



Analisis Algoritma Sequential Search Pada Aplikasi Pencarian Berita

Mhd Furqan^{1*}, Armansyah², Riski Askia Kurniawan³
^{1,2,3}Ilmu Komputer, Universitas Islam Negeri Sumatra Utara, Indonesia
Email: ¹mfurqan@uinsu.ac.id, ²armansyah@uinsu.ac.id, ³riskiaskiak@gmail.com

Abstract

The algorithm is an approach to be able to compile and manage data efficiently. The Sequential Search algorithm is used to build a mobile-based news search application. The search function is used to validate the data itself. The Sequential Search algorithm has 2 possibilities, namely the best possibility (best case) and the worst case (worst case). In determining a possibility, it takes the complexity of the time algorithm. This study will analyze the Sequential Search algorithm in determining the 2 possibilities that occur in mobile applications. Data is one of the things needed in the development of an application. There are 100 data used by researchers as keywords and researchers will take 5 keywords, namely Earthquake, PDI, Indonesian Education, Floods, and Online Sales to determine the best case and worst case from the Sequential Search algorithm. In determining the speed of time required running time program in units of milliseconds (ms). So that the average time for the earthquake keyword is 0.014189 ms, the PDI keyword is 0.073763 ms, the Indonesian Education keyword is 0.169640 ms, the Flood keyword is 0.206307 ms, and the Online Selling keyword is 0.284086 ms. By obtaining the time from the test, the results of the complexity are also obtained, namely $T_{min}(n) = 0.014189$ ms so that the best case is found in the Earthquake keyword and $T_{max}(n) = 0.284086$ ms so that the worst case is found in the Online Selling keyword. And $T_{avg}(n) = 0.1491375$ ms. News API is HTTP REST API which is used to access news after keywords are found

Keywords: Algorithm Sequential Search, Complexity algorithm, and News API

Abstrak

Algoritma merupakan sebuah pendekatan untuk dapat menyusun dan mengelola data secara efisien. Algoritma Sequential Search digunakan untuk membangun sebuah aplikasi pencarian berita berbasis mobile. Fungsi pencarian digunakan untuk memvalidasi data sendiri. Dalam algoritma Sequential Search memiliki 2 kemungkinan yaitu best case dan worst case dalam menentukan sebuah kemungkinan dibutuhkan kompleksitas algoritma waktu. Penelitian ini akan melakukan Analisa terhadap algoritma Sequential Search dalam menentukan 2 kemungkinan yang terjadi pada aplikasi mobile. Data adalah salah satu hal yang dibutuhkan dalam pembangunan sebuah aplikasi. Ada 100 data yang digunakan peneliti sebagai kata kunci dan peneliti akan mengambil 5 kata kunci yaitu Gempa, PDI, Pendidikan Indonesia, Banjir, dan Jualan Online untuk menentukan best case dan worst case dari algoritma Sequential Search. Dalam menentukan kecepatan waktu dibutuhkan running time program dalam satuan millisecond (ms). Sehingga didapatkan waktu rata - rata untuk kata kunci Gempa yaitu 0.014189 ms, kata kunci PDI yaitu 0.073763 ms, kata kunci Pendidikan Indonesia yaitu 0.169640 ms, kata kunci Banjir yaitu 0.206307 ms, dan kata kunci Jualan Online yaitu 0.284086 ms. Dengan didapatkan waktu dari pengujian maka hasil dari kompleksitas juga didapat yaitu $T_{min}(n) = 0.014189$ ms sehingga best case terdapat pada kata kunci Gempa dan $T_{max}(n) = 0.284086$ ms sehingga worst case terdapat pada kata kunci Jualan Online. Dan $T_{avg}(n) = 0.1491375$ ms. News API adalah HTTP REST API yang digunakan untuk mengakses berita setelah kata kunci ditemukan.

Kata kunci: Pencarian Beruntun, Kompleksitas algoritma, dan News API

1. PENDAHULUAN

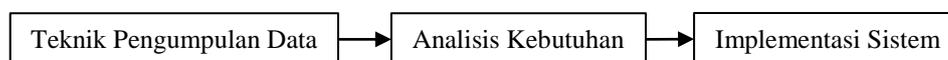
Algoritma dan struktur data memiliki peran yg sangat besar dalam tahap pengembangan sebuah perangkat lunak, baik dalam pembuatan tampilan serta penerapannya. Sebuah algoritma digunakan untuk menyelesaikan tugas atau

fungsi tertentu sehingga tercapai tujuan yang diinginkan dengan proses jalan menghabiskan waktu tertentu. Ada beberapa algoritma yang bisa digunakan untuk melakukan pencarian, salah satunya yg digunakan adalah *Sequential Search* [1]. Algoritma *Sequential Search* atau yang dikenal sebagai Linear Search suatu algorithm search yang prosesny beruntun atau membandingkan setiap elemn secara bertahap hingga akhir, dimana data - data tidak perlu diurutkan terlebih dahulu, mulai dari elemen pertama sampai elemen yang di cari ditemukan atau seluruh elemen sudah di periksa [2]. Algoritma *Sequential Search* memiliki 2 kemungkinan yaitu best case dan worst case dalam menentukan suatu kemungkinan dibutuhkannya kompleksitas waktu algoritma atau $T(n)$. Penerapan *Sequential Search* ini fokus pada berpusat aplikasi yang disiapkan dalam proses pencarian untuk mencari data atau keyword (kata kunci) dari dataset sebagai pencarian berita. [3] algoritma *Sequential Search* dapat digunakan pada array 1 dimensi yang terurut atau maupun yang belum beurut [2].

Kompleksitas algoritma menggunakan salah satu fungsi yang disimbolkan $g(n)$ akan memberikan batas atas terhadap jumlah operasi yang berada dalam algorithm terkait saat sebuah inputan nilai sebanyak n . [1] Ada dua kompleksitas algoritma, yaitu kompleksitas waktu dan kompleksitas ruang. Kompleksitas waktu disimbolkan $T(n)$ dan kompleksitas ruang disimbolkan $S(n)$. Kompleksitas waktu diukur dari jumlah tahapan komputasi yang dibutuhkan untuk menjalankan algoritma sebagai fungsi dari ukuran masukan n [4]. Dalam perancangan aplikasi juga dibutuhkan REST API untuk mengakses berita yaitu News API. News API yaitu web API yang memberikan layann berisi kumpulan berita yng nantinya dapat digunakn oleh para developer untuk mengembangkn sebuah aplikasi, news API dapat diakses melalui link <https://newsapi.org/>. Layann jenis ini bersifat terbuka dn memiliki berfungsi untuk integrse data dn kolaborasi informsi menggunakn teknologi yng dimiliki oleh setiap pengguna [5].

2. METODOLOGI PENELITIAN

Tahap – tahap proses yang dilakukan 3 tahap dalam penelitian ini, berikut 3 tahap akan digambarkan dalam diagram seperti berikut:



Gambar 1. Perencanaan

2.1. Teknik Pengumpulan Data

Ada 2 teknik metode yang dilakukan untuk pengumpulan data, yang di lakukan dalam penelitian, berikut teknik yang digunakan:

a) Penelitian Kepustakaan (*Library Research*).

Penelitian kepustakaan atau library. Penelitian dilakukan dengan mengumpulkan informasi atau data yang diperlukan untuk penlitian, yang akan dignakan sebagai acuan dlam proses pembangunan system dan penyusunn tugas akhir. Publikasi terkait sains, makalah, laporan penelitian, tesis, dan karya referensi tertulis lainnya, seperti teori tentang Algoritma Pencarian Berurutan, dapat memberikan informasi yang dibutuhkan dan

digunakan dalam penelitian ini.

b) Observasi.

Observasi adalah metode yang dengan cara melihat secara langsung pada objek penelitian. Objek yang dilakukan untuk dituju sebagai sasaran pengamatan adalah Google trends dengan link <https://trends.google.co.id/> dan berbagai media sosial seperti instagram, tiktok, dan twiter. Yang dijadikan sebagai data atau kata kunci yang sedang viral atau yang sedang trend selama beberapa hari atau beberapa bulan belakangan. Pengumpulan dilakukan secara kolektif atau secara selektif, yang berarti pemilihan kata kunci yang dipilih merupakan kata kunci yang sudah di filter secara pemilihan seperti kata kunci sepak bola, kata kunci sepak bola dipilih karena yang sedang trending di google trends yaitu spanyol vs jerman, kamerun vs serbia, dan korea selatan vs ghana. Maka penulis menentukan untuk kata kunci yang dipilih sebagai data yang digunakan untuk pencarian ialah sepak bola karena kata kunci tersebut sangat berkaitan tentang sepak bola.

2.2. Analisis Kebutuhan

Bagian ini memiliki tujuan dengan kebutuhn sistem yang baru dan mengembangkan sebuah sistem aplikasi yang memadai kebutuhan tersebut atau memutuskan bahwa pengembangan sistem yang baru tidak dibutuhkan. Analisis kebutuhan Sistem adalah dimana beberapa kebutuhan bahan dalam sistem yang akan dipergunakan untuk menambah dan membantu jalan proses pembuatan suatu obyek. Analisis Kebutuhan dibagi menjadi 2 yaitu:

a) Fungsional.

Fungsional berisi kebutuhan proses - proses apa saja yang nanti dilakukan oleh sistem. Proses ini membutuhkan *user* mencari berita dengan data yang ada. *User* hanya mencari berita dengan data. Dan berita yang dibutuhkan juga dari sumber dan portal berita dan yang dibutuhkan untuk menampilkan berita ialah API yang berasal dari NewsApi sebagai penyempurna aplikasi.

b) Non Fungsional.

Non fungsional pendukung kebutuhan yang dimiliki sistem. Kebutuhan sistem ialah dengan data yang digunakan sebagai kata kunci (*keyword*) dan yang dibutuhkan sistem ialah Algoritma *Sequential Search* untuk pencarian data.

2.3. Implementasi Sistem

Bukan hanya algoritma *Sequential Search* yang akan diimplementasikan kedalam sistem tapi ada juga beberapa penunjang lain agar *Sequential Search* berjalan lancar pada aplikasi seperti: Dataset yang digunakan untuk pencarian dan dibutuhkan juga *running time* untuk melihat waktu yang dibutuhkan algoritma *Sequential Search*. Untuk melancarkan implementasi pada sistem penulis menggunakan android studio versi terbaru atau versi *chipmunk* dan juga internet sebagai alat atau *tools* yang digunakan untuk mengakses *firebase* sebagai tempat

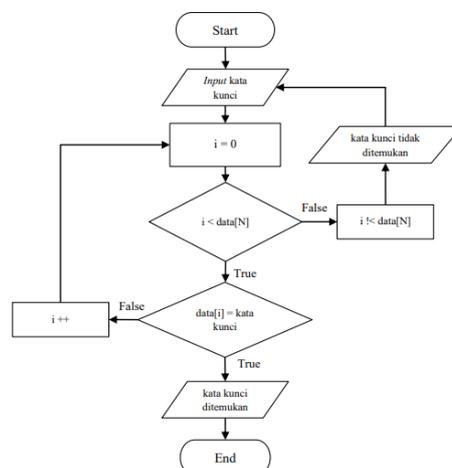
menyimpan dataset dan *news* API yang membutuhkan internet untuk koneksi aplikasi.

3. HASIL DAN PEMBAHASAN

3.1. Algoritma *Sequential Search*

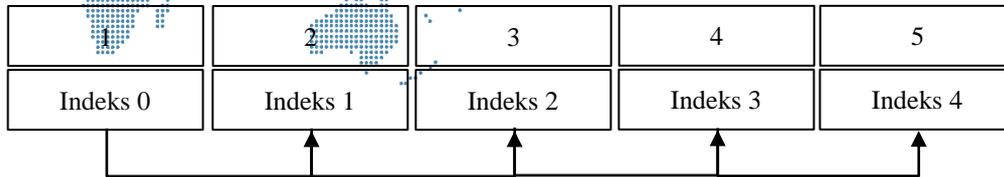
Algoritma *Sequential Search* merupakan model pencarian yang paling mudah digunakan pada kumpulan data. *Sequential Search* menggunakan strategi membandingkan data secara berurutan satu per satu dengan objek pencarian sampai data ditemukan atau tidak [6]. Algoritma ini bekerja dengan mencocokkan data pada database dengan data yang dicari pencarian dilakukan secara beruntun dari data awal hingga data terakhir. Algoritma *Sequential Search* atau bisa juga disebut pencarian linear atau linear search adalah algoritma pencarian sederhana yang pencariannya dilakukan secara beruntun [7]. Pencarian beruntun bisa dilakukn dengn barisan bilangan atau karakter yang beruurtan dari besar ke kecil atau kecil ke besar dan tidak terurut dari kecil ke besar. Pencarian sequential dengn melakukan perbandingan data yang akan dicari dengan kata kunci dalam barisan $A[1] \dots A[n]$ dengn memulai elemen awal pada barisan A. Perbandingan yang bernilai sama, akan dinyatakan berhasil atau sukses dan pencarian dihentikan [3].

Larik atau yang disimbolkan L merupakn deret yang memiliki n nilai data $(L[0], L[1], \dots, L[n+1])$ dan k yng akan mau dicari. Pencarian dilakukn untk menari nilai $L[i] = k$. Dengn yng memiliki nilai i adalah bilangan indeks terkecil yang memnuhi syarat kondisi $0 \leq k \leq n-1$. Di tentukn apabila ada kemungkinn bahwa ada kata kunci yng dicari tidakketemu [6]. Algoritma pencarian beruntun adalah bagian perulangan (loop) yaitu for dan While dengan dua kondisi yakni \leq yang mengontrol agar looping jangan sampai melewati batas yang ditentukan dan not (ditemukan) yang mengontrolsearch apabila data sudh ditemukn maka pencariin tidak perlu lagi diteruskan. Jadi, hal yang mengakibatkan proses pencarian keluar dari bagian looping adalah barisan sudah habis yakni \geq atau lebih besar atau data yang dicari sudah ditemukn yakni ketemu = True [6]. Untuk *flowchart* algoritma *Sequential Search* dapat dilihat berikut:



Gambar 2. Flowchart Algoritma *Sequential Search*

Berikut contoh proses algoritma *Sequential Search*:



Gambar 3. Proses Algoritma *Sequential Search*

1. Tahap awal data melakukan perbandingan satu persatu secara beruntun sampai data tersebut ditemukan atau tidak ditemukan.
2. Pada dasarnya, pencarian ini hanya melakukan pengulangan data dari 1 sampai dengan jumlah data (n).
3. Setiap pengulangan, dibandingkan data ke-i dengan data yang sedang dicari.
4. Apabila data sama dengan yang dicari, berarti data telah berhasil di temukan. Sebaliknya apabila sampai akhir melakukan pengulangan tidak ada data yang sama dengan yang dicari, berarti data tidak ada yang ditemukan. [9]

3.2. Kompleksitas algoritma

Kompleksitas algoritma adalah fungsi yang disimbolkan dengan $g(n)$ yang memberikan batasan up dari jumlah tiap operasi yang berada dalam algoritma tersebut saat masuk sebanyak nilai n. Kompleksitas sendiri dibagi menjadi 2, yang pertama kompleksitas waktu yang disimbolkan fungsi $T(n)$ adalah jumlah waktu yang diperlukan komputer saat menjalankan algoritma tersebut. Yang kedua yaitu kompleksitas ruang disimbolkan $S(n)$, yaitu jumlah ruang memori komputer yang dibutuhkan saat kondisi algoritma itu melakukan proses dari awal sampai selesai [1].

Untuk menentukan rumus kompleksitas algoritma dibutuhkan pseudocode untuk melihat jalannya algoritma:

```

Algoritma Pencarian beruntun
Deskripsi
Var data = array["buku", "pulpen", "penghapus", "tas"]
Var position = 0
Var ketemu = false
Var key = string
Implementasi
For (var i= 0, i < data; i++) do
If (data = key) then
Position = i
Ketemu = true
End
If (position = true) then
Print "data ketemu"
Else
Print "data tidak ketemu"
End
Return i
    
```

Gambar 4. Pseudocode Algoritma *Sequential Search*

- a) Kemungkinan Terbaik terjadi karena proses pencarian diawali dari indeks 0 maka $a_1 = x$, berikut rumusnya:

$$T_{min}(n) = 1 \tag{1}$$

- b) Kemungkinan terburuk terjadi karena proses pencarian melawati proses semua secara berurut sampai akhir atau hingga ketemu maka $a_n = x$ atau x tidak ketemu. Berikut rumusnya:

$$T_{max}(n) = n \tag{2}$$

- c) Avarange jika x ditemukn pada position ke - n , maka proses perbandingn ($a_k = x$) akan dihitung sebanyak n kali.

$$T_{avg}(n) = \frac{(1+2+3.....+n)}{n} = \frac{\frac{1}{2}n(1+n)}{n} = \frac{(n+1)}{2} \tag{3}$$

Perlu diketahui bahwa kompleksitas algoritma memiliki kemungkinan terbaik dan terburuk nya sendiri tergantung dari algoritma yang digunakan seperti pada kasus ini algoritma yang digunakan adalah *Sequential Search*.

3.3. Pengujian

Setelah semua selesai diimplementasikan selanjutnya untuk pengujian. Pengujian ini akan dilakukan beberapa pencarian dengan keyword yang sama dikarenakan setiap *running time* memiliki waktu yang berbeda maka dibutuh kan rumus untuk menghitung nilai rata - rata yaitu:

$$x = \frac{\text{jumlah semua nilai}}{\text{banyak data nilai}} \text{ atau } x = \frac{\sum x}{n} \tag{4}$$

Untuk satuan waktu yang digunakan adalah *millisecond* atau *ms*.

Pengujian ini menggunakan dataset yang sudah dikumpulkan untuk menjadi keyword, berikut kata kunci yang digunakan beserta position indexnya dalam bentuk tabel:

Tabel 1. Tabel Kata Kunci

No	Keyword	Position Index
1	Gempa	0
2	PDI	24
3	Pendidikan Indonesia	49
4	Banjir	74
5	Jualan Online	99

Pada tabel kata kunci diatas mengambil data sesuai dengan urutan awal sampai akhir yang dapat dilihat pada position index. Telah dijelaskan bahwa *Sequential Search* memiliki kemungkinan terbaik jika ditemukan di awal dan memiliki kemungkinan terburuk jika ditemukan diakhir. Maka pengujian ini akan dilihat waktu yang didapat untuk pencarian. Berikut pengujian:

0.013492

gempa

×

Gambar 5. Pengujian Kata Kunci Gempa Pertama

Pada gambar 5 diatas mendapatkan hasil 0.013492 ms. Pengujian terhadap masing - masing kata kunci akan dilakukan sebanyak 3 kali untuk mendapatkan hasil yang lebih akurat. Dikarenakan waktu *running time* tidak memiliki waktu tetap.

0.016092 gempa ✕

Gambar 6. Pengujian Kata Kunci Gempa Kedua

Untuk pengujian kedua mendapatkan waktu 0.016092 ms, tidak jauh dari pencarian sebelum nya.

0.012985 gempa ✕

Gambar 7. Pengujian Kata Kunci Ketiga

Untuk pengujian ketiga mendapatkan waktu 0.012985 ms. Setelah dilakukan percobaan maka akan ditentukan nilai rata - rata pada kata kunci gempa dengan rumus yang sudah ditentukan sebelumnya.

$$\text{nilai rata - rata} = \frac{0.013492 + 0.016092 + 0.012985}{3} = 0.014189 \text{ ms}$$

Untuk waktu rata - rata yang didapatkan pada kata kunci gempa adalah 0.014189 ms. Untuk kata kunci yang lain juga dilakukan hal yang sama sehingga mendapatkan waktu sebagai berikut:

Tabel 2. Hasil *Running Time* Pencarian

No	Keyword	Running Time
1	Gempa	0.014189 ms
2	PDI	0.073763 ms
3	Pendidikan Indonesia	0.169640 ms
4	Banjir	0.206307 ms
5	Jualan Online	0.284086 ms

Setelah semua waktu didapatkan kemudian maka bisa ditentukan best case dan worst case dari algoritma *Sequential Search* dengan kompleksitas algoritma waktu.

- a) $T_{\min}(n) = \text{Running time } 0.014189 \text{ ms.}$
Kemungkinan terbaik atau *best case* didapatkan oleh kata kunci Gempa.
- b) $T_{\max}(n) = \text{Running time } 0.284086 \text{ ms.}$
Kemungkinan terburuk atau *worst case* didapatkan oleh kata kunci Jualan Online
- c) $T_{\text{avg}}(n) = \frac{0.284086+0.014189}{2} = 0.1491375 \text{ ms.}$
Untuk waktu rata - rata dari T_{\min} dan T_{\max} adalah 0.1491375 ms.

4. SIMPULAN

Dari pemaparan penelitian ini dapat diberi kesimpulan bahwa penelitian berhasil menerapkan algoritma *Sequential Search* dalam aplikasi pencarian berita.

Hasil yang didapatkan setelah dianalisa dari pengujian maka hasil *running time* mendapatkan hasil dengan kata kunci Gempa memiliki waktu rata - rata 0.014189 ms, kata kunci PDI dengan waktu rata - rata 0.073763 ms, kata kunci Pendidikan Indonesia dengan waktu rata - rata 0.169640 ms, kata kunci Banjir dengan waktu rata - rata 0.206307 ms, dan kata kunci Jualan Online dengan waktu rata - rata 0.284086 ms. Dengan didapatkan waktu dari pengujian maka hasil dari kompleksitas juga didapat yaitu $T_{\min}(n) = 0.014189$ ms atau *best case* terdapat pada kata kunci Gempa dan $T_{\max}(n) = 0.284086$ ms atau *worst case* terdapat pada kata kunci Jualan Online. Dan $T_{\text{avg}}(n) = 0.1491375$ ms. Dari penelitian ini didapatkan algoritma *Sequential Search* memiliki kemungkinan terbaik (*best case*) jika pencarian itu didapatkan diawal atau lebih sedikit dan kemungkinan terburuk (*worst case*) jika pencarian itu didapatkan diakhir atau data lebih banyak. Maka dapat disimpulkan bahwa sesuai dengan yang telah dijelaskan tentang algoritma *Sequential Search* akan lebih unggul jika pencarian ditemukan diawal dan kelemahan bagi *Sequential Search* jika pencarian ditemukan pada indeks terakhir atau data sangat banyak karena memakan waktu yang cukup lama

DAFTAR PUSTAKA

- [1] Y. Religia, 2019, "Analisis Algoritma *Sequential Search* Dan Binary Search Pada Big Data," *J. Ilm. Inform. Arsit. dan Lingkung.*, vol. 14, no. 1, 74-79.
- [2] W. S. Wahyuni, S. Andryana, and B. Rahman, 2022, "Penggunaan Algoritma *Sequential Searching* Pada Aplikasi Perpustakaan Berbasis Web," *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 7, no. 2, 294-302.
- [3] A. Srirahayu and F. Suryani, 2021, "Implementasi Algoritma *Sequential Search* Pada Aplikasi Ensiklopedia Dermatologi berbasis Mobile," *J. Ilm. Rekam Medis dan Inform. Kesehat.*, vol. 11, no. 1, 49-53.
- [4] M. H. Pristyawan, 2018, "Analisis Algoritma *Sequential Search* Untuk Mencari Data Mahasiswa Teknik Informatika Pada Sekolah Tinggi Teknologi Pelita Bangsa," Sekolah Tinggi Teknologi Pelita Bangsa.
- [5] R. R. Hada and E. F. Hariadi, 2020, "Analisa Perbandingan Hasil Text Mining Dengan Fitur Tf Idf dan SQL Like Operator Pada Pencarian Berita berbahasa Indonesia," *JOINCS (Journal Informatics, Network, Comput. Sci.*, vol. 3, no. 1, 1-11.
- [6] H. Umam, S. Hardienata, and A. Chairunnas, 2018, "Impelementasi Algoritma Pencarian *Sequential Search* Pada Ensikloperia Ikan Hias Air Tawar Berbasis Android," *ilmu komput.*, 1-12.
- [7] D. Markuci and C. Prianto, 2022, "Analisis Perbandingan Penggunaan Algoritma *Sequential Search* Dan Binary Search Pada Aplikasi Surat Perjalanan Dinas," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 6, no. 1, 110-119.
- [8] N. Imamah and M. I. Bahari, 2021, "Perbandingan Algoritma *Sequential Search* Dan Algoritma Binary Search Pada Aplikasi Kamus Bahasa Indonesia Menggunakan Php Dan JQuery," *J. Inform.*, vol. 08. 1-6.
- [9] A. Sonita and M. Sari, 2018, "Implementasi Algoritma *Sequential Searching* Untuk Pencarian Nomor Surat Pada Sistem Arsip Elektronik," *J. Pseudocode*, vol. 5, no. 1, 1-9.