APLIKASI PENGHITUNG NILAI KEMIRIPAN DOKUMEN DENGAN ALGORITMA WINNOWING: SKENARIO PENGGUNAAN, RANCANGAN, DAN PURWARUPA

Berliyanto
Program Studi Teknik Informatika, FTI, Institut Teknologi Budi Utomo Jakarta, berli@itbu.ac.id

Abstrak
Setiap karya ilmiah, termasuk skripsi atau tugas akhir mahasiswa, harus dipastikan terbebas dari plagiarisme sebelum dipublikasikan. Salah satu indikator untuk mengindikasikan apakah sebuah karya ilmiah merupakan plagiarisme atau tidak adalah tingkat kesamaan (*similarity*) terhadap dokumen lain. Nilai similarity tertentu biasanya menjadi syarat yang harus dipenuhi agar sebuah tulisan layak dipublikasikan. Penelitian bertujuan untuk membuat rancang bangun aplikasi perangkat lunak untuk menghitung tingkat kesamaan (*similarity score*) suatu dokumen teks dengan dokumen lain yang ada di dalam basis data lokal. Metode penelitian yang dilakukan meliputi pengumpulan requirement aplikasi, menentukan algoritma penghitung nilai similarity, membuat rancangan aplikasi, mengembangkan purwarupa, dan menguji purwarupa yang dihasilkan.
Kata kunci: analisis software, perancangan software, *algoritma winnowing*, rekayasa perangkat lunak

1. PENDAHULUAN

Plagiarisme atau penjiplakan yang melanggar hak cipta, merupakan hal yang perlu dihindari oleh perguruan tinggi ketika mempublikasikan karya ilmiah. Isu plagiarisme ini perlu diperhatikan terutama pada publikasi skripsi atau tugas akhir yang merupakan syarat kelulusan mahasiswa pada suatu perguruan tinggi. Dampak plagiarisme tentu saja sangat serius bagi dunia pendidikan. Karya hasil plagiat, tidak hanya melanggar etika akademik tetapi juga merupakan tindak pidana di Indonesia. Sanksi untuk pelaku plagiarisme pada skripsi adalah pencabutan gelar akademik, pembatalan ijazah, dan bahkan ancaman penjara. Pada akhirnya, perguruan tinggi harus memastikan bahwa setiap skripsi atau karya ilmiah yang dipublikasikan terbebas dari plagiarisme. Melakukan tindakan pencegahan terhadap plagiarisme pada skripsi atau tugas akhir adalah hal yang perlu dilakukan oleh setiap perguruan tinggi di Indonesia.

Salah satu cara yang bisa dilakukan oleh perguruan tinggi untuk mencegah plagiarisme adalah dengan menghitung nilai kesamaan atau *similarity score* suatu dokumen sebelum mempublikasikannya. *Similarity score* mengacu pada seberapa mirip suatu dokumen dengan dokumen-dokumen lain yang sudah dipublikasikan. Biasanya *similarity score* dinyatakan dalam persentase, semakin besar nilainya maka semakin besar kemungkinan adanya plagiarisme. Misalnya, dokumen dengan *similarity score* 85% memiliki indikasi plagiarisme yang jauh lebih tinggi dibandingkan dengan dokumen dengan *similarity score* 20%. Nilai similarity ini sudah umum digunakan sebagai syarat publikasi suatu karya ilmiah. Kebanyakan perguruan tinggi di Indonesia mensyaratkan *similarity score* maksimal untuk skripsi antara 15 - 30% tergantu dari metode penghitungan yang digunakan.

Penentuan *similarity score* dapat dilakukan dengan bantuan perangkat lunak, salah satunya adalah yang disediakan oleh turnitin. Memanfaatkan aplikasi penghitung *similarity score* yang sudah ada memiliki keuntungan seperti hasil yang akurat, mudah untuk digunakan, dan menyediakan informasi bagian dokumen yang mengindikasikan plagiarisme. Namun demikian, aplikasi yang sudah ada tersebut tidak dapat digunakan dengan bebas. Perguruan tinggi perlu membayar sejumlah biaya setiap kali menghitung *similarity score* untuk sebuah dokumen. Beberapa aplikasi juga menetapkan biaya berdasarkan jumlah kata pada dokumen. Dengan kata lain, semakin banyak dokumen yang diperiksa maka semakin besar biaya

yang dikeluarkan. Biaya yang sama juga berlaku mengulangi penghitungan *similarity score* untuk dokumen yang sama. Biaya tersebut semakin besar pada skenario menghitung *similarity score* untuk skripsi karena jumlah dokumen dan jumlah kata di dalamnya sangat banyak. Perguruan tinggi yang memiliki keterbatasan anggaran mungkin saja terbebani dengan hal tersebut. Solusi alternatif yang mungkin dilakukan adalah mengembangkan aplikasi untuk menghitung *similarity score* secara mandiri.

Proses pengembangan aplikasi penghitung *similarity score* secara mandiri bukan lah pekerjaan yang sederhana. Ada banyak aktivitas yang terlibat dalam pengembangan aplikasi ini. Pendekatan yang disarankan untuk pengembangan perangkat lunak adalah mengikuti setiap aktivitas pada tahapan *software development life cycle* (SDLC) (Pressman & Maxim, 2020). Menentukan kebutuhan aplikasi, menyusun algoritma yang tepat, dan membuat berbagai rancangan perangkat lunak adalah beberapa contohnya (Dennis, Wixom, & Roth, 2018). Rumitnya mengembangkan aplikasi penghitung *similarity score* merupakan fokus masalah dalam penelitian ini. Tujuan dari penelitian ini adalah membuat rancang bangun aplikasi penghitung *similarity* dokumen. Hasil penelitian ini dapat digunakan sebagai acuan pengembangan aplikasi penghitung *similarity score* sebagai solusi alternatif dalam pencegahan plagiarisme pada skripsi atau tugas akhir.

Ada tiga pertanyaan penelitian yang dijawab dalam penelitian ini. Pertama, apa saja persyaratan (*requirements*) yang harus dipenuhi oleh aplikasi? Kedua, algoritma apa yang sesuai untuk digunakan dalam penghitungan *similarity score*? Ketiga, bagaimana rancangan aplikasi yang sesuai dengan *persyaratan* yang ada? Setelah menjawab pertanyaan penelitian tersebut dihasilkan sebuah cetak biru (*blue print*) rancangan aplikasi penghitung *similarity score*. Lingkup penelitian ini adalah penghitungan *similarity score* untuk skripsi mahasiswa di perguruan tinggi swasta.

2. METODOLOGI

Penelitian ini merupakan penelitian kualitatif yang mengikuti urutan proses pada siklus hidup perangkat lunak (*software development life cycle*). Metodologi yang digunakan tersusun atas lima tahap yang disusun untuk menjawab tiga pertanyaan penelitian. Fokus penelitian ini terletak pada tahap analisis dan perancangan perangkat lunak. Gambar 1 menunjukkan urutan langkah-langkah penelitian yang dilakukan.

Gambar 1. Langkah-langkah Penelitian
Sumber: penulis

Penelitian ini diawali dengan langkah perumusan masalah. Dari tahap ini dihasilkan pertanyaan-pertanyaan penelitian untuk dijawab. Langkah kedua penelitian ini adalah penentuan kebutuhan aplikasi. Kebutuhan yang dimaksud di sini adalah *software requirement* yang harus dipenuhi. Langkah selanjutnya adalah pemilihan dan konfigurasi algoritma penghitung *similarity*. Kemudian dilanjutkan ke langkah keempat yaitu pembuatan rancangan aplikasi. Setiap rancangan yang dihasilkan mengacu pada kebutuhan aplikasi yang sudah lebih dahulu diidentifikasi. Langkah terakhir adalah pembuatan purwarupa aplikasi yang bertujuan untuk menguji rancangan dan juga implementasi dari algoritma yang dipilih.

2.1 Perumusan Masalah

Langkah pertama dalam penelitian ini adalah perumusan masalah. Fokus penelitian ini adalah membuat rancang bangun aplikasi penghitung *similarity score* untuk skripsi. Luaran dari tahap perumusan masalah ini adalah tiga pertanyaan penelitian yang sudah disebutkan di bagian pendahuluan. Tujuan penelitian ini tercapai setelah semua pertanyaan penelitian tersebut berhasil

dijawab. Langkah-langkah penelitian setelah ini pada dasarnya dilakukan untuk menjawab tiga pertanyaan penelitian tersebut.

2.2 Penentuan Kebutuhan Aplikasi

Langkah selanjutnya setelah perumusan masalah selesai dilakukan adalah penentuan kebutuhan aplikasi. Kebutuhan (atau persyaratan) aplikasi merujuk pada kebutuhan fungsional dan non-fungsional perangkat lunak. Penelitian ini menggunakan *user story* untuk merepresentasikan kebutuhan aplikasi tersebut. *User story* merupakan salah satu teknik yang efektif untuk mendeskripsikan kebutuhan aplikasi (*software requirement*) (Sharp, Rogers, & Preece, 2019). Teknik ini sudah banyak digunakan dalam berbagai metodologi pengembangan aplikasi, terutama pada pendekatan agile. Luaran dari langkah penelitian yang kedua ini adalah daftar *user story* yang relevan untuk aplikasi penghitung *similarity score*.

2.3 Pemilihan Algoritma *Similarity Score*

Algoritma untuk menghitung *similarity score* merupakan bagian terpenting di dalam aplikasi ini. Saat ini sudah ada berbagai pilihan algoritma yang bisa dimanfaatkan untuk menghitung *similarity score*. Penelitian ini tidak mengembangkan algoritma *similarity score* sendiri, melainkan memilih yang paling sesuai dengan kebutuhan aplikasi. Pada langkah ketiga ini dilakukan studi literatur untuk memilih algoritma yang sesuai. Hasil studi literatur mendapatkan dua kandidat algoritma yang sesuai yaitu *Rabin Karp Algorithm* dan *Winnowing Algorithm*. Keduanya memiliki tingkat akurasi yang baik, bahkan ada penelitian yang menyatakan bahwa keduanya mendekati hasil pengujian *similarity* dengan turnitin. Pada akhirnya, penelitian ini memilih algoritma Winnowing karena memiliki lebih banyak bukti pendukung keberhasilan implementasinya untuk skenario penggunaan yang mirip dengan permasalahan penelitian ini, contohnya adalah penelitian yang dilakukan Sunardi, Yudhana, & Mukaromah (2019). Namun demikian, pengujian antar algoritma berada di luar lingkup penelitian ini.

2.4 Pembuatan Rancangan Aplikasi

Langkah keempat penelitian ini adalah membuat rancangan aplikasi. Rancangan yang dibuat mengacu kepada kebutuhan aplikasi dan mempertimbangkan keberhasilan implementasi algoritma *winnowing*. Rancangan yang dihasilkan meliputi rancangan data, rancangan antarmuka pengguna, dan rancangan arsitektur aplikasi. Dari luaran-luaran ini lah kemudian dikembangkan sebuah purwarupa aplikasi pada langkah selanjutnya.

2.5 Pembuatan Purwarupa Aplikasi

Langkah kelima sekaligus langkah terakhir sebelum disusun kesimpulan adalah pembuatan purwarupa aplikasi. Purwarupa aplikasi ini dibuat dengan tujuan untuk menguji apakah rancangan yang dihasilkan dalam penelitian ini dapat dikembangkan menjadi aplikasi yang siap beroperasi. Kode-kode program ditulis untuk mensimulasikan fungsionalitas aplikasi sesuai rancangan yang ada. Setelah berhasil dibuat, purwarupa tersebut kemudian diuji dengan teknik scenario-based testing dengan *test case* yang dibuat dari *user story*. Kesimpulan dari hasil penelitian ini kemudian disusun setelah langkah terakhir ini dilakukan.

3. HASIL DAN PEMBAHASAN

Setiap langkah penelitian yang dilakukan menghasilkan luaran-luaran yang merupakan hasil penelitian ini. Hasil penelitian ini mencakup user story dan model-model kebutuhan aplikasi, konfigurasi lagoritma winnowing yang disarankan, rancangan aplikasi, dan juga purwarupa aplikasi. Bagian ketiga ini berisi hasil-hasil penelitian tersebut beserta pembahasannya.

3.1 Daftar Kebutuhan Aplikasi

Proses penentuan kebutuhan aplikasi seperti yang disebutkan pada Gambar 1., menghasilkan daftar user story untuk aplikasi penghitungan *similarity score*. Pengembangan aplikasi dianggap berhasil apabila memenuhi semua *user story* yang tersebut. Penelitian ini menggunakan proses bisnis memeriksa *similarity score* proposal skripsi untuk dibandingkan dengan skripsi yang sudah diterbitkan oleh suatu perguruan tinggi. Pemilihan proposal skripsi sebagai objek yang diukur *similarity score* dilakukan dengan dua pertimbangan. Pertama, aplikasi ini diperuntukkan untuk tindakan pencegahan

plagiarisme terhadap skripsi sejak dini dan sebelum melakukan penelitian dan menyusun skripsi, mahasiswa harus membuat proposal terlebih dahulu. Apabila ada indikasi plagiarisme pada proposal skripsi maka mahasiswa yang bersangkutan masih memiliki waktu yang cukup untuk memperbaiki penelitian dan menyusun skripsinya. Pertimbangan kedua adalah karena dokumen proposal skripsi tidak dipublikasikan oleh perguruan tinggi. Dengan demikian, membandingkannya dengan skripsi yang sudah dipublikasikan sudah cukup. Pertanyaan yang terjawab dari proses bisnis ini adalah apakah proposal skripsi yang diajukan memiliki kesamaan yang tinggi dengan skripsi yang sudah dilakukan sebelumnya. Dengan kata lain dapat diketahui apakah penelitian yang diusulkan (melalui proposal skripsi) sudah pernah dilakukan sebelumnya atau tidak. Dari lingkup proses bisnis ini lah kemudian diidentifikasi *user story* yang relevan.

Tabel 1. Daftar *User Story* Bisnis

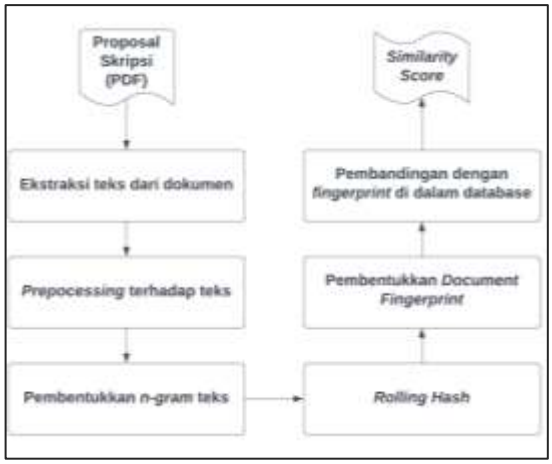
Kode	User Story
[US01]	Sebagai **mahasiswa** saya bisa menghitung *similarity score* proposal skripsi secara daring untuk mengantisipasi plagiarisme.
[US02]	Sebagai **operator** saya bisa menambahkan skripsi yang sudah dipublikasikan ke dalam basis data aplikasi untuk dijadikan acuan pembanding proposal skripsi.
[US03]	Sebagai **operator** saya bisa melihat hasil penghitungan similarity score proposal skripsi mahasiswa untuk memudahkan penentuan kelayakan penelitiannya.
[US04]	Sebagai **operator** saya bisa melakukan konfigurasi aplikasi untuk memilih pengaturan yang sesuai kebutuhan perguruan tinggi.
[US05]	Sebagai **operator** saya bisa melakukan semua hal yang bisa dilakukan oleh mahasiswa di dalam aplikasi

Sumber: penulis

Daftar *user stories* yang berhasil diidentifikasi adalah seperti yang disebutkan pada Tabel 1. Sebelum menentukan *user story*, terlebih dahulu dilakukan identifikasi aktor atau *user role* untuk aplikasi ini. Ada dua aktor di dalam aplikasi ini yaitu operator dan mahasiswa. Operator memiliki user stories yang lebih banyak daripada mahasiswa karena bertindak sebagai admin yang bertanggung jawab terhadap berbagai konfigurasi aplikasi. Bahkan seperti yang dinyatakan pada *user story* [US05], operator dapat melakukan semua hal yang bisa dilakukan oleh mahasiswa di dalam aplikasi yang diusulkan ini. Sementara itu, hanya ada satu user story yang dimiliki oleh mahasiswa yaitu menghitung *similarity score* dokumen proposal skripsinya. Total ada lima *user stories* yang dihasilkan dalam penelitian ini.

3.2 Konfigurasi Algoritma Winnowing

Kebutuhan fungsional utama dari aplikasi ini adalah mampu menghitung *similarity score* sebuah dokumen terhadap dokumen lain yang ada di dalam database. Kebutuhan tersebut dapat dipenuhi dengan algoritma *winnowing*. Penelitian ini tidak dihasilkan algoritma penghitung *similarity* baru, tetapi melakukan konfigurasi terhadap algoritma *winnowing* yang selama ini sudah terbukti secara akurat menghitung nilai kemiripan dokumen teks. Konfigurasi atau modifikasi algoritma ini perlu dilakukan agar sesuai dengan rancangan aplikasi yang memenuhi kelima *user stories* pada Tabel 1.



Gambar 2. Implementasi Algoritma Winnowing
Sumber: penulis

Alur proses dari algoritma winnowing yang diimplementasikan dalam aplikasi ini, secara umum dapat dilihat melalui Gambar 2. Dokumen yang dijadikan input adalah proposal skripsi mahasiswa dalam bentuk file PDF. Kemudian ekstraksi teks dilakukan dari dokumen tersebut sehingga dihasilkan dokumen baru yang hanya berisi teks.

Selanjutnya, preprocessing dilakukan terhadap teks tersebut untuk menghilangkan spasi, karakter spesial, dan karakter yang tidak perlu lainnya. Setiap teks yang menggunakan karakter *uppercase* juga diubah menjadi bentuk *lowercase*. Setelah preprocessing adalah tahap pembentukkan n-gram. Tahap ini membagi teks dalam kelompok-kelompok yang terdiri atas *n* karakter. *Rolling hash* kemudian dilakukan terhadap setiap kelompok n-gram yang terbentuk. Dari sini, selanjutnya dibentuk *document fingerprint* melalui proses *windowing*. *Fingerprint* ini bersifat unik untuk setiap dokumen. Tahap setelahnya adalah membandingkan *fingerprint* dokumen proposal skripsi dengan *fingerprint* dokumen yang ada di dalam *database*. Tahap ini menghasilkan nilai *similarity* proposal dengan dokumen-dokumen yang sudah ditambahkan ke dalam *database*. Semakin tinggi nilai similarity yang diperoleh menunjukkan semakin tinggi tingkat kesamaannya. Sebagai contoh, nilai *similarity* sebesar 100% berarti dokumen tersebut sama persis dengan salah satu dokumen yang ada di dalam *database*.

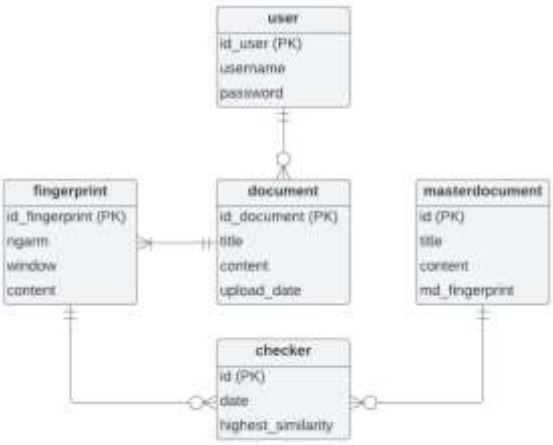
Implementasi algoritma winnowing seperti yang terlihat pada Gambar 2.. memungkinkan kita untuk melakukan beberapa konfigurasi sesuai dengan kebutuhan. Konfigurasi dapat dilakukan dalam tahap pembentukkan *n-gram*, *rolling hash*, dan pembentukkan *fingerprint*. Pada penelitian ini, bentuk n-*gram* yang digunakan adalah 45-*gram*. Pemilihan bentuk 45-gram didasarkan atas jumlah karakter rata-rata dari sebuah dokumen proposal skripsi yaitu 8.000 kata. Dengan jumlah kata sebesar itu, tidak disarankan untuk menggunakan bentuk n-gram yang lebih kecil dari 30-gram. Penelitian ini secara khusus membandingkan antara pengaruh n-gram yang dipilih dengan akursi *similarity score* yang dihasilkan.

$$H(C1...Ck) = c_1 * b_{(k-1)} + c_2 * b_{(k-2)} + \cdots + c_{(k-1)} * b_k + c_k \quad (1)$$

Konfigurasi selanjutnya yang dilakukan adalah pada tahap *hashing*. Penelitian ini menggunakan prosedur *rolling hash* seperti rumus (1). Notasi *c* berarti nilai ASCII dari karakter teks. Sedangkan *b* merupakan bilangan prima yang dipilih. Pada penelitian ini bilangan prima yang digunakan adalah 3. Sementara itu k merujuk pada bentuk n-gram dari kata yang dimasukkan dalam proses *hashing*. Konfigurasi lain yang dilakukan dalam penelitian ini adalah penentuan jumlah *window* saat pembentukkan *fingerprint*. Sama halnya dengan penentuan n-gram, penelitian ini juga membandingkan nilai *window* yang dipilih dengan hasil yang didapat dan diputuskan nilai window yang digunakan adalah 30. Konfigurasi terakhir, pada proses pembandingan *document fingerprint* dilakukan dengan menghitung *Jaccard Coeficient Similarity* untuk penentuan nilai *similarity* dokumen yang diuji.

3.3 Rancangan Aplikasi

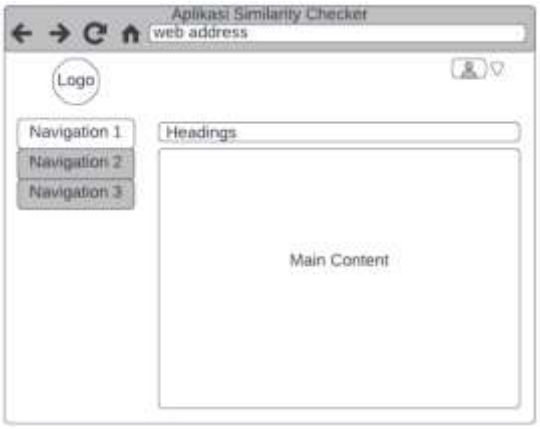
Rancangan aplikasi dibuat untuk memenuhi setiap *user stories* yang telah diidentifikasi sebelumnya. Proses utama yang ada di dalam aplikasi ini adalah meghitung similarity score prosposal skripsi dengan dokumen yang sudah dipublikasikan. Rancangan *database* perlu dibuat untuk mendukung proses tersebut. Penelitian ini menghasilkan rancangan database dan juga rancangan antarmuka aplikasi.



Gambar 3. Rancangan Database Aplikasi
Sumber: penulis

Rancangan *database* untuk aplikasi ini adalah seperti yang terlihat pada Gambar 3. Database yang digunakan adalah relational database, dengan demikian model rancangannya dibuat dengan menggunakan *entity relationship diagram* (ERD). *Database* aplikasi ini terdiri atas 5 buah tabel, yaitu: *user*, *document*, *fingerprint*, *checker*, dan *masterdocument*. Tabel *user* menyimpan informasi pengguna yang menggunakan aplikasi. *Username* dan *password* yang digunakan untuk login ke dalam aplikasi,

disimpan pada tabel ini. Data terkait dokumen yang sudah dipublikasikan disimpan pada tabel *masterdocument*. Sementara itu, dokumen proposal skripsi yang ingin dihitung *similarity score*-nya disimpan pada tabel *document* sedangkan *fingerprint* dokumennya disimpan pada tabel *fingerprint*. Ketika proses penghitungan similarity dilakukan, setiap data yang terkait disimpan pada tabel *checker*. Kelima tabel ini sudah cukup untuk memenuhi setiap skenario penggunaan yang ada di dalam *user stories*.



Gambar 4. Wireframe Antarmuka Pengguna
Sumber: penulis

Penelitian ini juga menghasilkan rancangan antarmuka pengguna aplikasi. Gambar 4 menunjukkan *wireframe* untuk antarmuka. Aplikasi ini dirancangan sebagai aplikasi berbasis *website*. *Wireframe* pada Gambar 4 menunjukkan pembagian *layout interface* aplikasi. Secara umum layar antarmuka pengguna terbagi menjadi tiga bagian utama. Bagian pertama adalah *header* yang berada di area atas antarmuka. Komponen *header* ada dua, yaitu logo sebagai petunjuk identitas di sebelah kiri dan menu pengguna di sebelah kanan. Bagian kedua adalah navigasi aplikasi. Navigasi terletak di area kiri dengan ukuran lebar 30% dari total lebar halaman antarmuka. Sedangkan bagian yang terakhir adalah konten utama. Konten utama secara dinamis menampilkan informasi sesuai dengan halaman yang sedang diakses. Konten utama ini menggunakan sekitar 70% lebar layar antarmuka. Proporsi pembagian konten utama dengan navigasi ini memperhatikan saran yang diberikan oleh Shneiderman et al (2016).

3.4 Purwarupa Aplikasi

Pengembangan purwarupa aplikasi dilakukan setelah semua rancangan yang diperlukan berhasil dibuat. Proses pengembangan purwarupa mengacu kepada rancangan, model kebutuhan, dan juga user stories yang dihasilkan pada langkah-langkah sebelumnya.

Purwarupa aplikasi dikembangkan dalam bentuk aplikasi berbasis *website*. Bahasa pemrograman yang digunakan untuk *backend* aplikasi adalah PHP. Sedangkan frontend aplikasi dikembangkan dengan HTML, CSS, dan *Javascript*. Database Management System (DBMS) yang digunakan oleh purwarupa ini adalah MySQL. Purwarupa ini dikembangkan oleh dua orang *programmer* secara *pair programming*.

Gambar 5. Tampilan Ekstraksi Teks dari PDF
Sumber: penulis

Contoh tangkapan layar dari purwarupa yang dikembangkan dapat dilihat pada Gambar 5 dan 6. Gambar 5 menunjukkan tampilan aplikasi saat proses ekstraksi teks dari dokumen proposal yang berbentuk file PDF. Sesuai nama prosesnya, hanya teks yang diambil untuk dihitung *similarity score*. Komponen lain seperti gambar atau grafik tidak diikut sertakan dalam proses ini.

Gambar 6. Tampilan Hasil Penghitungan *Similarity Score*
Sumber: penulis

Gambar 6 menunjukkan tampilan hasil pembandingan dokumen proposal dengan dokumen yang ada di dalam database. Proses pembandingan ini menghasilkan nilai *similarity score*. Operator sebelumnya sudah menentukan berapa batas nilai *similarity* yang ingin digunakan sebagai acuan. Apabila nilai similarity yang didapatkan oleh proposal skripsi berada di bawah batas yang ditentukan tersebut, maka dokumen tersebut dinyatakan tidak ada indikasi plagiarisme. Informasi tersebut ditampilkan di halaman hasil pembandingan seperti yang terlihat pada Gambar 6.

Di bagian akhir penelitian, purwarupa diuji dengan *scenario-based testing*. Kelima *user stories* yang sebelumnya sudah ditampilkan pada Tabel 1 dijadikan sebagai *test case*. Secara umum, hasil pengujian menunjukkan bahwa purwarupa ini berhasil lulus pengujian untuk setiap skenario penggunaan. Hal tersebut menunjukkan bahwa purwarupa aplikasi berhasil memenuhi kebutuhan yang ada. Hal tersebut diinterpretasikan bahwa rancangan, model kebutuhan, dan *user stories* yang dihasilkan oleh penelitian ini bisa dikembangkan menjadi aplikasi yang berjalan. Penelitian ini berakhir setelah hasil pengujian terhadap purwarupa selesai dilakukan. Beberapa poin kesimpulan berhasil didapatkan setelah semua langkah penelitian ini dilakukan.

4. KESIMPULAN

Setelah penelitian ini selesai dan setiap langkah penelitian yang direncanakan sudah dilakukan, maka tahap terakhir sebagai penutup adalah penyusunan kesimpulan. Kesimpulan disusun berdasarkan hasil penelitian. Poin-poin kesimpulan ini secara umum menjawab semua pertanyaan penelitian yang dirumuskan di bagian awal penelitian.

Penelitian ini mengidentifikasi sebanyak lima *user stories* yang sesuai dengan kebutuhan bisnis perguruan tinggi untuk menghitung *similarity score* dokumen yang akan dipublikasikan. Kelima user stories ini merepresentasikan *requirement* yang harus dipenuhi oleh aplikasi pemeriksaan kemiripan dokumen. *User stories* ini merupakan jawaban pertanyaan penelitian yang pertama.

Penelitian ini juga mengusulkan konfigurasi algoritma *winnowing* untuk pemeriksaan dokumen proposal skripsi. Bentuk n-gram yang disarankan adalah 45-gram atau paling tidak lebih banyak dari 30-gram. Sedangkan jumlah *windows* yang disarankan adalah 30 *windows*. Konfigurasi tersebut disarankan berdasarkan jumlah karakter dari dokumen proposal skripsi yang rata-rata berjumlah sekitar 8.000 kata. Dengan konfigurasi tersebut diharapkan aplikasi memiliki kinerja yang seimbang antara waktu proses dengan akurasi hasilnya. Konfigurasi algoritma *winnowing* ini merupakan jawaban untuk pertanyaan penelitian yang kedua.

Pada bagian akhir penelitian ini dihasilkan sebuah purwarupa aplikasi. Purwarupa ini telah melewati *scenario-based testing* dengan hasil yang sesuai harapan. Dengan demikian, ini menunjukkan bahwa rancangan yang dihasilkan oleh penelitian ini bisa dikembangkan menjadi aplikasi yang berjalan dan layak untuk dipertimbangkan sebagai acuan pengembangan aplikasi yang sejenis. Purwarupa yang dihasilkan, berikut dengan hasil pengujiannya, merupakan jawaban pertanyaan penelitian yang terakhir. Pada akhirnya penelitian ini menjawab tiga pertanyaan penelitian yang dirumuskan dari masalah yang coba diatasi.

DAFTAR PUSTAKA

Dennis, A., Wixom, B.H., Roth, R.M. (2018). *System analysis and design* (7th edition). Wiley

Pressman, R.S., & Maxim, B.R. (2020). *Software Engineering A Practicioner's Approach Ninth Edition*. McGraw-Hill Education.

Sharp H, Rogers Y, & Preece J. (2019). *Interaction Design Beyond Human-Computer Interaction 5th Edition*. Wiley. 2019, 385-417

Shneiderman B, Plaisant C, Cohen M, Jacobs S, Elmqvist N, & Diakopoulos N. (2016). *Designing the User Interface: Strategies for Effective Human-Computer Interaction 6th Edition. Pearson*.

Sunardi S, Yudhana A, & Mukaromah LA. (2019). *Indonesia Words Detection Using Fingerprint Winnowing Algorithm*. Jurnal Informatika.