

## **Pengembangan Sistem Pencarian Pada Aplikasi Skripsi Untuk Meningkatkan Hasil Pencarian Judul**

**Roni Ardianzah; Husni Thamrin, S.T., M.T., Ph.D.  
Teknik Informatika, Fakultas Teknik dan Informatika  
Universitas Muhammadiyah Surakarta**

### **Abstrak**

Sistem informasi adalah kombinasi terorganisir dari orang, perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber daya yang mengumpulkan, mengubah, dan mendistribusikan informasi dalam suatu organisasi. Penerapan teknologi Informasi salah satunya dalam bidang skripsi, mampu memudahkan mahasiswa mencari tahu mengenai informasi skripsi sebagai acuan mahasiswa agar bisa menempuh studi di perguruan tinggi. Sering mahasiswa kebingungan dalam pemilihan judul skripsi, karena judul tersebut belum pasti apakah sudah dikerjakan maupun belum diaplikasi myskripsi. Oleh karena itu perlu adanya suatu sistem yang dapat memberikan informasi kepada mahasiswa dalam terkait judul skripsi yang didalamnya berisikan informasi tentang judul, tingkat kemiripan judul dan bisa menjadi referensi mahasiswa dalam mengembangkan judul. Pembuatan sistem menggunakan bahasa pemrograman python, framework Django, serta PostGreSQL sebagai basis data. Mengembangkan fitur pencarian sistem informasi skripsi Universitas Muhammadiyah Surakarta dikembangkan dengan metode waterfall, menggunakan pembobotan TF-IDF serta memanfaatkan algoritma cosine similarity sebagai tingkat kemiripan judul. Pengembangan sistem ini agar dapat meningkatkan kinerja sehingga memberikan informasi untuk mahasiswa serta dosen dalam proses penyusunan skripsi. Hasil yang diperoleh merupakan sistem yang mampu menampilkan hasil kemiripan judul antara judul mahasiswa dengan data judul. Pengujian dengan black box dan System Usability Scale (SUS) menunjukkan nilai rata-rata sebesar 77,17 membuktikan bahwa sistem layak diterima oleh pengguna.

**Kata Kunci:** Pencarian, Skripsi, TF-IDF, Cosine Similarity, Django

### **Abstract**

Information systems are an organised combination of people, hardware, software, communication networks, and resources that collect, modify, and distribute information within an organization. Application of information technology is one in the field of thesis, able to make it easier for students to find out about the thesis information as a student reference to be able to pursue studies in college. Often, students are confused in the selection of the title of the thesis because the title is not sure whether it has been worked on or has not been applied. Therefore, there is a need for a system that can provide information to students in relation to the title of the thesis, which contains information about the title, the degree of similarity of the title, and can be a student's reference in developing the title. System creation using the Python programming language, the Django framework, and PostgreSQL as a database. Developed with the waterfall method, using TF-IDF lubrication as well as using cosine similarity algorithms as title similarity levels. Development of this system in order to improve performance so as to provide information to students as well as lecturers in the process of drafting a thesis. The result obtained is a system that is able to display the result of the title similarity between the student's title and the title data. Testing with the black box and the System Usability Scale (SUS) showed an average score of 77.17, proving that the system deserves acceptance by the user.

**Keywords:** Pencarian, Skripsi, TF-IDF, Cosine Similarity, Django

## 1. PENDAHULUAN

Sistem informasi adalah kombinasi terorganisir dari orang, perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber daya yang mengumpulkan, mengubah, dan mendistribusikan informasi dalam suatu organisasi (Jumardi et al., 2015). Sistem informasi diperlukan demi meningkatkan produktivitas yang baik, meningkatkan mutu layanan suatu organisasi, mengkoordinir setiap bagian dalam organisasi, kualitas kebijakan dari organisasi tersebut (Palumpun et al., 2015). Penerapan teknologi informasi sangatlah penting untuk menghasilkan kualitas sistem informasi yang efektif, efisien, dan pemrosesan data yang akurat (Salaby, 2017).

Penerapan teknologi Informasi salah satunya dalam hal skripsi, sistem informasi skripsi memudahkan mahasiswa mencari tahu tentang informasi skripsi sebagai acuan mahasiswa agar bisa menempuh studi di perguruan tinggi. Disertasi adalah suatu karya ilmiah yang membahas suatu topik permasalahan dalam suatu bidang keilmuan tertentu dengan menggunakan kaidah yang berlaku, yang harus diselesaikan mahasiswa secara mandiri agar mendapat gelar studi S1 (Patimah, 2018). Secara umum sistem informasi skripsi adalah sistem untuk memudahkan mahasiswa untuk mengetahui bagaimana tata cara pengajuan judul skripsi, penulisan skripsi, informasi dosen pembimbing, informasi kemiripan judul yang telah dikerjakan oleh mahasiswa.

Universitas Mummadiyah Surakarta selalu meningkatkan mutu kualitas pelayanannya. Salah satunya dalam bentuk pelayanan administrasi dan konsultasi skripsi berbasis *website*, itu dapat diakses melalui <http://skripsi.ums.ac.id>. Tahapan proses skripsi yang dilakukan mahasiswa paling penting yaitu memiliki judul penelitian dan tentunya memiliki referensi yang cukup untuk menentukan judul skripsi. Mahasiswa dituntut untuk menentukan judul skripsinya sesuai teori yang sudah didapatkan, karena Judul skripsi yang baik dan benar akan dilihat dari seberapa implementasi teori oleh mahasiswa tersebut (Ramadhan et al., 2017). Permasalahan yang terjadi dalam pemilihan suatu judul skripsi yaitu kemiripan antara judul mahasiswa dan judul sudah dikerjakan. Informasi semacam ini belum tersedia di aplikasi MySkripsi. Oleh karena itu diperlukan suatu sistem yang dapat memberikan informasi kepada mahasiswa terkait pemilihan judul skripsi yang didalamnya berisikan informasi tentang judul dan bisa menjadi suatu referensi mahasiswa dalam mengembangkan judul yang ada maupun yang belum (Pristiwati & Tomi, 2020).

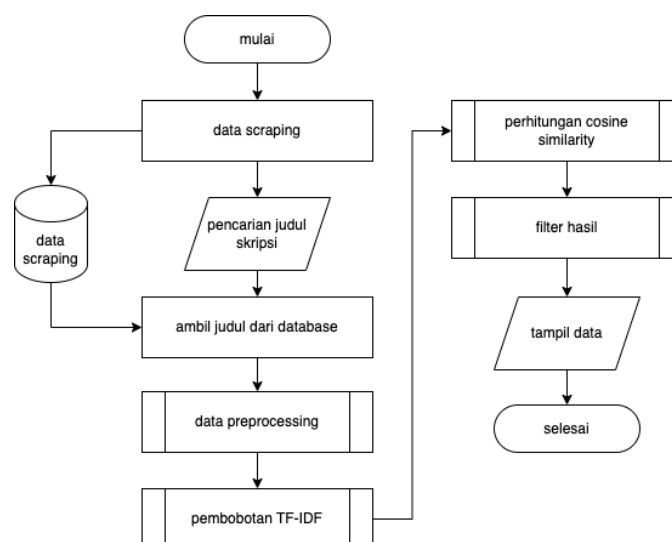
Penelitian mengenai sistem pencarian judul skripsi sudah pernah dilakukan oleh Salaby (2017), yaitu penerapan pemrograman php MySQL dan pengolahan data dengan menggunakan

microsoft word dan microsoft excel dalam pengembangan aplikasi pencarian. Prasetyo (2015) juga melakukan penelitian untuk membantu pengguna menemukan laporan skripsi dan Praktek Kerja Lapangan (PKL) dengan teknologi *website* semantik agar mendapatkan hasil pencarian sesuai dengan kebutuhan. Sistem diuji secara black box dan white box testing dan hasilnya valid. Selain itu, ada juga yang melakukan penelitian dengan menggunakan metode N-Gram dan *query extension* untuk mendeteksi persamaan pada dokumen skripsi (Iswanto & Thamrin, 2014). Perbedaan penting antara ketiga penelitian tersebut dengan penelitian yang diusulkan ini adalah integrasi fitur pencarian judul skripsi dengan sistem pengelolaan skripsi. Selain itu terdapat perbedaan bahasa pemrograman dan data base yang digunakan. Penelitian yang diusulkan ini menggunakan *framework* Django dan *database* PostgreSQL.

Tujuan penelitian ini adalah mengembangkan fitur pencarian sistem informasi skripsi yang terpadu dengan sistem pengelolaan skripsi (*myskripsi*) di Universitas Muhammadiyah Surakarta. Sistem nantinya akan memberikan informasi judul skripsi, abstrak, kata kunci, penulis, pembimbing serta menampilkan tingkat kemiripan terhadap judul yang di *input* oleh pengguna, dan dapat dipergunakan oleh pengguna yaitu mahasiswa untuk mengetahui tingkat kemiripan judul dengan data yang selesai dikerjakan oleh mahasiswa lainnya.

## 2. METODE

Bagan alur, juga dikenal sebagai *flowchart*, adalah diagram yang menunjukkan langkah-langkah dan pilihan yang harus diambil untuk menjalankan sebuah proses dalam suatu kegiatan(Sukhodimtseva et al., 2021). Dalam bentuk diagram, setiap Langkah dihubungkan dengan garis atau arah panah. *Flowchart* digunakan peneliti untuk menggambarkan alur penelitian sehingga mudah dipahami dari mulai penelitian sampai proses selesai ditunjukkan pada gambar 1.



Gambar 1. *Flowchart* pengumpulan data

## 2.1 Data Collection

Pengembangan sistem pencarian judul ini membutuhkan data-data untuk ditampilkan sesuai kebutuhan pengguna, maka perlu dilakukan pengumpulan data yang dibutuhkan oleh sistem. Data tersebut diperoleh dari situs *eprints.ums.ac.id* yang ingin di ambil datanya, dari proses *scraping* yaitu teknik pengambilan informasi web secara otomatis tanpa perlu menyalinnya dengan tangan atau manual (A. Yani et al., 2019). Langkah ini mengakses dari situs *eprints.ums.ac.id* untuk mendapatkan data kode prodi, tahun, judul skripsi, serta abstrak dalam bentuk format Json.

## 2.2 Data Preprocessing

Proses *preprocessing* dilakukan karena data skripsi yang diperoleh dari *eprints.ums.ac.id* masih dalam bentuk mentah. Proses ini dilakukan untuk mendapatkan data bersih sehingga dapat dilanjutkan ke tahap berikutnya. Proses yang dilakukan adalah (Gunawan et al., 2018):

1. Tahap *Case Folding* adalah mengubah semua huruf A hingga Z dalam sebuah dataset menjadi huruf a hingga z.
2. *Remove punctuation*, tahap untuk menghilangkan tanda baca atau simbol pada sebuah dataset.
3. *Remove whitespace*, tahap untuk menghilangkan spasi di dalam sebuah dataset.

Setelah melalui tahap pengumpulan dan *data preprocessing* data akan disimpan dalam *database* dan kemudian diproses ke dalam tahap klasifikasi kemiripan judul menggunakan pembobotan TF-IDF dan analisis menggunakan algoritma *cosine similarity*.

### 2.2.1 Term Frequency Invers Document Frequency

Metode pembobotan secara statistik, *Term Frequency Invers Document Frequency*, menunjukkan seberapa penting sebuah kata pada suatu dokumen dan dimana dokumen berada dalam kelompok dokumen (Adi et al., 2018). Frekuensi banyaknya kata yang muncul di sebuah dokumen yang di berikan merupakan seberapa penting kata dalam dokumen: jumlah kata yang muncul dalam dokumen menunjukkan seberapa umum kata tersebut. Oleh karena itu, bobot hubungan antara kata dan dokumen akan lebih besar jika ada banyak kata muncul dalam dokumen, dan jika ada sedikit kata muncul dalam jumlah total dokumen (Wahyuni et al., 2017). Rumus membobotkan TF-IDF adalah seperti yang ditunjukkan pada (3).

$$tf = \frac{tf}{\max(tf)} \quad (1)$$

$$idf_t = \log\left(\frac{D}{df_t}\right) + 1 \quad (2)$$

$$W_{d,t} = tf_{d,t} \times IDF_{d,t} \quad (3)$$

Keterangan:

$tf$  = banyaknya kata pada sebuah dokumen

$\max(tf)$  = jumlah kemunculan terbanyak *term* pada dokumen yang sama

$D$  = jumlah total dokumen

$df_t$  = jumlah dokumen yang mengandung *term*  $t$

$IDF$  = nilai *inverse dokumen frequency*

$d$  = dokumen ke- $d$

$t$  = kata ke- $t$  dari kata kunci

$W$  = nilai bobot dokumen ke- $d$  terhadap ke- $t$

### 2.2.2 Cosine Similarity

Metode ini dapat digunakan untuk menghitung tingkat kesamaan judul. Umumnya metode ini perhitungannya didasarkan dengan *vector space similarity measure* yaitu menggunakan titik dalam ruang untuk menunjukkan seberapa mirip dokumen dengan kueri dalam sebuah koleksi (Nurgiyatna, Sujalwo, 2017). *Cosine similarity* ini merupakan sudut kesamaan antara dokumen dengan kueri yang dinyatakan dalam dua buah vektor. Berikut rumus *cosine similarity* (4).

$$\cosSim(d_j, q_k) = \frac{\sum_{i=1}^n (td_{ij} \times tq_{ik})}{\sqrt{\sum_{i=1}^n td_{ij}^2 \times \sum_{i=1}^n tq_{ij}^2}} \quad (4)$$

Keterangan:

$\cosSim(d_j, q_k)$  = tingkat kesamaan dokumen dengan *query* tertentu

$td_{ij}$  = *term* ke- $i$  dalam vektor untuk dokumen ke- $j$

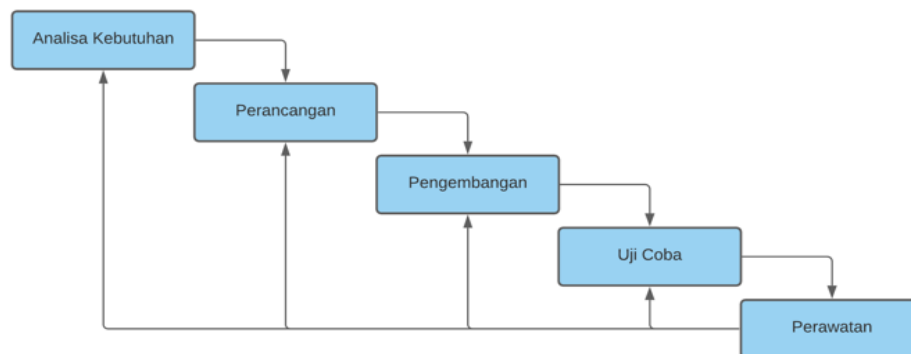
$td_{ik}$  = *term* ke- $i$  dalam vektor untuk dokumen ke- $k$

$n$  = jumlah *term* yang unik dalam *dataset*.

## 2.3 Perancangan Sistem

Perancangan sistem pencarian ini menggunakan metode *Waterfall*. Metode ini merupakan pengembangan perangkat atau sistem yang mudah dan terjadwal secara linier serta berurutan

yang menyelesaikan tahapan sebelum tahapan selanjutnya(Sasmito, 2017). Pengembangannya meliputi tahapan-tahapan sebagai berikut(Ristian & Anggoro, 2020):



Gambar 2. Diagram metode *waterfall*

### 2.3.1. Analisa Kebutuhan

Langkah pertama dalam proses “PENGEMBANGAN SISTEM Pencarian pada Aplikasi Skripsi untuk Meningkatkan Hasil Pencarian Judul” adalah analisis kebutuhan. Hal-hal yang dibutuhkan yaitu :

#### A. Wawancara

Tahap ini penulis penelitian melakukan wawancara kepada *developer* sistem informasi skripsi Universitas Muhammadiyah Surakarta terkait sistem informasi yang telah dibuat. Hasil wawancara tersebut didapatkan sistem pencarian judul skripsi di sistem informasi skripsi kurang efektif, maka dari itu peneliti perlu mengembangkan sistem pencarian di sistem tersebut agar dapat mengeluarkan hasil yang diinginkan dan menjadi sistem pencarian yang efektif bagi mahasiswa lingkup Universitas Muhammadiyah Surakarta. Hasil wawancara dapat disimpulkan bahwa sistem pencarian di aplikasi skripsi Universitas Muhammadiyah Surakarta membutuhkan seperti berikut :

- a) Sistem pencarian yang efektif dan efisien.
- b) Sistem dapat menampilkan hasil informasi yang lengkap mengenai judul yang dicari dan kemiripan judul yang dicari oleh mahasiswa.

#### B. Software

- a) Python
- b) Basisdata PostgreSQL
- c) Chrome Browser
- d) Text Editor Visual Studio Code
- e) Sistem operasi MacOS

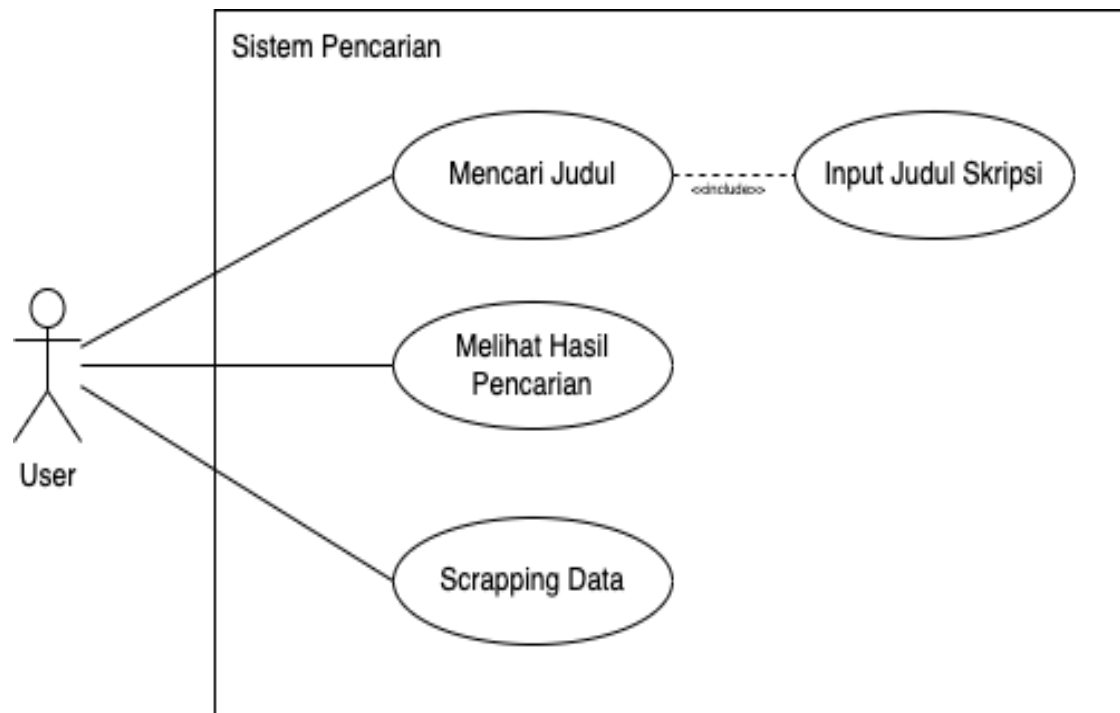
#### C. Hardware

- a) Intel dual-core i5 2,3Ghz
- b) Intel Iris Plus Graphic 640 1536Mb
- c) RAM 8GB LPDDR3
- d) SSD 256GB

## 2.3.2. Perancangan

### 2.3.2.1 Use Case Diagram

Peneliti menggunakan *Use Case Diagram* untuk menentukan interaksi antara pengguna dengan sistem secara umum. Salah satu jenis diagram UML (*Unified Modeling Language*) yang digunakan untuk menunjukkan hubungan interaksi antara actor dan sistem adalah diagram ini. Tipe interaksi yang dilakukan oleh pengguna sistem dengan sistem dapat dijelaskan dengan menggunakan kasus (Aquino et al., 2020).

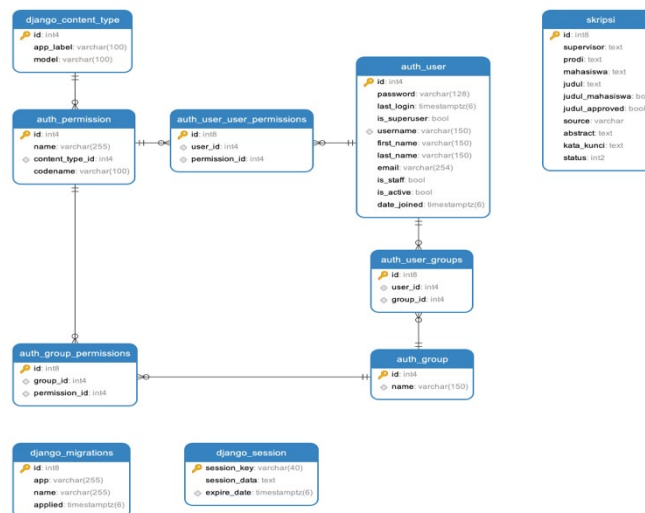


Gambar 3. *Use Case Diagram*

Gambar 3 merupakan skema *Use Case Diagram* dalam penelitian. *User* dalam diagram tersebut merupakan mahasiswa dan dosen yang dapat melakukan scraping data dan melakukan input judul untuk mencari kemiripan judul dalam database.

### 2.3.2.2 Entity Relationship Diagram

Untuk merancang suatu *database*, diagram relasi entitas adalah cara yang bagus untuk menggambarkan sebuah entitas (Hutagalung, 2016).

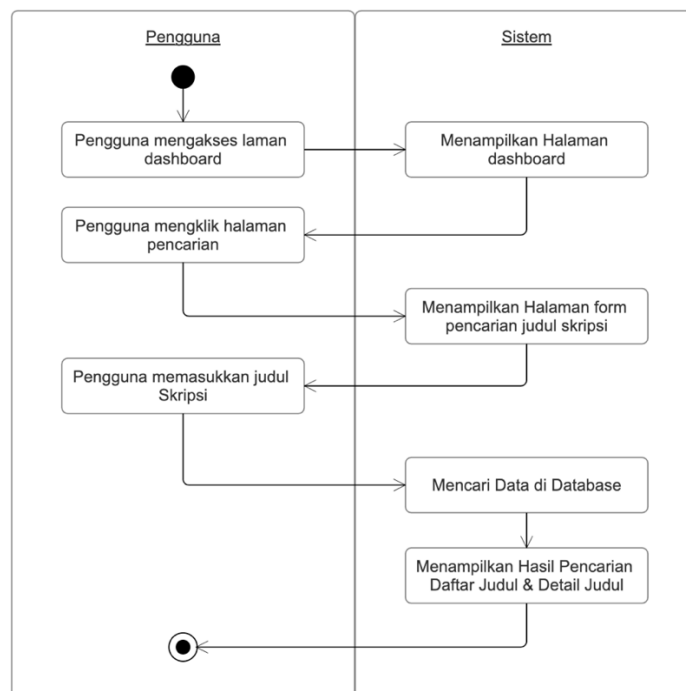


Gambar 4. *Entity Relationship Diagram* Sistem

Gambar 3 menjelaskan rancangan database yang digunakan dalam penelitian pengembangan sistem. Rancangan tersebut terdiri dari tabel skripsi yang bertugas untuk menyimpan data skripsi dari hasil *scraping* data dari situs *eprints.ums.ac.id* dan selebihnya merupakan bawaan dari sistem *framework* django.

### 2.3.2.3 Activity Diagram

*Activity diagram* merupakan penggambaran berbagai alur aktifitas dalam sistem yang dikembangkan. Pengembangan sistem pencarian ini meliputi bagian awal dari pengguna dan bagaimana berakhirnya dapat dilihat pada gambar 5.



Gambar 5. *Activity Diagram* Pencarian Judul

Gambar 5. Merupakan aktifitas dalam mencari judul skripsi yang dilakukan oleh dosen maupun mahasiswa. Pengguna mengakses halaman skripsi, berikutnya memasukkan judul skripsi di halaman dashboard pencarian. Setelah itu, sistem akan mencari judul sesuai judul yang dimasukkan oleh pengguna dan menampilkan pencarian berupa judul, abstrak, deskripsi, kata kunci, penulis, pembimbing serta presentase tingkat kemiripan antara judul mahasiswa dengan judul yang ditampilkan.

### 2.3.3. Pengembangan

Penelitian ini mengembangkan yang sudah ada sebelumnya dengan memanfaatkan fitur dalam PostgreSQL serta menggunakan *framework* Django. Pertama kali nya dikembangkan basisdata dianggap tidak memenuhi standar basis data berkala besar, tujuannya hanya menjadikan PostgreSQL sebagai suatu basis data mempunyai banyak fitur dan keluwesan yang sangat tinggi (Munawaroh, 2005). PostgreSQL yang memungkinkan untuk mendefinisikan SQL sendiri, terutama pembuatan *function*. Memungkinkan informasi yang tersimpan tidak hanya tabel dan kolom melainkan tipe, fungsi, akses, dan informasi lainnya. Dengan PostgreSQL akan lebih mudah dikembangkan untuk penelitian yang dilakukan saat ini. Kelebihannya yang lain yakni dukungan dari berbagai macam bahasa pemrograman salah satunya Python dan masih banyak lagi. Dalam pengembangan penelitian ini juga menggunakan salah satunya *framework* dari Python yakni Django. Django ini merupakan salah satu *framework* yang serbaguna dan terstruktur dengan sedemikian rupa (Prasetyo, 2022).

### 2.3.4. Uji Coba

Tahap pengujian dilakukan setelah sistem selesai dibuat oleh peneliti. Uji coba akan dilakukan dengan *blackbox testing* guna mengetahui bahwa program yang telah dibuat berfungsi dengan baik tanpa ada kendala, serta penyebaran kuesioner melalui *link google form* kepada pengguna guna menguji sistem untuk pengujian *System Usability Scale* (SUS) untuk proses evaluasi sistem yang dibuat. Sehingga, dengan 2 metode pengujian tersebut pengembangan sistem pencarian dapat disimpulkan tingkat kelayakannya terhadap pengguna.

### 2.3.5 Perawatan

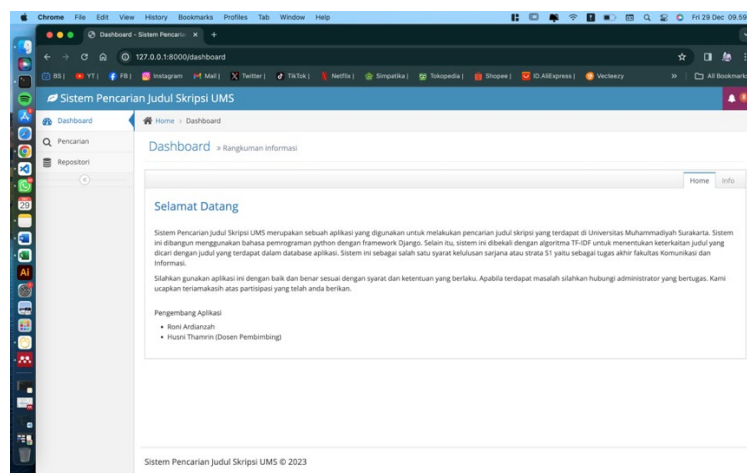
Perawatan adalah tahapan terakhir dari metode *waterfall*. Ini memungkinkan pengembang untuk melakukan perbaikan atau kesalahan yang toleran pada tahap sebelumnya. Perawatan juga dapat dilakukan untuk meningkatkan dan menyesuaikan sistem jika diperlukan.

## 3. HASIL DAN PEMBAHASAN

Berdasarkan metode penelitian penerapan algoritma dalam pengembangan sistem pencarian pada aplikasi skripsi telah dilakukan dan mendapatkan sebuah hasil menampilkan data dari

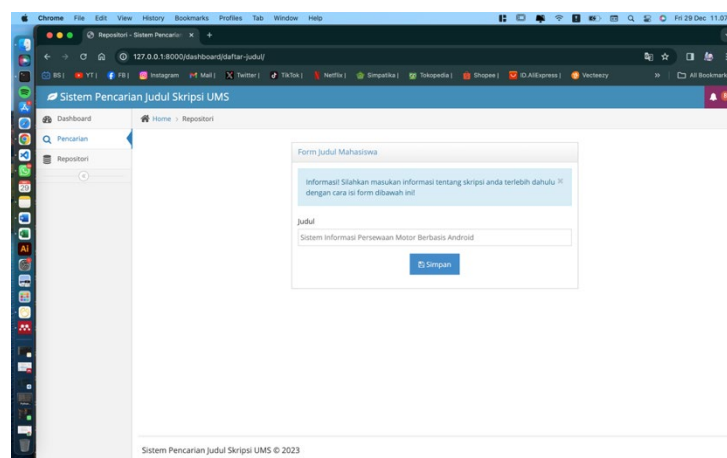
data *scraping* sebagai repositori judul yang sudah selesai dan presentase tingkat kemiripan judul mahasiswa. Adanya peneltian sistem ini digunakan untuk sebuah acuan mahasiswa dalam menentukan referensi judul skripsi dari tingkat kemiripan judul yang ada dalam *repository database* sehingga dapat menentukan judul skripsi mahasiswa tersebut. Berikut hasil yang diperoleh dari penelitian.

Laman yang akan di akses pertama oleh pengguna saat menggunakan sistem yaitu *dashboard* seperti pada gambar 6. Halaman ini ada 2 fungsi berupa *home* yang berisi informasi singkat tentang peneliti dalam pengembangan sistem, serta fungsi *info* berisi tentang versi aplikasi yang di kembangkan. Halaman ini dibuat sederhana agar dapat diterima pengguna saat menggunakan aplikasi.



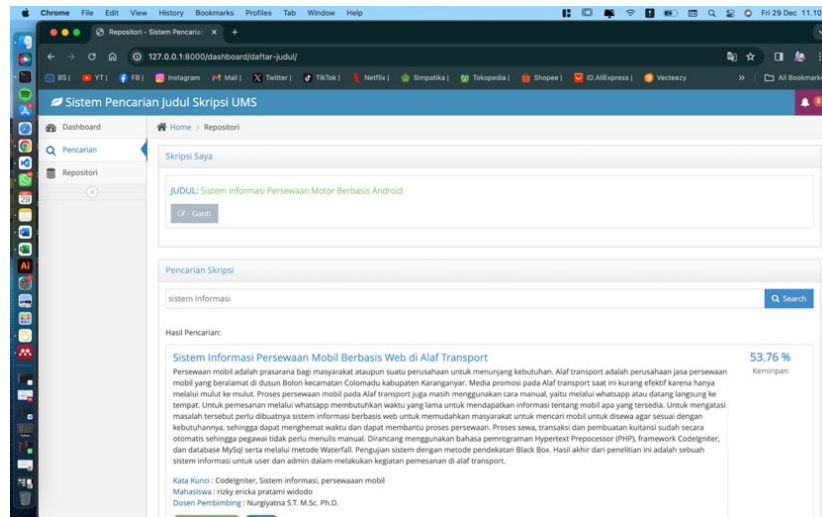
Gambar 6. Halaman *Dashboard*

Halaman yang akan tampil setelah pengguna mengklik *tab* pencarian adalah halaman pencarian seperti pada gambar 7. Halaman pencarian terdapat informasi memasukkan judul skripsi. Pengguna harus mengisi *form* “Masukan judul skripsi” secara lengkap terlebih dahulu.



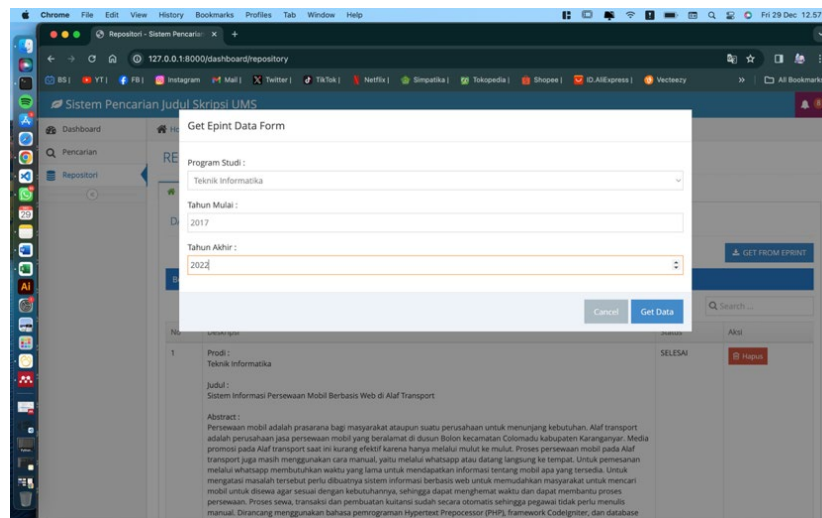
Gambar 7. Halaman Pencarian

Pengguna yang telah mengisi *form* judul skripsinya akan ditampilkan ke halaman hasil pencarian seperti pada gambar 8. Halaman hasil pencarian ada fitur pencarian “Apa yang sedang anda cari ?” digunakan untuk menyaring topik yang sedang dicari pengguna, setelah itu pengguna mengklik tombol “*Search*”, sistem akan menampilkan hasil dari topik yang dicari pengguna beserta tingkat kemiripan judul mahasiswa dengan judul yang ada di basis data.



Gambar 8. Halaman Hasil Pencarian

Halaman repositori merupakan halaman dimana untuk mengumpulkan data dengan proses *scraping* serta menampilkan datanya.



Gambar. 9 Halaman repositori.

Halaman ini terdapat tombol “GET FROM EPRINT” akan muncul *form* untuk di isi dari Program Studi, Tahun Mulai, serta Tahun Akhir bilamana pengguna sudah mengisi tersebut, maka klik tombol “GET DATA” untuk proses pengambilan data di *eprints*. Setelah itu akan ditampilkan data dari proses pengambilan data dari *eprints*.

### 3.2 Pengujian

### 3.2.1 Pengujian *Black Box*

Peneliti menguji sistem untuk 2 tujuan. Pertama, peneliti menguji sistem untuk memastikan bahwa sistem itu berjalan dengan baik dan kedua, peneliti mencari kesalahan atau eror saat menggunakannya. Metode pengujian *Black Box* digunakan dalam pengujian ini. Peneliti memilih pengujian ini karena fokus dalam pengujian fungsionalitas, *testing* semua halaman yang tersedia di sistem, serta memastikan halaman tersebut sudah berjalan dengan baik. Dapat dilihat pada tabel 1.

Tabel 1 Pengujian *Black Box*

| Halaman    | Skenario                             | Aksi                                                                           | Hasil yang diharapkan                                                                                 | Hasil |
|------------|--------------------------------------|--------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|-------|
| Dashboard  | Membuka halaman dashboard            | Klik tombol “dashboard” pada navbar                                            | Sistem akan tetap di halaman dashboard                                                                | Lolos |
|            | Membuka tab info                     | Klik tombol “info”                                                             | Sistem akan menginformasikan halaman info                                                             | Lolos |
|            | Membuka tab home                     | Klik tombol “home”                                                             | Sistem akan menginformasikan halaman home                                                             | Lolos |
| Pencarian  | Membuka halaman dashboard            | Klik tombol “dashboard” pada navbar                                            | Sistem akan menginformasikan halaman dashboard                                                        | Lolos |
|            | Membuka halaman pencarian            | Klik tombol “Pencarian” pada navbar                                            | Sistem akan menginformasikan halaman pencarian                                                        | Lolos |
|            | Mengisi judul skripsi                | Mengisi judul skripsi pada kolom “Masukan judul Skripsi”                       | Sistem akan menginformasikan notifikasi “Judul Berhasil Disimpan” dan hasil pencarian                 | Lolos |
|            | Tidak mengisi judul skripsi          | Tidak Mengisi judul skripsi pada kolom “Masukan judul Skripsi”                 | Sistem akan menginformasikan notifikasi “Judul Harus Diisi” dan kembali kolom “Masukan judul skripsi” | Lolos |
|            | Memfilter data skripsi               | Mengisi topik pada kolom “Apa yang sedang anda cari?” dan Klik tombol “search” | Sistem akan menginformasikan halaman hasil cari                                                       | Lolos |
| Repositori | Membuka halaman dashboard            | Klik tombol “dashboard” pada navbar                                            | Sistem akan menginformasikan halaman dashboard                                                        | Lolos |
|            | Membuka halaman repositori           | Klik tombol “repositori” pada navbar                                           | Sistem akan menginformasikan halaman repositori                                                       | Lolos |
|            | Membuka kolom dan melakukan scraping | Klik tombol “GET FROM EPRINT” dan mengisi kolom                                | Sistem akan menginformasikan kolom “GET EPRINT DATA                                                   | Lolos |

|  |  |  |                                  |  |
|--|--|--|----------------------------------|--|
|  |  |  | FORM” dan akan hasil<br>scraping |  |
|--|--|--|----------------------------------|--|

### 3.2.2 System Usability Scale

Pengujian ini melibatkan sepuluh pernyataan yang diberikan kepada pengguna setelah menggunakan sistem, yang dapat dilihat di tabel 2. Setelah itu, pengguna diminta untuk memberikan jawaban antara sangat tidak setuju, ragu-ragu, setuju, atau sangat setuju dengan pernyataan yang telah diberikan. Angka 1-5 akan digunakan untuk menunjukkan tingkat respons pengguna.

Tabel 2. Pengujian *System Usability Scale* (SUS)

| Pernyataan                                                                    |
|-------------------------------------------------------------------------------|
| Saya tertarik untuk menggunakan sistem ini lagi.                              |
| Saya berfikir sistem ini tidak praktis jika digunakan.                        |
| Saya berfikir sistem ini mudah untuk digunakan.                               |
| Saya perlu meminta bantuan orang teknis, saat menggunakan sistem.             |
| Saya berfikir bahwa fitur atau tombol pada sistem berjalan dengan semestinya. |
| Saya melihat banyak ketidakserasian pada sistem.                              |
| Saya yakin pengguna lain akan mudah memahami cara menggunakan sistem ini.     |
| Saya merasa bingung selama menggunakan sistem.                                |
| Saya sangat yakin bisa menggunakan sistem.                                    |
| Saya perlu belajar dan membiasakan diri, sebelum menggunakan sistem.          |

Setelah itu peneliti menghitung skor nilai SUS yang didapat untuk mendapatkan nilai rata-rata SUS. Cara menghitung nilai skor pada metode SUS adalah nomor pernyataan ganjil dikurangi 1, nomor pernyataan genap nilai 5 dikurangi nilai responden, dan setelah nilai diakumulasi dikalikan 2,5 (Setemen et al., 2019). Dapat dilihat pada tabel 3.

Tabel 3. Nilai Hasil Pengujian *System Usability Scale* (SUS)

| Responden | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | JML | NILAI<br>JML*2,5 |
|-----------|----|----|----|----|----|----|----|----|----|-----|-----|------------------|
| 1         | 4  | 4  | 3  | 4  | 3  | 2  | 3  | 3  | 3  | 1   | 30  | 75               |
| 2         | 3  | 2  | 3  | 3  | 4  | 3  | 3  | 3  | 3  | 3   | 30  | 75               |
| 3         | 4  | 3  | 4  | 4  | 4  | 3  | 3  | 3  | 3  | 3   | 34  | 85               |
| 4         | 4  | 3  | 4  | 4  | 4  | 3  | 3  | 4  | 4  | 3   | 36  | 90               |
| 5         | 3  | 2  | 3  | 3  | 3  | 2  | 3  | 3  | 3  | 3   | 28  | 70               |
| 6         | 3  | 2  | 3  | 3  | 3  | 2  | 3  | 3  | 4  | 3   | 29  | 72,5             |
| 7         | 2  | 2  | 2  | 4  | 3  | 2  | 3  | 3  | 2  | 3   | 26  | 65               |
| 8         | 3  | 3  | 4  | 4  | 3  | 3  | 3  | 3  | 4  | 3   | 33  | 82,5             |
| 9         | 4  | 1  | 3  | 2  | 4  | 3  | 3  | 2  | 3  | 3   | 28  | 70               |
| 10        | 4  | 1  | 3  | 3  | 4  | 3  | 2  | 4  | 4  | 3   | 31  | 77,5             |
| 11        | 3  | 1  | 4  | 3  | 4  | 4  | 4  | 3  | 4  | 4   | 34  | 85               |
| 12        | 3  | 1  | 3  | 2  | 3  | 4  | 3  | 3  | 3  | 2   | 27  | 67,5             |
| 13        | 3  | 3  | 3  | 4  | 2  | 1  | 3  | 2  | 4  | 1   | 26  | 65               |
| 14        | 4  | 4  | 3  | 3  | 4  | 3  | 0  | 4  | 4  | 3   | 32  | 80               |
| 15        | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4   | 40  | 100              |
| 16        | 3  | 3  | 3  | 3  | 3  | 2  | 3  | 3  | 3  | 2   | 28  | 70               |

| Responden          | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | JML | NILAI<br>JML*2,5 |
|--------------------|----|----|----|----|----|----|----|----|----|-----|-----|------------------|
| 17                 | 2  | 3  | 0  | 3  | 4  | 3  | 3  | 1  | 2  | 3   | 24  | 60               |
| 18                 | 3  | 4  | 4  | 4  | 4  | 3  | 3  | 4  | 4  | 3   | 36  | 90               |
| 19                 | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 0   | 36  | 90               |
| 20                 | 3  | 3  | 4  | 3  | 2  | 3  | 3  | 3  | 4  | 2   | 30  | 75               |
| 21                 | 4  | 4  | 3  | 3  | 2  | 3  | 4  | 2  | 3  | 3   | 31  | 77,5             |
| 22                 | 3  | 3  | 4  | 2  | 3  | 3  | 4  | 3  | 3  | 3   | 31  | 77,5             |
| 23                 | 4  | 1  | 3  | 2  | 4  | 4  | 3  | 3  | 4  | 2   | 30  | 75               |
| <b>RATA - RATA</b> |    |    |    |    |    |    |    |    |    |     |     | <b>77,17</b>     |

Rumus (5) digunakan untuk menghitung nilai SUS rata-rata.

$$\bar{X} = \frac{\sum x}{n} \quad (5)$$

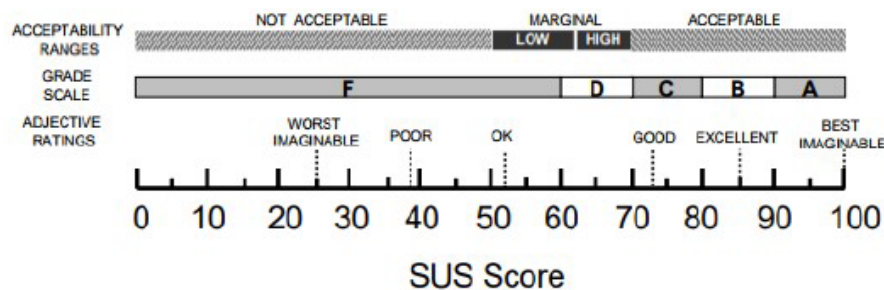
Keterangan:

$\bar{X}$  = nilai rata – rata

$\sum x$  = jumlah Nilai SUS

n = jumlah responden

Hasil rata – rata yang diperoleh dengan menggunakan rumus (5) tersebut adalah 77,17 dan hasil ini akan diuji kelayakan dengan menggunakan diagram nilai yang telah ditentukan, seperti yang ditunjukkan pada gambar 10.



Gambar 10. Nilai *System Usability Scale*

Menurut nilai rata-rata yang di peroleh, untuk sistem ini masuk ke dalam kategori penilaian “*GOOD*”, maka kelayakannya termasuk dalam kategori “*ACCEPTABLE*” yang menandakan bahwa sistem diterima oleh pengguna dengan baik.

#### 4. PENUTUP

Seperti yang diharapkan, proyek ini berjalan dengan baik. Ini mencakup pengembangan sistem pencarian untuk aplikasi skripsi untuk meningkatkan hasil pencarian judul. Hasil yang diperoleh merupakan sistem yang dapat menampilkan hasil kemiripan judul antara judul mahasiswa dengan data judul yang sudah selesai di web eprints. Pengujian *Black Box* dan

*System Usability Scale* (SUS), dengan nilai rata-rata 77,17, menunjukkan bahwa sistem ini dapat diterima dengan baik oleh pengguna. Hasil pengujian ini menunjukkan bahwa sistem ini berjalan dengan baik.

## Daftar Pustaka

- A. Yani, D. D., Pratiwi, H. S., & Muhandi, H. (2019). Implementasi Web Scraping untuk Pengambilan Data pada Situs Marketplace. *Jurnal Sistem Dan Teknologi Informasi (JUSTIN)*, 7(4), 257. <https://doi.org/10.26418/justin.v7i4.30930>
- Adi, K., #1, N., & Sebastian, D. (2018). Pembentukan Dataset Topik Kata Bahasa Indonesia pada Twitter Menggunakan TF-IDF & Cosine Similarity. *Jurnal Teknik Informatika Dan Sistem Informasi*, 4, 2443–2229. <http://dx.doi.org/10.28932/jutisi.v4i3.862>
- Aquino, E., de Saqui-Sannes, P., & Vingerhoeds, R. A. (2020). A Methodological Assistant for Use Case Diagrams. *International Conference on Model-Driven Engineering and Software Development*, 2020(February), 227–236. <https://doi.org/10.5220/0008938002270236>
- Gunawan, D., Sembiring, C. A., & Budiman, M. A. (2018). The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents. *Journal of Physics: Conference Series*, 978(1). <https://doi.org/10.1088/1742-6596/978/1/012120>
- Hutagalung, D. M. (2016). Skenario Pengaruh Dalam Pemodelan Er Diagram. *CESSJournal Of Computer Engineering, System And Science*, 1(2), 26–31.
- Iswanto, D., & Thamrin, H. (2014). Perbandingan Hasil Deteksi Kemiripan Topik Skripsi Dengan Menggunakan Metode N-Gram Dan Ekspansi Kueri. *Perbandingan Hasil Deteksi Kemiripan Topik Skripsi Dengan Menggunakan Metode N-Gram Dan Ekspansi Kueri*.
- Jumardi, R., Nugroho, E., & Hidayah, I. (2015). Analisis Kesuksesan Implementasi Sistem Informasi Skripsi pada Program Studi Teknik Informatika Universitas Pembangunan Nasional “ Veteran ” Yogyakarta. 7–13.
- Munawaroh, S. (2005). Mengeksplorasi Database PostgreSQL dengan PgAdmin III. *Jurnal Teknologi Informasi DINAMIK*, X(2), 103–107.
- Nurgiyatna, Sujalwo, S. H. (2017). Aplikasi Pencarian Kamar Dalam Sistem Reservasi Hotel Dengan Metode Vector Space Model Menggunakan Antarmuka Bahasa Sehari-Hari. *Symposium Nasional RAPI XVI – 2017 FT UMS*, 2008, 328–334.
- Palumpun, A. N., Lomi, A., & Handoko, F. (2015). Perancangan Sistem Informasi Akademik Untuk Meningkatkan Kinerja Manajemen (Studi Kasus: Universitas Satya Wiyata Mandala Nabire). *Jurnal Teknologi Dan Manajemen Industri*, 1(1), 15–24. <http://eprints.itn.ac.id/3151/1/248-49-483-2-10-20171107.pdf>
- Patimah, S. (2018). Aplikasi Manajemen E-Skripsi Online (Studi Kasus Prodi Sistem Informasi Universitas Islam Indragiri). *Sistemasi*, 7(3). <https://doi.org/10.32520/stmsi.v7i3.358>
- Prasetyo, D. A. B. (2022). Implementasi Information Schema Database Pada PostgreSQL untuk Pembuatan Tabel Informasi dengan Menggunakan Python Di PT XYZ. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 9(3), 1961–1972.

- <https://doi.org/10.35957/jatisi.v9i3.2221>
- Pristiwati, F., & Tomi, A. (2020). *Sistem Pendukung Keputusan dengan Metode WASPAS, COPRAS, dan EDAS: Menentukan Judul Skripsi*. 4, 56.  
<https://doi.org/10.30865/mib.v4i4.2431>
- Ramadhan, M. R., Nugroho, L. E., & Sulisty, S. (2017). Perancangan Sistem Informasi Monitoring Skripsi. *Proceedings of Conference on Information Technology and Electrical Engineering*, 290–295.
- Ristian, Y., & Anggoro, D. A. (2020). Sistem Pendukung Keputusan Penyeleksian Jadwal Kegiatan ORMAWA FKI UMS Dengan Metode Weighted Product Berbasis Web. *Emitor: Jurnal Teknik Elektro*, 21(1), 10–19.  
<https://doi.org/10.23917/emitor.v21i01.11989>
- Salaby, A. F. (2017). *Membangun Aplikasi Pencarian Judul LTA dan Skripsi Fakultas Ilmu Komputer Universitas Dehasen Bengkulu*. 13(1), 1–11.
- Sasmito, G. W. (2017). Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, 2(1), 6–12.
- Setemen, K., Erawati Dewi, L. J., & Purnamawan, I. K. (2019). PAON usability testing using system usability scale. *Journal of Physics: Conference Series*, 1165(1).  
<https://doi.org/10.1088/1742-6596/1165/1/012009>
- Sukhodimtseva, A., Sergeeva, M., Lukashenko, D., & Pyankov, M. (2021). The methodology of the creation of a flowchart of teacher professional development programme in functional literacy aspect. *Revista Tempos e Espaços Em Educação*, 14(33), e15525.  
<https://doi.org/10.20952/revtee.v14i33.15525>
- Wahyuni, R. T., Prastiyanto, D., & Suprpto, E. (2017). Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi. *Jurnal Teknik Elektro Universitas Negeri Semarang*, 9(1), 18–23.  
<https://journal.unnes.ac.id/nju/index.php/jte/article/download/10955/6659>