

## **Perancangan Aplikasi Kode Etik Profesi Dengan Menerapkan Algoritma Raita**

**Yulytawaty Napitupulu**

Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia

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

**Abstrak**– Kode Etik Profesi merupakan suatu tataan etika yang telah disepakati oleh suatu kelompok masyarakat tertentu. Dapat diartikan juga sebagai pola aturan, tata cara, tanda, pedoman etis dalam melakukan suatu kegiatan atau pekerjaan. Namun pada kenyataannya pelaku profesi banyak yang tidak mengetahui atau memperdulikan kode etik dari profesinya sendiri. Masalah yang terjadi adalah kode etik profesi umumnya tertulis dalam buku kode etik yang sulit dibawa kemana-mana jika dibanding dengan buku lain yang sudah berbasis aplikasi, baik berbasis web atau berbasis mobile. Untuk itu dengan adanya aplikasi kode etik profesi ini, para pelaku profesi dan seluruh masyarakat dapat dengan mudah mengetahui kode etik profesi yang diinginkan dengan proses pencarian kode etik profesi yang telah ditentukan sesuai dengan pattern yang ada. Aplikasi pencarian kode etik profesi ini dirancang dengan menggunakan metode String Matching. Pengembangan aplikasi menggunakan metode perencanaan Unified Modeling Language (UML) dan dibangun pada platform android.

**Kata Kunci:** Kata Kunci : Kode Etik, Profesi, Metode, String Matching, Android..

**Abstract**– Professional Code of Ethics is an ethical order that has been agreed upon by a certain community group. It can also be interpreted as a pattern of rules, procedures, signs, ethical guidelines in carrying out an activity or work. But in fact many professional actors do not know or care about the code of ethics of their own profession. The problem that occurs is that the professional code of ethics is generally written in a code of ethics book which is difficult to carry everywhere when compared to other books that are already application-based, either web-based or mobile-based. For this reason, with the application of this professional code of ethics, professional actors and the entire community can easily find out the desired professional code of ethics by searching for a professional code of ethics that has been determined according to the existing pattern. This professional code of ethics search application is designed using the string matching method. Application development uses the Unified Modeling Language (UML) planning method and is built on the android platform.

**Keywords:** Code of Ethics, Profession, Method, String Matching, Android.

### **1. PENDAHULUAN**

Kode etik profesi merupakan suatu tatanan etika yang telah disepakati oleh suatu kelompok masyarakat tertentu. Dapat diartikan juga sebagai pola aturan, tata cara, tanda, pedoman etis dalam melakukan suatu kegiatan atau pekerjaan. Namun pada kenyataannya pelaku profesi banyak yang tidak mengetahui atau memperdulikan kode etik dari profesinya sendiri. Kode etik profesi umumnya tertulis dalam buku kode etik. Masalah yang terjadi adalah dalam kode etik profesi banyak yang masih menggunakan buku yang sulit dibawa kemana-mana jika dibandingkan dengan buku lain yang sudah berbasis aplikasi, baik berbasis *web* atau berbasis *mobile*. Oleh karena itu, penelitian ini bertujuan untuk merancang sebuah aplikasi kode etika profesi serta memanfaatkannya sebagai media belajar bagi masyarakat umum. Maka untuk mempermudah masyarakat mempelajari kode etik profesi diperlukan pengembangan sistem dengan merancang aplikasi yang memiliki fitur pencarian yang baik guna meminimalisir waktu yang masyarakat butuhkan untuk menemukan hasil pencarian kode etik profesi yang akan dipelajari.

Berdasarkan uraian diatas penulis merancang aplikasi kode etik profesi untuk masyarakat menggunakan *Software Eclipse Juno*. *Eclipse Juno* merupakan sebuah *IDE* yang bersifat *open source* atau yang dapat dikembangkan dan digunakan untuk membangun sebuah program computer dan dapat dijalankan pada *system* operasi apa saja (*multi platform*). *Eclipse* juga dilengkapi dengan adanya *Plug-in Development Environment* untuk mengembangkan *plug-in* baru. Untuk mempermudah proses pencarian pada aplikasi kode etik profesi yang dirancang, maka diimplementasikan algoritma *Raita*. Algoritma *Raita* adalah algoritma pencarian string yang meningkatkan kinerja algoritma *Boyer Moore Horspool*. Algoritma ini diterbitkan oleh Tim Raita pada tahun 1991. Diciptakan oleh Robert S. Boyer dan J. Strother Moore pada tahun 1977.

Ditinjau dari beberapa kutipan jurnal yang sudah melakukan penelitian, dengan judul “Implementasi Algoritma *Raita Search* Pada Aplikasi Filsafat Berbasis *Android*” yang disusun oleh Ratih Anggraini, Nelly Astuti Hasibuan, Suginam, Fince Tinus Waruwu menyimpulkan bahwa:

- Pencarian arti istilah filsafat dilakukan berdasarkan istilah filsafat
- Penerapan algoritma *raita* dapat mempercepat proses penemuan hasil pencarian arti istilah filsafat
- Aplikasi filsafat berbasis *android* yang dibangun pada penelitian ini dapat mempermudah masyarakat umum untuk memperoleh informasi mengenai arti istilah filsafat dengan mudah[1]. Kemudian ditinjau dari judul penelitian “Perbandingan Algoritma *Binary Search* Dan *Raita* Dalam Pencarian Data” yang disusun oleh Dedi Rudi Bawanto, Nidia Rosmawanti, menyimpulkan bahwa hasil penelitian yang dilakukan dalam hal ini membandingkan keakuratan berdasarkan kecepatan dan ketepatan pencarian antara algoritma *Binary Search*, *Boyer moore* dan *Raita*, maka dapat disimpulkan bahwa dari 8 teknik pengujian yaitu match case, part case, 32 record data, 64 record data, 96 record data, tipe data string, tipe data numeric dan tipe data date dengan 5 kali simulasi dari setiap kata kunci pada teknik pengujian, algoritma *raita* lebih cepat 3% dari algoritma *binary search* dalam pencarian data dari keseluruhan data sampel yang

diuji dengan total waktu 2.5116 ms dibandingkan penelitian terdahulu yang mana algoritma boyer moore dengan total waktu 2.8150 ms lebih lambat 1% dari algoritma binary search dengan total waktu 2.7830 ms. Hal tersebut diperoleh dengan menghitung total waktu rata-rata pada setiap kategori teknik pengujian kemudian dijumlahkan menjadi total waktu setiap algoritma pencarian[2].

## 2. METODOLOGI PENELITIAN

### 2.1 String Matching

*String matching* adalah proses pencarian semua kemunculan *query* yang selanjutnya di sebut *Pattern* kedalam *string* yang lebih panjang (text). *Pattern* di lambangkan dengan  $x=x[0..m-1]$  dan panjangnya adalah  $m$ . Teks dilambangkan dengan  $y=y[0..n-1]$  dan panjangnya adalah  $n$ . Kedua *string* terdiri dari sekumpulan karakter yang disebut alphabet yang di lambangkan dengan  $\Sigma$  dan mempunyai ukuran  $\sigma$ . Algoritma *string matching* mempunyai tiga komponen yaitu:

- Pattern, yaitu deretan karakter yang di cocokkan dengan teks
- Teks, yaitu tempat pencocokkan pattern dilakukan
- Alfabet, yang berisi semua simbol yang digunakan oleh bahasa pada *text pattern*.

*String* merupakan bagian penting dari sebuah proses pencarian *string* (*string searching*) dalam sebuah dokumen hasil dari pencarian sebuah *string* dalam dokumen tergantung dari teknik atau cara pencocokan *string* yang di gunakan[3][4][5].

### 2.2 Algoritma Raita

Algoritma *Raita* merupakan bagian dari algoritma *exact string matching* yaitu pencocokan *string* secara tepat dengan susunan karakter dalam *string* yang dicocokkan memiliki jumlah maupun urutan karakterdalam *string* yang sama. Raita merancang sebuah algoritma dengan membandingkan karakter yang terakhir daripola yang diawali dari karakter paling kanan dari "*jendela*". Jika mereka cocok, kemudian karakter pertama daripola teks paling kiri dari jendela juga dibandingkan. Jika mereka cocok, maka akan dibandingkan karakter tengahpola dengan karakter teks tengah jendela. Pada akhirnya, jika mereka benar-benar cocok, maka algoritmamembandingkan karakter lain mulai dari pola karakter kedua ke karakter kedua terakhir, dan akan membandingkan dengan karakter tengah lagi[1]. Algoritma *Raita* mencari pola "P" dalam teks yang diberikan "T" dengan membandingkan setiap karakter pola dalam teks yang diberikan. Pencarian akan dilakukan sebagai berikut. Jendela untuk teks "T" didefinisikan sebagai panjang "P".

- Pertama, karakter terakhir dari pola dibandingkan dengan karakter paling kanan dari jendela.
- Jika ada kecocokan, karakter pertama dari pola dibandingkan dengan karakter paling kiri dari jendela.
- Jika mereka cocok lagi, itu membandingkan karakter tengah pola dengan karakter tengah jendela.

Jika semuanya berhasil, maka perbandingan asli dimulai dari karakter kedua hingga yang terakhir. Jika ada ketidaksesuaian pada tahap apa pun dalam algoritma, ia melakukan fungsi pergeseran karakter buruk yang dihitung pada fase pra-pemrosesan [4].

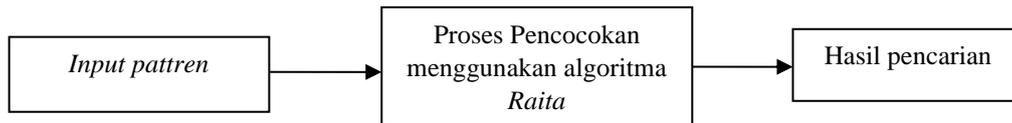
### 2.3 Kode Etik

Kode etik merupakan prinsip-prinsip yang merupakan kesatuan moral yang melekat pada suatu profesi sesuai kesepakatan organisasi profesi yang disusun secara sistematis[6]. Pengertian kode etik lainnya adalah suatu aturan yang tertulis, secara sistematis dengan sengaja dibuat, berdasarkan prinsip-prinsip moral yang ada serta ketika dibutuhkan bisa difungsikan sebagai alat yang dapat digunakan menghakimi berbagai macam dari tindakan yang pada umumnya dinilai menyimpang dari kode etik yang ada. Dalam pembentukannya, kode etik tertentu memiliki tujuan didalamnya yaitu,

- Agar professional dapat memberikan jasa dengan sebaik-baiknya kepada para pemakai ataupun para nasabahnya.
- Sebagai pelindung dari perbuatan yang tidak profesional.

## 3. HASIL DAN PEMBAHASAN

Kode etik profesi merupakan suatu tatanan etika yang telah disepakati oleh suatu kelompok masyarakat tertentu. Dapat diartikan juga sebagai pola aturan, tata cara, tanda, pedoman etis dalam melakukan suatu kegiatan atau pekerjaan. Namun pada kenyataannya pelaku profesi banyak yang tidak mengetahui atau memperdulikan kode etik dari profesinya sendiri. Kode etik profesi umumnya tertulis dalam buku kode etik. Masalah yang terjadi adalah dalam kode etik profesi banyak yang masih menggunakan buku yang sulit dibawa kemana-mana jika dibandingkan dengan buku lain yang sudah berbasis aplikasi, baik berbasis *web* atau berbasis *mobile*. Berdasarkan uraian masalah tersebut penulis merancang aplikasi kode etik profesi yang berbasis *Android* menggunakan *Eclipse Juno Software*. Dalam penelitian ini penulis menerapkan algoritma *Raita* untuk menyederhanakan proses pencarian kode etik profesi dalam aplikasi perancangan kode etik profesi yang berbasis *android*. Aplikasi kode etik profesi berbasis *android* yang di rancang dalam penelitian ini dapat dioperasikan pada *Smartphone* yang di akhiri oleh sistem operasi berbasis *android* sehingga penggunaan aplikasi kode etik profesi berbasis *android* ini sangat mudah karena tidak memiliki ukuran yang terlalu besar sehingga bisa digunakan di mana saja dan kapan saja.



Gambar 1. Tampilan Prosedur

Pada penelitian ini penulis menggunakan algoritma *Raita* dalam pencarian *pattern* JURNAL pada teks KODE ETIK JURNALISTIK

T = KODE ETIK JURNALISTIK

m = JURNAL

diketahui bahwa :

m = Panjang pola

T = Teks yang akan dicari

Maka :

m = 6

Dibuatlah tabel *BmBc* untuk melakukan perhitungan dengan persamaan sebagai berikut:

$$m - 2 \dots\dots\dots(1)$$

$$6 - 2 = 4$$

Mencari nilai *BmBc* (a)

$$m - 1 - i \dots\dots\dots(2)$$

Sebagai pencarian nilai karakter pada tabel *BmBc*

Tabel 1. Perhitungan tabel *BmBc*

|                       |          |          |          |          |          |          |
|-----------------------|----------|----------|----------|----------|----------|----------|
| <b>I</b>              | <b>0</b> | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> |
| <b>A</b>              | <b>J</b> | <b>U</b> | <b>R</b> | <b>N</b> | <b>A</b> | <b>L</b> |
| <b><i>BmBc</i>(a)</b> | <b>5</b> | <b>4</b> | <b>3</b> | <b>2</b> | <b>1</b> | <b>6</b> |

Pencarian nilai *BmBc* (a) dengan menggunakan rumus persamaan

$$m - 1 - i \dots\dots\dots(2)$$

6 - 1 - 0 = 5 maka nilai diletakkan pada indeks ke-0 dengan karakter J

6 - 1 - 1 = 4 maka nilai diletakkan pada indeks ke-1 dengan karakter U

6 - 1 - 2 = 3 maka nilai diletakkan pada indeks ke-2 dengan karakter R

6 - 1 - 3 = 2 maka nilai diletakkan pada indeks ke-3 dengan karakter N

6 - 1 - 4 = 1 maka nilai diletakkan pada indeks ke-4 dengan karakter A.

Nilai L adalah 6 sesuai dengan panjang pola, karena abjad yang tidak ada pada tabel maka diinisialisasikan dengan tanda (\*) kemudian lainnya sesuai dengan panjang pola. Jadi, untuk perhitungan Algoritma *Raita* sesuai tabel *BmBc* adalah sebagai berikut :

Tabel 2. Hasil *BmBc* (a)

|               |   |   |   |   |   |   |
|---------------|---|---|---|---|---|---|
| Karakter Teks | J | U | R | N | A | * |
| <i>BmBc</i>   | 5 | 4 | 3 | 2 | 1 | 6 |

Perhitungan mencari nilai hasil *BmBc* (a) dengan menggunakan rumus persamaan

$$m - 2 \dots\dots\dots(1)$$

$$6 - 2 = 4$$

Maka, dapatlah hasil tabel *BmBc* (a) menjadi 5 karakter.

T = KODE ETIK JURNALISTIK

m = JURNAL

Langkah selanjutnya adalah melakukan pencarian menggunakan Algoritma *Raita* dengan tahap-tahap berikut ini :

1. Tahap pertama, yaitu mencocokkan pola akhir pada teks. Jika tidak cocok maka pola akan bergeser kekanan sebanyak nilai teks.

Tabel 3. Proses Pencarian Pada Teks Ke-1

|      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Teks | K | O | D | E | E | T | I | K | J | U | R | N | A | L | I | S | T | I | K |
| Pola | J | U | R | N | A | L |   |   |   |   |   |   |   |   |   |   |   |   |   |

Pada proses dinyatakan terjadi ketidakcocokan pada teks, maka pada pola terakhir teks terdapat huruf terakhir yaitu L. Pada tabel hasil *BmBc* bahwa L bernilai 6 karakter.

2. Tahap kedua, yaitu pola akan bergeser sebanyak 5 karakter

Tabel 4. Proses Pencarian Pada Teks ke-2

|      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Teks | K | O | D | E | E | T | I | K | J | U | R | N | A | L | I | S | T | I | K |
|      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

|      |   |   |   |   |   |   |
|------|---|---|---|---|---|---|
| Pola | J | U | R | N | A | L |
|------|---|---|---|---|---|---|

- Pada pola terakhir terdapat huruf terakhir yaitu N. Pada tabel hasil *BmBc* bahwa N bernilai 2.
- Tahap ketiga, yaitu jika pergeseran pola terdapat ketidakcocokan, maka pola akan terus bergeser sebanyak 2 karakter.

**Tabel 5.** Proses Pencarian Pada Teks Ke-3

|      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Teks | K | O | D | E | E | T | I | K | J | U | R | N | A | L | I | S | T | I | K |
| Pola |   |   |   |   |   |   |   |   | J | U | R | N | A | L |   |   |   |   |   |
|      |   |   |   |   |   |   |   |   | 0 | 1 | 2 | 3 | 4 | 5 |   |   |   |   |   |

Pada tahap pencarian ketiga semua pola memiliki kecocokan dengan teks. Maka pencarian berhenti.

#### 4. KESIMPULAN

Berdasarkan hasil akhir penerapan algoritma raita pada kode etik profesi maka penulis menguraikan beberapa kesimpulan yaitu Aplikasi perancangan kode etik profesi dilakukan untuk mempercepat user dalam pencarian kode etik profesi. Diman dengan algoritma Raita dapat digunakan untuk mempermudah proses pencarian kode etik profesi..

#### REFERENCES

- [1] R. Anggraini, N. A. Hasibuan, dan F. T. Waruwu, "IMPLEMENTASI ALGORITMA RAITA SEARCH PADA APLIKASI FILSAFAT," vol. 2, hal. 471–475, 2018.
- [2] D. R. Bawanto dan N. Rosmawanti, "Perbandingan Algoritma Binary Search Dan Raita Dalam Pencarian Data," hal. 1311–1317.
- [3] G. D. Hantoro, "Wifi 2015 Wireless Lan." jakarta erlanga , 2015, 2015.
- [4] A. Azhar, N. Marbun, S. Aripin, dan E. Buulolo, "Implementasi Algoritma Horspool Pada Aplikasi Istilah Fashion," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 3, no. 1, hal. 549–551, 2019, doi: 10.30865/komik.v3i1.1641.
- [5] I. Al Murtada, N. Marbun, E. Buulolo, dan S. Aripin, "PENERAPAN ALGORITMA TURBO BOYER MOORE PADA APLIKASI KAMUS BAHASA INDONESIA - ALAS," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 3, no. 1, 2019, doi: 10.30865/komik.v3i1.1639.
- [6] Y. A. Mangesti, "KEINDONESIAAN," hal. 11–22.