



Optimalisasi Aplikasi Pengendalian Skripsi Menggunakan Algoritma *Dynamic Priority Scheduling* dan *Sequential Search*

Arya Arief ^{1*}, Ratih Titi Komala Sari ², Eri Mardiani ³

^{1*,2,3} Program Studi Informatika, Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta, Indonesia.

article info

Article history:

Received 6 February 2024
Received in revised form
12 March 2024
Accepted 1 May 2024
Available online July 2024.

DOI:

<https://doi.org/10.35870/jti.k.v8i3.2219>

Keywords:

Dynamic Priority Scheduling;
Sequential Search; Thesis
Management.

Kata Kunci:

Dynamic Priority Scheduling;
Sequential Search;
Pengendalian Skripsi.

abstract

Thesis registration in the academic environment often faces priority and time management problems, therefore an effective and responsive approach is needed. This study suggests optimizing the Dynamic Priority Scheduling and Sequential Search algorithms to improve the performance of thesis control applications. Dynamic characteristics of students and service requests regulate submission priorities and a sequential search algorithm improves the search for thesis-related information in the database. This research discusses the ideas of both algorithms and how they can improve system efficiency by integrating them.

abstrak

Pendaftaran skripsi dilingkungan akademik seringkali menghadapi masalah manajemen prioritas dan waktu, karena itu pendekatan yang efektif dan responsif diperlukan. Studi ini menyarankan pengoptimalan algoritma Dynamic Priority Scheduling dan Sequential Search untuk meningkatkan kinerja aplikasi pengendalian skripsi. Karakteristik dinamis mahasiswa dan permintaan layanan mengatur prioritas pengajuan dan algoritma pencarian sequential meningkatkan pencarian informasi terkait skripsi dalam basis data. Penelitian ini membahas ide-ide kedua algoritma dan bagaimana mereka dapat meningkatkan efisiensi sistem dengan mengintegrasikannya.

Corresponding Author. Email bozzaryarief20@gmail.com ^{1}.

© E-ISSN: 2580-1643.

Copyright © 2024 by the authors of this article. Published by Lembaga Otonom Lembaga Informasi dan Riset Indonesia (KITA INFO dan RISET). This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License. 



Association for
Computing Machinery

ACM Computing Classification System (CCS)

EBSCOhost

Communication and Mass Media Complete (CMMC)

1. Latar Belakang

Teknologi merupakan media manusia untuk menyelesaikan permasalahan dan pemanfaatan untuk keberlangsungan hidup manusia. Teknologi di iringi oleh pembaruan sesuai perkembangan zaman dan adaptasi untuk setiap masalah yang di hadapi. Salah satu pemanfaatan teknologi yaitu di bidang pendidikan di dunia universitas. Sebagai salah satu syarat penilaian akhir dan penentuan kelulusan mahasiswa adalah mengerjakakan skripsi. Skripsi adalah karya tulis ilmiah yang didukung oleh fakta-fakta yang tepat dan keahlian khusus. Menggunakan bukti-bukti yang akurat, dan fakta-fakta ini disatukan untuk menciptakan pemecah masalah yang luas. Dalam praktiknya kegiatan skripsi membutuhkan sistem sehingga membuat kegiatan skripsi ini berjalan dengan efektif dan efisien. Mulai dari bimbingan, dosen, serta pengujian dan sidang akhir. Oleh karena itu penulis mendefinisikan masalah yang ada dan di kemas dalam judul “Optimalisasi Aplikasi Pengendalian Skripsi Menggunakan Algoritma *Dynamic Priority Scheduling* dan *Sequential Search*”. Skripsi juga merupakan salah satu tugas ilmiah yang harus diselesaikan oleh mahasiswa, terutama yang sudah duduk di semester VIII. Dalam praktiknya kegiatan skripsi membutuhkan usaha yang lebih, seperti bimbingan serta revisi jika ada dalam proses bimbingan. Sebagai media diskusi terkait skripsi yang di tulis mahasiswa kepada dosen membutuhkan salinan tulisan dalam bentuk cetak yang berarti mahasiswa harus mengeluarkan biaya dalam mencetak semua halaman untuk di diskusikan kepada dosen. Hal ini menjadi yang menjadi masalah utama dalam penelitian ini, selain pemborosan biaya dan juga pemborosan kertas. Pada kesempatan ini penelitian yang di lakukan berfokus dalam pembuatan aplikasi skripsi. Hal ini di dukung terkait dengan masalah efisiensi dan keefektifan kegiatan skripsi.

Dalam kegiatan skripsi ini tidak luput dengan kesalahan manusia salah satunya lalai dari tanggung jawab baik dosen dan mahasiswa. Umumnya banyak masalah terjadi di kedua pihak, salah satu contoh ialah dengan mahasiswa tidak bimbingan. Untuk contoh yang satu ini di butuhkan monitoring bimbingan sehingga aktivitas mahasiswa terkait bimbingan dapat terkendali, begitupun juga dengan pihak dosen. Selain monitor bimbingan rencana fitur

yang akan di bangun yaitu fitur pengumpulan *softcopy* file skripsi dalam hal revisi. Fitur ini di tujukan untuk mengurangi kertas dan menghemat biaya pengeluaran mahasiswa untuk setiap kertas yang di cetak. Adapun aplikasi skripsi yang eksisting sekarang mempunyai kekurangan yaitu fitur bimbingan untuk mahasiswa belum tersedia. Aplikasi ini yang akan di bangun dengan menggunakan *framework* Codeigniter 3, Javascript dan MySQL, aplikasi skripsi ini juga mengadaptasi algoritma *sequential* dalam pencarian data dalam aplikasi skripsi. Data dicari secara berurutan menggunakan Algoritma *Sequential Search*, baik dari awal hingga akhir atau dari depan ke belakang. berdasarkan kunci pencarian yang digunakan. Algoritma *Sequential Search* ini memiliki keuntungan untuk menemukan data dengan cepat jika informasi yang Anda cari berada di bagian depan secara teknis proses tersebut memeriksa setiap karakter yang di inputkan dan mencocokkan data karakter pertama sesuai dengan data yang tersedia. Terlepas dari manfaat *Algoritma Sequential*, ada juga kekurangan dari metode ini yaitu, jika data yang di butuhkan berada di bagian paling akhir atau di belakang, proses pencarian akan memakan waktu lama, dan jika larik berisi banyak data, beban komputer akan bertambah [1]. Dalam penelitian dan pengembangan aplikasi ini mengadaptasi metode *agile*, pendekatan metode *agile* terbilang sangat flexible terkait perubahan fitur, rancangan dan sistem yang akan di bangun [2]. Dalam pembuatan aplikasi skripsi ini juga memakai algoritma *Dynamic Priority Schedule* berfungsi untuk mahasiswa ketika mengajukan judul dan memilih bidang konsentrasi dan mata kuliah yang relevan sebagai hasil output nya adalah ketika mahasiswa memilih salah satu konsentrasi pada proses pengajuan pembimbing, di dalam list data maka pembimbing yang tersedia akan terfilter pada masing masing bidang konsentrasinya

Dalam uji coba aplikasi ini, peneliti menggunakan aplikasi katalon sebagai *testing* aplikasi yang di buat. Alasan peneliti menggunakan katalon karena dengan tingkatan penggunaan Katalon yang bebas biaya, individu dan tim kecil dapat mengeksplorasi dan mempelajari platform tanpa dibatasi oleh periode uji coba. Tester dapat beradaptasi dan meningkatkan prosedur kualitas dengan memanfaatkan solusi hemat biaya dan waktu. Pendekatan *low-code* katalon, yang menarik bagi pengguna non-teknis, memungkinkan tim untuk merasionalisasi solusi spesialis yang terpisah

ke dalam platform *all-in-one* [3]. Harapan besar, aplikasi ini dapat diimplementasi untuk fakultas serta universitas lain yang masih menggunakan sistem *manual* dalam kegiatan skripsi.

Identifikasi Masalah

Ada beberapa hal yang menjadi masalah atau kekhawatiran terkait kegiatan skripsi, mulai dari aplikasi serta alurnya

- 1) Aplikasi yang eksisting atau aplikasi yang sudah berjalan, memiliki kekurangan dalam fitur pemilihan dosen pembimbing.
- 2) Aplikasi skripsi membutuhkan algoritma yang tepat, cepat dan mudah untuk kebutuhan pencarian data di aplikasi skripsi.
- 3) Kegiatan skripsi yang masih manual akan memperlambat proses keseluruhan kegiatan skripsi, dan juga tidak terkendalinya mahasiswa dan dosen dalam proses bimbingan.

Algoritma Sequential Search

Sequential Search juga dikenal sebagai pencarian linier, adalah model pencarian paling dasar yang dapat digunakan dengan pengumpulan data tertentu. Perihal proses mencari data yang diinginkan secara berurutan dari awal daftar hingga akhir. Dari data pertama hingga data terakhir, pencarian dilakukan selangkah demi selangkah. Hasilnya, dengan menelusuri daftar dari depan sampai bagian yang relevan ditemukan, nilai yang dicari akan ditemukan. Mengambil data menjadi sangat mudah dengan cara ini. Pengaturan dan pengorganisasian dapat digunakan untuk mencapai hal ini. Data yang berhasil dicocokkan akan ditampilkan jika hasilnya sama, data yang berhasil dicocokkan akan ditampilkan jika tidak, nilai kosong akan ditampilkan.

Agile

Agile yang menerapkan DevOps membutuhkan kolaborasi dan pertukaran informasi di antara para praktisi di berbagai posisi untuk menghasilkan nilai. Cara para praktisi bekerja sama dipengaruhi oleh adopsi pendekatan pengembangan baru, yang memerlukan hubungan yang lebih erat antara pengembangan perangkat lunak dan strategi bisnis. Sangat penting untuk menentukan jenis manajemen informasi, penjualan, dan pengembangan yang diperlukan untuk merencanakan, mengevaluasi fitur, dan menyeimbangkan ekspektasi pemangku kepentingan.

Codeigniter 3

Kerangka kerja pengembangan aplikasi, atau toolkit, untuk pembangun situs web PHP ialah *CodeIgniter*. Tujuannya adalah untuk memfasilitasi pengembangan proyek secara signifikan lebih cepat daripada membuat kode dari awal dengan menawarkan *library* yang komprehensif untuk operasi yang sering diperlukan bersama dengan *interface* yang mudah digunakan dan struktur logis untuk mengakses *library* ini. *CodeIgniter 3* mengurangi jumlah kode yang diperlukan untuk aktivitas tertentu, sehingga dapat berkonsentrasi penuh pada proyek.

Dynamic Priority Scheduling

Algoritma penjadwalan prioritas adalah algoritma penjadwalan yang memprioritaskan setiap tugas sesuai dengan tingkat kepentingannya. Ada dua jenis algoritma pendekatan yaitu, pendekatan statis dan pendekatan dinamis. Perbedaan utamanya adalah bahwa dalam sistem statis, prioritas yang telah ditentukan tidak dapat diubah. sementara dengan pendekatan dinamis, tidak demikian halnya. Proses dengan prioritas yang lebih tinggi, atau kegiatan, akan diproses terlebih dahulu dalam metode ini. Di sisi lain, lebih banyak proses atau aktivitas akan tersedia jika semua proses memiliki prioritas yang sama. Siapa cepat dia dapat.

2. Metode Penelitian

Lokasi Penelitian

Pada lokasi penelitian yang akan di lakukan berada di Gedung Universitas Nasional, Jl. Sawo Manila No.61, RT.14/RW.7, Pejaten Bar., Ps. Minggu, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12520

Waktu Penelitian

Waktu yang di butuhkan untuk menyelesaikan proses penelitian dan pengembangan aplikasi adalah 20 minggu atau 5 bulan.

Penentuan Subjek Penelitian

Sistem aplikasi skripsi ini di bangun dari beberapa sumber seperti refrensi jurnal penelitian, komunitas Web lainnya dan metode yang di pakai serta algoritma yang di gunakan untuk membangun fitur yang ada dalam web ini. Penggunaannya Web ini di tujuan untuk mahasiswa dan dosen yang berkepentingan dalam kegiatan skripsi ini. Di harapkan dengan di

bangunnya Web ini aktivitas kegiatan skripsi di Universitas Nasional berjalan dengan efektif dan efisien.

Fokus Penelitian

Fokus penelitian ini adalah melakukan pembaruan web aplikasi skripsi yang sudah ada, Penerapan fitur *tracking progress* bimbingan menjadi fokus utama dalam penelitian ini, dan implementasi algoritma *dynamic priority* terkait pemilihan dosen pembimbing yang relevan dengan topik skripsi mahasiswa. Penerapan algoritma Sequential terkait pencarian data mahasiswa dan dosen. Mengimplementasikan automation testing menggunakan katalon sebagai UI test pada aplikasi skripsi ini.

Teknik Pengumpulan Data

Observasi

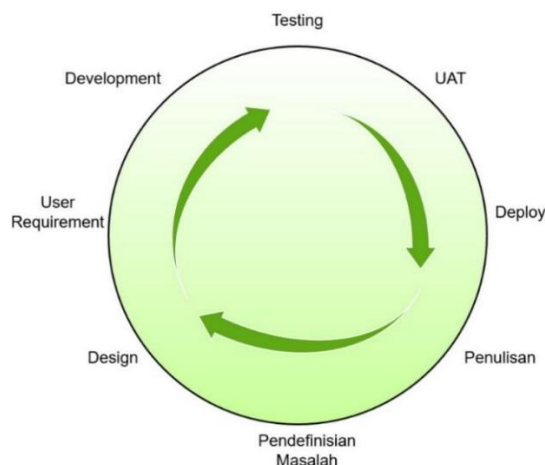
Pada fase ini, peneliti melakukan observasi ketempat penelitian yaitu Universitas Nasional, peneliti meninjau terkait proses kegiatan skripsi, serta mencari data untuk kebutuhan penelitian ini.

Kuisioner

Sebagai awalan untuk observasi peneliti melakukan penyebaran kuisioner kepada Mahasiswa dan dosen yang berada di Universitas Nasional. Ada 5 pertanyaan yang di ajukan seputar aplikasi skripsi dan sistem yang digunakan.

Arsitektur Desain Penelitian

Arsitektur ini di buat untuk memudahkan proses development aplikasi Skripsi yang akan di bangun.



Gambar 1. Diagram Siklus Hidup Pengembangan Perangkat Lunak (*Software Development Life Cycle* atau *SDLC*)

Penulisan dan penelitian, di tahap ini di lakukannya penelitian mulai dari judul latar belakang dan sebagainya, tahap ini beriringan dengan pendefinisian masalah.

1) Pendefinisian masalah

Pendefinisian masalah merupakan tahap yang paling penting dari proses *development* aplikasi hingga penelitian, tahap ini mendefinisikan masalah yang terjadi untuk acuan dalam implementasi aplikasi. *Tech Stack* di definisikan di tahap ini.

2) Design

Setelah terdefinisi masalahnya, selanjutnya adalah design. Membuat *design* yang *user friendly* dan fungsionalitas nya tidak membingungkan para *user* saat menggunakannya.

3) User Requirement

Tahap *user requirement* adalah tahap dimana peneliti mewancarai pihak yang berkepentingan terkait fitur, *design*, dan flow dari Web yang di bangun.

4) Development

Tahap development di lakukan setelah *user requirement* terpenuhi, sehingga dapat membangun fitur sesuai dengan *user requirement*.

5) Testing

Setelah tahap *development* selesai lanjut ke tahap testing, tahap testing ini terpenuhi jika dari suatu *test case* yang di buat di katalon studio berhasil atau passed, jika tidak terpenuhi kembali lagi ke tahap *development*.

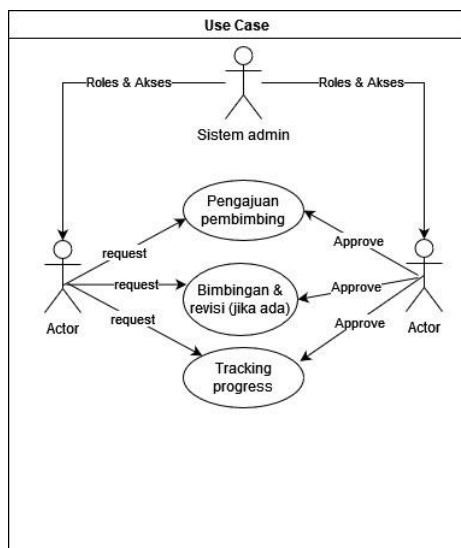
6) UAT (User Acceptance Test)

Setelah berhasil dari Testing adalah di lakukan nya UAT atau *User Acceptance Test* tahap ini di lakukan, untuk melihat apakah aplikasi dapat di terima oleh para user. Tahap ini juga merupakan penilaian dari user terkait Web yang di bangun.

7) Deploy

Setelah tahap UAT selesai, Web dalam keadaan *deploy*, atau siap di hosting dan di sebarakan di lingkungan penelitian.

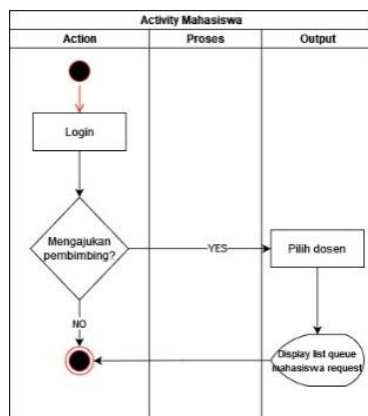
Rencana Sistem Use Case



Gambar 2. Use Case Diagram

Berikut berupa *Use case* sistem aplikasi skripsi yang akan di buat. Ada 3 roles yaitu dosen, mahasiswa, dan system administrator. Pada fitur utama terdapat pengajuan pembimbing, bimbingan dan revisi (jika ada), dan tracking progress. Alurnya adalah system administrator memberikan hak akses kepada dosen dan mahasiswa terkait fitur yang akan di gunakan. Pada tiap fitur yang ada mahasiswa melakukan *action request* kepada dosen, dan dosen akan menerima *request* dari mahasiswa dan dapat di *approve*.

Activity Diagram

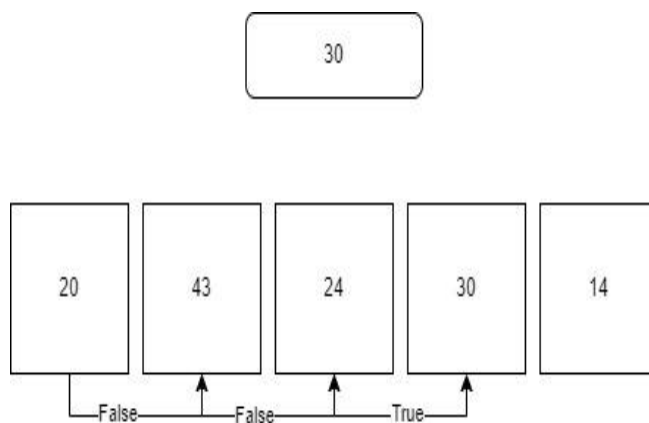


Gambar 3. Activity Diagram

Pada menu pengajuan pembimbing mahasiswa dapat memilih dosen, terdapat tombol *Request* pembimbing baik dosen 1 dan dosen 2, setelah mahasiswa melakukan pemilihan dosen *queue* atau antrian list dosen yang di ajukan terlihat, dan mempunyai status "Sedang di proses".

3. Hasil dan Pembahasan

Pengujian Algoritma Sequential Search



Gambar 4. Pengujian Algoritma Sequential Search

Algoritma *Sequence* bekerja dengan cara mengecek kondisi dari data array yang telah diatur, sebagai contoh gambar di atas adalah ketika ingin mencari data "30" maka Algoritma *Sequence* langsung mengeksekusinya. dalam kondisi yang bernilai salah, maka Algoritma berkesinambungan mengecek indeks array berikutnya hingga kondisi *True*, Algoritma *Sequence* berhenti dan mengembalikan nilai indeks tersebut. Contoh di atas adalah kondisi sederhana dari Algoritma *Sequence*.

Pengujian Algoritma Dynamic Priority Scheduling

Dalam pengujian Algoritma *Dynamic Priority Scheduling* yang di terapkan pada aplikasi apsta di tentukan sesuai dengan konsentrasi bidang skripsi yang di ambil oleh mahasiswa. Setiap mata kuliah berisi data dosen yang berkorelasi pada skill set dosen. Maka pada rekomendasi setiap dosen adalah bernilai 100. Berikut penjelasan Algoritma *Dynamic Priority Scheduling* dengan tabel di bawah ini.

Tabel 1. Penjelasan Algoritma *Dynamic Priority Scheduling*

No	Value	Dosen	Mata kuliah
1	100	Dosen [N]	Rekayasa Perangkat Lunak
2	100	Dosen [N]	Pemrograman Web
3	100	Dosen [N]	Pengolahan citra
4	100	Dosen [N]	<i>Mobile Programming</i>
5	100	Dosen [N]	<i>Cloud Computing</i>
6	100	Dosen [N]	<i>Data Science</i>
7	100	Down [N]	<i>Deep Learning</i>
8	100	Down [N]	Kecerdasan Artifisial
9	100	Dosen [N]	Manajemen Proyek
10	100	Dosen [N]	Pemrograman Game
11	100	Dosen [N]	Perancangan Analisis Algoritma
12	100	Dosen [N]	<i>Internet Of Things</i>

Pada penerapan ini parameter penilaian dan ketentuan dosen adalah skill set dosen yang ahli dalam bidangnya, tidak ada pengaruh dalam variable lain maka dapat di rumuskan:

$$Priority = 100 * Mata\ kuliah\ Dosen$$

Hasil Data dan Observasi

Dari Hasil data observasi dan wawancara yang dilakukan pada lokasi Universitas Nasional, peneliti membagikan sebuah kuisioner dengan 5 pertanyaan yang sudah di siapkan, kuisioner tersebut membahas tentang aplikasi skripsi peneliti untuk di implementasikan di Universitas Nasional.

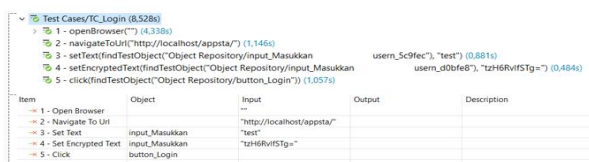
Tabel 2. Kuisioner

No	Seberapa efektifnya kah sistem aplikasi skripsi digunakan sekarang?	Apakah proses pengajuan judul, dan bimbingan berjalan dragan lancar dan cepat?	Apakah informasi mengenai skripsi di ditampilkan dengan tepat?	Apakah perlu adanya <i>Improvement</i> pada aplikasi skripsi ini?	Seberapa puas dengan proses kegiatan skripsi melalui aplikasi yang sudah eksisting
1	4	4	5	4	5
2	3	4	3	3	5
3	4	5	4	5	4
4	4	3	5	3	4
5	4	5	4	5	5
6	5	4	5	4	5
7	4	3	5	5	5
8	5	5	5	5	5
9	4	5	5	5	5
10	5	5	5	3	5

Bedasarkan tabel tersebut 5 pertanyaan dengan hasil nilai di kalikan dengan 2 maka dapat disimpulkan nilai untuk pertanyaan pertama adalah 84, pertanyaan kedua adalah 86, pertanyaan ketiga adalah 92, pertanyaan keempat 84 dan pertanyaan kelima adalah 96, nilai tersebut menunjukan seberapa puas pengguna dalam menggunakan aplikasi ini.

Testing

Testing merupakan bagian yang paling penting untuk memastikan produk atau aplikasi yang di buat berjalan dengan sesuai apa yang di harapkan, testing juga guna memberitahu kepada pengembang jika terdapat *bug minor* atau *major* untuk di *fixing*. Berikut di bawah ini merupakan beberapa testcase dan hasil pengujian pada fitur yang di buat.



Item	Object	Input	Output	Description
1 - Open Browser				
2 - Navigate To Url		"http://localhost/appsta"		
3 - Set Text	input_Masukkan	"test"		
4 - Set Encrypted Text	input_Masukkan	"tzh6RvftStg="		
5 - Click	button_Login			

Gambar 5. Test Case Login

Pada test case login, test di buat dengan sesuai ekspetasi output yang di ingin kan pengguna, pada test case ini tahap pertama adalah membuka browser, lalu mengarah ke URL localhost/appsta, menginput username dan password yang sudah terdaftar dan menekan tombol login. Hasilnya pada fitur login dari beberapa rangkaian script yang di jalankan telah berhasil melalui proses pengujian.

4. Kesimpulan

Penerapan algoritma *Dynamic Priority Scheduling* pada aplikasi pengendalian skripsi dapat meningkatkan efisiensi pengelolaan tugas dan sumber daya, dengan memprioritaskan tugas berdasarkan tingkat urgensi dan pentingnya untuk memastikan penyelesaian skripsi dilakukan secara efektif. Selain itu, penggunaan metode Sequential Search yang dioptimalkan dapat meningkatkan kecepatan dan akurasi dalam pencarian informasi yang dibutuhkan. Algoritma *Dynamic Priority Scheduling* juga membantu dalam pengelolaan waktu penyelesaian skripsi dengan menangani tugas-tugas berurgensi tinggi lebih awal, sehingga meminimalkan risiko keterlambatan dan meningkatkan produktivitas. Implementasi algoritma ini juga meningkatkan kemudahan penggunaan aplikasi, memberikan pengguna akses cepat terhadap informasi yang dibutuhkan, serta proses yang lebih responsif dan lancar. Rekomendasi pengembangan masa depan mencakup evaluasi performa mendalam terhadap algoritma yang digunakan, perbandingan dengan metode pengendalian skripsi lainnya, dan integrasi algoritma dalam pemilihan dosen pembimbing skripsi untuk optimasi penempatan mahasiswa.

5. Daftar Pustaka

Pengabdian Masyarakat Global, 2(2), 78-87. DOI: <https://doi.org/10.30640/cakrawala.v2i2.100>.

- [2] Sukma, C. D., & Hasya, N. A. (2023). Tingkat Efikasi Diri Dalam Menyusun Skripsi Pada Mahasiswa Bimbingan Dan Konseling Universitas Muhammadiyah Prof. Dr. Hamka Angkatan 2018. *Jurnal Mahasiswa BK An-Nur: Berbeda, Bermakna, Mulia*, 9(1), 177-185. DOI: <http://dx.doi.org/10.31602/jmbkan.v9i1.10220>.
- [3] Aminnur, M., Pakpahan, R. S., Alfarizi, D. G., Apriana, D., Rahmat, S. M., Fauzi, A., & Rosyani, P. (2023). Implementasi Metode Sequential Search Untuk Pengelolaan Data Barang Pada Sistem Aplikasi Sikilat Cargo. *LOGIC: Jurnal Ilmu Komputer dan Pendidikan*, 1(2), 283-287.
- [4] Bomström, H., Kelanti, M., Annanperä, E., Liukkunen, K., Kilamo, T., Sievi-Korte, O., & Systä, K. (2023). Information needs and presentation in agile software development. *Information and Software Technology*, 162, 107265.
- [5] Lasriana, L., & Gunaryati, A. (2022). Sistem Informasi Apotek Berbasis Web Menggunakan Algoritma Sequential Search Dan Selection Sort. *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 7(2), 392-401.
- [6] Samli, R., & Orman, Z. (2023). A comprehensive overview of web-based automated testing tools. *İleri Mühendislik Çalışmaları ve Teknolojileri Dergisi*, 4(1), 13-28.
- [7] Ritonga, P., Masrizal, M., & Muti'ah, R. (2023). Design of a Drug Inventory Information System at Janji Primary Health Center Using a MySQL Database. *Budapest International Research and Critics Institute-Journal (BIRCI-Journal)*, 6(3), 1705-1714. DOI: <https://doi.org/10.33258/birci.v6i3.7696>.

- [1] Bafadal, R. (2023). Peningkatan Kualitas Penulisan Skripsi Bagi Mahasiswa STIA Muhammadiyah Selong. *Cakrawala: Jurnal*