

# Implementasi Algoritma *Knuth-Morris-Pratt* pada Perancangan Aplikasi *Chatbot* Kamus Istilah Komputer Berbasis *Web*

Arif Iik Haerudin<sup>1</sup>

<sup>1</sup>Fakultas Teknik, Teknik Informatika, Universitas Pamulang, Jl. Raya Puspiptek No. 46, Kel. Buaran, Kec. Serpong, Kota Tangerang Selatan, Banten 15310, Indonesia

Email: [arifbj@gmail.com](mailto:arifbj@gmail.com)

(\* : coresponding author)

**Abstrak**—Proses pencarian dalam kamus istilah komputer akan sangat membantu bagi siapapun yang ingin mengetahui suatu istilah komputer, akan tetapi proses pencariannya lambat dan memakan waktu di mana hal tersebut menyulitkan pengguna karena harus mencari istilah secara manual dengan membuka setiap halaman satu per satu. Sejumlah besar halaman kamus, dari alfabet A hingga alfabet Z. Untuk mempercepat dan menyederhanakan proses pencarian, diperlukan suatu algoritma guna memaksimalkan proses pencarian. Pada penelitian ini akan mengimplementasikan algoritma *knuth-morris-pratt* pada perancangan aplikasi *chatbot* kamus istilah komputer yang berbasis *web* dengan menggunakan model pengembangan sistem yaitu model *waterfall*. Hasil dari penelitian ini berupa aplikasi *chatbot* kamus istilah komputer berbasis *web*, dengan adanya aplikasi *chatbot* ini pencarian suatu istilah komputer menjadi lebih cepat dan efisien.

**Kata Kunci:** Kamus Istilah; String Matching; *Knuth-Morris-Pratt*; *Web*; *Waterfall*

**Abstract**—The search process in the computer term dictionary will be very helpful for anyone who wants to know a computer term, but the search process is slow and time-consuming where it makes it difficult for users because they have to search for terms manually by opening each page one by one. A large number of dictionary pages, from alphabet A to alphabet Z. To speed up and simplify the search process, an algorithm is needed to maximize the search process. In this study, we will implement the *knuth-morris-pratt* algorithm in the design of a web-based computer term dictionary *chatbot* application using a system development model, namely the *waterfall* model. The results of this study are in the form of a web-based computer term dictionary *chatbot* application, with the existence of this *chatbot* application the search for a computer term becomes faster and more efficient.

**Keywords:** Dictionary of Terms; String Matching; *Knuth-Morris-Pratt*; *Web*; *Waterfall*

## 1. PENDAHULUAN

Kamus merupakan buku yang memuat kumpulan istilah yang disusun menurut abjad beserta dengan penjelasan maknanya dan memiliki berbagai macam jenis sesuai dengan isi yang terkandung di dalamnya, seperti kamus istilah dan ada juga jenis kamus yang menjadi pedoman disiplin ilmu tertentu, misalnya kamus satwa, kamus kedokteran, kamus bahasa daerah dan kamus komputer (Rifqo & Andilala, 2020:69). Kamus komputer dalam bentuk aplikasi sangat praktis dan tidak lagi sulit untuk membawa-bawa buku kamus yang sangat tebal atau beberapa buku untuk menemukan suatu istilah di dunia komputer dan menemukan pemahaman tentang istilah komputer itu sendiri. Tentu saja, mudah untuk mencari dan menghemat waktu dalam penggunaannya.

Istilah ilmu komputer merupakan kata atau gabungan kata yang mengungkapkan suatu makna, konsep, proses dan keadaan yang berhubungan dengan ilmu komputer (Farid & Maliki, 2022:13). Kamus terminologi komputer yang tersebar luas saat ini masih dalam bentuk buku, di mana hal tersebut menyulitkan pengguna karena harus mencari istilah dan arti komputer secara manual, yang memakan waktu lama karena harus membuka setiap halaman satu per satu. sejumlah besar halaman kamus, dari alfabet A hingga alfabet Z.

Proses pencarian di kamus istilah komputer sangat membantu bagi siapapun yang ingin mengetahui suatu istilah komputer, akan tetapi proses pencariannya lambat dan memakan waktu. Untuk mempercepat dan menyederhanakan proses pencarian, diperlukan suatu algoritma untuk memaksimalkan proses pencarian. Algoritma merupakan susunan yang logis dan sistematis untuk memecahkan suatu masalah atau untuk mencapai tujuan tertentu, dalam dunia komputer algoritma sangat berperan dalam masalah pengembangan atau pembangunan suatu software (Maulana,

2017:70). Algoritma pencarian juga berkembang dari hari ke hari, salah satu algoritma pencarian yaitu algoritma *string matching*. Algoritma *string matching* merupakan algoritma pencocokan *string*, di mana sifat algoritma *string matching* adalah mencari sebuah *string* yang terdiri dari beberapa karakter yang biasa disebut *pattern* dalam sejumlah besar *text* (Matondang, 2018:102). Terdapat beberapa algoritma pada *string matching* ini diantaranya *Brute Force*, *Winnowing*, *Knuth-Morris-Pratt*, *Booyer-Moore* dan sebagainya.

Pada penelitian ini penulis akan menerapkan algoritma *knuth-morris-pratt* dalam perancangan aplikasi *chatbot* kamus istilah komputer. Algoritma *knuth-morris-pratt* merupakan algoritma pencarian *string* dengan melakukan perbandingan karakter mulai dari karakter paling kanan dari *string* yang dicari (Nursobah & Pahrudin, 2019:112). Algoritma *KMP* memiliki keunggulan pencarian kecocokan pada file yang berukuran besar (Sa'diah, 2017:115).

Pada penelitian ini akan mengimplementasikan algoritma *knuth-morris-pratt* pada perancangan aplikasi *chatbot* kamus istilah komputer yang berbasis web dengan menggunakan model pengembangan sistem yaitu model *waterfall* karena dengan model ini akan memudahkan dalam tahap pengembangan seperti yang dilakukan secara bertahap. Model *waterfall* merupakan proses mengembangkan perangkat lunak yang langkahnya dilakukan secara berurut dan bertahap meliputi tahapan perencanaan, pemodelan, implementasi dan pengujian (Trisianto, 2018:12).

## 2. METODOLOGI PENELITIAN

### 2.1 Metode Waterfall

Dalam merancang aplikasi *chatbot* kamus istilah komputer ini penulis menggunakan *Sistem Development Live Cycle (SDLC)* dengan model *waterfall*. Di bawah ini akan dijelaskan langkah-langkah pengembangan *software* dengan menggunakan model *waterfall*:

#### a. Requirement

Pada tahap ini, penulis menganalisis masalah apa yang dihadapi untuk mengetahui kebutuhan pengguna. Berikut adalah daftar *requirement* pada penelitian ini:

##### 1. User Requirement (Kebutuhan Pengguna)

Kebutuhan pengguna untuk aplikasi *chatbot* kamus istilah komputer yaitu *software* dirancang dengan *user interface* Bahasa Indonesia yang mudah dipahami, berupa web *application* yang dirancang seminimalis mungkin dan *button* menggunakan ikon yang sesuai juga mudah dipahami.

##### 2. System Requirement (Kebutuhan Sistem)

Kebutuhan sistem pada aplikasi *chatbot* kamus istilah komputer yaitu sistem dibangun dengan model *client-server* dan web *application* menggunakan *framework laravel 8*, sistem dapat menampilkan daftar istilah, sistem dapat menjawab pertanyaan mengenai istilah apa yang dicari pengguna dan menampilkan referensi foto atau video yang berkaitan dengan jawaban.

#### b. System and Software Design

Pada titik ini spesifikasi sistem permintaan pengguna dari *requirement analysis* penulis kemudian mengimplementasikannya dalam bentuk pengembangan desain berupa *Unified Modeling Language (UML)* sedangkan pada perancangan basis datanya menggunakan *Entity Relationship Diagram (ERD)*. *UML* merupakan kumpulan diagram-diagram yang sudah memiliki standar untuk membangun perangkat lunak berbasis objek” (Sulianta, 2017) sedangkan *ERD* adalah sebuah diagram yang menggambarkan setiap entitas yang terkait pada suatu sistem dimana pada *Physical Data Model*, setiap komponen yang ada pada *ERD* akan menjadi objek-objek yang disebut tabel, atribut, *data type*, *relationship*, *primary key*, *foreign key* dan objek lain (Simanjuntak dkk, 2017:15).

#### c. Implementation

Pada tahap ini, rencana aplikasi sistem diimplementasikan ke dalam bentuk program dengan menggunakan bahasa pemrograman *PHP*, *Framework Laravel* serta algoritma pencarian yang digunakan yaitu algoritma *Knuth-Morris-Pratt*. *PHP* adalah bahasa pemrograman yang digunakan bersamaan dengan *HTML* dimana bahasa tersebut dijalankan di sisi web *browser* (Nurmalasari, Anna & Arissusandi, 2019:10).

Sedangkan *framework* adalah kumpulan dari fungsi-fungsi/prosedur-prosedur dan *class-class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau *class* dari awal (Mualim & Putra, 2017:35). Algoritma *knuth-morris-pratt* merupakan algoritma pencarian *string* dengan melakukan perbandingan karakter mulai dari karakter paling kanan dari *string* yang dicari (Nursobah & Pahrudin, 2019:112), algoritma *KMP* memiliki keunggulan pencarian kecocokan pada file yang berukuran besar (Sa'diah, 2017:115). Langkah-langkah yang dilakukan algoritma Knuth-Morris-Pratt pada saat pencocokan *string* adalah sebagai berikut (Ramadhani, 2017:26):

1. Masukkan *query* kata yang akan dicari. Dengan permisalan  $P = Pattern$  atau pola susunan kata yang dijadikan sebagai contoh atau pola teks yang akan dicari  $T = Teks$  atau judul dokumen.
  2. Algoritma *KMP* mulai mencocokkan *pattern* atau pola susunan kata yang dijadikan sebagai contoh pada awal teks.
  3. Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter, *pattern* dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi: Karakter di *pattern* dan di teks yang dibandingkan tidak cocok (*mismatch*) dan semua karakter di *pattern* cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
  4. Algoritma kemudian menggeser *pattern* atau pola susunan kata yang dijadikan sebagai contoh berdasarkan tabel *next*, lalu mengulangi langkah kedua sampai *pattern* atau pola susunan kata yang dijadikan sebagai contoh berada di ujung teks.
- d. *Integration and System Testing*

Dalam tahap ini semua urutan program menjadi satu kesatuan kemudian program diuji dan diperiksa secara menyeluruh untuk mengetahui apakah masih terdapat *bug* atau kesalahan pada sistem, pengujian yang dilakukan menggunakan *blackbox testing*. Metode uji coba *Black Box* memfokuskan pada keperluan fungsional dari *software*, di mana Uji coba *Black Box* berusaha untuk menemukan kesalahan dalam beberapa kategori diantaranya fungsi-fungsi yang salah atau hilang, kesalahan *interface*, kesalahan dalam struktur data atau akses *database* eksternal, kesalahan performa dan kesalahan inisialisasi dan terminasi (Hidayat, 2017:93).

## 2.2. Metode Pengumpulan Data

Metode pengumpulan data merupakan langkah yang paling strategis dalam penelitian, karena tujuan utama dari penelitian adalah mendapatkan data (Sugiyono, 2017:224). Dalam tahap pengumpulan data penulis menggunakan penelitian lapangan (*field research*). Tahap-tahap yang digunakan yaitu wawancara, observasi dan studi kepustakaan.

## 3. ANALISA DAN PEMBAHASAN

Analisis kelayakan penulis lakukan pada kelayakan program yang sudah penulis buat agar dapat dipergunakan dengan baik oleh pengguna. Proses pencarian suatu istilah komputer yang semula dilakukan secara manual, kini dapat dilakukan secara efektif dan efisien menggunakan aplikasi chatbot kamus istilah komputer yang telah penulis buat. Dimana dalam aplikasi ini admin mendapatkan hak akses penuh dalam pengelolaan kamus istilah yang terdapat pada aplikasi chatbot ini. Sedangkan pengguna dapat mengetahui suatu istilah setelah memasukan istilah yang ingin dicari/diketahui.

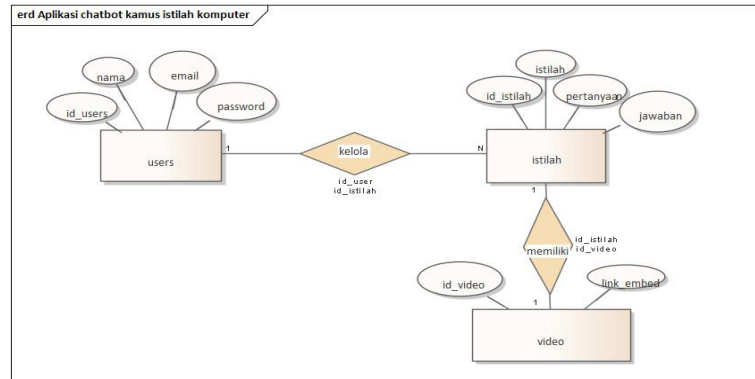
Seperti yang telah dijelaskan di atas berikut ini perancangan basis data dan perancangan sistem pada aplikasi chatbot kamus istilah komputer menggunakan ERD, use case & squence diagram:

### 3.1 Perancangan Sistem

Seperti yang telah dijelaskan di atas berikut ini perancangan basis data dan perancangan sistem pada aplikasi *chatbot* kamus istilah komputer menggunakan ERD, use case & squence diagram.

### 3.1.1 Perancangan *Entity Relationship Diagram (ERD)*

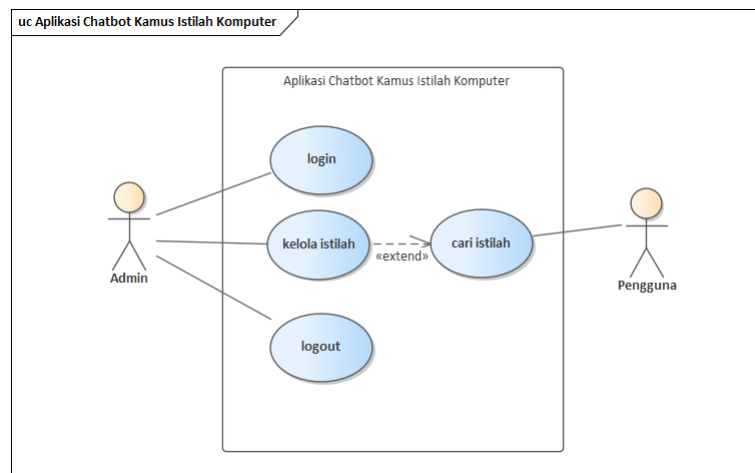
Sebelum membuat *ERD*, penulis terlebih dahulu mengidentifikasi entitas serta atribut yang akan digunakan dalam perancangan *database* pada aplikasi chatbot kamus istilah komputer ini. Dari entitas serta atribut yang telah diidentifikasi, akan dibentuk relasi antar entitas terkait ke dalam bentuk *ERD*. Pada penelitian ini terdapat 3 buah tabel yaitu tabel *user*, istilah dan video. Di mana dalam tabel *user* terdapat kardinalitas 1 dan M untuk tabel istilah itu menunjukkan kalau satu user dapat mengelola banyak istilah dan dalam 1 istilah terdapat 1 video. *ERD* aplikasi *chatbot* kamus istilah komputer terlihat seperti pada gambar 1.



**Gambar 1.** ERD Aplikasi Chatbot Kamus Istilah Komputer

### 3.1.2 Perancangan *Use Case Diagram*

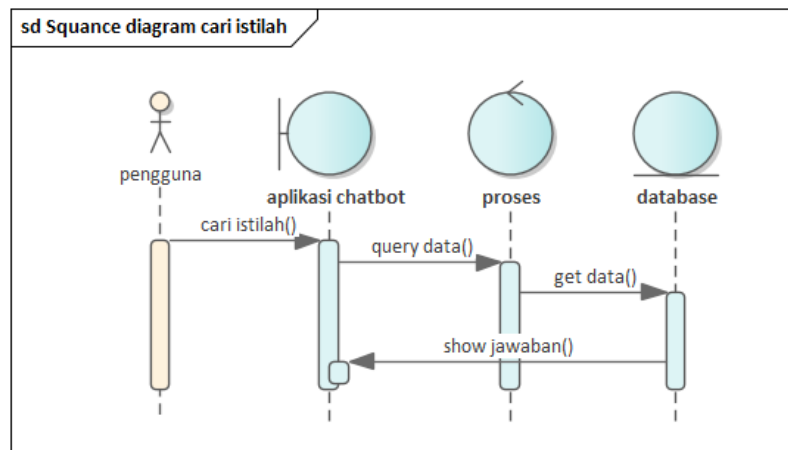
*Use case diagram* menggambarkan interaksi antara *user* dan sistem. Dalam aplikasi *chatbot* kamus istilah komputer ini terdapat dua aktor yang terlibat yaitu Admin dan Pengguna. Admin dapat mengelola istilah setelah berhasil melalui proses *login* dengan cara memasukkan *username* dan *password* yang tersedia dengan benar, sementara Pengguna dapat langsung mengetikkan istilah yang ingin dicari/diketahui pada *form chatbot* tanpa harus melakukan proses *login* terlebih dahulu. *Use case diagram* aplikasi *chatbot* kamus istilah komputer terlihat pada gambar 2



**Gambar 2.** Use Case Aplikasi Chatbot Kamus Istilah Komputer

### 3.1.3 Perancangan *Squence Diagram*

Seperti yang telah dijelaskan pada perancangan *use case* di atas, pada aplikasi *chatbot* kamus istilah komputer ini terdapat dua aktor yaitu admin dan pengguna. Pada *squence diagram* pengguna ini menggambarkan proses pengguna dalam mencari suatu istilah komputer dengan cara memasukkan kata atau kalimat pada *form chat* kemudian sistem akan melakukan *query* dan *get* data lalu menampilkan jawaban. *Squence diagram* pengguna terlihat seperti pada gambar 3.



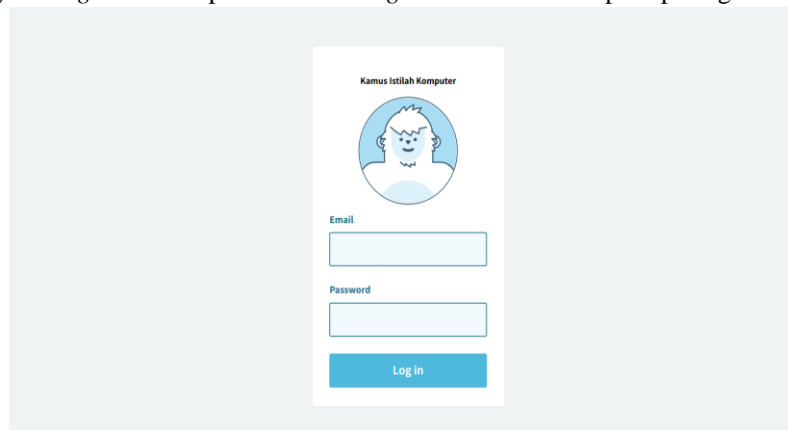
**Gambar 3.** *Sequence Diagram* Cari Istilah

## 4. IMPLEMENTASI

Berikut adalah hasil penerapan algoritma *knuth-morris-pratt* dan model *waterfall* pada perancangan aplikasi *chatbot* kamus istilah komputer yang sebelumnya telah dijelaskan pada bab 2 (metodologi) dan juga telah dilakukan uji serta diperiksa secara menyeluruh untuk mengetahui apakah masih terdapat *bug* atau kesalahan pada sistem. Terdapat beberapa tampilan seperti halaman *login* admin, kelola istilah dan *chatbot* untuk halaman pengguna.

### 1. Tampilan Halaman *Login* Admin

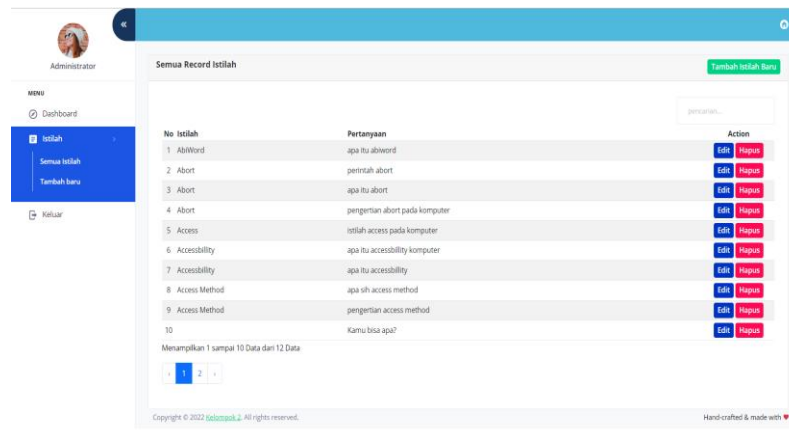
Tampilan halaman *login* admin adalah tampilan yang muncul pertama kali saat admin mengakses sistem. Pada halaman ini proses *login* dilakukan untuk dapat masuk ke dalam aplikasi kamus istilah komputer, jika admin memasukkan *username* dan *password* dengan benar maka sistem akan menampilkan halaman *dashboard*, namun jika *username* dan *password* yang dimasukan salah maka akan muncul pesan kesalahan pada *form login* ini. Tampilan halaman *login* admin terlihat seperti pada gambar 4.



**Gambar 4.** Tampilan Halaman *Login* Admin

### 2. Tampilan Halaman Kelola Istilah

Halaman kelola istilah dapat diakses oleh admin setelah berhasil melalui proses *login*. Pada saat pertama mengakses halaman ini akan ditampilkan *list* istilah yang telah dikelola sebelumnya dan terdapat *button* tambah istilah untuk menambahkan istilah baru, *button* cari untuk mencari suatu istilah, *button* ubah untuk mengubah data istilah dan *button* hapus untuk menghapus suatu istilah yang ada pada sistem. Tampilan halaman kelola istilah terlihat seperti pada gambar 5.



**Gambar 5.** Tampilan Halaman Admin Kelola Istilah

### 3. Tampilan Halaman Cari Istilah

Tampilan halaman pengguna berupa *chatbot*, di mana pengguna dapat langsung mengetikkan istilah apa yang ingin dicari untuk dapat mengetahui suatu istilah yang terdapat pada kamus istilah komputer tanpa harus melalui proses *login* terlebih dahulu. Tampilan halaman pengguna terlihat seperti pada gambar 6.



**Gambar 6.** Tampilan Halaman Cari Istilah

## 5. KESIMPULAN

Berdasarkan analisa, hasil penerapan metode untuk menyelesaikan masalah pada penelitian ini maka dapat diambil kesimpulan bahwa penelitian ini telah menghasilkan sebuah aplikasi chatbot kamus istilah komputer berbasis web dan dalam penggunaannya, aplikasi ini mampu menghasilkan informasi tentang istilah-istilah komputer yang belum banyak diketahui.

## REFERENCES

Farid, & Maliki, O. (2022). Aplikasi Kamus Istilah Komputer Pada Perangkat Mobile Menggunakan fitur Voice, Abjad dan Kategori Berbasis Android. *JSAI: Journal Scientific and Applied Informatics*, V(1), 12-19.

Hidayat, R. (2017). Aplikasi Penjualan Jam Tangan Secara Online Studi Kasus: Toko JAMBORESHOP. *Jurnal Teknik Komputer*, III(2), 90-96.

Matondang, Z. A. (2018). Implementasi Algoritma String Matching Pencarian Kata Dari Makna Rambu Lalu Lintas Berbasis Android. *Jurnal Sistem Informasi Kaputama (JSIK)*, II(1), 101-106.



- Maulana, G. G. (2017). Pembelajaran Dasar Algoritma dan Pemrograman Menggunakan Algoritma Berbasis Web. *Jurnal Teknik Mesin (JTM)*, VI, 69-73.
- Mualim, W., & Putra, G. U. (2017). Implementasi Framework MVC Pada Sistem Informasi Akademik Di STMIK Yadika Bangil. *Jurnal SPIRIT*, IX(2), 35-39.
- Nurmalasari, Anna, & Arissusandi, R. (2019). Rancang Bangun Sistem Informasi Akuntansi Laporan Laba Rugi Berbasis Web Pada PT United Tractors Pontianak. *Evolusi: Jurnal Sains dan Manajemen*, VII(2), 6-14.
- Nursobah, & Pahrudin, P. (2019). Penerapan Algoritma Pencarian Knuth-Morris-Pratt (KMP) Dalam Sistem Informasi Perpustakaan SMK TI Pratama. *SEBATIK*, XXIII(1), 112-115.
- Ramadhani, D. (2017). Perancangan aplikasi pencarian Buku Pada Perpustakaan Islamic Internasional School Darul Ilmi Murni Dengan Algoritma Knuth Morris Pratt. *Pelita Informatika Budi Darma*, XVI(7), 25-27.
- Rifqo, M. H., & Andilala. (2020). Implementasi Algoritma Boyer-Moore Pada Kamus Istilah Berbasis Android. *Jurnal Pseudocode*, III(1), 69-77.
- Sa'diah, H. T. (2107). Implementasi Algoritma Knuth-Morris-Pratt Pada Fungsi Pencarian Judul Tugas Akhir Repository. *Jurnal Komputasi*, XIV(1), 115-124.
- Simanjuntak, H., Lumbantoruan, R., Banjarnahor, W., Sitorus, E., Panjaitan, M., & Panjaitan, S. (2017). Penilaian Kesamaan Entity Relationship Diagram dengan Algoritme Tree Edit Distance. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, VI(1), 15-24.
- Sugiyono. (2017). *Metode Penelitian Kuantitatif, Kualitatif, dan R&D – MPKK*. Bandung: Alfabeta.
- Sulianta, F. (2107). *Teknik Perancangan Arsitektur Sistem Informasi*. Yogyakarta: Andi.
- Trisianto, C. (2018). Penggunaan Metode Waterfall Untuk Pengembangan Sistem Monitoring dan Evaluasi Pembangunan Pedesaan. *Trisianto*, XII(1), 7-21.