 2 kali iterasi, dihasilkan aplikasi PWA '*Speed Reader*' yang bertujuan untuk membantu berlatih membaca cepat secara senyap dengan materi bacaan berbahasa Inggris. Hasil evaluasi pengujian menunjukkan bahwa aplikasi ini kompatibel pada perangkat mobile dan dapat berjalan secara *cross-browser*. Setiap fitur berhasil dijalankan dengan tingkat validitas 100% pada pengujian efektivitas. Melalui pengujian efisiensi, didapatkan waktu penyelesaian tugas sebesar 0,156 tugas per detik. Hasil pengujian kepuasan pengguna melalui pengukuran *System Usability Scale* (SUS) mencapai nilai 84 dengan kategori grade B dan tergolong *acceptable*. Dengan demikian, dapat disimpulkan bahwa pengguna berhasil mencapai tujuan mereka dan aplikasi ini layak untuk digunakan.

Kata kunci: *membaca cepat, bahasa Inggris, progressive web app, prototyping, pengujian usability*

THE DEVELOPMENT OF A PROGRESSIVE WEB APP TO TRAIN SPEED READING IN ENGLISH USING THE PROTOTYPING METHOD

Abstract

Reading is a process of understanding, interpreting, and extracting information from a series of words that takes time. The internet has significantly broadened reading access, providing abundant content like articles, e-books, journals, predominantly in English. A survey conducted among 30 students explored their engagement in speed reading and its relevance to English. Surprisingly, 76.7% of participants were unaware of their reading speed, while 56.7% acknowledged frequent reading of English texts. To foster extensive digital literacy, it is advisable to train speed reading skills specifically using the English language. Based on this observation, a Progressive Web App (PWA) was developed utilizing the SDLC prototyping method. The development of an app called 'Speed Reader' was aimed to facilitate the practice of silent speed reading with English materials. Evaluation tests demonstrated the application's compatibility with mobile devices and its cross-browser functionality. Each feature achieved a 100% validity rate in effectiveness testing. Moreover, efficiency testing revealed a task completion time of 0.156 tasks per second. User satisfaction assessments, conducted using the SUS method, yielded a score of 84, categorizing it as grade B and considered acceptable. It can be concluded that the users have successfully achieved their goals, and this application is deemed suitable.

Keywords: *speed reading, english language, progressive web app, prototyping, usability testing*

1. PENDAHULUAN

Kegiatan membaca merupakan keterampilan yang telah dilatih sejak kecil. Dengan membaca,

seseorang dapat mengasah pemikiran yang kritis dengan mengembangkan kerangka berfikir yang sistematis. Pembentukan pemikiran tersebut

didapatkan melalui proses memahami, menafsirkan, dan mengekstraksi informasi dari serangkaian kata.

Pertumbuhan era modern yang masif merubah kebiasaan membaca konvensional menjadi digital. Dibutuhkan literasi digital yang baik untuk menyikapi fenomena ledakan informasi. Literasi digital menjadi kunci untuk berdaya saing tinggi dan unggul di era globalisasi (Rahmadi dan Hayati, 2020).

Internet menjadi sumber yang sering dirujuk oleh mahasiswa karena memiliki peranan yang besar dalam ketersediaan informasi. Mayoritas konten di internet dimuat dalam bentuk teks, tersusun dalam berbagai format, seperti *e-book*, jurnal penelitian, berita, artikel, dan masih banyak lagi. Penting bagi pembaca untuk mengolah informasi secara efisien dengan membangun keterampilan membaca cepat. Pembaca yang kurang kompeten akan cenderung membaca lebih sedikit sehingga menurunkan produktivitas dalam membaca.

Bahasa kerap menjadi batasan linguistik bagi pembaca yang mengakses informasi melalui internet. Diperlukan bahasa universal yang dapat menjadi bahasa pengantar di antara banyaknya bahasa lain. Melalui data yang dikumpulkan oleh (Statista, 2023), Bahasa Inggris merupakan bahasa yang paling populer sebagai konten yang dimuat di internet dengan estimasi sekitar 58.8% dari total seluruh situs web. Hal ini memperkuat bahwasanya Bahasa Inggris digunakan sebagai bahasa pengantar atau sebagai *lingua franca*. Ada baiknya jika seorang pembaca yang merupakan pembelajar bahasa kedua (L2) terbiasa dengan membaca konten berbahasa Inggris. Menurut (Nushi, 2017) kegiatan tersebut dapat mengembangkan keterampilan pengenalan kata yang diharapkan dapat memperluas wawasan dan kemampuan dalam memahami informasi di luar dari bahasa ibu (L1).

Mengamati penelitian meta-analisis oleh (Brysaert, 2019), ditemukan bahwa kecepatan membaca rata-rata untuk teks nonfiksi adalah 238 kata per menit (KPM), sementara pada teks fiksi mencapai 260 KPM untuk teks berbahasa Inggris. Ditemukan juga perbedaan yang signifikan dalam kecepatan membaca antara pembaca L2 dengan pembaca L1. Pembaca L2 cenderung memiliki kecepatan membaca yang rata-ratanya di bawah angka tersebut, bahkan tidak jarang ditemukan kecepatan di bawah 100 KPM. Melalui survey yang dilakukan oleh peneliti terhadap 30 responden mahasiswa Universitas Brawijaya, sebanyak 76.7% responden tidak mengetahui kecepatan membaca mereka. Sebesar 56.7% responden juga merasa bahwa mereka cukup sering membaca teks berbahasa Inggris dan 93% darinya menganggap penggunaan Bahasa Inggris di internet merupakan hal yang penting. Kemahiran membaca cepat diperlukan bagi pelajar untuk menciptakan motivasi membaca, meningkatkan penalaran deskriptif, dan khususnya untuk menerima informasi dalam jumlah yang banyak (Sirait, Hutauruk dan Herman, 2020).

Saat ini teknologi yang dikembangkan untuk melatih membaca cepat masih belum tersegmentasi secara penuh untuk kategori pengguna tertentu (Abdul-Rab et al., 2023). Pengembangan perangkat lunak yang ada saat ini juga banyak yang mengadopsi metode *rapid serial visual representation* (RSVP), membantu membaca dengan bantuan mekanis dengan cara memotong kalimat dan memperlihatkan kata demi kata (Acklin dan Papesch, 2017). Metode ini perlu dikaji lebih lanjut karena kurang sesuai dengan keadaan membaca secara normal, dari kiri ke kanan.

Dalam upaya untuk meningkatkan literasi digital melalui kegiatan berlatih membaca cepat dan mengimplementasi pendekatan metode membaca yang berbeda, peneliti merumuskan pengembangan aplikasi web bernama *Speed Reader*. Aplikasi ini ditujukan untuk melatih kemampuan membaca cepat seperti melakukan simulasi membaca cepat berdasarkan konfigurasi pengguna pada kecepatan tertentu, mengubah konfigurasi tampilan visual simulasi, dan panduan terkait konsep membaca cepat melalui fitur yang akan dikembangkan. Akan digunakan metode pengembangan perangkat lunak *prototyping* dan melalui proses iterasi yang akan divalidasi melalui tahap evaluasi prototipe. Adanya iterasi diharapkan dapat memperbaiki capaian sistem sesuai dengan kebutuhan pengguna dan mengetahui apakah solusi yang disarankan layak untuk diterapkan (Lang dan Mjöberg, 2020). Teknologi *progressive web app* (PWA) dimanfaatkan karena memiliki fleksibilitas yang luas untuk berbagai orientasi layar dan dapat dijalankan pada berbagai perangkat dengan fitur serupa layaknya aplikasi *native*. PWA juga dapat mengeliminasi penggunaan bahasa pemrograman yang *platform-specific* (Majchrzak, Biørn-Hansen dan Grønli, 2018).

2. DASAR TEORI

2.1 Membaca Cepat

Membaca cepat memiliki definisi yaitu kegiatan membaca yang lebih cepat dari kecepatan normal secara efisien dengan tetap mementingkan pemahaman. Kemampuan ini dapat dilatih menggunakan teknik tertentu yang dapat dipelajari (Armstrong, 2015). Terdapat beberapa faktor yang mempengaruhi kemampuan membaca cepat, diantaranya adalah *subvocalization*, *backtracking*, dan fiksasi kata.

Subvocalization merupakan keadaan ketika seseorang membaca kembali kata atau frasa di dalam pikiran yang sebaiknya dihindari. *Backtracking* atau regresi pada saat membaca merupakan aktivitas membaca kembali kata atau kalimat yang telah dibaca sebelumnya. Hal ini tidak dianjurkan karena dapat mempengaruhi pemahaman konteks secara utuh. Terakhir, yaitu fiksasi kata yang merupakan kemampuan untuk mengarahkan dan memfokuskan mata pada titik tertentu secara akurat. Fiksasi diusahakan tidak dilakukan pada satu kata, melainkan

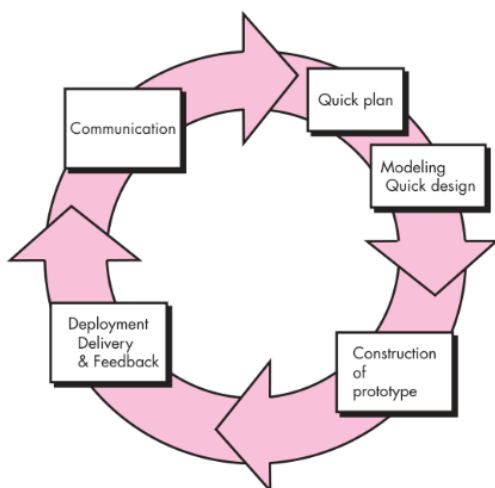
memanfaatkan penglihatan tepi untuk menangkap beberapa kata sekaligus dalam sekali fiksasi.

2.2 Progressive Web App

Progressive web app (PWA) merupakan istilah buatan Google untuk sebuah aplikasi web yang dibangun menggunakan teknologi khusus untuk berfungsi layaknya web dan juga aplikasi *native* (MDN, 2022). PWA bekerja dengan bantuan *Service Worker API* untuk menyimpan berkas data pada *cache storage* sehingga penggunaan aplikasi seterusnya dapat berjalan lebih intuitif. Setidaknya, PWA harus dapat dipasang, dibagikan melalui URL, bekerja secara *offline* atau dengan koneksi minimal dan menerapkan desain yang responsif.

2.3 Metode Prototyping

Metode *prototyping* dapat digunakan untuk membangun purwarupa dengan menyesuaikan kebutuhan dan spesifikasi yang dibutuhkan (Pressman, 2010). Sebuah prototipe akan melalui serangkaian proses perancangan, pengujian, evaluasi, dan perbaikan hingga dihasilkan prototipe yang sesuai kebutuhan dan dapat diterima. Pengerjaan dalam beberapa iterasi juga diharapkan dapat memudahkan pembangunan solusi produk yang direncanakan. Paradigma tahapan metode *prototyping* diperlihatkan pada Gambar 1.

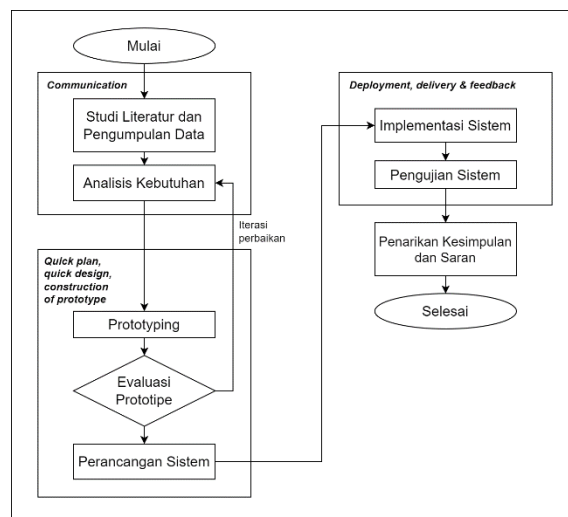


Gambar 1. Paradigma Model Prototyping (sumber: Pressman, 2010)

3. METODOLOGI

Untuk menyelesaikan masalah penelitian, telah dirancang metodologi untuk memfokuskan tahapan yang dikerjakan. Penerapan metode *prototyping* akan menyesuaikan paradigma model Pressman yang mencakup aspek *communication*, *quick plan*, *quick design*, *construction of prototype*, dan *deployment, delivery & feedback*. Tahap penelitian dimulai dari studi literatur, pengumpulan data penelitian, analisis kebutuhan, melakukan *prototyping*, mengevaluasi prototipe, merancang sistem mengimplementasi

sistem, menguji sistem, sampai penarikan kesimpulan. Pengambilan data primer didapat melalui hasil survey terhadap 30 mahasiswa Universitas Brawijaya menggunakan *convenience sampling* terkait konteks membaca cepat, Bahasa Inggris, dan relevansi penggunaannya di internet. Sementara, data sekunder didapat melalui studi literatur terhadap konsep membaca cepat dan referensi teknologi membaca cepat yang sudah ada sebelumnya. Tahapan penelitian diperjelas melalui diagram metodologi pada Gambar 2.



Gambar 2. Diagram Metodologi Penelitian

4. ANALISIS KEBUTUHAN

4.1 Gambaran Umum Sistem

Speed Reader merupakan aplikasi web berbasis PWA untuk meningkatkan kecepatan membaca secara senyap pada bahasa Inggris khusus untuk pembaca L2. Pengguna dapat melatih kemampuan membaca cepat pada teks berbahasa Inggris sesuai dengan level kemahiran bahasa mereka dan menguji pemahaman terkait bacaan melalui tes komprehensif. Pengguna juga dapat mempelajari konsep membaca cepat dan melakukan evaluasi terhadap progres latihan mereka. Tujuan utama *Speed Reader* adalah untuk melatih kemampuan membaca cepat melalui metode *meta-guiding* dan *chunk reading* dengan tampilan yang dirancang seperti membaca dengan natural, berbeda dengan teknik RSVP yang umum digunakan pada aplikasi berlatih membaca cepat terdahulu.

4.2 Identifikasi Aktor

Aktor merupakan entitas yang akan berkecimpungan dan berinteraksi langsung dengan sistem untuk mencapai tujuannya. Setidaknya terdapat beberapa aktor yang teridentifikasi yang diperlihatkan pada Tabel 1 **Error! Reference source not found..**

Tabel 1. Deskripsi Aktor

Aktor	Deskripsi
Guest	Seluruh pengguna yang belum mendaftar atau pengguna yang sudah terdaftar namun belum melakukan autentikasi.
Trainee	Merupakan aktor yang berinteraksi langsung pada aplikasi <i>Speed Reader</i> sebagai pengguna yang berlatih membaca cepat.

4.3 Daftar Kebutuhan Fungsional

Kebutuhan fungsional merupakan proses yang akan dikerjakan dan dapat dilakukan pada sebuah sistem. Kebutuhan didefinisikan secara rinci sebagai panduan terkait tujuan yang harus dikerjakan sistem dan aktor mana yang dapat melakukannya. Kebutuhan fitur yang ada didapat dari hasil survey sebagai data primer. Pemilihan kebutuhan fitur didasarkan pada batasan penelitian yang ditentukan, yaitu terkait berlatih membaca cepat secara senyap menggunakan konten berbahasa Inggris. Daftar kebutuhan fungsional ditampilkan pada Tabel 2.

Tabel 2. Daftar Kebutuhan Fungsional

No.	Kode	Nama Fungsi
1.	SR-F-01	Register
2.	SR-F-02	Login
3.	SR-F-03	Lihat Profile
4.	SR-F-04	Edit Profile
5.	SR-F-05	Lihat Progres Latihan
6.	SR-F-06	Lihat Panduan Membaca Cepat
7.	SR-F-07	Lihat Riwayat Latihan
8.	SR-F-08	Melakukan <i>Normal Test</i>
9.	SR-F-09	Melakukan <i>Blind Test</i>
10.	SR-F-10	Melakukan <i>Custom Test</i>
11.	SR-F-11	Edit Tampilan Simulasi
12.	SR-F-12	Logout

4.4 Daftar Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan batasan perilaku yang diharapkan dari sistem. Sistem harus memiliki *usability* SUS di atas 68, kompatibel terhadap *browser* modern, dan menjalankan fitur PWA. Batasan kebutuhan non-fungsional dapat dilihat pada Tabel 3.

Tabel 3. Daftar Kebutuhan Non-Fungsional

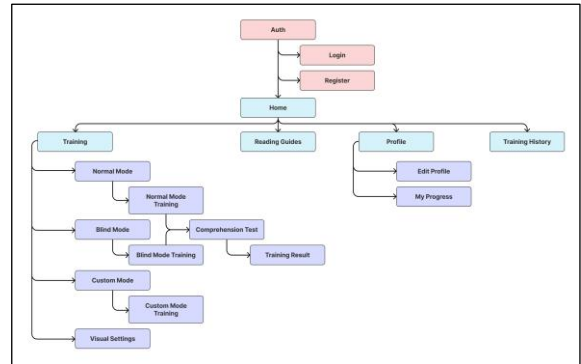
No.	Kode	Nama Fungsi
1.	SR-NF-01	<i>Usability</i>
2.	SR-NF-02	<i>Compatibility</i>

5. PERANCANGAN DAN IMPLEMENTASI

5.1 Prototyping

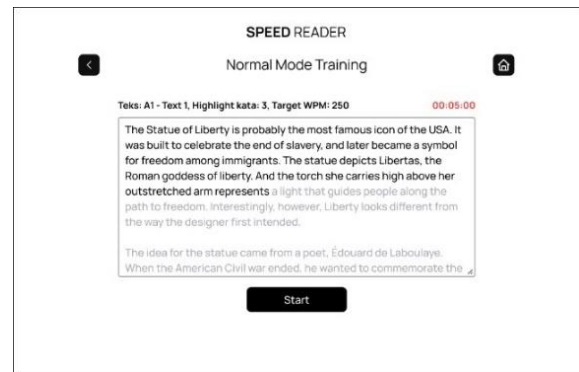
Pembuatan prototipe dilakukan pada tahap *prototyping* dan menghasilkan prototipe *high-fidelity*. Proses pembuatan prototipe terusun atas 2 kali iterasi, dimulai dari perancangan *sitemap*, perancangan antarmuka dan perancangan *screen flow*. Prototipe akan dievaluasi menggunakan metode *Cognitive Walkthrough*. Evaluasi prototipe akan memvalidasi masalah *usability* yang ada pada iterasi sebelumnya yang dilakukan oleh pakar di bidang membaca cepat.

Gambar 3 menampilkan perancangan diagram *sitemap* untuk memperjelas hierarki halaman.



Gambar 3. Perancangan Sitemap

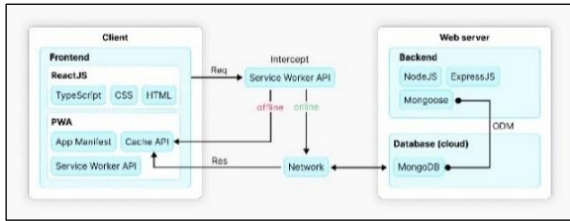
Berikutnya, visualisasi prototipe dirancang dalam bentuk antarmuka *high-fidelity*. Antarmuka akan memuat tata letak komponen *button*, *form input*, *image placeholder* dan lainnya. Kemudian, runtunan halaman antarmuka akan dihubungkan melalui perancangan *screen flow* untuk menggambarkan alur transisi halaman. Gambar 4 menampilkan contoh rancangan antarmuka halaman *Normal Mode Training* ketika pengguna melakukan simulasi latihan membaca cepat pada mode *Normal*.



Gambar 4. Perancangan Antarmuka Prototipe

5.2 Perancangan Arsitektur

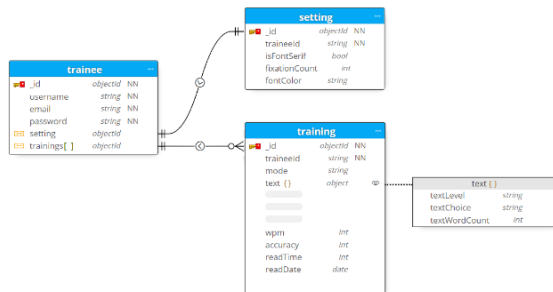
Arsitektur sistem tersusun dari dua bagian utama, yakni *client* dan *web server*. Bagian *frontend* dikelola menggunakan ReactJS, sementara pada bagian *backend* menggunakan NodeJS yang terhubung dengan platform basis data MongoDB. Fitur PWA akan dikonfigurasi oleh *Service Worker* API dan menerapkan strategi *Network First* pada *Cache Storage* API. Rancangan arsitektur sistem diperlihatkan pada Gambar 5.



Gambar 5. Perancangan Arsitektur

5.3 Implementasi Basis Data

Implementasi basis data secara NoSQL akan menggunakan platform MongoDB dalam bentuk dokumen yang disimpan dalam data dengan tipe BSON *Object*. Data yang dapat tersimpan pada MongoDB sangat fleksibel sehingga diperlukan *library* Mongoose sebagai *object data modeler*. Implementasi basis data diperlihatkan dalam bentuk *physical data model* pada Gambar 6.



Gambar 6. Physical Data Model

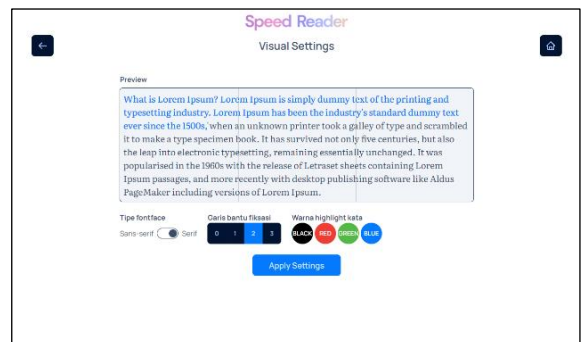
5.4 Implementasi Antarmuka

Bagian implementasi antarmuka dibangun menggunakan *library* ReactJS yang terdiri atas susunan komponen yang modular. Tampilan antarmuka menyesuaikan dengan rancangan *high-fidelity* yang sebelumnya digambarkan pada prototipe yang sudah final. Akan ditampilkan beberapa halaman dari hasil implementasi antarmuka. Implementasi halaman *Normal Mode Training* yang menggambarkan pengguna saat melakukan simulasi latihan membaca cepat pada mode *Normal*. Sistem dapat menampilkan animasi *meta-guiding* yang memandu pergerakan mata saat membaca. Proses *highlight* warna teks pada animasi *meta-guiding* akan ditampilkan secara bertahap, dengan jumlah *word chunk* yang ditentukan di awal. Terdapat juga garis bantu fiksasi yang menyarankan pergerakan fiksasi mata untuk menjaga ritme gerakan mata *saccadic* agar tidak berantakan. Halaman ini dapat dilihat pada Gambar 7.



Gambar 7. Halaman Normal Mode Training

Berikutnya, implementasi halaman *Visual Settings* yang menampilkan keadaan pada saat pengguna melakukan penyesuaian tampilan visual. Terdapat komponen tipe *fontface* untuk mengubah bentuk *font*, antara *serif* atau *sans-serif*. Terdapat juga komponen untuk memilih jumlah garis bantu fiksasi dan warna *highlight* animasi *meta-guiding*. Seluruh opsi pengaturan tampilan bersifat personal dan disesuaikan sendiri oleh pengguna. Hal ini dikarenakan preferensi pengguna yang berbeda sehingga kenyamanan saat latihan membaca dapat lebih maksimal. Implementasi halaman ini ditampilkan pada Gambar 8.



Gambar 8. Halaman Visual Settings

Lalu, contoh implementasi halaman *Reading Guides* pada saat pengguna melihat panduan membaca cepat yang berisi kurasi materi dan konsep dalam membaca cepat. Implementasi halaman ini ditampilkan pada Gambar 9.



Gambar 9. Halaman Reading Guides

6. PENGUJIAN

6.1 Pengujian Validasi

Dilakukan pengujian validasi yang termasuk dalam pengujian *black box* untuk mengetahui apakah aplikasi yang dibuat sudah sesuai yang direncanakan. Pengujian dilakukan dengan membangun sejumlah kasus uji. Pengujian validasi mendapatkan hasil valid 100% untuk total 17 kasus uji.

6.2 Pengujian Usability

Pengujian *usability* dilakukan untuk mengevaluasi aspek efektivitas, aspek efisiensi, dan aspek kepuasan pengguna pada aplikasi *Speed Reader*. Hasil pengujian *usability* didapat dari 12 skenario tugas yang diujikan terhadap 15 responden. Jumlah dan alur skenario tugas akan disesuaikan berdasarkan daftar kebutuhan fungsional yang ada. Kriteria responden yang terlibat merupakan mahasiswa aktif yang bukan merupakan penutur asli bahasa Inggris untuk mengetahui hasil pelatihan membaca cepat terhadap pembaca L2. Rangkaian pengujian *usability* dilakukan dengan menggunakan bantuan *tools* Useberry.

Pengujian pada aspek pertama, yaitu aspek efektivitas bertujuan untuk mengetahui ketercapaian atau kegagalan pengerjaan skenario tugas. Akan dihitung *completion rate* dari setiap tugas. Nilai 1 menandakan tugas berhasil dan nilai 0 menandakan tugas gagal untuk dilakukan. Didapatkan hasil perhitungan efektivitas sebesar 100%. Hasil pengujian efektivitas ditampilkan pada Tabel 4.

Tabel 4. Hasil Pengujian Usability (Efektivitas)

Total Tugas	Total Tugas Berhasil	$\frac{\text{Total Tugas Berhasil}}{\text{Total Tugas}} \times 100\%$
180	180	100%

Berikutnya pada pengujian efisiensi, akan diukur waktu penyelesaian setiap tugas untuk mendapatkan nilai efisiensi pengerjaan tugas. Didapatkan hasil efisiensi sebesar 0,156 tugas per detik. Hasil perhitungan pengujian aspek efisiensi ditampilkan pada Tabel 5.

Tabel 5. Hasil Pengujian Usability (Efisiensi)

$\sum_{j=1}^R \sum_{i=1}^N \frac{n_{ij}}{t_{ij}}$	NR	$\frac{\sum_{j=1}^R \sum_{i=1}^N n_{ij}}{NR}$
28,067	180	0,156

Selanjutnya pada pengujian kepuasan, didapatkan hasil SUS sebesar 84 dan sudah diatas nilai 68 untuk standar minimum bagi aplikasi web. Menurut indikator penilaian SUS, tingkat *acceptability*, skala *grade* dan rating *adjective* masing-masing dinilai *acceptable*, berskala *grade B*, dan dengan rating *excellent*. Hasil perhitungan pengujian aspek kepuasan ditampilkan pada Tabel 6.

Tabel 6. Hasil Pengujian Usability (Kepuasan)

Responden	Total Skor * 2.5	$\frac{\text{Total SUS}}{\text{Total Responden}}$

R1	90	
R2	95	
R3	82,5	
R4	92,5	
R5	82,5	
R6	75	
R7	80	
R8	75	84
R9	80	
R10	85	
R11	82,5	
R12	92,5	
R13	67,5	
R14	87,5	
R15	100	

6.3 Pengujian Compatibility

Pengujian *compatibility* dilakukan dengan menguji sistem dalam lingkungan yang berbeda melalui uji kompatibilitas *cross-browser* dan kemampuan sistem menerapkan teknologi PWA. *Browser* yang diuji antara lain, Edge 111, Firefox 111, Safari 16, Opera 96 dan Chrome 111. Adapun kriteria pengujian kompatibilitas PWA, yaitu inspeksi keberadaan *App Manifest* dan uji instalasi pada *real device*. Tabel 7 menampilkan hasil pengujian kompatibilitas pada dua metrik yang dinyatakan kompatibel.

Tabel 7. Hasil Pengujian Compatibility

No.	Metrik Pengujian	Hasil Pengujian
1	<i>Cross-browser</i>	Kompatibel
2	PWA	Kompatibel

7. KESIMPULAN

Penelitian ini bertujuan untuk mengetahui penerapan aplikasi web berbasis PWA untuk melatih kemampuan membaca cepat pada pembaca L2. Melalui studi literatur, pengumpulan data dan evaluasi dari prototipe sebanyak 2 kali iterasi, telah teridentifikasi 12 macam tujuan pengguna, diantaranya adalah melihat kurasi panduan membaca cepat, melakukan simulasi membaca cepat dengan metode *chunk reading* dan *meta-guiding* pada mode dan level kesulitan berstandar *Common European Framework of Reference (CEFR)*, mengubah tampilan simulasi latihan, dan melihat progres latihan. Hasil pengujian *compatibility* membuktikan *Speed Reader* berfungsi selayaknya aplikasi PWA dan kompatibel secara *cross-browser*. Pengujian *usability* menghasilkan sistem yang efektif dalam menyelesaikan setiap tugas dan dinilai *acceptable* berdasarkan penilaian SUS yang didapat, yaitu sebesar 84 dari 100.

Speed Reader menggunakan standar bahasa CEFR dengan 6 tingkatan level kemahiran bahasa yang ditentukan sendiri oleh pengguna. *Speed Reader* belum memiliki sistem rekomendasi level sehingga tidak dapat menentukan tingkat kesulitan secara otomatis berdasarkan hasil latihan pengguna. Penelitian teknologi membaca cepat berikutnya dapat

mengimplementasi standar bahasa lain atau dengan melatih pembaca pada konten bacaan selain bahasa Inggris. Sistem penilaian komprehensif pemahaman bacaan juga dapat dikembangkan menggunakan metode asesmen yang berbeda.

st-common-languages-on-the-internet/>
[Diakses 23 Juni 2023].

DAFTAR PUSTAKA

- ABDUL-RAB, S.D., ABDUL-HAMID, S., ROMLY, R., TOTI, U.S., dan MOHAMED, A.W.A., 2023. Transformational Development of Speed-Reading Technology: Tools, Machines and Software Applications. *Theory and Practice in Language Studies*, 13(6), pp.1452-1463.
- ACKLIN, D., dan PAPESH, M., 2017. Modern Speed-Reading Apps Do Not Foster Reading Comprehension. *The American Journal of Psychology*, 130(2), pp.183-199.
- ARMSTRONG, N., 2015. *Speed reading. The comprehensive guide to speed reading*. CreateSpace Independent Publishing Platform.
- BRYSAERT, M., 2019. How many words do we read per minute? A review and meta-analysis of reading rate. *Journal of Memory and Language*, 109, pp.1-23.
- LANG, F. dan MJÖBERG, A., 2020. *Prototyping as a Requirements Engineering Technique*. Master's Thesis. Lund University.
- MAJCHRZAK, T.A., BIØRN-HANSEN, A. dan GRØNLI, T.M., 2018. Progressive Web Apps: the Definite Approach to Cross-Platform Development? *Proceedings of the 51st Hawaii International Conference on System Sciences*. Oslo, Norway.mdn
- MDN. 2022. *Introduction to progressive web apps*. [online] MDN Web docs. Tersedia di: <https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Tutorials/js13kGames/Introduction> [Diakses 23 Juni 2023].
- NUSHI, M., 2017. Spreeder: A Web App to Develop and Enhance Reading Speed. *Reading in a Foreign Language*, 29(1), pp.178-184.
- PRESSMAN, R.S., 2010. *Software engineering: a practitioner's approach*. 7th ed ed. New York: McGraw-Hill Higher Education.
- RAHMADI, I.F. dan HAYATI, E., 2020. Literasi Digital, Massive Open Online Courses, dan Kecakapan Belajar Abad 21 Mahasiswa Generasi Milenial. *Jurnal Studi Komunikasi dan Media*, 24(1), p.91.
- SIRAIT, M.F., HUTAURUK, B.S., dan HERMAN, 2020. The Effect of Using Speed Reading Technique to the Student's Ability in Comprehending a Text. *Jurnal Ilmu Pendidikan*, 3(3), pp.485-498.
- STATISTA, 2023. *Most used languages online by share of websites 2023*. [online] Statista. Tersedia di: <<https://www.statista.com/statistics/262946/mo>

Halaman ini sengaja dikosongkan.