

# ***Software Development Dengan Extreme Programming (XP) Pada Aplikasi Deteksi Kemiripan Judul Skripsi Berbasis Android***

Imam Ahmad<sup>1</sup>, Rohmat Indra Borman<sup>2</sup>, Jafar Fakhrurozi<sup>3</sup>, Gavan Gorbi Caksana<sup>4</sup>  
Fakultas Teknik dan Ilmu Komputer, Universitas Teknokrat Indonesia  
Jl. ZA. Pagar Alam No.9 -11, Labuhan Ratu, Kota Bandar Lampung, Lampung, Indonesia  
Email: [imamahmad@teknokrat.ac.id](mailto:imamahmad@teknokrat.ac.id)<sup>1</sup>, [rohmat\\_indra@teknokrat.ac.id](mailto:rohmat_indra@teknokrat.ac.id)<sup>2</sup>,  
[jafar.fakhrurozi@teknokrat.ac.id](mailto:jafar.fakhrurozi@teknokrat.ac.id)<sup>3</sup>, [gavangorbicaksana26@gmail.com](mailto:gavangorbicaksana26@gmail.com)<sup>4</sup>

**Abstrack** – To determine the topic or research title for the student's thesis, it is possible that the similarity of the title to the previous title, either accidentally or unintentionally. Therefore, the relevant section must check the titles submitted by students, so as to avoid plagiarism. The development of a thesis title similarity detection application is one of the efforts that can be made in solving this problem. The development of software requires the right method to develop applications according to user needs. Extreme Programming (XP) is a part of the agile development approach based on iterative development, which is based on needs and solutions through collaboration among the development team. So, in this study, the development of a similarity detection application for the title of a thesis with the XP software development approach was carried out.

**Keywords** – agile development, extreme programming, thesis

**Intisari** – Untuk menentukan topik penelitian maupun judul untuk skripsi mahasiswa dimungkinkan terjadi kemiripan judul dengan judul yang sebelumnya, baik secara disengaja maupun tidak sengaja. Maka dari itu, bagian terkait harus mengecek terhadap judul yang diajukan oleh mahasiswa, sehingga dapat menghindari plagiat. Pengembangan aplikasi deteksi kemiripan judul skripsi, merupakan salah satu upaya yang dapat dilakukan dalam penyelesaian masalah tersebut. Dalam *software development* perlu adanya metode yang tepat untuk mengembangkan aplikasi yang sesuai dengan kebutuhan pengguna. *Extreme Programming (XP)* adalah bagian pendekatan *agile development* yang berbasiskan *iterative development*, yang berdasarkan dari kebutuhan dan pemecahan solusi melalui kerjasama diantara tim pengembang. Maka, pada penelitian ini dilakukan pengembangan aplikasi deteksi kemiripan judul skripsi dengan pendekatan pengembangan *software XP*.

**Kata Kunci** – agile development, extreme programming, skripsi

## **I. PENDAHULUAN**

*Software development* merupakan suatu aktivitas untuk menyempurnakan sistem yang sebelumnya telah dengan sistem berbasis komputer. Ketika melaksanakan proyek *software development* tidak menutup kemungkinan terjadinya perubahan mendadak yang berdampak pada tim pengembang perangkat lunak akan menghadapi kondisi maupun situasi yang mendesak. Pengembangan sistem pada umumnya melalui proses tertentu dan tahapan dalam pengembangannya melalui sebuah metode. Maka diperlukan metode pengembangan sistem untuk pengembangan perangkat lunak yang sesuai dengan kasus yang dihadapi. Metode pengembangan sistem merupakan kerangka pijakan utama dalam merancang dan mengembangkan *software* untuk menghasilkan sebuah sistem yang sesuai keinginan dan kebutuhan bisnis suatu organisasi atau pengguna sistem [1]. Sebuah *software development* sebaiknya dilakukan melalui tahapan yang dilakukan dengan matang dan terencana [2]. Terdapat beberapa metodologi pengembangan *software* yang bisa diterapkan oleh pengembang *software*, diantaranya *waterfall*, RAD, *prototype* serta *agile development*. Metode *agile development* pengembangan *software engineering* yang dikenal efisien dan fleksibel [3]. *Agile*

*software development* merupakan metode rancang bangun *software* yang berdasarkan pada iteratif *development*, di mana kebutuhan pada setiap tahapan dan solusi yang ditawarkan berbasiskan pendekatan kerjasama antar tim [4]. Salah satu metode *agile development* yang dapat digunakan penyesuaian persyaratan pengembangan adalah *extreme programming* (XP). XP adalah pengembangan *software* yang sasaran dari metode ini adalah untuk pembangunan melalui penetapan kebutuhan yang kurang jelas atau adanya perubahan terhadap kebutuhan yang sangat cepat dan melalui tim yang berskala kecil sampai menengah [5].

Beberapa penelitian yang menggunakan XP dalam pengembangan system menunjukkan hasil yang baik. Penelitian penerapan metode (XP pada perancangan aplikasi perhitungan kuota SKS mengajar dosen [5], menunjukkan bahwa pada hasil pengujian *black box testing* aplikasi telah sesuai dengan kebutuhan. Tahapan Pengembangan XP mampu membantu dalam menghasilkan *software* yang sesuai dengan *functional requirement*. Penelitian lainnya, yang meneliti XP dalam merancang aplikasi pengelolaan surat keterangan kependudukan [6], mendapatkan hasil bahwa pendekatan yang diajukan dapat diimplementasikan dalam membangun aplikasi sederhana yang tidak membutuhkan proses yang diulang-ulang. Hasil dari pengujian *black-box testing* didapat bahwa XP aplikasi telah berfungsi dengan baik dan sesuai kebutuhan. Sedangkan penelitian yang mengimplementasikan metode XP pada *e-commerce* berbasis *m-commerce* [7], memperlihatkan bahwa teknik pendekatan menggunakan XP dalam setiap tahapnya memberikan percepatan pengembangan pada setiap perubahan yang diinginkan oleh pengguna tanpa adanya batasan.

Pada penelitian ini XP diterapkan pada aplikasi deteksi kemiripan judul skripsi berbasis android. Skripsi merupakan istilah yang digunakan untuk menyebut karya tulis ilmiah mahasiswa jenjang Strata-1 (S1). Skripsi berisi hasil riset membahas suatu masalah dalam bidang ilmu tertentu [8]. Salah satu tujuan pembuatan skripsi adalah untuk membentuk mahasiswa agar mendapatkan sikap dan mental ilmiah dan mampu menerapkan ilmunya dalam penyelesaian masalah pada dunia nyata [9]. Sebelum membuat karya ilmiah atau skripsi, mahasiswa diharuskan membuat usulan judul untuk kemudian diajukan dengan memaparkan topik penelitian. Pada Universitas Teknokrat Indonesia, untuk pengajuan judul skripsi diajukan melalui form pengajuan judul dengan mengisi topick judul dan paparan singkat penelitian yang akan dilakukan. Dalam menentukan judul skripsi dimungkinkan terjadi kemiripan judul antara mahasiswa satu dengan yang lain baik secara disengajaja maupun tidak sengaja. Program Studi harus mengecek judul yang diajukan oleh mahasiswa, sehingga dapat mengambil keputusan apakah judul diterima ataupun ditolak berdasakan dengan salah satunya adalah tingkat kemiripan judul. Hal ini bertujuan untuk menghindari dari plagiasi dan kesamaan judul. Mahasiswa yang mengajukan judul setiap semesternya sangat banyak, hal ini menuntut ketelitian dan upaya yang lebih untuk melakukan pengecekan tingkat kemiripan judul. Mengecek dengan cara manual akan dibutuhkan tenaga dan waktu yang lebih. Untuk membantu dalam pengecekan tingkat kemiripan judul maka dikembangkan aplikasi deteksi kemiripan judul. Salah satu pendekatan dalam deteksi kemiripan adalah dengan pencocokan *string* Boyer-Moore. Pencocokan dengan Boyer-Moore yaitu mencocokkan dan mengamati pola *string* diawali dari posisi sebelah kanan kemudian ke kiri dari *string* yang akan dicari [10][11]. Penelitian ini menghasilkan aplikasi berupa persentase kemiripan judul berdasarkan dengan judul skripsi sebelumnya yang diambil dari database skripsi yang ada, dengan pendekatan pengembangan sistem XP dan pencocokan kata menggunakan Boyer-Moore berbasis Andorid.

## II. SIGNIFIKANSI STUDI

### A. Studi Literatur

#### 1. Agile Development

*Agile development* merupakan kumpulan metodologi rancang bangun *software* yang berdasarkan dari persyaratan dan solusi melalui kolaborasi dan kerjasama tim [12]. Metode ini juga disebut sebagai pendekatan *incremental* yang berfokus pada pembangunan yang singkat,

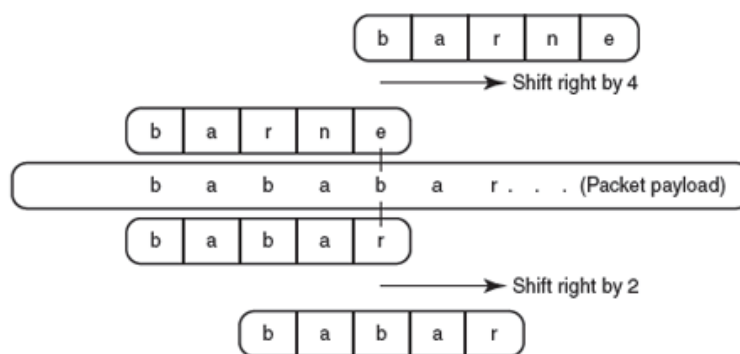
dengan merilis *software* tahap demi tahap, pengurangan proses *overhead*, serta dihasilkan *coding* yang memiliki kualitas tinggi dan pada tahap pengembangan melibatkan *user* secara langsung [4]. *Agile development* berusaha untuk dapat mengatasi permasalahan perkembangan *software* diantaranya:

- a) *Software* memiliki kebutuhan yang sulit untuk diprediksi dan selalu akan berubah. Tidak hanya itu, keinginan *user* juga sering berubah beriringan dengan jalannya proyek.
- b) Tumpang tindihnya rancangan dan pembangunan, sehingga sulit diprediksi seberapa jauh rancangan yang diperlukan sebelum pengembangan.
- c) Tidak dapat diprediksi analisis, rancangan, implementasi dan pengujian sistem seperti yang direncanakan dan diinginkan.

Terdapat beberapa pendekatan *software development* yang termasuk kedalam metode *agile software* diantaranya: 1) *Agile Modeling*; 2) *Extreme Programming*; 3) *Model Scrum*; 4) *Dynamic Systems Development Method*; dan 5) *Adaptive Software Development*.

### 2. Boyer-Moore String Matching

*String matching* atau mencocokkan *string* merupakan pendekatan pemecahan masalah untuk mendapatkan pola berdasarkan susunan dari karakter *string* yang terdapat pada bagian dari isi teks atau *string* lainnya [13]. Sedangkan Boyer-Moore merupakan pendekatan yang mencocokkan *string* dengan cara melihat dari dari kanan kemudian menuju ke kiri dengan melakukan pelacakan karakter pada teks [14]. *Good-suffix shift* dan *bad-character shift* merupakan dua fungsi *shift* yang digunakan Boyer-Moore dalam mengambil langkah dalam pencarian setelah terdapat ketidaksamaan antara karakter *pattern* dan karakter teks yang dilakukan pencocokan [10]. Algoritma ini memiliki kelebihan proses lebih cepat secara rata-rata jika dibandingkan dengan algoritma *string matching* yang lain [15]. Ide utama dari Boyer-Moore adalah mencocokkan *string* berdasarkan *string* yang paling kanan.



Gambar 1. Porses Pencocokan Boyer-Moore

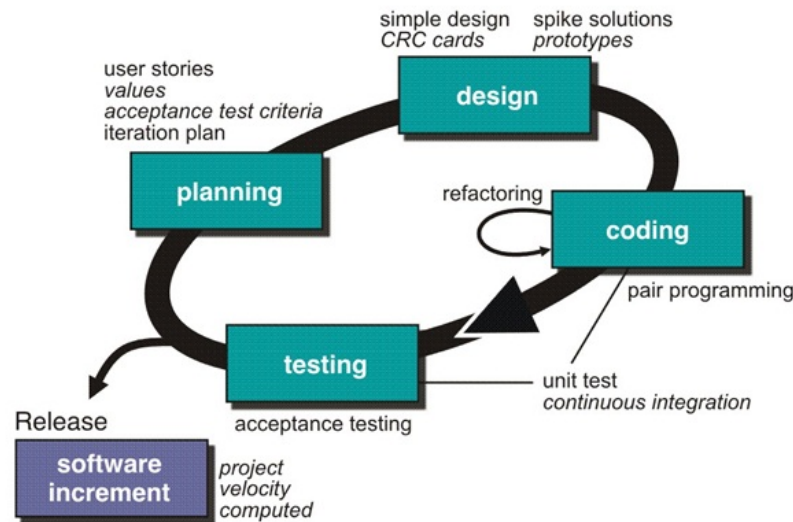
### 3. Andorid

Android merupakan *operating system*, *middleware*, dan *mobile application* yang berbasis pada Linux Kernel, dibangun oleh Google dan *Open Handset Alliance*. Android sistem operasi mobile yang memiliki sifat *open source*. Google memiliki tujuan untuk mengembangkan produk *real-world* pengalaman para pengguna dapat meningkat, itu sebabnya Android bersifat *open source project*. Android dibangun dengan memanfaatkan SDK (*Software Development Kit*) secara komprehensif menggunakan alat yang cukup untuk membangun *software* yang kaya akan fitur dan *powerful*.

#### B. Metode Penelitian

Dalam *software development* aplikasi deteksi kemiripan judul skripsi, pada penelitian ini menggunakan pendekatan metodologi pengembangan sistem *Extreme Programming* (XP). XP merupakan pendekatan yang berfokus pada *coding* yang merupakan aktivitas utama pada semua tahapan atau siklus pengembangan sistem [16]. Pada XP memiliki kelebihan pada tahapannya membutuhkan waktu yang cepat dan dapat diulangi pada bagian yang berbeda sesuai dengan

fokus tujuan pengembangan sistem [17]. Selain itu dengan pengembangan system yang menyederhanakan setiap proses dalam tahapannya, maka metode XP menjadi lebih adaptif dan fleksibel. Berikut ini tahapan-tahapan XP dalam pengembangan sistem yang dilakukan.



Gambar 1. Tahapan Pada *Extreme Programming* (XP)

Dari tahapan pengembangan XP diatas, berikut adalah penjelasan dari setiap tahapannya.

#### 1. *Planning* (Perencanaan)

Tahapan ini merupakan langkah awal pada pengembangan XP, dimana akan dilakukan identifikasi alur proses dari *software* yang akan dibuat, mendefinisikan luaran *software*, fasilitas-fasilitas yang ada pada *software*, fungsi dari aplikasi, dan alur proses pengembangan *software* [18][19]. Selain itu pada tahapan ini *software developer* dituntut untuk membuat user stories yang menggambarkan output, fitur, dan fungsi-fungsi dari *software* yang akan dibuat [19]. Tidak hanya itu, pada tahap *planning* juga memuat tahapan dimana *developer* menentukan fungsionalitas keseluruhan dari *software* yang akan dikembangkan [6].

#### 2. *Design* (Perancangan)

Tahapan selanjutnya adalah mendesain sistem. *Design* sistem pada penelitian ini menggunakan *Unified Modelling Language* (UML). UML adalah bahasa visual untuk memodelkan dan menggambarkan komunikasi sebuah sistem dengan mempergunakan diagram [20]. Pada penelitian ini, *use case diagram* dipergunakan dalam tahap perancangan. *Use case* adalah sebuah diagram yang memiliki kemampuan dalam memvisualkan interaksi diantara aktor dan *software* yang akan dikembangkan [21].

#### 3. *Coding* (Pengkodean)

*Coding* atau pengkodean adalah tahapan dimana rancangan yang telah dibuat kemudian diimplementasikan kedalam bentuk bahasa pemrograman yang dapat dikenali oleh komputer [22].

#### 4. *Testing* (Pengujian)

Untuk menegaskan bahwa *software* dapat berfungsi dengan baik, maka sistem harus melalui tahap pengujian agar dapat menemukan kesalahan-kesalahan [23]. Pengujian atau *testing* menggunakan pendekatan *black box testing*. Pengujian kotak hitam atau *black box testing* merupakan *software testing* yang menguji pada spesifikasi fungsional untuk mengetahui apakah fitur, *input* dan *output* dari *software* telah sesuai spesifikasi telah ditetapkan [24]. Pada pengujian ini *software* diuji tanpa memperhatikan struktur logika internal perangkat lunak [25].

### III. HASIL DAN PEMBAHASAN

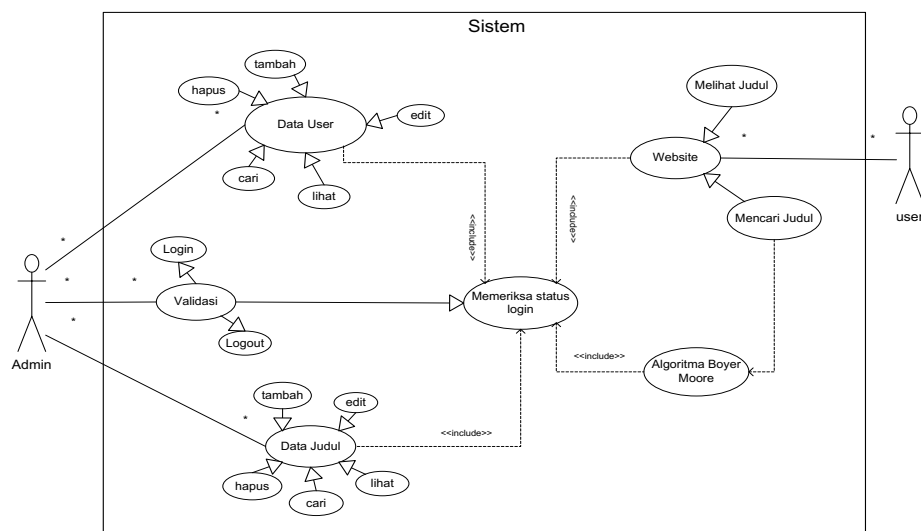
#### A. Planning (Perencanaan)

Kegiatan ini dimulai dengan pengumpulan data melalui wawancara dengan tujuan mendengarkan pelanggan, hal itu berguna untuk mengumpulkan informasi mengenai permasalahan yang dialami pengguna. Dari identifikasi masalah yang didapatkan, kemudian disusun kebutuhan sistem yang akan dibangun. Permasalahan utama pada penelitian adalah proses pengecekan judul skripsi pada Universitas Teknokrat Indonesia yang mahasiswa ajukan apakah terdapat kemiripan atau indikasi adanya plagiasi dari judul sebelumnya. Dari permasalahan tersebut kebutuhan akan aplikasi yang akan dikembangkan adalah sistem yang mampu menentukan tingkat kemiripan judul berupa presentasi jumlah kemiripan berdasarkan judul sebelumnya yang diambil dari database judul skripsi pada Universitas Teknokrat Indonesia. Hasil identifikasi kebutuhan kemudian disusun dalam bentuk analisa kebutuhan fungsional. *Functional requirement* atau kebutuhan fungsional dibutuhkan untuk identifikasi proses atau fasilitas apa saja yang dapat dikerjakan oleh sistem, serta siapa pengguna sistem yang dapat menjalankan sistem tersebut [26]. Pengguna sistem ada dua, yaitu: pengguna atau mahasiswa dan admin atau petugas administrasi. Berikut ini adalah kebutuhan fungsional dari aplikasi deteksi kemiripan judul skripsi pada Universitas Teknokrat Indonesia:

1. Admin
  - a) Admin dapat melakukan logine dengan *user name* dan *password*;
  - b) Admin dapat mengelola data judul skripsi, dengan cara menambah, menghapus dan mengupdate judul apabila ada perubahan;
  - c) Admin dapat melakukan pencarian judul;
  - d) Admin dapat melihat presentase kemiripan judul yang diajukan oleh mahasiswa;
  - e) Admin dapat melakukan manajemen *user*.
2. User / Mahasiswa
  - a) Mahasiswa dapat mengajukan judul dan melihat hasil deteksi kemiripan judul berupa persentase kemiripan;
  - b) Mahasiswa dapat melakukan pencarian judul dan melihat persentase kemiripan.

#### B. Design (Perancangan)

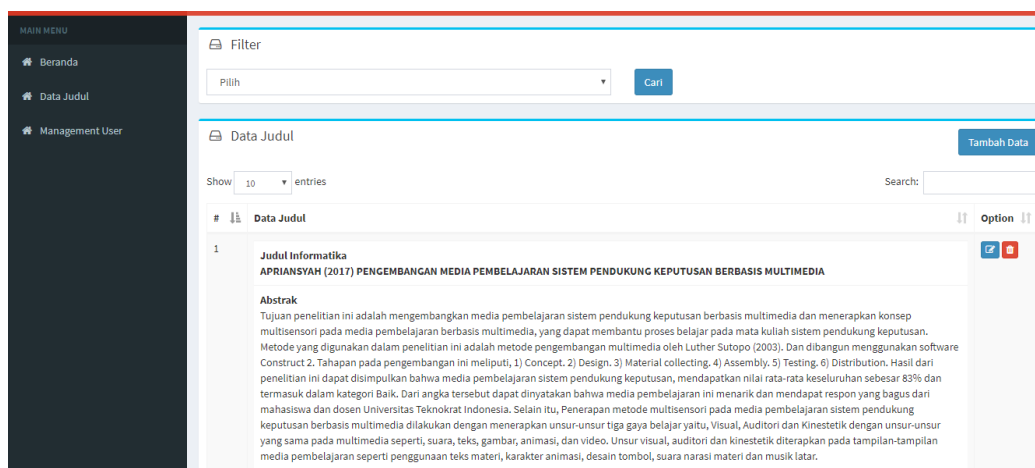
Setelah dilakukan tahapan *planning*, selanjutnya dilakukan rancangan berdasarkan analisa kebutuhan fungsional yang telah dilakukan. Sesuai dengan analisa kebutuhan fungsional, pengguna atau aktor pada aplikasi ini ada dua, yaitu: Admin dan *user* atau mahasiswa. Berikut ini adalah hasil rancangan dengan menggunakan *use case diagram* (Gambar 2).



Gambar 2. Use Case Diagram Aplikasi Deteksi Kemiripan Judul Skripsi

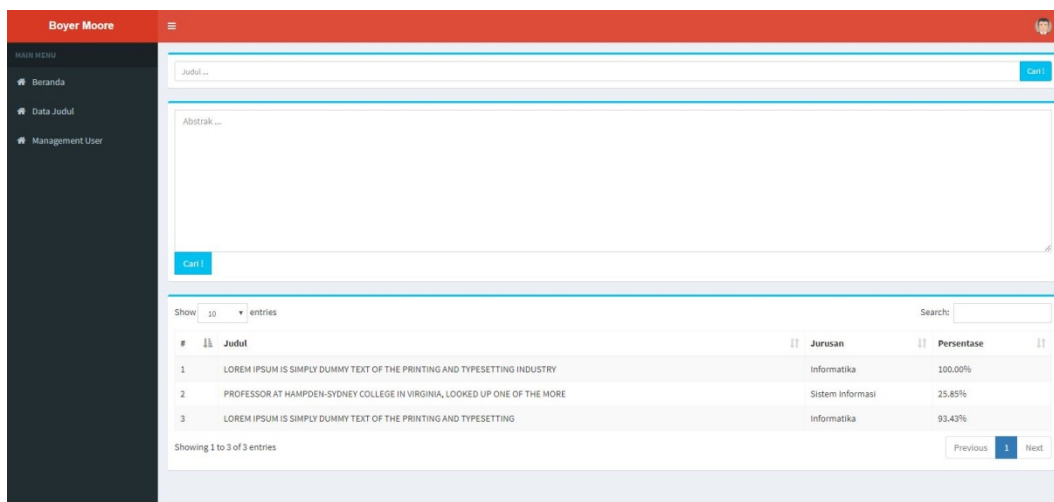
C. Coding (Pengkodean)

Sistem diimplementasikan berbasis android untuk akses *frontend* atau pengguna dan web untuk *backend* atau admin. Bahasa pemrogramana yang digunakan adalah bahasa pemrograman Java dengan compailer untuk aplikasi Android yang digunakan oleh *user* / mahasiswa dengan Andorid Studio, sedangkan aplikasi yang digunakan oleh Admin dengan menggunakan Sublime Text 3. Untuk admin, pertama yang diakses adalah login. Admin diminta untuk memasukan *user name* dan *password*. Setelah *login*, Admin dapat menambah, mengubah, menghapus dan melihat judul yang ada.



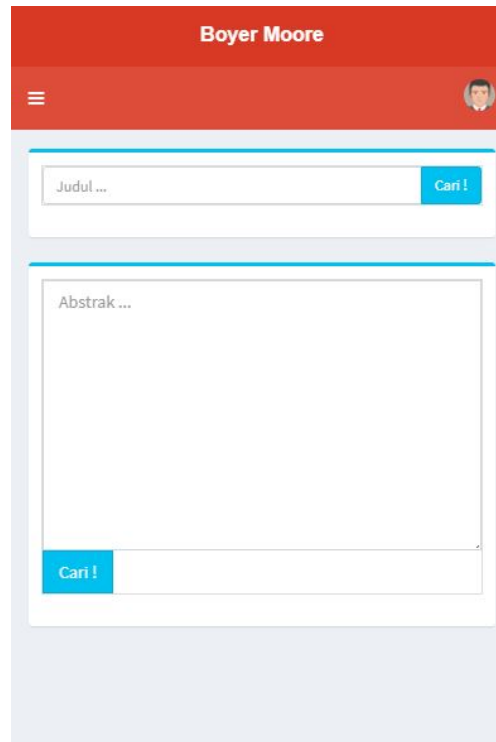
Gambar 3. Form Admin Untuk Mengelola Judul Skripsi

Selain itu, admin dapat melakukan pengecekan tingkat kemiripan judul skripsi pada menu pencarian. Admin tinggal memasukan judul yang diajukan oleh mahasiswa untuk dicek tingkat kemiripanya.



Gambar 4. Form Pengecekan Tingkat Kemiripan Judul Skripsi

Sedangkan untuk *user* atau mahasiswa setelah melakukan *login*, *user* dapat melakukan pencarian judul yang diajukan. Tampilan Berikut digunakan untuk melakukan pencarian judul skripsi.



Gambar 5. Pencarian Judul

Pencocokan karakter yang digunakan adalah dengan pendekatan Boyer Moore. Pendekatan ini memiliki cara kerja dengan melakukan *matching* atau pencocokan antara *pattern* dan *text* dimulai dari karakter paling kanan terlebih dahulu kemudian ke kiri, dimana pencocokan berdasarkan dari occurrence heuristic (OH) dan match heuristic (MH). Sebagai contoh kasus pencocokan *string* “PAKAR” dan “SISTEM PAKAR” sebagai kata pencarian. Langkah pertama hitung OH dan MH dari *string pattern* terhadap teks yang ada.

TABEL I  
NILAI OCCURENCE HEURISTIC (OH) DAN MATCH HEURISTIC (MH)

<i>PATTERN</i>	P	A	K	A	R
<i>OH</i>	4	3	2	1	0
<i>MH</i>	5	5	5	5	1

Untuk pencocokan string pada text “SISTEM PAKAR” dengan pattern “PAKAR” dapat dilihat bahwa karakter paling kanan adalah “E” tidak cocok dengan karakter “R”, sehingga akan bergeser nilainya berdasarkan tabel OH dan MH pada tabel I. Langkah ini dapat dilihat pada table II.

TABEL II  
PENCOCOKAN TEXT DAN PATTERN

<i>INDEX</i>	0	1	2	3	4	5	6	7	8	9	10	11
<i>TEXT</i>	S	I	S	T	E	M		P	A	K	A	R
<i>PATTERN</i>	P	A	K	A	R							

Pada tabel OH dan MH karakter “E” tidak ada, maka jumlah pergeseran dilakukan sebanyak 5 langkah sesuai dengan nilai MH pada table I. Langkah ini dapat dilihat pada table III.

TABEL III  
PEGESERAN PATERN SEBANYAK 5 PEGESERAN

<i>INDEX</i>	0	1	2	3	4	5	6	7	8	9	10	11
<i>TEXT</i>	S	I	S	T	E	M		P	A	K	A	R
<i>PATTERN</i>												
						P	A	K	A	R		

5 Pergeseran

Setelah mengalami pegeseran, karakter “R” setelah mengalami pegeseran ternyata tidak sama dengan karakter “K”, maka berikutnya adalah melakukan pergeseran merujuk pada nilai OH. Pada tabel OH untuk karakter “K” memiliki nilai 2, untuk itu selanjutnya akan dilakukan pergeseran sebanyak 2 langkah. *String* pada teks paling kanan yaitu karakter “K” telah sama atau cocok dengan string pada pattern yaitu karakter “K”, begitu pula dengan string pada teks yang lainnya.

TABEL IV  
NILAI OCCURENCE HEURISTIC (OH) DAN MATCH HEURISTIC (MH)

INDEX	0	1	2	3	4	5	6	7	8	9	10	11
<b>TEXT</b>	S	I	S	T	E	M	P	A	K	A	R	
<b>PATTERN</b>												

2 Pegeseran

Pada aplikasi, setelah dilakukan pencarian maka selanjutnya akan muncul tampilah hasil judul yang dicari beserta persentase kemiripan dengan judul yang ada pada *database* judul skripsi Universitas Teknokrat Indonesia yang sebelumnya yang pernah diajukan. Tampilan Berikut digunakan untuk menampilkan daftar hasil pencarian dan persentase tingkat kemiripan.

#	Judul	Jurusan	Persentase
1	APRIANSYAH (2017) PENGEMBANGAN MEDIA PEMBELAJARAN SISTEM PENDUKUNG KEPUTUSAN BERBASIS MULTIMEDIA	Informatika	17.39%
2	NURKHOFIYAH (2017) RANCANG BANGUN MEDIA PEMBELAJARAN PEMROGRAMAN WEB MENGGUNAKAN CONSTRUCT 2 (STUDI KASUS TEKNIK KOMPUTER DAN JARINGAN (TKJ) DI SMK N4 BANDAR LAMPUNG)	Informatika	17.50%
3	ANTONI (2018) SISTEM INFORMASI NILAI SISWA MENGGUNAKAN FRAMEWORK CODEIGNITER (STUDI KASUS : SMK N 1 LIWA LAMPUNG BARAT)	Informatika	15.13%
4	Edy Warsito (2018) GAME PETUALANGAN UNTUK ANAK TUNAGRAHITA MENGGUNAKAN CONSTRUCT 2	Informatika	23.38%
5	NURUL FEBRIYANI UTAMI (2019) PENGEMBANGAN APLIKASI GAME PENGENALAN HURUF DAN ANGKA MENGGUNAKAN CONSTRUCT 2	Informatika	17.58%
6	Muhammad Fadil Akbar (2018) PENGEMBANGAN GAME EDUKASI PENGENALAN HEWAN LANGKA BERBASIS ANDROID MENGGUNAKAN CONSTRUCT 2	Informatika	15.38%

Gambar 6. Hasil Pencarian Judul

**D. Testing (Pengujian)**

Uji aplikasi dilakukan menggunakan pendekatan *black box testing*. Pengujian ini berfokus pada fungsionalitas dari aplikasi. Pada pengujian ini akan dicoba semua fungsi fitur yang ada dalam aplikasi deteksi tingkat kemiripan judul skripsi. Berikut ini hasil dari *black box testing* yang telah dilaksanakan (Tabel V).

TABEL V  
HASIL PENGUJIAN BLACK BOX TESTING

No.	Kasus Uji	Hasil yang diharapkan	Keterangan
1	Admin login dengan <i>username</i> dan <i>password</i> yang benar	Berhasil login, selanjutnya muncul tampilan beranda / halaman utama.	Valid



No.	Kasus Uji	Hasil yang diharapkan	Keterangan
2	Login admin dengan <i>username</i> dan <i>password</i> yang salah	Gagal login, selanjutnya muncul pesan <i>username</i> atau <i>password</i> salah.	Valid
3	Admin dapat menambahkan data judul pada menu data judul dan menekan <i>button</i> tambah judul	Muncul <i>form</i> tambah judul, kemudian admin diminta memasukkan data judul dan abstrak	Valid
4	Admin dapat mengubah data judul pada menu data judul dan menekan logo edit judul	Muncul <i>form</i> ubah judul, kemudian admin dapat mengubah data judul dan abstrak yang ingin diubah, setelah itu tekan tombol simpan dan akan muncul pesan data berhasil dirubah	Valid
5	Admin dapat menghapus data judul pada menu data judul dan menekan logo hapus judul	Muncul pesan “apakah anda ingin menghapus judul ini” apabila tekan “ya” data akan terhapus, apabila tekan “tidak” akan kembali ke menu data judul	Valid
6	Admin dapat melakukan pengecekan kemiripan judul dengan melakukan pencarian judul yang dicek kemiripannya	Admin mengetikan judul yang akan dilakukan pengecekan kemudian menekan <i>button</i> cari, selanjutnya akan tampil hasil persentase kemiripan	Valid
7	Admin dapat menambahkan <i>user</i> pada menu manajemen user	Admin diminta mengisi data <i>user</i> setelah tekan tombol simpan <i>user</i> akan bertambah	Valid
8	<i>User</i> login dengan <i>username</i> dan <i>password</i> yang benar	Berhasil login, kemudian muncul tampilan beranda / halaman utama.	Valid
9	<i>User</i> login dengan <i>username</i> dan <i>password</i> yang salah	Gagal login, kemudian muncul pesan <i>username</i> atau <i>password</i> salah.	Valid
10	<i>User</i> dapat melakukan pengecekan persentase kemiripan judul yang diajukan	<i>User</i> dapat mengetikan judul yang diajukan untuk dicek tingkat kemiripannya, setelah itu tekan <i>button</i> cari maka akan muncul persentase tingkat kemiripan	Valid
11	<i>User</i> dapat melihat hasil pengecekan kemiripan judul	<i>User</i> disuguhkan tampilan data tingkat kemiripan judul	Valid

Aplikasi yang dibangun dapat melakukan pencocokan dengan menghasilkan kemiripan berdasarkan persentase. Berdasarkan *black box testing* yang telah dilakukan terlihat bahwa semua fitur pada aplikasi dapat berfungsi dan berjalan dengan baik. Ini menunjukkan bahwa aplikasi deteksi kemiripan judul skripsi dapat digunakan. Hal ini menunjukkan bahwa pendekatan yang digunakan pencocokan *string* dengan Boyer Moore juga mampu melakukan pencocokan dengan baik pada kasus ini dan pengembangan system yang digunakan yaitu *extreme programming* (XP) dapat diterapkan dalam pengembangan sistem untuk aplikasi kemiripan usulan judul skripsi.

#### IV. KESIMPULAN

Penelitian ini menghasilkan aplikasi yang mampu menentukan tingkat kemiripan usulan judul skripsi mahasiswa. Pendekatan pencocokan *string* Boyer Moore yang digunakan mampu melakukan pencocokan dengan menemukan *pattern* yang sama persis dari satu teks.

Penggunaan pendekatan Boyer Moore dapat melakukan pencocokan *pattern* dan *text* dimulai dari karakter paling kanan terlebih dahulu kemudian ke kiri terhadap teks pada judul skripsi. Sedangkan metode XP telah menghasilkan *software* yang sesuai dengan kebutuhan. Hal ini didukung dari hasil *black box testing* yang menunjukkan bahwa semua fitur dari aplikasi dapat berfungsi dengan baik, artinya aplikasi dapat digunakan.

#### REFERENSI

- [1] M. Bolung and H. R. K. Tampangela, "Analisa Penggunaan Metodologi Pengembangan Perangkat Lunak," *J. ELTIKOM*, vol. 1, no. 1, pp. 1–10, 2017.
- [2] A. Fatoni and D. Dwi, "Rancang Bangun Sistem Extreme Programming Sebagai Metodologi Pengembangan Sistem," *Prosisko*, vol. 3, no. 1, pp. 1–4, 2016.
- [3] K. C. Dewi, P. I. Ciptayani, and I. W. R. Wijaya, "Agile Project Management Pada Pengembangan E-Musrenbang Kelurahan Benoa Bali," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 6, pp. 723–730, 2018.
- [4] I. Mahendra and D. T. E. Yanto, "Agile Development Methods Dalam Pengembangan Sistem Informasi Pengajuan Kredit Berbasis Web (Studi Kasus : Bank Bri Unit Kolonel Sugiono)," *J. Teknol. Dan Open Source*, vol. 1, no. 2, pp. 13–24, 2018.
- [5] I. Carolina and A. Supriyatna, "Penerapan Metode Extreme Programming Dalam Perancangan Aplikasi Perhitungan Kuota SKS Mengajar Dosen," *J. Khatulistiwa Inform.*, vol. 3, no. 1, pp. 106–113, 2019.
- [6] L. Rusdiana, "Extreme Programming untuk rancang bangun aplikasi pengelolaan surat keterangan kependudukan," *Regist. J. Ilm. Teknol. Sist. Inf.*, vol. 4, no. 1, p. 49, 2018.
- [7] E. B. Pratama, "Pendekatan Metodologi Extreme Programming pada Aplikasi e-Commerce Berbasis M-Commerce Studi Kasus: Toko Buku An'Nur di Pontianak," *J. Khatulistiwa Inform.*, vol. V, no. 2, pp. 92–102, 2017.
- [8] W. Astuti, "Analisis String Matching Pada Judul Skripsi Dengan Algoritma Knuth-Morris Pratt (KMP)," *Ilk. J. Ilm.*, vol. 9, no. 2, pp. 167–172, 2017.
- [9] S. Hadi, "Pemeriksaan Keabsahan Data Penelitian Kualitatif Pada Skripsi," *J. Ilmu Pendidik. Univ. Negeri Malang*, vol. 22, no. 1, pp. 74–79, 2016.
- [10] R. I. Borman and A. Pratama, "Penerapan String Matching Dengan Algoritma Boyer Moore Pada Aplikasi Font Italic Untuk Deteksi Kata Asing," *J. Teknoinfo*, vol. 10, no. 2, pp. 1–5, 2016.
- [11] R. I. Darmawan, A. H. Setianingrum, and A. Arini, "Implementasi Algoritma Boyer Moore Pada Aplikasi Kamus Istilah Kebidanan Berbasis Web," *QUERY J. Sist. Inf.*, vol. 02, no. 01, pp. 53–62, 2018.
- [12] R. Wulan, "Pengembangan Konfigurasi Model Analisis Arsitektur Agile Pada Perusahaan Bisnis IT Online (Studi kasus Lazada dan Bhineka.com)," *Fakt. Exacta*, vol. 9, no. 2, pp. 166–177, 2016.
- [13] I. Mulyawati, R. T. Subagio, and D. Martha, "Implementasi Metode String Matching Untuk Aplikasi Pengarsipan Dokumen ( Studi Kasus : SMPN 3 Sumber Kab. Cirebon)," *J. Digit*, vol. 7, no. 1, pp. 50–61, 2017.
- [14] D. Kartini, R. Y. Tanjung, and I. Lasmana, "Penerapan Algoritma Boyer Moore Horspool Pada Pencarian Katalog Buku," in *Seminar Nasional Sains & Teknologi Informasi (SENSASI)*, 2019, pp. 341–344.
- [15] I. J. Sinaga, R. C. G. I. Kembaren, D. M. Br Bangun, and N. Marbun, "Penerapan Algoritma Boyer Moore Pada Aplikasi Filosof Berbasis Website," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 3, no. 1, pp. 436–438, 2019.
- [16] T. Gumelar, R. Astuti, and A. T. Sunarni, "Sistem Penjualan Online Dengan Metode Extreme Programming," *J. Telemat.*, vol. 9, no. 2, pp. 87–90, 2017.
- [17] R. I. Borman, A. T. Priandika, and A. R. Edison, "Implementasi Metode Pengembangan Sistem Extreme Programming ( XP ) pada Aplikasi Investasi Peternakan Implementation of Extreme Programming ( XP ) System Development Method in Livestock Investment

- Application,” *JUSTIN (Jurnal Sist. dan Teknol. Informasi)*, vol. 8, no. 3, pp. 272–277, 2020.
- [18] I. G. N. Suryantara, “Merancang Aplikasi dengan Metodologi Extreme Programming.” PT. Elex Media Komputindo, Jakarta, 2017.
- [19] I. Carolina and A. Rusman, “Penerapan Extreme Programming Pada Sistem Informasi Penjualan Pakaian Berbasis Web (Studi Kasus Toko ST Jaya),” *INOVTEK Polbeng - Seri Inform.*, vol. 4, no. 2, p. 157, 2019.
- [20] R. D. Gunawan, T. Oktavia, and R. I. Borman, “Perancangan Sistem Informasi Beasiswa Program Indonesia Pintar (PIP) Berbasis Online (Tudi Kasus : SMA N 1 Kota Bumi),” *J. Mikrotik*, vol. 8, no. 1, pp. 43–54, 2018.
- [21] Y. Wulandari and H. Sulistiani, “Rancang Bangun Aplikasi Presensi SMS Gateway Berbasis Web Dengan Framework Codeigniter Pada SMKN 1 Trimurjo,” *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 1, no. 1, pp. 43–50, 2020.
- [22] M. Melinda, R. I. Borman, and E. R. Susanto, “Rancang Bangun Sistem Informasi Publik Berbasis Web (Studi Kasus : Desa Durian Kecamatan Padang Cermin Kabupaten Pesawaran),” *J. Tekno Kompak*, vol. 11, no. 1, p. 1, 2018.
- [23] A. E. Kumala, R. I. Borman, and P. Prasetyawan, “Sistem Informasi Monitoring Perkembangan Sapi Di Lokasi Uji Performance (Studi Kasus : Dinas Peternakan Dan Kesehatan Hewan Provinsi Lampung),” *J. Tekno Kompak*, vol. 12, no. 1, p. 5, 2018.
- [24] W. N. Cholifah, Y. Yulianingsih, and S. M. Sagita, “Pengujian Black Box Testing Pada Aplikasi Action & Strategy Berbasis Android Dengan Teknologi Phonegap,” *J. String*, vol. 3, no. 2, pp. 206–210, 2018.
- [25] T. Arnova and I. Ahmad, “Sistem Informasi E-Document Korespodensi Pada Korem 043/Gatam,” *J. Ilm. Rekayasa dan Manaj. Sist. Inf.*, vol. 1, no. 2, pp. 15–18, 2015.
- [26] T. Monica and R. I. Borman, “Implementasi Konsep Media Sosial Dalam Sistem Informasi Kegiatan Kesiswaan (Studi Kasus : SMK XYZ),” *J. Tekno Kompak*, vol. 11, no. 2, p. 33, 2017.

#### UCAPAN TERIMA KASIH

Peneliti mengucapkan terima kasih kepada “Universitas Teknokrat Indonesia dan LPPM Universitas Teknokrat Indonesia” atas dukungannya melalui skim penelitian internal.