

Rancang Bangun Website untuk Memeriksa Plagiat E-Journal Fakultas Kedokteran Universitas Palangka Raya

Widiatry¹, Nova Noor Kamala Sari²

^{1,2}Teknik Informatika, Universitas Palangka Raya

Kampus UPR Tunjung Nyaho Jalan Yos Sudarso, Palangka Raya 73111

widiatry@it.upr.ac.id¹, novanoorks@it.upr.ac.id²

Abstrak – Jurnal ilmiah berisi kumpulan artikel yang dipublikasikan secara periodik dan ditulis oleh ilmuwan atau peneliti untuk melaporkan hasil-hasil penelitian terbarunya. Masalah yang dihadapi oleh Fakultas Kedokteran Universitas Palangka Raya, jurnal dikumpulkan dan diperiksa secara manual serta penerbitannya masih berupa media cetak. Oleh sebab itu, penyelesaian masalah tersebut adalah dengan membangun sebuah website untuk memeriksa plagiat pada *E-Journal* Fakultas Kedokteran. *Website* ini menerapkan Algoritma Rabin Karp untuk mendeteksi plagiarisme pada abstrak. Tahapan metode penelitian yang digunakan dalam penelitian ini adalah : (1) pengumpulan dan analisis data yang dilakukan dengan cara studi pustaka dan observasi, (2) metode yang diterapkan adalah metode *waterfall* yang sudah dimodifikasi melalui analisis sistem, desain sistem, implementasi dan pengujian menggunakan *blackbox testing*. Dengan adanya *Website* untuk memeriksa plagiat pada E-Journal diharapkan dapat meningkatkan efektifitas Fakultas Kedokteran dalam melakukan pengelolaan jurnal yang terbit sehingga jurnal tidak perlu dicetak. Jurnal dapat diunduh dan diunggah melalui *website* ini dan fasilitas yang ada dapat pula membantu dalam mendeteksi plagiat sehingga kedepannya bisa menghasilkan jurnal dengan kualitas yang baik. Diharapkan untuk pengembangan dimasa akan datang, *website* ini dapat dikembangkan dengan menggunakan algoritma *stemming* untuk mendapatkan hasil perhitungan kemiripan yang lebih akurat dan memperhatikan kesalahan ejaan yang ada.

Kata Kunci – Algoritma Rabin-Karp, E-Journal, Plagiat

PENDAHULUAN

Jurnal sebagai salah satu hasil pengetahuan yang terwujud dan terangkum dalam tulisan-tulisan ilmiah memungkinkan siapa saja yang mempunyai karya tulis dapat memasukkan karyanya. Jurnal yang kita kenal biasanya berupa buletin atau majalah ilmiah yang diterbitkan oleh institusi tertentu dan bagi

siapapun yang ingin mempublikasikan hasil karya ilmiahnya harus terlebih dahulu mengirim ke institusi tersebut untuk dipublikasikan. Terdapat beberapa kelemahan apabila jurnal ini masih dikembangkan dan dijalankan dengan cara konvensional yang berarti karya ilmiah dipublikasikan dalam bentuk buku saja, keterbatasan jurnal konvensional ini antara lain yakni terbatasnya karya ilmiah yang diterima dan membutuhkan proses seleksi yang lama, terbatasnya pendanaan dalam penerbitan jurnal tersebut karena publikasi harus dicetak dalam bentuk buletin atau majalah yang sangat jelas apabila semakin banyak dicetak maka diperlukan biaya yang tidak sedikit dan lebih mahal.

Pada Fakultas Kedokteran Universitas Palangka Raya, jurnal dari penulis dikumpulkan dan diperiksa secara manual, penerbitannya masih berupa media cetak dan tidak terdapat sebuah aplikasi untuk mengecek atau mengetahui apakah jurnal yang dikirimkan plagiat atau tidak.

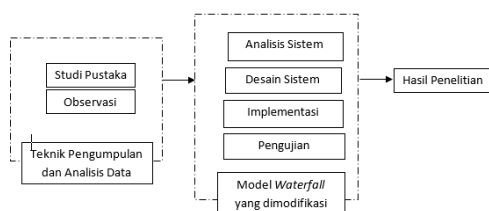
Oleh karena itu, proses pencegahan dan pendeteksian merupakan cara yang dapat dilakukan untuk mengurangi plagiaris yaitu dengan menggunakan algoritma Rabin-Karp pada *website* E-Journal. Algoritma Rabin-Karp adalah algoritma *multiple pattern search* yang sangat efisien untuk mencari *string* dengan pola banyak. *E-Journal* tidak berarti menggantikan model jurnal konvensional, tetapi memperkuat jurnal tersebut melalui pengelolaan penulis, karya tulis dan tanggapan atas karya tersebut, bahkan sampai mendiskusikan secara tak terbatas.

Peneliti sebelumnya juga telah melakukan penelitian diantaranya Pengembangan Media Informasi Ruang Kuliah pada Fakultas Teknik Universitas Palangka Raya berbasis android dan LBS [1] dan Penerapan Algoritma Levenshtein Distance Untuk Pencarian Pada Sistem Informasi Perpustakaan Pada Fakultas Kedokteran Universitas Palangka Raya [2] yang dapat menambah pemahaman dalam menganalisis dan mendesain suatu sistem di lingkungan Universitas Palangka Raya.

HASIL DAN PEMBAHASAN

METODE PENELITIAN

Model yang digunakan website untuk memeriksa plagiat pada E-Journal Fakultas Kedokteran Universitas Palangka Raya adalah model *Waterfall*. Model *Waterfall* adalah model klasik yang bersifat sistematis dan berurutan dalam Rekayasa Perangkat Lunak. Model ini merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE) dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 [3]. Pada perkembangannya banyak bermunculan modifikasi-modifikasi dari model tersebut. Sehingga model yang digunakan pada penelitian ini merupakan model *Waterfall* yang modifikasi.



Gambar 1. Tahapan Penelitian

Berikut ini merupakan beberapa tahapan penelitian, yaitu:

- 1) **Studi Pustaka dan Observasi**
 Metode ini melakukan studi pustaka terhadap buku, jurnal ilmiah nasional dan internasional sebagai pendukung dalam penelitian. Observasi dilakukan untuk mengambil data-data tentang jurnal pada Fakultas Kedokteran Universitas Palangka Raya.
- 2) **Analisis Sistem**
 Proses pencarian kebutuhan difokuskan pada *software* untuk mengetahui sifat dari website yang akan dibuat, seperti manajemen hak akses pengguna beserta fasilitas-fasilitas yang ada, pemodelan proses bisnis menggunakan *flowchart* dan *Unified Modeling Language* (UML)
- 3) **Desain Sistem**
 Proses ini digunakan untuk membuat “*blueprint*” *software* seperti, perancangan basis data dan perancangan antarmuka (*interface*).
- 4) **Implementasi**
 Proses ini akan dilakukan pembuatan perangkat lunak sesuai dengan perancangan basis data dan antar muka yang telah dilakukan pada tahap sebelumnya.
- 5) **Pengujian Sistem**
 Proses pengujian akan dilakukan dengan *Blackbox Testing* yang disertai dengan melakukan simulasi data.

A. Skenario Pengujian Algoritma *Rabin Karp*

Algoritma rabin karp merupakan metode yang digunakan untuk mendeteksi plagiarisme. Pada tahap ini dilakukan analisa terhadap algoritma *rabin karp*. Dengan menentukan nilai *k-gram* dan basis primanya. Hasil dari *hashing* asli dan *hashing* uji kemudian dibandingkan untuk dicari *hashing* yang sama. Jika *hashing* yang sama ditemukan, maka dari hasil *hashing* yang sama tersebut dihitung tingkat persentase kesamaannya (*similarity*).

Berikut ini adalah contoh perhitungan kesamaan dokumen yang akan di proses dengan menggunakan metode *Rabin Karp*.

Dokumen 1

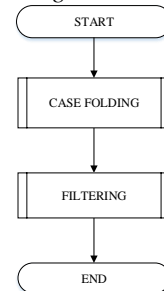
Glikasi merupakan reaksi ikatan antara glukosa dengan gugus amin pada protein atau lemak. Ekstrak etanol rimpang kunyit putih (*Kaempferia rotunda* Linn.) diyakini mampu menghambat reaksi glikasi karena memiliki kandungan flavonoid, kurkumin dan minyak atsiri yang berpotensi sebagai anti-glikasi secara *in vitro*.

Dokumen 2

Diabetes Mellitus mempunyai gejala berupa peningkatan kadar glukosa darah (hiperglikemia). Tanaman serai wangi merupakan salah satu tanaman yang mampu menurunkan kadarglukosa darah.

Adapun langkah-langkah algoritma *Rabin Karp* yang akan dilakukan untuk mendeteksi plagiarisme adalah sebagai berikut [4]:

1. Tahapan *Preprocessing* yang terdiri dari *case folding* dan *filtering*



Gambar 2. Flowchart *Preprocessing*

a. Proses *Case Folding*

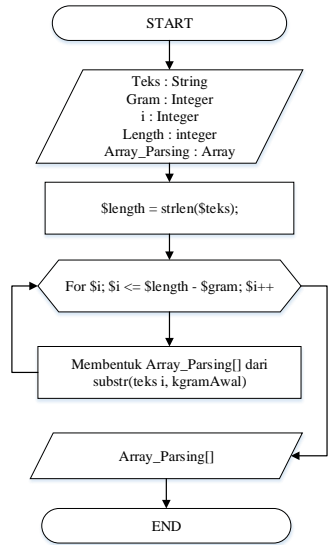
Proses *case folding* merupakan proses mengubah semua huruf menjadi huruf kecil.

b. Proses *Filtering*

Proses *filtering* yaitu proses membuang kata – kata yang tidak penting, membuang tanda baca dan membuang *spasi*.

2. Tahapan *Parsing K-Gram*

K-gram merupakan metode sebuah metode yang diaplikasikan untuk pembangkitkan kata atau karakter dengan memecah kata menjadi potongan – potongan yang mengandung karakter sebanyak *k*.



Gambar 3. Flowchart Parsing K-Gram

Berikut adalah proses *parsing k-gram k = 7*
 Hasil *k-gram* dokumen 1 yang telah melalui proses *parsing k-gram*

```
{glikasi} {likasim} {ikasime} {kasimer} {asimeru} {simerup}
{imerupa} {merupak} {erupaka} {rupakan} {upakanr} {pakanre}
{akanrea} {kanreak} {anreaks} {nreaksi} {reaksii} {eaksiik}
{aksiika} {ksiikat} {siikata} {iikatan} {ikatana} {katanan}
{atanant} {tananta} {anantar} {nantara} {antarag} {ntaragl}
{taraglu} {aragluk} {ragluko} {aglukos} {glukosa} {lukosag}
{ukosagu} {kosagug} {osagugu} {sagugus} {agugusa} {gugusam}
{ugusami} {gusamin} {usaminp} {saminpa} {aminpad} {minpada}
{inpadap} {npadapr} {padapro} {adaprot} {daprote} {aprotei}
{protein} {roteinl} {oteinle} {teinlem} {einlema} {inlemak}
{nlemake} {lemakek} {emakeks} {makekst} {akekstr} {kekstra}
{ekstrak} {kstrake} {straket} {traketa} {raketan} {aketano}
{ketanol} {etanolr} {tanolri} {anolrim} {nolrimp} {olrimpa}
{lrimpan} {rim pang} {impangk} {mpangku} {pangkun} {angkuny}
{ngkunyi} {gkunyit} {kunyitp} {unyitpu} {nyitput} {yitputi}
{itputih} {tputihk} {putihka} {utihkae} {tihkaem} {ihkaemp}
{hkaempf} {kaempfe} {aempfer} {empferi} {mpferia} {pferiar}
{feriaro} {eriarot} {riarotu} {iarotun} {arotund} {rotunda}
{otundal} {tundali} {undalin} {ndalinn} {dalinn} {alinndi}
{linndiy} {innndiyak} {ndiyak} {ndiyaki} {diyakin} {iyakini}
{yakininim} {akinima} {kinimam} {inimamp} {nimampu}
{imampum} {mampume} {ampumen} {mpumeng} {pumengh}
{umengha} {mengham} {enghamb} {nghamba} {ghambat}
{hambatr} {ambatre} {mbatre} {btreak} {atreaks} {treaksi}
{reaksig} {eaksigl} {aksigli} {ksiglik} {siglika} {iglikas}
{glikasi} {likasim} {ikasime} {kasimem} {asimemi}
{simemil} {imemili} {memilik} {emiliki} {milikik}
{ilikika} {likikan} {ikikand} {kikandu} {ikandun}
{kandung} {andunga} {ndungan} {dunganf} {unganfl}
{nganfla} {ganflav} {anflavo} {nflavon} {flavono}
{lavonoi} {avonoid} {vonoidk} {onoidku} {noidkur}
{oidkurk} {idkurku} {dkurkum} {kurkumi} {urkumin}
{rkuminm} {kuminmi} {uminmin} {minminy} {inminya}
{nminyak} {minyaka} {inyakat} {nyakats} {yakatsi}
{akatsir} {katsiri} {atsirib} {siriber} {iriberp}
{riberpo} {iberpot} {berpote} {erpoten} {rpotens}
{potensi} {otensia} {tensian} {ensiant} {nsianti}
{siantig} {iantigl} {antigli} {ntiglik} {tiglika}
{iglikas} {glikasi} {likasis} {ikasise} {kasisec}
{asiseca} {sisecar} {isecara} {secarai} {ecarain}
{carainv} {arainvi} {rainvit} {ainvitr} {invitro}
```

Hasil *k-gram* dokumen 2 yang telah melalui proses *parsing k-gram*

```
{diabete} {iabetes} {abetesm} {betesme} {etesmel} {tesmeli}
{esmilit} {smelitu} {melitus} {elitusm} {litusme} {itusmem}
{tusmemp} {usmempu} {smempun} {mempuny} {empunya}
{mpunyai} {punyaig} {unyaige} {nyaigej} {yaigeja} {aigejal}
{igejala} {gejalab} {ejalabe} {jalaber} {alaberu} {laberup}
{aberupa} {berupap} {erupape} {rupapen} {upapeni}
{papenin} {apening} {peningk} {eningka} {ningkat} {ingkata}
{ngkatan} {gkatank} {katanka} {atankad} {tankada}
{ankadar} {nkadarg} {kadargl} {adarglu} {dargluk} {argluko}
{rglukos} {glukosa} {lukosad} {ukosada} {kosadar} {osadara}
{sadarah} {adarahh} {darahhi} {arahhip} {rahhipe} {ahhiper}
{hhiperg} {hipergl} {ipergli} {perglik} {erglike} {rglikem}
{glikemi} {likemia} {ikemiat} {kemiat} {emiatan} {miatana}
{iatanam} {atanama} {tanaman} {anamans} {namanse}
{amanser} {mansera} {anserai} {nseraiw} {seraiwa}
{eraiwan} {raiwang} {aiwangi} {iwangim} {wangime}
{angimer} {ngimeru} {gimerup} {imerupa} {merupak}
{erupaka} {rupakan} {upakans} {pakansa} {akansal}
{kansala} {ansalah} {nsalahs} {salahsa} {alahsat} {lahsatu}
{ahsatut} {hsatuta} {satutan} {atutana} {tutanam} {utanama}
{tanaman} {anamann} {nannam} {nannam} {nannam}
{anmerup} {nmerupa} {merupak} {erupaka} {rupakan}
{upakans} {pakansa} {akansal} {kansala} {ansalah} {nsalahs}
{salahsa} {alahsat} {lahsatu} {ahsatut} {hsatuta} {satutan}
{atutana} {tutanam} {utanama} {tanaman} {anamann}
{nannam} {amanmam} {manmamp} {anmampu}
{nmampum} {mampume} {ampumen} {mpumenu}
{pumenor} {umenuru} {menurun} {enurunk} {nurunka}
{urunkan} {runkank} {unkanka} {nkankad} {kankada}
{ankadar} {nkadarg} {kadargl} {adarglu} {dargluk} {argluko}
{rglukos} {glukosa} {lukosad} {ukosada} {kosadar} {osadara}
{sadarah}
```

3. Tahapan Hashing

Proses *hashing* merupakan proses untuk mengkonvert setiap string menjadi bilangan. Dengan rumus

$$H_{c1...ck} = C_1 \cdot b^{(k-1)} + C_2 \cdot b^{(k-2)} + \dots \quad (1)$$

Contoh proses *hashing* untuk menghitung nilai *hash* dari kata {glikasi}, dengan nilai $k = 7$ dan $b = 7$.

$$H = (103.7^6) + (108.7^5) + (105.7^4) + (107.7^3) + (97.7^2) + (115.7^1) + (105.7^0)$$

$$H = 12117847 + 1815156 + 252105 + 36701 + 4753 + 805 + 105 = 14227472$$

Jadi, nilai *hash* pada dokumen 1 *k-gram* dari kata glikasi bernilai 14227472, proses perhitungan nilai *hash* diulang kembali hingga *k-gram* keseluruhan di hitung. Berikut ini adalah nilai *hash* dari dokumen 1 dan dokumen 2.

Nilai Hash Pada Dokumen 1

{14227472}	{14767484}	{14429845}	{14537014}	{13640114}	{15597239}
{14473325}	{14841367}	{14123479}	{15686620}	{15922552}	{15103434}
{13487319}	{14527669}	{13574697}	{15139313}	{15385566}	{13815167}
{13528423}	{14815406}	{15588838}	{14414531}	{14429799}	{14536688}
{13637831}	{15581243}	{13537827}	{14881215}	{13578878}	{15168583}
{15590468}	{13602395}	{15333205}	{13448648}	{14256962}	{14973908}
{15874829}	{14769375}	{15266641}	{15453329}	{13465955}	{14378123}
{15822037}	{14399838}	{15974049}	{15463909}	{13540018}	{14896552}
{14509789}	{15096622}	{15086735}	{13370445}	{13709545}	{13574547}
{15404779}	{15596745}	{15293414}	{15640734}	{13954247}	{14501993}
{15042037}	{14704636}	{15048764}	{14751734}	{13496065}	{14588881}
{14003173}	{14844469}	{15792298}	{15838738}	{15340288}	{13498225}
{14604012}	{14109097}	{15551641}	{13570708}	{15111397}	{15190146}
{14917859}	{15828172}	{14493509}	{14982665}	{15112578}	{13551351}
{14975891}	{14241623}	{14866522}	{15946824}	{15273353}	{16323846}
{14618323}	{15856353}	{15463580}	{16008345}	{15703993}	{14397075}
{14307612}	{14504913}	{13415404}	{14024262}	{14992088}	{15178543}
{14013096}	{14090402}	{15455088}	{14301824}	{13640853}	{15602397}
{15332985}	{15917727}	{15893211}	{14898056}	{13696762}	{13523139}
{14778423}	{14506414}	{15072990}	{14921305}	{13859515}	{14662410}
{16164964}	{13472528}	{14659432}	{14497035}	{15007347}	{14461808}
{14760742}	{13559117}	{15030251}	{15445674}	{15882999}	{14826571}
{14019908}	{14961610}	{14141656}	{14166777}	{13519068}	{14749902}
{13483234}	{13675539}	{15845207}	{15385564}	{13815154}	{13528340}
{14814816}	{15584708}	{14385626}	{14227472}	{14767484}	{14429845}
{14537009}	{13640067}	{15596906}	{14471002}	{14825106}	{14009660}
{14889884}	{14463090}	{14769781}	{15445923}	{14649563}	{14427910}
{14523738}	{13542162}	{14946533}	{14036389}	{15900489}	{14948989}
{14200609}	{13548319}	{14954952}	{14095045}	{14664034}	{13705694}
{16056294}	{15216101}	{15099548}	{15813107}	{14337335}	{13926570}
{14871875}	{15984155}	{15534663}	{14858844}	{15892917}	{14896009}
{14505973}	{15084565}	{14899688}	{14531745}	{15250315}	{16162180}
{13489431}	{14542731}	{13680114}	{15877228}	{15578852}	{14559692}
{15413143}	{14288215}	{13545591}	{14112033}	{15366403}	{15361724}
{15295349}	{15654280}	{14049128}	{15165878}	{15571519}	{14293296}
{13581162}	{15184570}	{15702357}	{14385626}	{14227472}	{14767490}
{14429887}	{14537293}	{13642047}	{15610772}	{14568056}	{15504482}
{13824039}	{13590550}	{13603198}	{15338831}	{13488029}	{14532643}

Jadi $\sum Hash = 228$

Nilai Hash Dokumen 2

{13802008}	{14259871}	{13347191}	{13546767}	{14120263}	{15664106}
{14117859}	{15647286}	{13999646}	{14819788}	{14795973}	{14629276}
{15933029}	{16000332}	{15647903}	{14827997}	{14029889}	{15031485}
{15454311}	{15943462}	{15249809}	{16159030}	{13464615}	{14368731}
{14109200}	{13939572}	{14399275}	{13499484}	{14612829}	{13347256}
{13547233}	{14123518}	{15686893}	{15924454}	{15116757}	{13580586}
{15180538}	{14027047}	{15011602}	{14491581}	{14969162}	{14194511}
{14536745}	{13638214}	{15583924}	{13556594}	{15321590}	{14498508}
{13370572}	{13710440}	{13618891}	{15448681}	{14256962}	{14973905}
{15874788}	{15874788}	{14769099}	{15264689}	{15439654}	{13370237}
{13708093}	{13602463}	{15333671}	{13451909}	{14279795}	{14310201}
{14523040}	{15189372}	{14088889}	{15444489}	{14227626}	{14768550}
{14437322}	{14589336}	{14006361}	{14866781}	{14301389}	{13637805}
{15581074}	{13536645}	{14872945}	{13520999}	{14763419}	{13577851}
{15161405}	{15540202}	{14074079}	{15340813}	{13501894}	{14629696}
{15935958}	{13550203}	{14967867}	{14185451}	{14473325}	{14841367}
{14123479}	{15686620}	{15922553}	{15103437}	{13487351}	{14527883}
{13576184}	{15149732}	{15458491}	{13502108}	{14631202}	{13475886}
{14447628}	{15485034}	{1368790}	{15931668}	{15990785}	{15581074}
{13536639}	{14872899}	{13520672}	{14761145}	{13561945}	{15050053}
{14760742}	{13559117}	{15030265}	{15445782}	{15883775}	{14832004}
{14057948}	{15227890}	{16005610}	{15684846}	{15910117}	{15016388}
{14525083}	{13556594}	{15012590}	{14498508}	{13370572}	{13710440}
{13618891}	{15448681}	{14256962}	{14973905}	{15874788}	{14769099}
{15264689}	{15439654}				

Jadi $\sum Hash = 146$

Hash Data 1 dan Data 2 yang Sama :

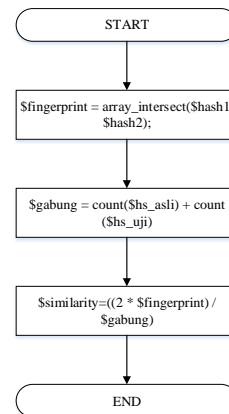
{14473325}	{14841367}	{14123479}	{1558620}
{14256962}	{1476074}	{13559117}	

Jadi $\sum Hash = 7$

4. Tahapan String Matching

Untuk menghitung kesamaan digunakan perhitungan *similarity*. *Similarity* didapat dari 2 kali hasil *hash* yang sama dibagi dengan jumlah *hash* kedua dokumen dikali dengan seratus persen.[5]

$$Similarity = \frac{2C}{A+B} \times 100\% \quad (2)$$



Gambar 4. Flowchart Similarity

Setelah dilakukan tahapan *Preprocessing*, *Parsing K-Gram*, dan *Hashing* di atas dapat dihitung kesamaan contoh dua dokumen dengan memasukkan ke rumus *similarity*

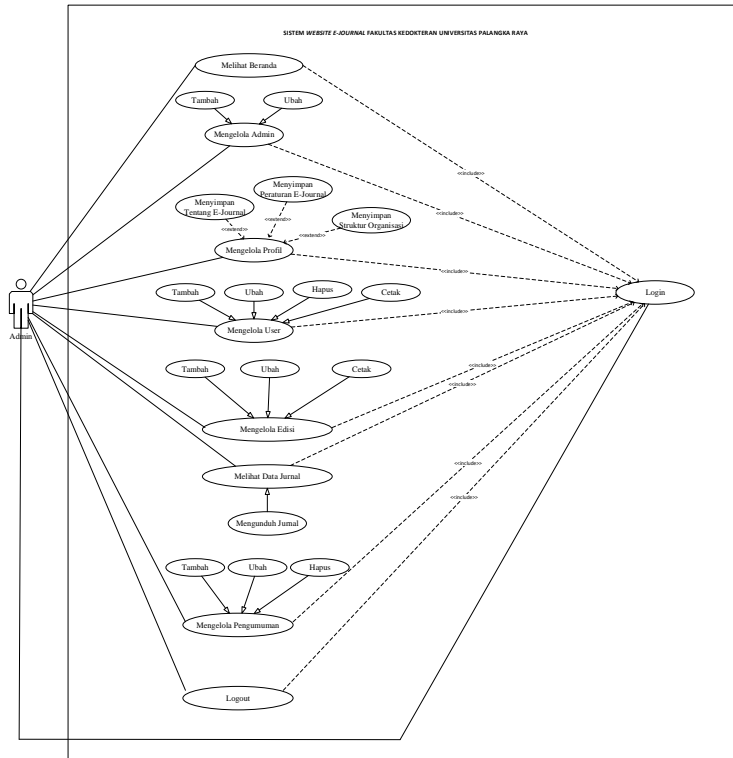
$$Similarity = \frac{2 \times 7}{228+146} \times 100\% = 3,7 \%$$

B. Hak Akses pada Website E-Journal

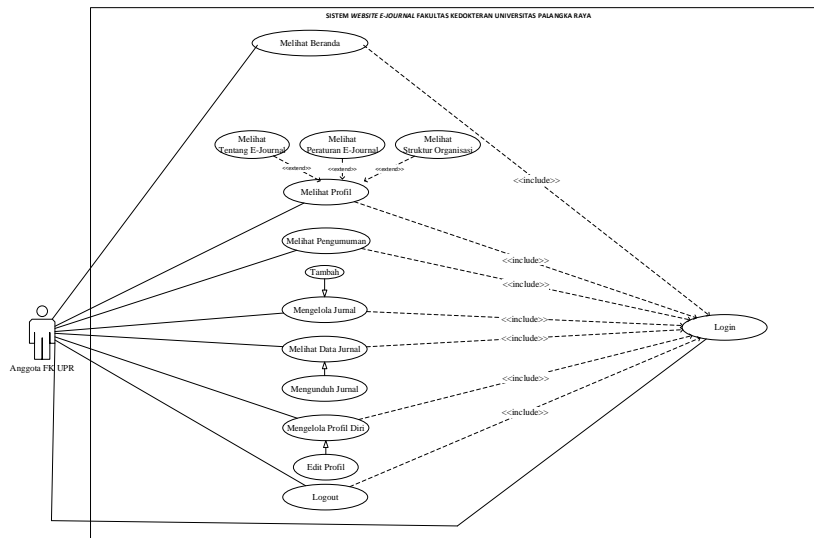
Manajemen hak akses pada *Website E-journal* Fakultas Kedokteran Universitas Palangka Raya akan dibagi menjadi 6 (enam) bagian, yaitu : Admin, Anggota FK UPR, Anggota Non FK UPR, Dewan Redaksi, Reviewer, Mitra Bestari

C. Use Case Diagram

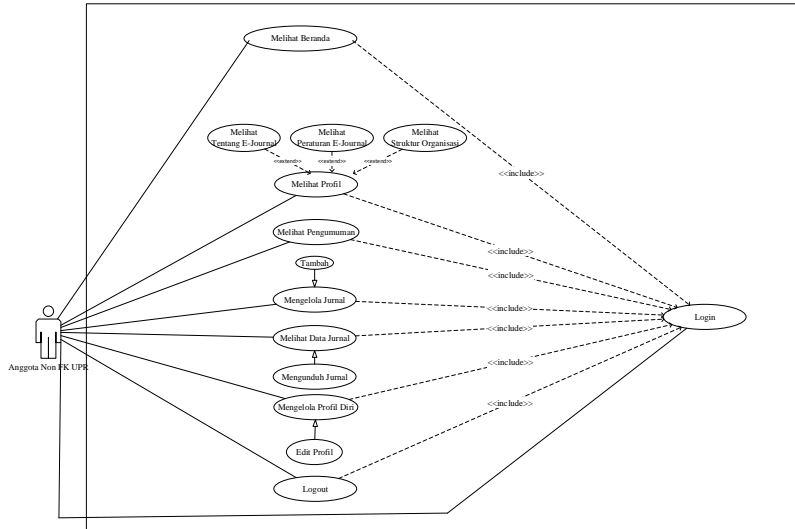
Use case atau diagram use case merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.[6]



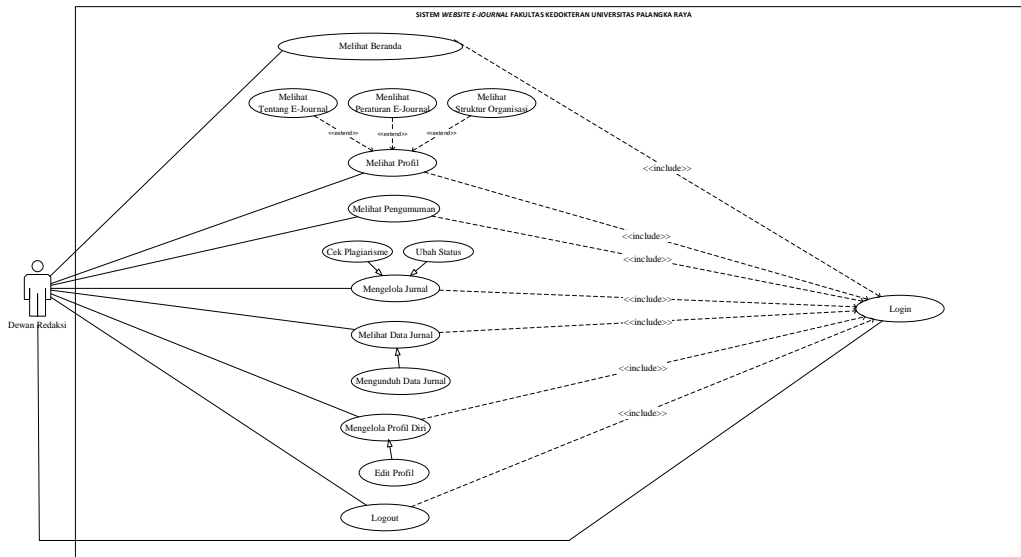
Gambar 5. Use Case Admin



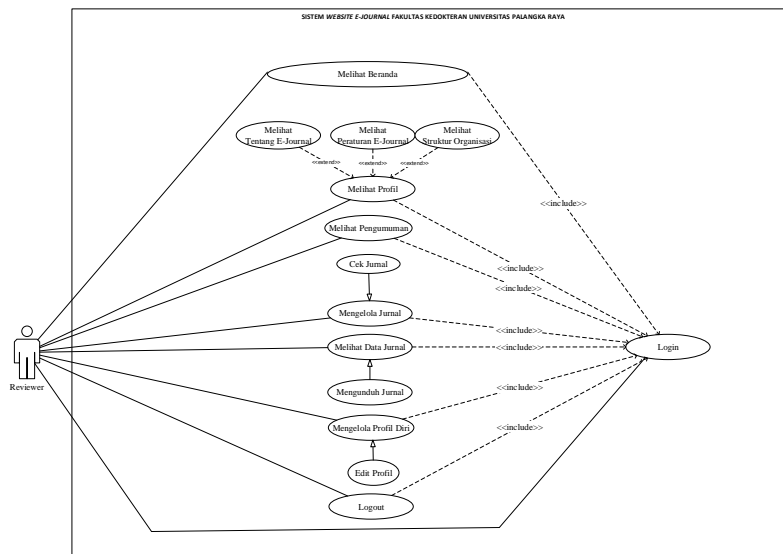
Gambar 6. Use Case Anggota FK UPR



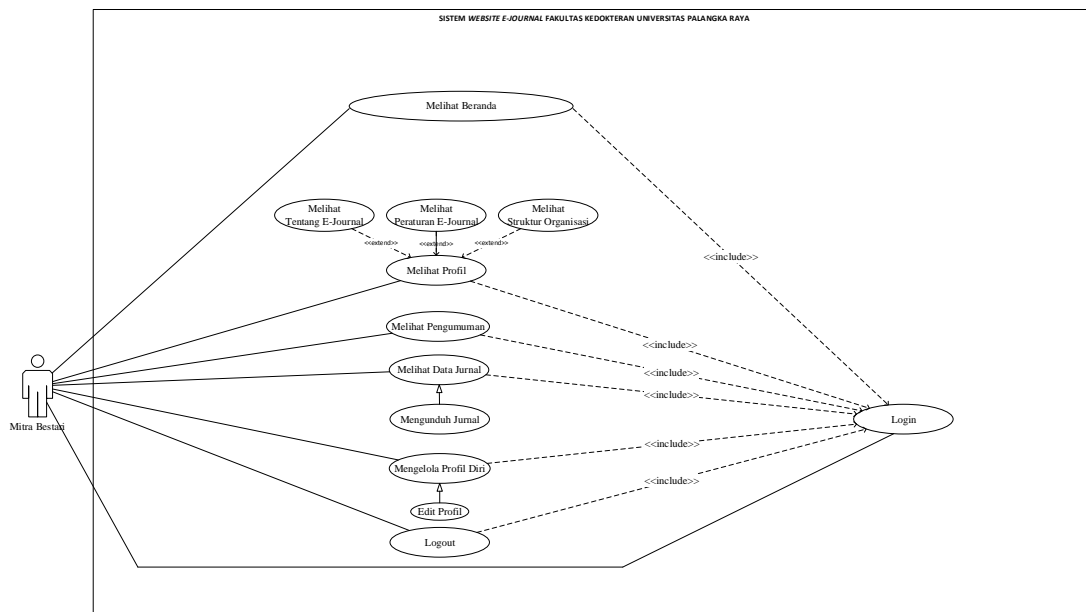
Gambar 7. Use Case Anggota Non FK UPR



Gambar 8. Use Case Dewan Redaksi



Gambar 9. Use Case Reviewer

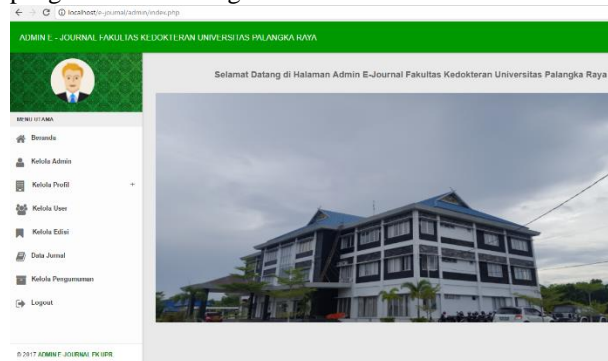


Gambar 10. Use Case Mitra Bestari

D. Implementasi

Fitur Menu Utama Admin

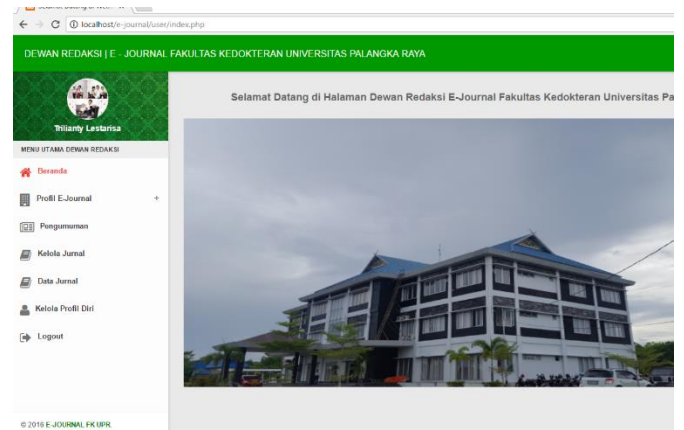
Pada gambar 11 merupakan beranda dan fitur-fitur yang terdapat pada halaman admin *Website E-Journal* Fakultas Kedokteran Universitas Palangka Raya. Fitur – fitur yang ada pada halaman admin *Website E- Journal* Fakultas Kedokteran Universitas Palangka Raya yaitu beranda, kelola admin, kelola profil, kelola user, kelola edisi, data jurnal, kelola pengumuman dan logout.



Gambar 11. Tampilan Menu Utama Admin

Fitur Menu Utama Dewan Redaksi

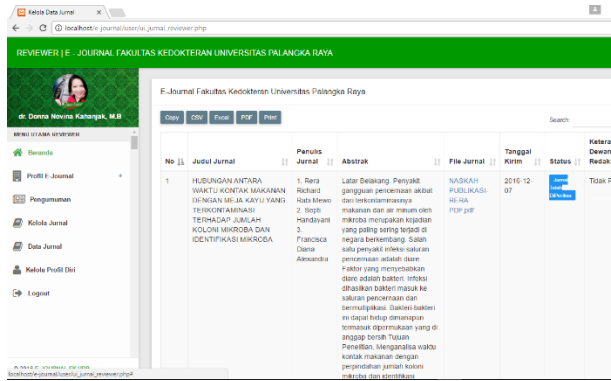
Pada gambar 12 merupakan beranda dan fitur-fitur yang terdapat pada halaman dewan redaksi *Website E- Journal* Fakultas Kedokteran Universitas Palangka Raya. Fitur – fitur yang ada pada halaman dewan redaksi *Website E- Journal* Fakultas Kedokteran Universitas Palangka Raya yaitu beranda, profil *e-journal*, pengumuman, kelola jurnal, data jurnal, kelola profil diri dan logout.



Gambar 12. Tampilan Menu Utama Dewan Redaksi

Fitur Kelola Jurnal

Fitur kelola jurnal merupakan fitur yang berfungsi untuk mengelola jurnal yang dikirimkan oleh anggota atau penulis jurnal yang sudah diseleksi oleh dewan redaksi. Pada fitur kelola jurnal ini *reviewer* berfungsi memberikan penilaian terhadap jurnal yaitu dengan cara menentukan jurnal tersebut diterima, ditolak dan di revisi. Pada gambar 13 adalah tampilan dari fitur kelola jurnal.



Gambar 13. Tampilan Fitur Kelola Jurnal Pada Reviewer

KESIMPULAN DAN SARAN

Kesimpulan yang dapat diambil adalah Pembuatan website *E-Journal* Fakultas Kedokteran Universitas Palangka Raya ini menerapkan Algoritma Rabin Karp untuk mendeteksi plagiat pada Jurnal. Model yang digunakan dalam pengembangan aplikasi ini adalah model *Waterfall* yang telah dimodifikasi. Langkah – langkah *waterfall* dalam metode ini yaitu tahap analisis, desain, implementasi dan pengujian.

Berdasarkan hasil uji dengan beberapa dokumen dapat disimpulkan bahwa algoritma Rabin Karp dapat digunakan untuk menghitung tingkat kemiripan sebuah abstrak. Perhitungan persentase kemiripan dihitung dengan menggunakan rumus *Dice Coefficient Similarity*.

Saran

Beberapa saran yang bertujuan untuk perbaikan atau peningkatan *website* ini yang diperlukan adalah :

1. *Website* ini dikembangkan hanya dengan menggunakan Algoritma Rabin Karp, diharapkan untuk pengembangan selanjutnya dapat menggunakan algoritma *stemming* untuk mendapatkan hasil perhitungan kemiripan yang lebih akurat.
2. *Website* ini tidak memperhatikan kesalahan ejaan sehingga diharapkan dapat dikembangkan dengan memperhatikan kesalahan ejaan.

REFERENSI

[1] N. N. K. Sari, “Pengembangan Media Informasi Ruang Kuliah Pada Fakultas Teknik Universitas Palangka Raya Berbasis Android dan Location Based Service,” *J. Teknol. Inf.*, vol. 13, no. 2, pp. 30–36, 2019.

[2] W. Widiatry, V. H. Pranatawijaya, N. N. K.

Sari, and P. B. A. A. Putra, “Penerapan Algoritma Levenshteindistance untuk Pencarian pada Sistem Informasi Perpustakaan Fakultas Kedokteran Universitas Palangka Raya,” *J. Saintekom*, vol. 9, no. 1, pp. 66–82, 2019.

- [3] S. Pressman, Roger, *Rekayasa Perangkat Lunak.Pendekatan Praktisi. Edisi 7*. Yogyakarta: Andi, 2012.
- [4] S. Ramadhani, “Sistem Pencegahan Plagiarism Tugas Akhir Menggunakan Algoritma Rabin-Karp (Studi Kasus : Sekolah Tinggi Teknik Payakumbuh),” *J. Teknol. Inf. Komun. Digit. Zo.*, vol. 6, no. 1, pp. 44–52, 2015.
- [5] B. Zaka, *Theory and Applications of Similarity Detection Techniques*. Graz University of Technology, 2009.
- [6] R. A. S. and M. Shalahuddin, *Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek*. Bandung: Informatika, 2013.

