

PERANCANGAN APLIKASI AUTOMATIC CORNELL NOTE BERBASIS PROGRESSIVE WEB APP (PWA) MENGGUNAKAN METODE EXTREME PROGRAMMING

Arin Novia W, Sri Lestanti, Saiful Nur Budiman

Program Studi Teknik Informatika S1, Fakultas Teknologi Informasi
Universitas Islam Balitar Blitar, Jalan Majapahit No.2-4 Blitar, Indonesia
arinnoviaaaa@gmail.com

ABSTRAK

Penelitian ini merespon kompleksitas dalam mencatat mahasiswa dengan merancang sebuah *Progressive Web App* (PWA) yang secara otomatis mengubah catatan menjadi format *cornell note*. Mahasiswa saat ini memiliki fleksibilitas dalam mencatat, tetapi metode *cornell note* diakui sebagai yang paling efektif dalam meningkatkan pemahaman dan persiapan ujian. Meskipun demikian, banyak mahasiswa enggan menggunakannya karena kompleksitas dalam pembuatan manual. Tujuan penelitian adalah menciptakan solusi untuk mahasiswa dengan mengembangkan aplikasi yang dapat mengubah catatan menjadi format *cornell* secara otomatis. Metode penelitian melibatkan algoritma *TextRank* untuk mengidentifikasi kata kunci dan ringkasan, dengan aplikasi berbasis PWA yang dipilih untuk kemudahan akses dan performa yang lebih baik. Penelitian ini merupakan penelitian R&D (*Research & Development*) dan pengembangan aplikasi menggunakan metode *Extreme Programming*. Hasil pengujian *alpha* pada aplikasi ini dengan metode *blackbox* menunjukkan tingkat kesesuaian sebesar 97.5% dan *whitebox* dengan tingkat resiko rendah, serta tingkat kepuasan pengguna sebesar 84.98% dalam pengujian *beta*.

Kata kunci : *Cornell Note, Extreme Programming, Progressive Web App, TextRank*

1. PENDAHULUAN

Mahasiswa memiliki fleksibilitas dalam mencatat, dengan teknologi seperti laptop dan tablet tersedia sebagai alat pencatatan yang efisien. Mencatat meningkatkan daya ingat, pemahaman, persiapan ujian, dan nilai mahasiswa secara keseluruhan [1]. Metode *cornell note*, yang dikemukakan oleh Professor Walter Pauk dari *Cornell University, New York*, diakui sebagai salah satu yang paling efektif dalam meningkatkan pemahaman materi dan persiapan ujian [2]. Metode ini telah direkomendasikan oleh berbagai perguruan tinggi Amerika karena kegunaan dan fleksibilitasnya [3]. Namun, kompleksitas dalam pembuatan manualnya menyebabkan banyak mahasiswa enggan mengadopsi metode ini.

Penelitian ini bertujuan merancang sebuah solusi dengan menghadirkan fitur yang memungkinkan transformasi catatan secara otomatis yang akan diimplementasikan dalam bentuk *Progressive Web App* (PWA). Algoritma *TextRank* akan digunakan untuk mengidentifikasi kata kunci dan ringkasan, yang merupakan langkah penting dalam proses transformasi catatan [4]. Pengembangan aplikasi akan mengikuti metode *Extreme Programming*, yang memungkinkan pengembangan yang iteratif dan responsif terhadap umpan balik pengguna. Teknologi PWA dipilih karena memungkinkan akses yang mudah tanpa perlu instalasi, performa yang lebih baik, dan kemampuan untuk berfungsi dalam mode *offline*.

PWA adalah pendekatan modern dalam pengembangan aplikasi web yang menggabungkan kemudahan akses web dengan sejumlah fitur aplikasi

seluler, termasuk kemampuan untuk bekerja dalam mode *offline*. Fitur *offline* memungkinkan pengguna tetap produktif bahkan ketika tidak memiliki koneksi internet. Selain itu, PWA memberikan pengguna opsi untuk mengaksesnya melalui *browser* atau menginstalnya sebagai aplikasi, menciptakan pengalaman seolah menggunakan aplikasi *native*. Ini menyediakan alat mudah digunakan untuk memanfaatkan metode pencatatan *cornell*.

Untuk menganalisis kebutuhan pengguna dan mengembangkan aplikasi ini, penulis melakukan survei dan wawancara dengan mahasiswa FKIP Universitas Islam Balitar, yang membantu memastikan bahwa aplikasi yang dikembangkan sesuai dengan kebutuhan mahasiswa yang merupakan pengguna utama. Selain itu, penelitian ini akan melibatkan pengujian *alpha* dan *beta* untuk mengevaluasi kinerja aplikasi ini sebelum peluncuran resmi dan untuk mengukur tingkat kepuasan pengguna selama penggunaan aplikasi.

2. TINJAUAN PUSTAKA

2.1. Metode Cornell Note-Taking

Cornell note adalah metode catatan yang digunakan untuk meningkatkan pemahaman dan retensi informasi. Metode ini terdiri dari tiga bagian utama, yaitu kolom utama, kolom samping, dan bagian bawah, yang masing-masing memiliki fungsi yang berbeda. Kolom utama digunakan untuk menulis informasi utama, kolom samping untuk membuat catatan tambahan dan pertanyaan, dan bagian bawah untuk membuat kesimpulan dan ringkasan [5].



Sumber: goodnotes

Gambar 1. Catatan dengan metode cornell

Dari Gambar 1, dapat dilihat bahwa mahasiswa dapat menggunakan sistem kolom saat mencatat catatan. Kolom atas digunakan untuk informasi umum seperti nama, judul catatan, atau tanggal. Kolom kanan adalah tempat mencatat gagasan dan poin-poin penting dari perkuliahan, sedangkan kolom kiri digunakan untuk kata kunci dan pertanyaan relevan. Kolom bawah digunakan untuk merangkum poin-poin utama. Dengan penggunaan sistem kolom ini, mahasiswa dapat meningkatkan pemahaman dan pencapaian mereka.

2.2. Algoritma TextRank

TextRank adalah algoritma untuk ringkasan teks dan pencarian informasi dalam dokumen. Dikembangkan dari model *PageRank* Google, algoritma ini menggunakan pendekatan *unsupervised learning* dan tidak memerlukan data pelatihan [6]. Tahap awal adalah *preprocessing*, di mana teks dipersiapkan dengan melakukan *case folding*, *tokenisasi*, penghapusan *stopwords*, dan *stemming/lemmatization*. Selanjutnya, dilakukan pembuatan graf dengan kata-kata sebagai simpul dan hubungan antar kata sebagai sisi dalam graf. Setelah representasi graf terbentuk, dilakukan perangkangan *node* menggunakan rumus *TextRank*.

$$S(V_i) = (1 - d) + d * \sum \left(\frac{S(V_j)}{\text{Out}(V_j)} \right) \quad (1)$$

Keterangan:

$S(V_i)$ = skor dari *node* V_i

d = *damping factor*

$S(V_j)$ = skor *node* yang terhubung dengan V_i

$\text{Out}(V_j)$ = *edge* dari *node* V_j

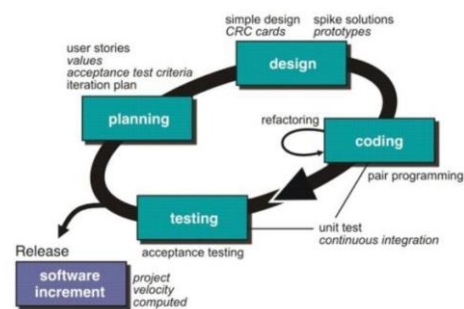
2.3. Progressive Web App (PWA)

PWA (*Progressive Web App*) adalah aplikasi web yang meniru pengalaman aplikasi *native* pada perangkat *mobile* dan *desktop*. Dibangun dengan HTML, CSS, dan *JavaScript*, PWA memiliki fitur seperti akses *offline*, notifikasi, dan akses ke perangkat keras. Pengguna dapat mengaksesnya melalui *browser* tanpa perlu mengunduh. Arsitektur PWA melibatkan elemen *App Shell* (UI dasar yang dapat di-cache), *Service Workers* (script latar belakang dengan

kemampuan *caching* dan notifikasi), dan *Web App Manifest* (informasi aplikasi web). Dengan arsitektur yang benar, PWA dapat memberikan pengalaman mirip aplikasi *native* dengan kinerja yang optimal pada berbagai platform [7].

2.4. Metode Extreme Programming

Extreme Programming (XP) adalah metodologi pengembangan perangkat lunak yang berfokus pada pengiriman perangkat lunak berkualitas dengan kecepatan, kerja tim efektif, pengembangan sederhana, dan keterlibatan pelanggan yang kuat. XP memiliki lima nilai utama: komunikasi, kesederhanaan, umpan balik, keberanian, dan penghargaan. Praktik-praktik XP melibatkan *pair programming*, *continuous integration*, *testing* otomatis, *refactoring*, dan lainnya [8].



Gambar 2. Tahapan extreme programming [9]

Dalam gambar 2 terdapat empat tahap utama: perencanaan, perancangan, pemrograman, dan pengujian. Tahap perencanaan melibatkan kolaborasi antara tim pengembang dan pelanggan untuk merencanakan pengembangan perangkat lunak, memahami kebutuhan pelanggan, menetapkan prioritas, dan mengidentifikasi persyaratan untuk setiap iterasi. Selanjutnya, tahap perancangan melibatkan pembuatan desain sistem, termasuk arsitektur dan struktur perangkat lunak yang akan dikembangkan. Tahap pemrograman adalah saat implementasi perangkat lunak berdasarkan desain yang telah dibuat, dan tahap pengujian adalah proses pengujian perangkat lunak untuk memastikan bahwa produk yang dihasilkan memenuhi kebutuhan dan persyaratan bisnis. Pada penelitian ini pengujian menggunakan metode *alpha* dan *beta*. Tahap-tahap ini berulang dalam siklus pengembangan XP yang berkelanjutan.

3. METODE PENELITIAN

Dalam suatu penelitian, pemilihan metode atau prosedur yang sesuai dengan topik penelitian sangat penting. Ini membantu dalam mengatur parameter penelitian seperti objek penelitian, teknik pengumpulan data, dan metode analisis data. Penelitian ini menggunakan metode pengembangan *Research and Development* (R&D), yang merupakan metode penelitian yang digunakan untuk menciptakan dan menguji produk tertentu [10]. Selanjutnya, untuk

mengembangkan aplikasi sistem, penelitian ini mengadopsi pendekatan *Extreme Programming* (XP). Berikut adalah langkah-langkah yang diterapkan dalam metode *Extreme Programming* ini:

3.1. Tahap Perencanaan (Planning)

3.1.1. User Story

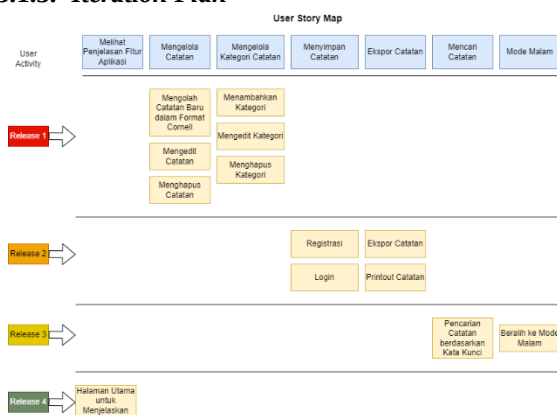
Penelitian ini melibatkan penentuan nilai (*priority*) pada setiap *user story* menggunakan teknik MOSCOW. Tahapan metode MOSCOW mencakup identifikasi *user stories*, penggolongan *user stories* dengan label (*Must Have*, *Should Have*, *Could Have*, atau *Won't Have*), diskusi intensif dalam tim proyek untuk mencapai konsensus mengenai label-label tersebut, dan terakhir, dokumentasi hasil klasifikasi ini. Teknik MOSCOW memberikan kerangka kerja yang jelas untuk mengatur prioritas dengan sangat rinci, sehingga memastikan bahwa elemen-elemen paling penting diutamakan dalam pengembangan aplikasi.

3.1.2. Acceptance Test Criteria

Acceptance test criteria digunakan untuk mengevaluasi apakah suatu fitur atau fungsi dalam perangkat lunak telah memenuhi persyaratan pengguna atau pemilik produk. *Acceptance test criteria* (kriteria uji penerimaan) sangat penting dalam pengembangan PWA karena memastikan bahwa aplikasi ini memenuhi persyaratan pengguna dan pemilik produk, terutama dalam hal responsivitas, kinerja, dan kemampuan offline.

Kriteria uji penerimaan ini adalah tolok ukur utama yang menentukan kesiapan PWA untuk dipublikasikan. Kriteria ini diwakili dalam bentuk skenario penggunaan dan berfungsi sebagai panduan dalam pengujian perangkat lunak serta menentukan kesiapan fitur atau fungsi untuk dipublikasikan.

3.1.3. Iteration Plan



Gambar 3. User story map

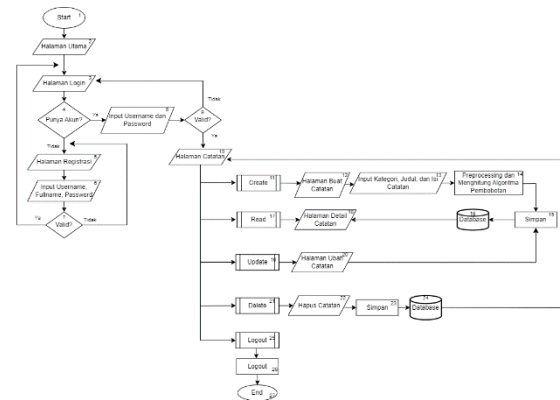
Pada gambar 3, merupakan *iteration plan* yang digunakan untuk menentukan urutan pengembangan fitur didalam perilsan aplikasi pada metode *extreme programming* untuk merancang aplikasi berbasis PWA ini. Di dalamnya, digunakan pendekatan yang

menggabungkan teknik MOSCOW dan *user story mapping* untuk merencanakan iterasi.

3.2. Tahap Perancangan (Design)

3.2.1. Flowchart

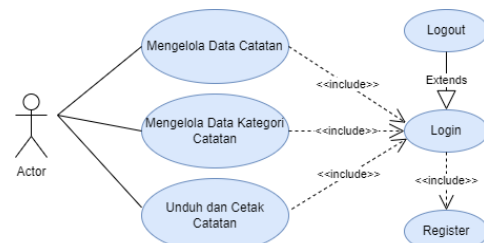
Flowchart berikut ini disajikan *flowchart* yang menggambarkan langkah-langkah utama dalam pengembangan aplikasi:



Gambar 4. Flowchart

Gambar 4 mengilustrasikan alur proses utama yang terlibat dalam pengembangan sistem, memberikan pandangan visual tentang bagaimana berbagai komponen berinteraksi satu sama lain dalam perangkat lunak yang dibangun.

3.2.2. Use Case Diagram



Gambar 5. Use case diagram

Gambar 5 merupakan diagram kasus pengguna yang menggambarkan interaksi antara pengguna dan perangkat lunak. Diagram ini memberikan gambaran tentang fungsi-fungsi yang tersedia dalam sistem dan cara pengguna berinteraksi dengan perangkat lunak tersebut.

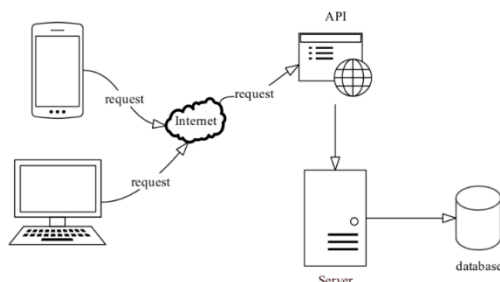
3.2.3. Class Diagram

Gambar 6 merupakan sebuah diagram kelas yang mengilustrasikan struktur kelas dalam perangkat lunak. Diagram ini memberikan gambaran tentang entitas-entitas utama dalam sistem dan interaksi di antara kelas-kelas tersebut.



Gambar 6. Class Diagram

3.2.4. Arsitektur Sistem



Gambar 7. Arsitektur Sistem

Pada gambar 7 diatas, terdapat komponen utama yang meliputi *Server*, *Database*, *Internet*, dan *Request*. Proses arsitektur sistem ini terdiri dari pengiriman permintaan dari sisi klien ke sisi server melalui API dengan protokol HTTP atau HTTPS. Selanjutnya, sisi server memproses permintaan tersebut untuk mengambil atau memodifikasi data di database, dan mengirimkan *respons* berisi data atau konfirmasi kembali ke sisi klien.

3.3. Tahap Pemrograman (Coding)

Tahap Pemrograman (*Coding*) merupakan fase kunci dalam pengembangan PWA, yang melibatkan implementasi desain konsep dan struktur yang telah direncanakan sebelumnya. Di tahap ini, penulis menggunakan berbagai teknologi dan alat, seperti *Node.js*, *React*, *Hapi*, dan *PostgreSQL*, untuk membangun aplikasi PWA. Selain menulis kode yang mengatur berbagai aspek, termasuk antarmuka pengguna, logika bisnis, pemrosesan permintaan pengguna, serta interaksi dengan database, juga diterapkan prinsip-prinsip seperti *Test-Driven Development (TDD)*, *refactoring*, dan *Continuous Integration (CI)*. Hal ini merupakan bagian integral dari pendekatan *Extreme Programming* yang

membantu mencapai kualitas dan keberlanjutan dalam pengembangan aplikasi PWA.

3.4. Tahap Pengujian (Testing)

Tahap Pengujian (*Testing*) adalah langkah penting dalam memastikan bahwa PWA berfungsi sesuai dengan yang diharapkan. Pengujian dilakukan dalam dua tahap utama:

a) Alpha Testing

Pada tahap ini, PWA diuji secara mendalam oleh tim pengembang untuk mengidentifikasi dan memperbaiki potensi masalah, bug, atau kerentanan keamanan. Tes ini mencakup pengujian *whitebox* dan *blackbox*. Pengujian *whitebox* melibatkan metode *basis path* dan pengujian *blackbox* menggunakan teknik *equivalence partitioning*.

b) Beta Testing

Setelah berhasil melewati tahap *alpha*, PWA juga diuji oleh sekelompok pengguna yang mencakup ahli IT, ahli bahasa, dan pengguna akhir, seperti mahasiswa. Hal ini bertujuan untuk mengumpulkan umpan balik dari pengguna yang mewakili berbagai aspek penggunaan aplikasi, termasuk aspek teknis dan bahasa. Dengan demikian, tahap *beta testing* memastikan bahwa PWA berfungsi dengan baik dalam situasi dunia nyata dan memenuhi berbagai kebutuhan pengguna.

3.5. Tahap Rilis (Release)

Pada tahap rilis, penulis melakukan *deployment* produk yang telah dibuat setelah melewati proses pengembangan dan pengujian. Rilis aplikasi ini dilakukan secara bertahap, yang berarti bahwa produk dirilis dalam beberapa iterasi. Jadwal rilis aplikasi ini dijelaskan secara terperinci dalam Tabel 1 di bawah ini.

Tabel 1. Jadwal Rilis Aplikasi

Release	Features	Deadline
1.	Mengkonversi Catatan Baru ke dalam Format <i>Cornell</i>	29 April 2023
	Mengedit Catatan	
	Menghapus Catatan	
	Menambahkan Kategori Catatan	
	Mengedit Kategori Catatan	
2.	Menghapus Kategori Catatan	20 Mei 2023
	Registrasi	
	Login	
	Ekspor Catatan	
3.	Printout Catatan	10 Juni 2023
	Pencarian Catatan berdasarkan Kata Kunci	
4.	Beralih ke Mode Malam	1 Juli 2023
	Halaman Utama untuk Menjelaskan Fitur Aplikasi	

Tabel 1 menunjukkan perencanaan rilis, fitur-fitur, dan batas waktu pengerjaan dalam metode *extreme programming*. Dalam XP, setiap rilis berfokus pada fitur-fitur penting yang harus selesai dalam waktu

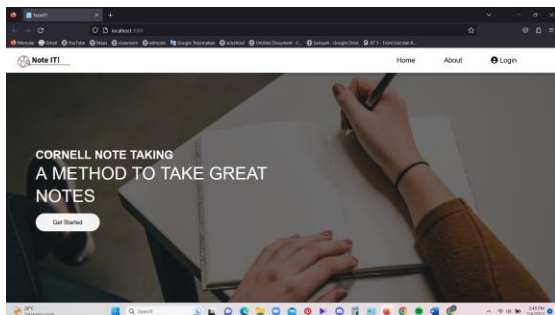
singkat untuk pengujian oleh pengguna. Setelah rilis, penulis dapat terus meningkatkan aplikasi pada rilis berikutnya.

4. HASIL DAN PEMBAHASAN

Berikut hasil penerapan dan pengujian aplikasi, serta evaluasi kinerjanya dalam memenuhi kebutuhan pengguna.

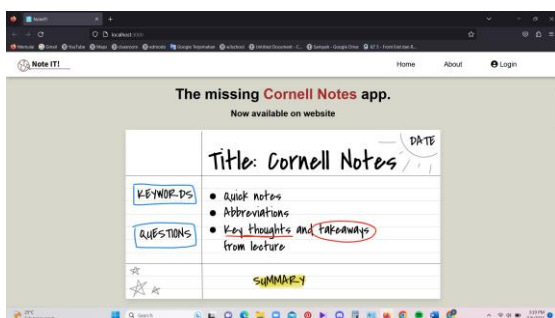
4.1. Halaman Utama

Halaman beranda atau halaman utama menampilkan informasi mengenai tampilan format *Cornell Note*, fitur yang tersedia, serta manfaat dari penggunaan website *Cornell Note*. Kontennya terstruktur dalam beberapa bagian.



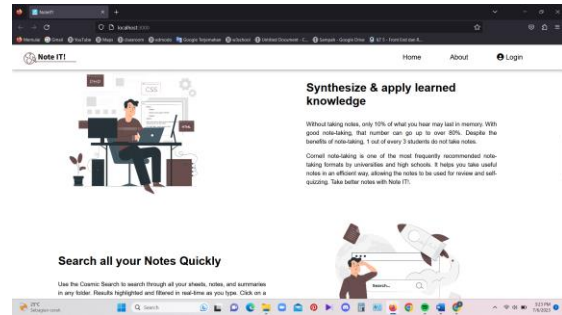
Gambar 8. Konten halaman utama

Pada gambar 8 halaman utama difokuskan pada metode *Cornell Note* dan mengajak pengguna untuk memulai pengalaman dengan tombol "Get Started." Halaman ini juga menyajikan contoh format *Cornell Note* dengan tiga kolom utama: catatan utama, kata kunci, dan ringkasan seperti pada gambar 9.



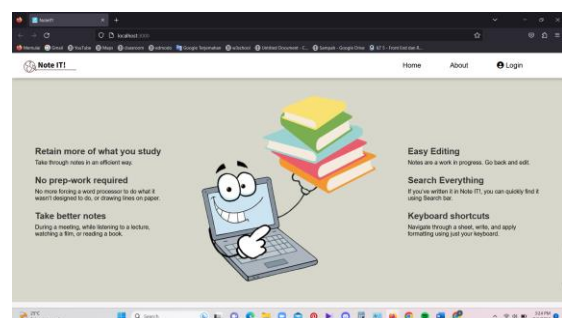
Gambar 9. Konten tampilan cornell notes

Kemudian, terdapat konten yang menjelaskan fitur-fitur aplikasi, memberikan informasi tentang berbagai fitur yang tersedia, seperti yang terlihat pada Gambar 9.



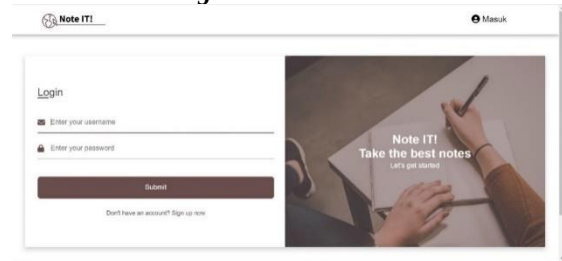
Gambar 9. Penjelasan fitur website

Terakhir, terdapat konten yang menjelaskan manfaat penggunaan aplikasi ini, memberikan informasi tentang manfaat bagi pengguna saat menggunakan aplikasi ini dalam aktivitas sehari-hari, seperti yang terlihat pada Gambar 10.



Gambar 10. Konten manfaat website

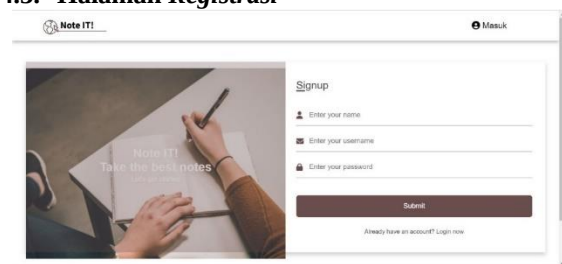
4.2. Halaman Login



Gambar 11. Tampilan halaman login

Pada gambar 11, halaman login di *cornell note* memungkinkan pengguna untuk memasukkan nama pengguna dan kata sandi melalui formulir login. Setelah memasukkan data yang valid, pengguna dapat mengklik "Masuk" untuk mengakses akun. Ini memberikan akses yang aman dan terkontrol ke aplikasi.

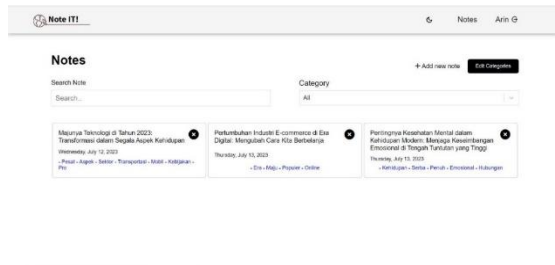
4.3. Halaman Registrasi



Gambar 12. Tampilan halaman registrasi

Pada gambar 12, halaman registrasi di *cornell note* memungkinkan pengguna membuat akun baru dengan mengisi formulir yang meminta informasi seperti nama, email, nama pengguna, dan kata sandi. Setelah mengklik "Sign Up", pengguna akan diarahkan ke halaman *login* untuk mengakses akun mereka.

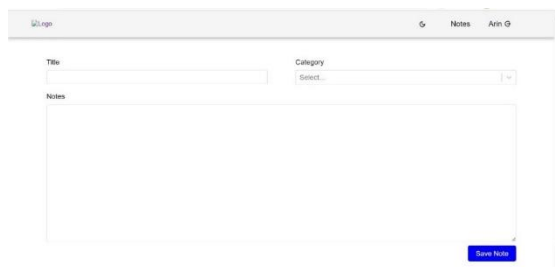
4.4. Halaman List Catatan



Gambar 13. Tampilan halaman list catatan

Pada gambar 13, halaman list catatan menampilkan catatan dalam bentuk kartu dengan judul, tanggal, kata kunci, dan kategori. Pengguna dapat mengklik kartu untuk mengakses catatan. Halaman ini juga memiliki fitur filter untuk mengorganisir catatan berdasarkan kategori.

4.5. Halaman Buat Catatan Baru



Gambar 14. Tampilan halaman buat catatan

Pada gambar 14, halaman buat catatan di *cornell note* memungkinkan pengguna untuk membuat catatan baru dengan judul dan isi utama. Setelah pengguna selesai, aplikasi akan mengatur catatan tersebut menjadi format *cornell note* dengan tiga kolom. Halaman ini mempermudah pembuatan catatan dalam format *cornell note*.

4.6. Halaman Ubah Catatan



Gambar 15. Tampilan halaman ubah catatan

Pada gambar 15, halaman ubah catatan memungkinkan pengguna untuk mengedit catatan yang sudah ada, termasuk mengubah isi catatan utama, kata kunci, dan kolom ringkasan. Pengguna dapat dengan mudah menyimpan perubahan yang telah dilakukan.

4.7. Halaman Hasil Catatan



Gambar 16. Tampilan halaman hasil catatan

Pada gambar 16 merupakan halaman hasil catatan yang mana pada halaman inilah algoritma *TextRank* digunakan. Setelah berhasil menyimpan catatan baru, pengguna akan diarahkan ke halaman hasil catatan. Halaman ini menampilkan hasil perhitungan dari algoritma *TextRank*, yang merupakan salah satu metode analisis teks. Dalam perhitungan *TextRank*, sistem menganalisis teks catatan dengan memperhatikan hubungan antar kata-kata. Sistem mengidentifikasi kata-kata yang saling terkait dan menghitung bobot kata-kata tersebut berdasarkan hubungan tersebut.

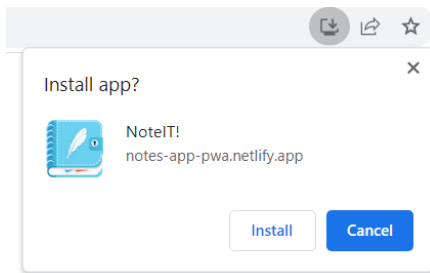
Ada beberapa langkah penting dalam proses ini. Pertama, sistem menggunakan daftar *stopwords* untuk mengabaikan kata-kata umum yang tidak relevan. Kemudian, teks dibagi menjadi token-token, yang merupakan unit-unit dasar dalam analisis. Sistem juga membangun graf yang menggambarkan hubungan antar kata-kata dalam teks.

Selama analisis, algoritma *TextRank* diterapkan untuk mengidentifikasi kata-kata kunci. Selain itu, algoritma ini juga digunakan untuk mengekstraksi ringkasan kalimat dari teks. Semua langkah ini adalah bagian penting dari proses analisis teks yang dilakukan oleh sistem untuk menghasilkan hasil yang relevan untuk pengguna.

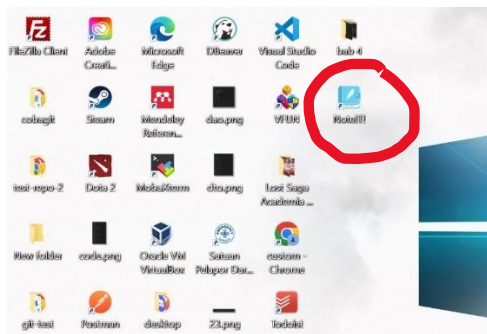
Setelah melalui proses analisis ini, hasil perhitungan dari algoritma *TextRank* akan disajikan kepada pengguna dalam bentuk kata-kata kunci dan ringkasan kalimat yang mencerminkan inti dari teks catatan yang telah disimpan.

4.8. Fitur Progressive Web App (PWA)

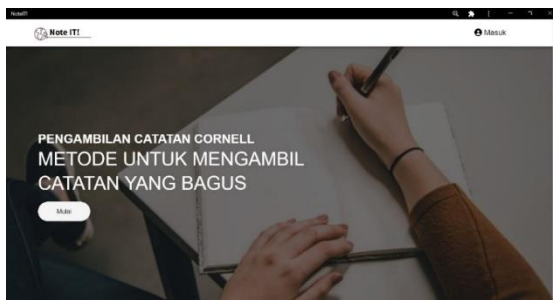
Seperti terlihat pada gambar 17, untuk menginstal PWA pada perangkat pengguna dapat memulai dengan mengunjungi situs web *cornell note*. Selanjutnya, perlu mencari ikon atau pesan yang mengundang untuk menginstal aplikasi, biasanya ditandai dengan "Install" atau "Add to Home Screen," dan kemudian mengklik atau menyentuhnya.



Gambar 17. Install aplikasi

Gambar 18. Aplikasi *cornell* pada desktop

Seperti terlihat pada gambar 18, setelah proses instalasi selesai, PWA akan muncul sebagai ikon aplikasi di layar beranda perangkat, baik itu pada desktop komputer atau perangkat seluler. Dengan demikian, pengguna dapat mengakses PWA dengan mudah melalui ikon aplikasi yang telah diinstal, menciptakan pengalaman mirip dengan aplikasi konvensional.



Gambar.19 Tampilan PWA

Pada gambar 19, hasil implementasi PWA (*Progressive Web App*) memberikan pengalaman seperti aplikasi mandiri tanpa tampilan browser atau URL yang terlihat. PWA juga memungkinkan akses *offline* atau dengan koneksi internet yang lemah, memungkinkan pengguna untuk membuka aplikasi tanpa tergantung pada ketersediaan koneksi yang stabil. Ini berguna saat pengguna ingin mengakses catatan saat bepergian atau di tempat dengan koneksi terbatas.

4.9. Black Box Testing

Pengujian *black box* pada penelitian ini menggunakan metode partisi yang setara (*equivalence partitions*) dengan mengelompokkan halaman-

halaman yang akan diuji. Kategori-kategori pengujian dan hasil pengujian dijabarkan pada tabel 2.

Tabel 2. *Black box testing*

Kategori	Halaman	Hasil Pengujian
Interface	Halaman Utama	100%
	Halaman Hasil Catatan	50%
Fitur PWA	Akses Aplikasi Mode <i>Offline</i>	100%
	Tambahkan aplikasi ke layar utama perangkat	100%
	Sinkronisasi data secara otomatis ketika terhubung ke koneksi	100%
	Caching konten untuk akses <i>offline</i>	100%
	Responsif pada berbagai perangkat	100%
	Mengakses PWA melalui ikon di layar	100%
	Mengakses PWA melalui browser	100%
Logika Pemrograman	Halaman Login	100%
	Halaman Sign Up	100%
	Halaman List Data Catatan	100%
	Halaman Tambah Catatan	100%
	Halaman Ubah Catatan	100%
Total		97.56%

Pada tabel 2, rumus perhitungan persentase hasil pengujian *black box* adalah sebagai berikut:

$$\text{Nilai} = \left(\frac{\text{pengujian berhasil}}{\text{total pengujian}} \right) \times 100\%$$

Dari hasil pengujian tersebut, dapat disimpulkan bahwa seluruh pengujian kecuali fitur *export* catatan pada halaman hasil catatan telah melewati pengujian dengan hasil yang tinggi, dengan total persentase keseluruhan sebesar 97.56%.

4.10. White Box Testing

Setelah melalui proses pengujian *black box*, aplikasi dilanjutkan proses pengujian menggunakan metode *white box*, khususnya *basis path analysis*, untuk memastikan kinerja aplikasi. *Cyclomatic complexity* dihitung berdasarkan *flowgraph* yang telah dibuat pada gambar 19. *Flowgraph* ini dibuat berdasarkan *flowchart* pada gambar 4.

Perhitungan *Cyclomatic Complexity*:

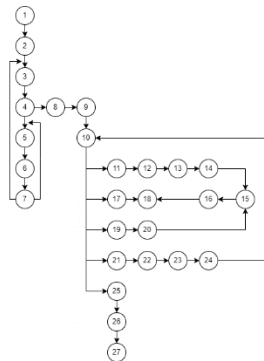
$$V(G) = E - N + 2$$

$$V(G) = 33 - 27 - 2$$

$$V(G) = 8$$

Hasil perhitungan menghasilkan nilai sebesar 8, yang menunjukkan bahwa ada 8 alur sistem yang berbeda. Nilai sebesar 8 masuk dalam kategori "5-10: Prosedur yang terstruktur dengan baik dan stabil"

berdasarkan tabel hubungan antara *Cyclomatic Complexity* dan tingkat risiko pada tabel 2.



Gambar 19. Flowgraph sistem

Tabel 3. Hubungan CC dan resiko [11]

Nilai CC	Tipe Prosedur	Tingkat Resiko
1-4	Prosedur Sederhana	Rendah
5-10	Prosedur yang terstruktur dengan baik dan stabil	Rendah
11-20	Prosedur yang lebih kompleks	Menengah
21-50	Prosedur yang kompleks dan kritis	Tinggi
>50	Rentan kesalahan, sangat mengganggu, prosedur tidak dapat diuji	Sangat tinggi

Berdasarkan perhitungan CC dan tabel hubungan yang terlihat pada tabel 3, kesimpulan yang dapat diambil adalah bahwa aplikasi ini memiliki tingkat risiko yang rendah dalam hal kesalahan manusia (*human error*) dalam penggunaannya.

4.11. Pengujian Ahli IT

Pengujian validasi ahli IT dilakukan untuk mengevaluasi aplikasi ini. Pengujian ini melibatkan 3 ahli IT untuk mengevaluasi berbagai aspek aplikasi secara keseluruhan berdasarkan standar ISO/IEC 25010 yang meliputi *functional suitability*, *performance efficiency*, *usability*, *reliability*, *maintainability*, *satisfaction*, *portability*, dan *security*. Hasil yang didapatkan dihitung dengan menggunakan rumus berikut:

$$Y = (\text{Nilai Persentase}) = \frac{\sum N.R}{R} \quad (2)$$

Di mana:

N = nilai jawaban responden.

R = jumlah total responden

$\sum N.R$ = jumlah nilai jawaban yang dikalikan dengan jumlah responden untuk setiap aspek yang diuji.

Hasil pengujian menunjukkan bahwa sistem telah memperoleh rata-rata skor sebesar 86.6% berdasarkan penilaian responden. Namun, ada catatan bahwa alur website dan tampilan websitenya memerlukan pengembangan lebih lanjut. Saran-saran yang

diberikan oleh penguji ahli dapat menjadi panduan dalam pengembangan selanjutnya.

4.12. Pengujian Ahli Bahasa

Dalam penilaian akurasi oleh sepuluh ahli bahasa Indonesia, algoritma *TextRank* menghasilkan kata kunci dengan tingkat keakuratan 62%. Meskipun kata kunci tersebut sesuai dengan inti dan konteks catatan, keterbatasan algoritma menyebabkan kata kunci terbatas pada satu kata.

Dalam penilaian keakuratan hasil ringkasan oleh responden, algoritma *TextRank* memiliki tingkat keakuratan 74%. Namun, terdapat kendala di mana ringkasan terkadang tidak akurat atau terlalu singkat. Hal ini disebabkan oleh pendekatan perangkikan kata kunci yang digunakan oleh algoritma.

Meskipun demikian, algoritma *TextRank* tetap memberikan kontribusi yang berarti dalam melakukan ringkasan teks secara otomatis dan dapat memberikan informasi yang berguna dalam beberapa konteks.

4.13. Beta Testing

Pengujian *beta* adalah tahap penting dalam pengembangan aplikasi yang dilakukan dengan metode *open beta testing*. Pengujian ini, melibatkan 49 responden dari mahasiswa FKIP untuk mengevaluasi berbagai aspek aplikasi, termasuk fungsionalitas, *usability*, performa, dan kompatibilitas. Hasil pengujian *beta* dinyatakan dalam nilai persentase kepuasan rata-rata, yang dihitung menggunakan rumus berikut:

$$Y = (\text{Nilai Persentase}) = \frac{\sum N.R}{R} \quad (3)$$

Di mana:

N = nilai jawaban responden.

R = jumlah total responden

$\sum N.R$ = jumlah nilai jawaban yang dikalikan dengan jumlah responden untuk setiap aspek yang diuji.

Hasil pengujian *beta* menunjukkan tingkat kepuasan rata-rata sebesar 84.98%. Ini mencerminkan persepsi positif responden terhadap aplikasi, menunjukkan penerimaan yang kuat dari pengguna potensial. Hasil ini memberikan kontribusi positif terhadap pengembangan dan penyempurnaan aplikasi di masa depan.

5. KESIMPULAN DAN SARAN

Kesimpulan dari penelitian ini adalah bahwa aplikasi *automatic cornell note* berbasis *Progressive Web App* (PWA) yang dikembangkan menggunakan metode *Extreme Programming* (XP) telah berhasil dalam pengujian *alpha* dan *beta*, dengan tingkat keberhasilan pada pengujian *alpha* menggunakan *blackbox* sebesar 97.5%, dan pengujian *whitebox* dengan tingkat resiko rendah. Tingkat kepuasan pengguna pada pengujian *beta* mencapai 84.98%. Saran untuk pengembangan selanjutnya adalah meningkatkan kemampuan pencarian kata kunci,

mendukung berbagai format file, mengembangkan algoritma ringkasan yang canggih, dan mengoptimalkan antarmuka pengguna untuk meningkatkan pengalaman pengguna.

DAFTAR PUSTAKA

- [1] A. Thompson, "Students ' Views on Strategic Note-taking and its Impact on Performance , Achievement , and Learning," vol. 13, no. 2, hal. 1–16, 2020.
- [2] N. Iman, "Efektivitas Gaya Mencatat Linear dan Non-Linear Bersetting Saintifik dalam Pembelajaran Matematika," *JNPM (Jurnal Nas. Pendidik. Mat.*, vol. 2, no. 2, hal. 233, 2018, doi: 10.33603/jnpm.v2i2.981.
- [3] B. O'Shea, L. Sundal, dan S. Yan, *College Success*. University of Minnesota Libraries Publishing, 2015.
- [4] M. Sholahuddin, "Analisa Perbandingan Algoritma Maximum Marginal Relevance dengan Textrank dan Lexrank pada Kasus Peringkasan Teks Otomatis," 2019.
- [5] E. E. Charles dan J. A. Weinstock, "Improving Student Learning through Cornell Note-Taking," *Coll. Stud. J.*, vol. 52, no. 2, hal. 233–238, 2018.
- [6] M. A. Zamzam, "Sistem Automatic Text Summarization Menggunakan Algoritma Textrank," *Matics*, vol. 12, no. 2, hal. 111–116, 2020, doi: 10.18860/mat.v12i2.8372.
- [7] D. Fortunato dan J. Bernardino, "Progressive web apps: An alternative to the native mobile Apps," in *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, 2018, hal. 1–6. doi: 10.23919/CISTI.2018.8399228.
- [8] K. Beck, *Extreme Programming Explained: Embrace Change*. 1999.
- [9] N. A. Septiani dan F. Y. Habibie, "Penggunaan Metode Extreme Programming Pada Perancangan Sistem Informasi Pelayanan Publik," *J. Sist. Komput. dan Inform.*, vol. 3, hal. 341–349, 2022, doi: 10.30865/json.v3i3.3931.
- [10] A. Chandra, K. P. K. R, dan D. F. H. P, "PERANCANGAN APLIKASI SOLUSI KERUSAKAN SEPEDA MOTOR KARBURATOR BERBASIS WEB MENGGUNAKAN METODE WATERFALL," vol. 6, no. 2, hal. 696–701, 2022.
- [11] J. B. L. Sie, Izmy Alwiah Musdar, dan Syamsul Bahri, "Pengujian White Box Testing Terhadap Website Room Menggunakan Teknik Basis Path," *KHARISMA Tech*, vol. 17, no. 2, hal. 45–57, 2022, doi: 10.55645/kharismatech.v17i2.235.