

---

# Penerapan Algoritma Dynamic Programming Dalam Pencarian Judul Skripsi

Gunawan<sup>1</sup>, Yuni Andriani<sup>2</sup>

<sup>1,2</sup> Program Studi Sistem Informasi, Universitas Muhammadiyah Bengkulu

email: <sup>1</sup>[gunawan@umb.ac.id](mailto:gunawan@umb.ac.id), <sup>2</sup>[yuniandriani@gmail.com](mailto:yuniandriani@gmail.com)

Email Penulis Korespondensi: [gunawan@umb.ac.id](mailto:gunawan@umb.ac.id)

---

## Article History:

Received Dec 12<sup>th</sup>, 2022

Revised Jan 03<sup>th</sup>, 2023

Accepted Jan 9<sup>th</sup>, 2023

## Abstrak

Dalam proses pencarian judul skripsi, terdapat permasalahan yang timbul yakni judul skripsi yang sama dari tahun sebelumnya atau double atau terjadi dua kali penginputan judul, tidak adanya suatu pembagian judul skripsi berdasarkan kategori-kategori masalah yang dibahas. Oleh karena itu, penulis ingin membenahi permasalahan tersebut dengan menerapkan Algoritma Dynamic Programming untuk melakukan pencarian judul Skripsi. Algoritma Dynamic Programming menyelesaikan suatu masalah dengan mengkombinasikan solusi dengan membagi menjadi sub-masalah yang independen, kemudian menyelesaikan submasalah secara terstruktur dan mengkombinasikan solusi tersebut untuk menyelesaikan masalah utama, yakni dalam hal ini masalah yang dimaksud adalah proses pencarian judul skripsi. Dengan penerapan Algoritma Dynamic Programming ini, agar dapat menghindari masalah yang timbul ketika melakukan proses pencarian judul skripsi. Dalam penerapan algoritma Dynamic Programming, penulis menentukan variabel pencarian untuk mempersempit pencarian judul skripsi, yaitu Kategori Skripsi, Judul Skripsi, Tahun, dan Abstrak.

**Kata Kunci :** Penerapan, Algoritma Dynamic Programming, Pencarian, Judul Skripsi.

---

## Abstract

*In the process of searching for the thesis title, there are problems that arise, namely the thesis title being the same from the previous year or double or the title inputting twice, there is no division of thesis titles based on the categories of problems discussed. Therefore, the author wants to fix this problem by applying the Dynamic Programming Algorithm to search for thesis titles. The Dynamic Programming Algorithm solves a problem by combining solutions by dividing into independent sub-problems, then solving the sub-problems in a structured way and combining these solutions to solve the main problem, namely in this case the problem in question is the process of finding the thesis title. With the implementation of this Dynamic Programming Algorithm, in order to avoid problems that arise when searching for thesis titles. In applying the Dynamic Programming algorithm, the writer determines the search variable to narrow down the search for thesis title, namely Thesis Category, Thesis Title, Year, and Abstract.*

**Keywords:** Application, Dynamic Programming Algorithm, Search, Thesis Title.

---

## 1. PENDAHULUAN

Penerapan teknologi yang sedang berkembang pesat hingga saat ini telah banyak digunakan oleh semua instansi pemerintahan, terutama dalam bidang pendidikan. Teknologi mampu menghasilkan alternatif pemecahan masalah yang ada dengan memberikan kemudahan atau bahkan menghasilkan produk yang bisa dimanfaatkan oleh semua pengguna. Misalnya saja permasalahan kampus seperti pengajuan judul skripsi mahasiswa. Masalah yang sering muncul yaitu seringnya penggunaan judul yang sama pada tahun-tahun berikutnya. Sehingga penulis berinisiatif untuk melakukan penelitian untuk dapat memberikan alternatif solusi untuk permasalahan tersebut dengan cara menghasilkan sebuah aplikasi pencarian judul skripsi, harapannya aplikasi ini nanti bisa dimanfaatkan oleh mahasiswa. Namun, aplikasi yang dibuat nantinya memerlukan algoritma. Algoritma yang cocok untuk aplikasi ini salah satunya algoritma *dynamic*

*programming*, karena algoritma ini lebih mudah digunakan seperti penelitian yang telah dilakukan sebelumnya. Pada penelitian pencarian rute terpendek pada game[1], peningkatan kapasitas gudang[2], mudah diterapkan pada aplikasi android dan efisien coding[3], mampu menghasilkan rute tercepat pengangkutan sampah[4], serta juga mampu di terapkan pada aplikasi sistem pendukung keputusan[5]. Dalam penelitian ini penulis akan menggunakan perancangan aplikasi[6] dan pengujian aplikasi dengan menggunakan pengujian beta. Pengujian beta telah digunakan pada penelitian sebelumnya diantaranya: pengujian aplikasi penerimaan siswa baru[7], pengujian sistem informasi *e-commerce*[8], pengujian aplikasi penjualan undangan pernikahan[9], pengujian pada game edukasi[10] dan pengujian dalam penggunaan website[11] dan pengujian pada aplikasi *arbook* media pembelajaran beladiri nusantara pencak silat[12].

## 2. METODOLOGI PENELITIAN

### 2.1 Metode Pengumpulan Data

Metode pengumpulan data digunakan untuk memperoleh data yang dapat mendukung permasalahan yang akan dibahas. Sehubungan dengan hal ini maka digunakan metode pengumpulan data yang meliputi :

1) Studi Pustaka

Studi pustaka adalah suatu metode pengumpulan data yang diambil dari perpustakaan atau instansi yang berupa karya ilmiah, jurnal, buku-buku serta dari internet yang berhubungan dengan penulisan ini. Tujuan dari studi pustaka ini adalah untuk mendalami dan memperoleh keterangan yang lengkap terhadap obyek yang diteliti.

2) Studi Praktikum

Studi praktikum adalah menguji coba teori algoritma Dynamic Programming dalam pencarian judul skripsi. Pada studi ini dilakukan tahapan-tahapan, yaitu analisis, implementasi, perancangan, coding, testing, dan evaluasi.

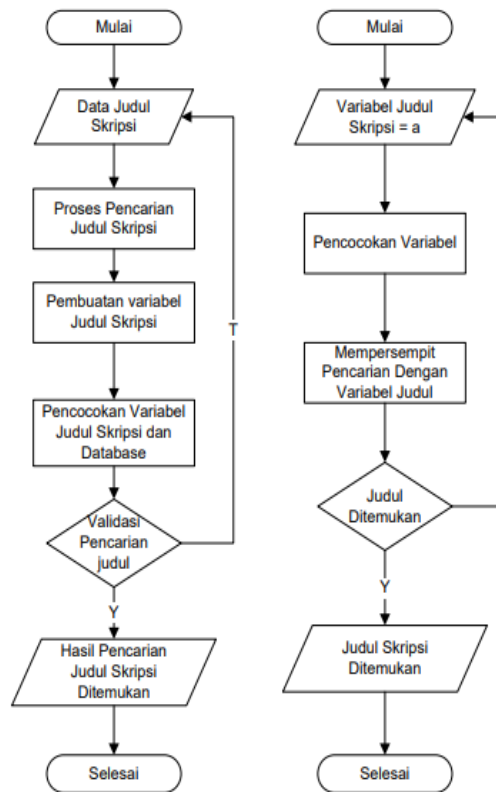
### 2.2 Perancangan Sistem

Dalam perancangan algoritma digunakan pendekatan terstruktur (*structured approach*). Pendekatan ini dilakukan dengan cara memecah-mecah suatu masalah yang besar dan rumit menjadi beberapa masalah yang lebih kecil dalam bentuk modul-modul sehingga menjadi cukup mudah ditangani. Suatu modul adalah kumpulan dari instruksi yang melakukan suatu fungsi dasar tertentu yang bersifat *independent*. Setiap modul harus mewakili suatu bagian masalah inti tertentu dari keseluruhan masalah.

Salah satu teknik pendekatan perancangan terstruktur yang paling terkenal adalah *top-down design*. Akan lebih realistis bila rancangan suatu program dimulai dari atas ke bawah (*top down*), yaitu dengan menentukan kebutuhan secara umum terlebih dahulu, memecahnya menjadi modul-modul yang lebih sederhana lagi dan seterusnya. Dengan pendekatan terstruktur diperoleh beberapa keuntungan dibandingkan dengan pendekatan secara tradisional, yaitu :

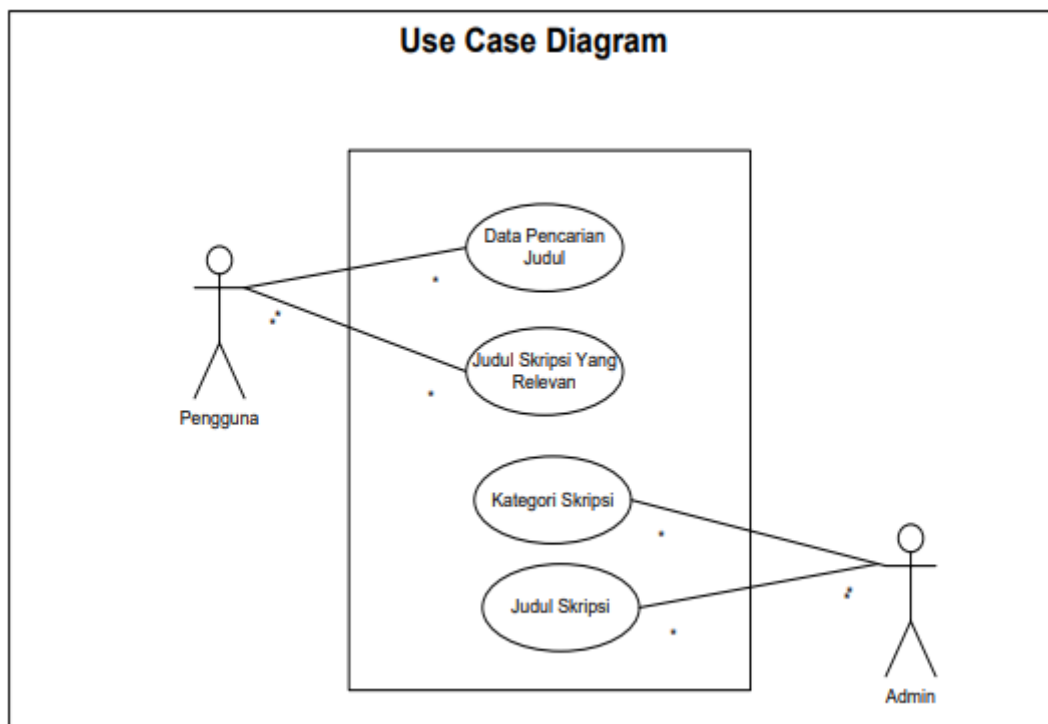
- 1) Kemampuan yang lebih untuk mempertemukan dengan kebutuhan pemakai secara keseluruhan.
- 2) Waktu dan biaya pembuatan perangkat lunak dapat lebih mudah diperhitungkan.
- 3) Memudahkan pengawasan dan pengendalian perangkat lunak.
- 4) Memperkecil kesalahan dan permasalahan yang mungkin timbul selama pembuatan perangkat lunak.
- 5) Perencanaan ini bersifat fleksibel, karena tidak dipaksakan dari modul-modul, tetapi yang umum terlebih dahulu.
- 6) Merupakan pendekatan yang standard untuk perancangan/pembuatan program.

Untuk penerapan algoritma Dynamic Programming dapat dilihat pada gambar 1 dibawah ini.



Gambar 1. Flowchart Algoritma Dynamic Programming

Untuk rancangan use case diagram dapat dilihat pada gambar 2 dibawah ini.



Gambar 2. Use Case Diagram

## 2.3 Metode Pengujian Sistem

Pengujian beta merupakan bagian dari pengujian non fungsional (khasanah) yang biasa digunakan dalam pengujian aplikasi(gun). Pengujian beta dilakukan dengan menyebarkan kuesioner yang diberikan kepada responden. Skala likert dirancang untuk menyakinkan responden menjawab dalam berbagai tingkatan dari setiap butir pertanyaan yang terdapat pada kuesioner

## 3. HASIL DAN PEMBAHASAN

### 3.1 Hasil

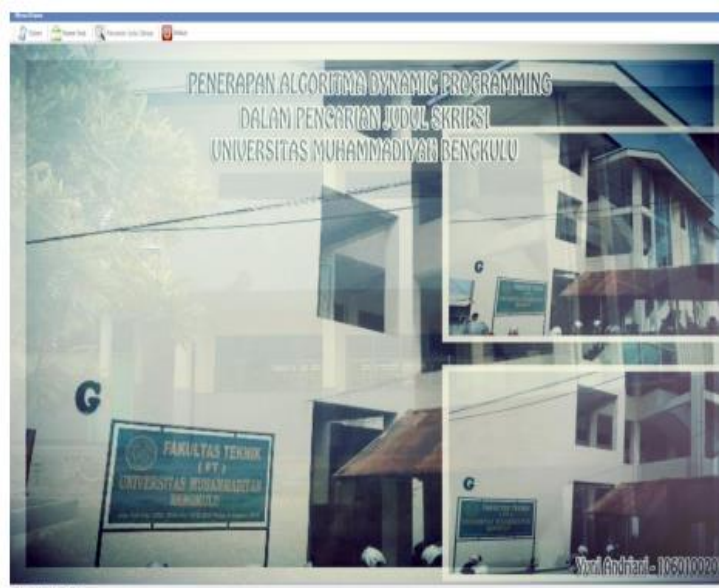
Penelitian ini menghasilkan sebuah aplikasi pencarian judul skripsi yang diterapkan pada fakultas teknik Universitas Muhammadiyah Bengkulu dengan menggunakan aplikasi Visual Studio, adapun hasil tampilan aplikasi sebagai berikut:

- 1) Halaman menu login, dapat dilihat pada gambar 3, terdiri dari *username* dan *password*.



Gambar 3. *Login* Aplikasi

- 2) Halaman menu utama menggunakan pemrograman desktop sesuai pada gambar 4 berikut.



Gambar 4. Menu Utama Aplikasi

- 3) Halaman menu administrator untuk input kategori skripsi sesuai pada gambar 5 dan input judul skripsi sesuai pada gambar 6.

Kategori	ID
Jaringan Komputer	003
Sistem Pakar	001
Sistem Pendukung Keputusan	002

Gambar 5. Input Kategori Skripsi

ID_Judul	ID_Kategori	Kategori Skripsi	Judul Skripsi	Penulis	Tahun	Abstrak	Dosen Pembimbing
001	001	Sistem Pakar	Sistem Pakar...	Sudarmadi...	2011	Sistem pak...	T. RG. Gunt...
002	003	Jaringan Ko...	Perancangan...	M. Widodo...	2011	Jaringan M...	T. R. Hakim...
003	003	Jaringan Ko...	Aplikasi Wla...	Andika Dwi...	2011	Wake On L...	T. RG. Gunt...

Gambar 6. Input Judul Skripsi

- 4) Halaman menu pengguna, tampilannya sesuai dengan gambar 7 berikut.

Kategori Skripsi	Judul Skripsi	Penulis	Tahun	Abstrak	Dosen Pembimbing
------------------	---------------	---------	-------	---------	------------------

Gambar 7. Menu Pengguna

## 3.2 Pembahasan

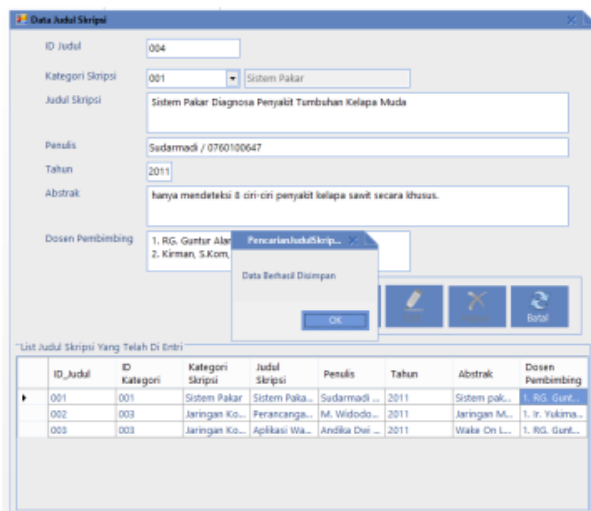
Dalam penerapan algoritma *Dynamic Programming*, penulis menentukan variabel pencarian untuk mempersempit pencarian judul skripsi, yaitu :

- 1) Kategori Skripsi
- 2) Judul Skripsi
- 3) Tahun
- 4) Abstrak

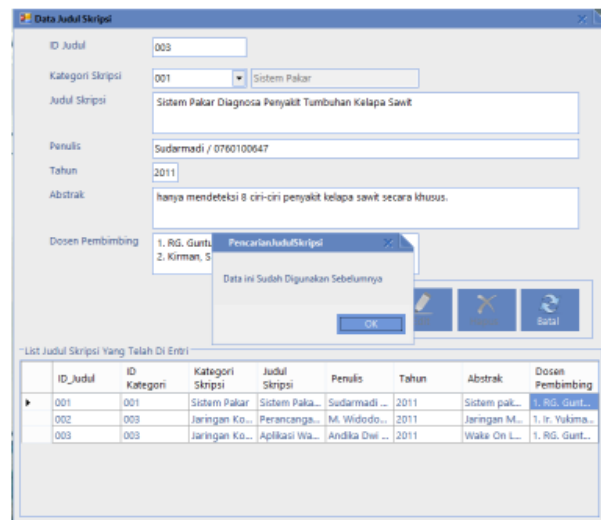
Berdasarkan 4 (empat) variabel tersebut, ditetapkan kriteria pencarian judul skripsi antara lain, tampilannya dapat dilihat pada gambar 8 dan 9.

- 1) Jika kategori skripsi aktif dan judul skripsi, tahun, abstrak tidak aktif, maka pencarian hanya berdasarkan variabel kategori skripsi.
- 2) Jika judul skripsi aktif dan kategori skripsi, tahun, abstrak tidak aktif, maka pencarian hanya berdasarkan variabel judul skripsi.
- 3) Jika tahun aktif dan kategori skripsi, judul skripsi, abstrak tidak aktif, maka pencarian hanya berdasarkan variabel tahun.
- 4) Jika abstrak aktif dan kategori skripsi, judul skripsi, tahun tidak aktif, maka pencarian hanya berdasarkan variabel abstrak.
- 5) Jika kategori skripsi, judul skripsi aktif dan tahun, abstrak tidak aktif, maka pencarian berdasarkan 2 variabel yaitu kategori skripsi dan judul skripsi.

- 6) Jika kategori skripsi, tahun aktif dan judul skripsi, abstrak tidak aktif, maka pencarian berdasarkan 2 variabel yaitu kategori skripsi dan tahun.
- 7) Jika kategori skripsi, abstrak aktif dan tahun, judul skripsi tidak aktif, maka pencarian berdasarkan 2 variabel yaitu kategori skripsi dan tahun.
- 8) Jika judul skripsi, tahun aktif dan kategori skripsi, abstrak tidak aktif, maka pencarian berdasarkan 2 variabel yaitu judul skripsi dan tahun.
- 9) Jika judul skripsi, abstrak aktif dan kategori skripsi, tahun tidak aktif, maka pencarian berdasarkan 2 variabel yaitu judul skripsi dan abstrak.
- 10) Jika abstrak, tahun aktif dan judul skripsi, kategori skripsi tidak aktif, maka pencarian berdasarkan 2 variabel yaitu abstrak dan tahun.
- 11) Jika kategori skripsi, judul skripsi, tahun aktif dan abstrak tidak aktif, maka pencarian berdasarkan 3 variabel yaitu kategori skripsi, judul skripsi dan tahun.
- 12) Jika kategori skripsi, judul skripsi, abstrak aktif dan tahun tidak aktif, maka pencarian berdasarkan 3 variabel yaitu kategori skripsi, judul skripsi dan abstrak.
- 13) Jika kategori skripsi, tahun, abstrak aktif dan judul skripsi tidak aktif, maka pencarian berdasarkan 3 variabel yaitu kategori skripsi, tahun dan abstrak.
- 14) Jika judul skripsi, abstrak, tahun aktif dan kategori skripsi tidak aktif, maka pencarian berdasarkan 3 variabel yaitu judul skripsi, abstrak dan tahun.
- 15) Jika kategori skripsi, judul skripsi, tahun, abstrak aktif, maka pencarian berdasarkan 4 variabel yaitu kategori skripsi, judul skripsi, tahun, dan abstrak.



Gambar 8. Jika Judul Belum Ada



Gambar 9. Jika Judul Sudah Ada

### 3.3 Pengujian aplikasi

Pada sistem aplikasi ini dilakukan pengujian Beta. Hal ini dilakukan untuk mengetahui apakah peneliti sudah sesuai dengan seperti yang penulis harapkan. Dalam pengujian beta penulis menggunakan 15 pertanyaan dengan menyebarkan kuis online melalui *google forms* kepada mahasiswa semester akhir fakultas teknik Universitas Muhammadiyah Bengkulu sebanyak 50 orang, kuis online dengan skala 1-4. Tabel 1 Skala Pengujian Beta

Tingkat Kepuasan	Skala
Sangat Setuju (SS)	4
Setuju (S)	3
Tidak Setuju (TS)	2
Sangat Tidak Setuju (STS)	1

Berdasarkan data hasil kuesioner, didapat persentase masing-masing jawaban dengan menggunakan rumus (1) :

$$Y = \frac{\Sigma(N.R)}{\text{Skor Ideal}} \times 100\% \quad (1)$$

Keterangan :

Y = Nilai persentase yang dicari

X = Jumlah nilai kategori jawaban dikalikan dengan frekuensi ( $\Sigma = N.R$ )

N = Nilai dari setiap jawaban

R = Frekuensi

Skor ideal = Nilai tertinggi dikalikan dengan jumlah sampel ( $4 \times 5 = 20$ )

Hasil pengujian beta dapat dilihat pada tabel 2 berikut.

Tabel 2. Hasil Pengujian Beta

No	Pertanyaan	Jawaban Responden	(N)	(R)	(N.R)	$\Sigma$ (N.R)	(Y)	Hasil
1.	Aplikasi yang dihasilkan mudah digunakan ( <i>user friendly</i> )?	SS	4	41	164	191	95.5%	Aplikasi <i>User Friendly</i>
		S	3	9	27			
		TS	2	0	0			
		STS	1	0	0			
2.	Interaksi dalam aplikasi jelas dan dimengerti?	SS	4	48	192	198	99%	Aplikasi jelas dan mudah dimengerti
		S	3	2	6			
		TS	2	0	0			
		STS	1	0	0			
3.	Tampilan aplikasi apakah menarik?	SS	4	32	128	174	87%	Tampilan aplikasi menarik
		S	3	10	30			
		TS	2	8	16			
		STS	1	0	0			
4.	Tata letak menu apakah baik dan tidak rumit	SS	4	20	80	168	84%	Letak menu tidak rumit
		S	3	28	84			
		TS	2	2	4			
		STS	1	0	0			
5.	Bisa digunakan dalam aplikasi desktop?	SS	4	50	200	200	100%	Aplikasi desktop
		S	3	0	0			
		TS	2	0	0			
		STS	1	0	0			
6.	Aplikasi sesuai yang ditampilkan berdasarkan menu?	SS	4	50	200	200	100%	Tampilan sesuai dengan menu
		S	3	0	0			
		TS	2	0	0			
		STS	1	0	0			
7.	Pengguna bisa mendapatkan inspirasi judul skripsi?	SS	4	34	136	178	89%	Aplikasi mendukung pengguna untuk mendapatkan inspirasi
		S	3	10	30			
		TS	2	6	12			
		STS	1	0	0			
8.	Apakah tampilan aplikasi memenuhi keinginan pengguna?	SS	4	33	132	173	86.5%	Aplikasi memenuhi
		S	3	8	24			
		TS	2	8	16			

	STS	1	1	1			keinginan pengguna
9. Teks yang ditampilkan apakah jelas?	SS	4	50	200	200	100%	Teks ditampilkan jelas
	S	3	0	0			
	TS	2	0	0			
	STS	1	0	0			
10. Menu yang ditampilkan apakah berfungsi dengan baik?	SS	4	50	200	200	100%	Fungsi menu baik
	S	3	0	0			
	TS	2	0	0			
	STS	1	0	0			
11. Pernahkah terjadi eror saat aplikasi digunakan?	SS	4	50	200	200	100%	Tidak ada eror pada aplikasi
	S	3	0	0			
	TS	2	0	0			
	STS	1	0	0			
12. Ada kontak person jika terjadi eror aplikasi?	SS	4	50	200	200	100%	Tersedia kontak person aplikasi
	S	3	0	0			
	TS	2	0	0			
	STS	1	0	0			
13. Database judul skripsi apakah lengkap setiap tahun?	SS	4	40	160	181	90.5%	Database <i>update</i>
	S	3	3	9			
	TS	2	5	10			
	STS	1	2	2			
14. Apakah aplikasi ini memberikan dampak terhadap pengguna?	SS	4	50	200	200	100%	Aplikasi berdampak pada pengguna
	S	3	0	0			
	TS	2	0	0			
	STS	1	0	0			
15. Apakah perlu pengembangan dari aplikasi?	SS	4	6	24	134	67%	Perlu pengembangan aplikasi
	S	3	23	69			
	TS	2	20	40			
	STS	1	1	1			

## 4. KESIMPULAN

Kesimpulan dari penelitian ini adalah telah berhasil pembuatan aplikasi pencarian judul skripsi dengan menggunakan aplikasi visual studio, aplikasi ini mampu menerapkan fungsi dari algoritma *dynamic programming* pada empat variabel pencarian yang telah ditentukan. Sehingga aplikasi ini telah memberikan informasi pengecekan judul skripsi yang akan diusulkan, jika judul skripsi sudah ada maka aplikasi akan menolak judul yang akan dimasukkan.

## UCAPAN TERIMA KASIH

Diucapkan terima kasih pihak yang terkait dalam hal ini tempat penelitian dan peneliti sebelumnya.

## DAFTAR PUSTAKA

- [1] I. Fawwaz, A. Winarta, Selvianna, J. J. Ramli, and L. M. Waruwu, "Implementation Of Dynamic Programming Algorithm For Npc Movement In Police And Thief Game," *J. Inf. Technol. Educ. Res.*, vol. 2, no. 2, pp. 114–121, 2019, doi: 10.31289/jite.v2i2.2169.
- [2] M. Riski, A. Yanuar, and B. Santosa, "Gudang Barang Jadi Pt Xyz Dengan Penerapan Racking System Untuk," *J. Rekayasa Sist. Ind.*, vol. 3, no. 4, pp. 25–31, 2016.



- [3] M. Ali, "Evaluasi Softskill Mahasiswa Berbasis Android Dengan Algoritma Dynamic Programming," *J. Elektron. Sist. Inf. Dan Komput.*, vol. 1, no. 1, p. 7, 2017.
- [4] K. Saputra S, N. H. Harahap, and J. S. Sitorus, "Analisis Transportasi Pengangkutan Sampah di Kota Medan Menggunakan Dynamic Programming," *J. Inform.*, vol. 7, no. 2, pp. 126–130, 2020, doi: 10.31294/ji.v7i2.7921.
- [5] P. A. Rahmadani, Y. Saintika, and W. A. Prabowo, "Pendukung Keputusan Untuk Menentukan Menu Diet Bagi Penderita Diabetes," pp. 978–979, 2018.
- [6] G. Gunawan and A. W. Nuri David Veronica, "Sistem Informasi Katalog Buku Dan Skripsi Dengan Pencarian Menggunakan Algoritme Interpolation Search," *Pseudocode*, vol. 6, no. 2, pp. 164–171, 2019.
- [7] S. Masripah and L. Ramayanti, "Penerapan Pengujian Alpha Dan Beta Pada Aplikasi Penerimaan Siswa Baru," *Swabumi*, vol. 8, no. 1, pp. 100–105, 2020, doi: 10.31294/swabumi.v8i1.7448.
- [8] A. Suandi, F. N. Khasanah, and E. Retnoningsih, "Pengujian Sistem Informasi E-commerce Usaha Gudang Cokelat Menggunakan Uji Alpha dan Beta," *Inf. Syst. Educ. Prof.*, vol. 2, no. 1, pp. 61–70, 2017, [Online]. Available: <https://media.neliti.com/media/publications/234474-pengujian-sistem-informasi-e-commerce-us-2bea597f.pdf>.
- [9] F. N. Salamah, U., & Khasanah, "Pengujian Sistem Informasi Penjualan Undangan Pernikahan Online Berbasis Web Menggunakan Black Box Testing," *Inf. Manag. Educ. Prof.*, vol. 2, no. 1, pp. 35–46, 2017.
- [10] F. N. Khasanah, S. Murdowo, T. Informatika, U. Bina, P. Beta, and P. N. Fungsional, "Pengujian Beta Pada Aplikasi Game Edukasi," *Infokam*, vol. 15, no. 2, pp. 83–89, 2019.
- [11] G. Gunawan and S. Fernandes, "Implementasi Template Web Responsive Dinamis (Studi Kasus: Website Lembaga Penjaminan Mutu Dan Pengembangan Pendidikan)," *J. Sist. Inf. dan Inform.*, vol. 4, no. 2, pp. 156–169, 2021, doi: 10.47080/simika.v4i2.1316.
- [12] A. Sucipto, Q. J. Adrian, and M. A. Kencono, "Martial Art Augmented Reality Book (Arbook) Sebagai Media Pembelajaran Seni Beladiri Nusantara Pencak Silat," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 10, no. 1, pp. 40–45, 2021, doi: 10.32736/sisfokom.v10i1.983.