

PENERAPAN ALGORITMA BRUTE FORCE PADA MENU SEARCH WEBSITE “CALONKU” DALAM RANGKA PEMILU BERBASIS WEB

¹Irwan Adi Pribadi, ²Yola Rahmawati, dan ³Yunda Heningtyas

^{1,2,3}Jurusan Ilmu Komputer Fakultas Matematika Ilmu Pengetahuan Alam,
Universitas Lampung, Bandar Lampung, 35141

e-mail : ¹irwan.adipribadi@fmipa.unila.ac.id, ²yolarahmawati16@gmail.com

Abstract — General elections are a means of people's involvement in the government process. Democratic countries, including Indonesia, conduct general elections. General election does not always go smoothly, at times tinged with some constraints. The lack of information has prompted a platform to contain the complete information that voters need. Website "Calonku" was created to meet the shortage of such information. In this study, the website "Calonku" implements the algorithm in its search feature, brute force deemed appropriate for solving problems using a straightforward approach. This research discusses the brute force algorithm's appropriate level and effectiveness on the "Calonku" system. Testing the brute force algorithm's performance is by calculating the recall, precision, and accuracy. This test calculates the similarity of each document to the given query. Similarity calculation will produce a weight on each paper that determines how relevant it is to the question. In this study, the average Accuracy value was 0.76 Recall of 0.85, and an average Precision of 0.83.

Keywords: Brute Force; Calonku; Query; Search

1. PENDAHULUAN

Pemilihan umum atau Pemilu merupakan sarana bentuk keterlibatan rakyat dalam proses pemerintahan. Sebagai negara demokrasi, Indonesia telah melaksanakan 11 kali pemilihan umum (pemilu), pemilu di Indonesia telah dilaksanakan sejak tahun 1955. Beberapa kendala seperti Golput (golongan putih) tak luput dijumpai pada kegiatan bernegara lima tahun sekali ini. Salah satu alasan Golput terjadi diindikasikan karena kesulitan akses informasi para calon pemimpin, hal ini yang menyebabkan para pemilih sulit untuk menentukan layak dan kompeten atau tidaknya seorang calon pemimpin. Karakteristik pemilih masa kini lebih kritis terhadap para kandidat, namun sifat kritis calon pemilih tidak dibarengi dengan banyaknya informasi yang bisa didapatkan, bahkan terkadang informasi yang adapun mengandung informasi dengan unsur keberpihakan.

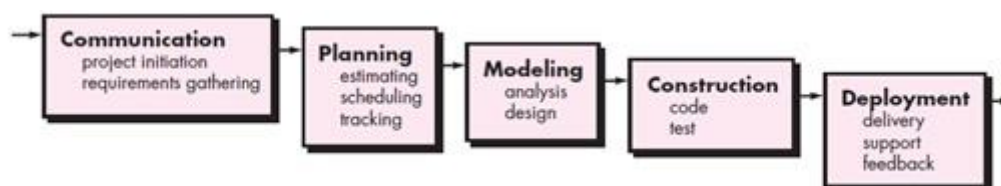
Minimnya informasi tentang calon pemimpin, mendorong dibuatnya website “Calonku”. Website “Calonku” dibuat dengan tujuan sebagai wadah informasi untuk calon pemilih guna mempertimbangkan layak atau tidaknya calon pemimpin tersebut dipilih, website “Calonku” memiliki banyak fitur didalamnya termasuk fitur *Search*. Pada awalnya fitur *search* pada “Calonku” dibangun dengan metode *exact matching*. Namun, metode *exact matching* memiliki kelemahan dimana karakter dari *query* yang dimasukkan harus sama persis dengan karakter yang ada, apabila terdapat sedikit perbedaan maka kata yang diinginkan tidak ditemukan [1]. Kelemahan *exact matching* tersebut bisa mengurangi akurasi hasil pencarian pada fitur *search* “Calonku”. Oleh karena itu, metode *exact matching* diganti dengan algoritma *brute force*. Algoritma *brute force* dipilih karena dinilai mampu memecahkan masalah dengan sangat sederhana, langsung, dan dengan cara yang jelas (*obvious way*) [2]. *Brute Force* adalah suatu pendekatan untuk memecahkan permasalahan, biasanya didasarkan pernyataan masalah dan definisi konsep yang dilibatkan [3]. *Brute Force* memiliki pola pikir yang sederhana, mampu menyelesaikan suatu permasalahan tanpa memakan banyak waktu. Pada penerapannya, algoritma *brute force* bekerja dengan pola pencocokan dari arah kanan ke kiri dalam pencocokan *pattern* [4].

Pada penelitian implementasi algoritma *brute force* sebagai mesin pencari (*search engine*) berbasis web pada *database* [3] disebutkan bahwa mesin pencari atau *search engine* dapat mempermudah memperoleh informasi yang dibutuhkan dalam segala bidang kehidupan termasuk dalam hal suster dan dokter pada RS. Husada. Pembuatan mesin pencarian dirasa cukup dibutuhkan untuk membantu petugas medis mengetahui kandungan zat pada obat tertentu. Tujuan penelitian ini mengurangi waktu yang dibutuhkan dalam mencari informasi yang dibutuhkan. Dalam pembuatan *search engine* ini, dilakukan dengan menambahkan algoritma *brute force*. Algoritma *brute force* dipilih karena mampu memecahkan masalah dengan pendekatan yang mudah [2]. *search engine* pada penelitian ini dibuat dengan menggunakan bahasa pemrograman PHP dengan basis data MySQL. Pada penelitian ini menghasilkan *search engine* yang dapat memasukkan kata kunci lebih dari satu serta mampu menentukan urutan paling atas berdasarkan banyak nya kata kunci yang ditemukan dalam dokumen tersebut [3]. Tujuan dari penelitian ini adalah untuk membuat aplikasi yang meningkatkan akurasi pencarian pada fitur searching aplikasi “Calonku”.

2. METODOLOGI PENELITIAN

2.1. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode *waterfall* [5]. Alur pengembangan sistem menggunakan *waterfall* dapat dilihat pada Gambar 1.



Gambar 1. *Waterfall* Pengembangan Sistem [5]

Metode *waterfall* memiliki tahap- tahap pengembangan sistem sebagai berikut :

2.1.1. Tahap *Communication (Project Initiation & Requirements Gathering)*

Tahap *Communication* merupakan tahap analisis kebutuhan sistem dengan menganalisis kebutuhan sistem serta observasi pengumpulan data [5]. Terdapat dua jenis kebutuhan sistem yang dibutuhkan yaitu : kebutuhan fungsional dan kebutuhan non fungsional.

a. Kebutuhan Fungsional

Kebutuhan fungsional adalah bagaimana *website* Calonku merespon *input* yang dimasukkan. *Website* Calonku dapat melakukan pencarian berdasarkan subjek nama, dapat melakukan *login*, serta pengelolaan data bagi pihak admin.

b. Kebutuhan Non Fungsional

Kebutuhan Non Fungsional. Diartikan sebagai fungsi yang ditawarkan oleh *website* Calonku seperti keakuratan hasil pencarian, pilihan warna yang ringan dan desain *website* yang *user friendly*.

c. Metode Pengumpulan Data

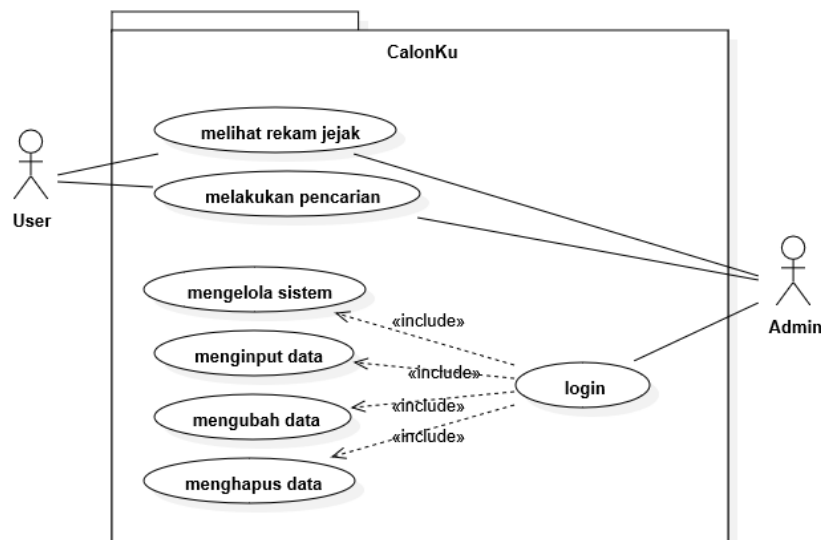
Dalam proses pengumpulan data pada penelitian ini, dilakukan dengan Metode Observasi dan Studi Literatur. Data terkait yang dikumpulkan didapatkan dari laman KPU resmi dan mengutip dari berbagai literatur buku, jurnal dan thesis terkait. Pengumpulan studi literatur dibutuhkan untuk menyusun penelitian dengan teori yang relevan dan dapat dipertanggung jawabkan

2.1.2. Planning (Estimating, Scheduling, Tracking)

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko-resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan *tracking* proses pengerjaan sistem [5]. Pada penelitian ini digunakan basis data menggunakan MySQL *database*, penulisan *script* sistem menggunakan teks editor *Sublime Text 3*, dan rancangan diagram menggunakan *Star UML*.

2.1.3. Modeling (Analysis & Design)

Tahapan ini adalah tahap perancangan dan pemodelan sistem yang berfokus pada perancangan struktur data, arsitektur *software*, tampilan *interface*, kerangka *usecase* dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan [5]. Tampilan *usecase* pada *website Calonku* dapat dilihat pada Gambar 2.



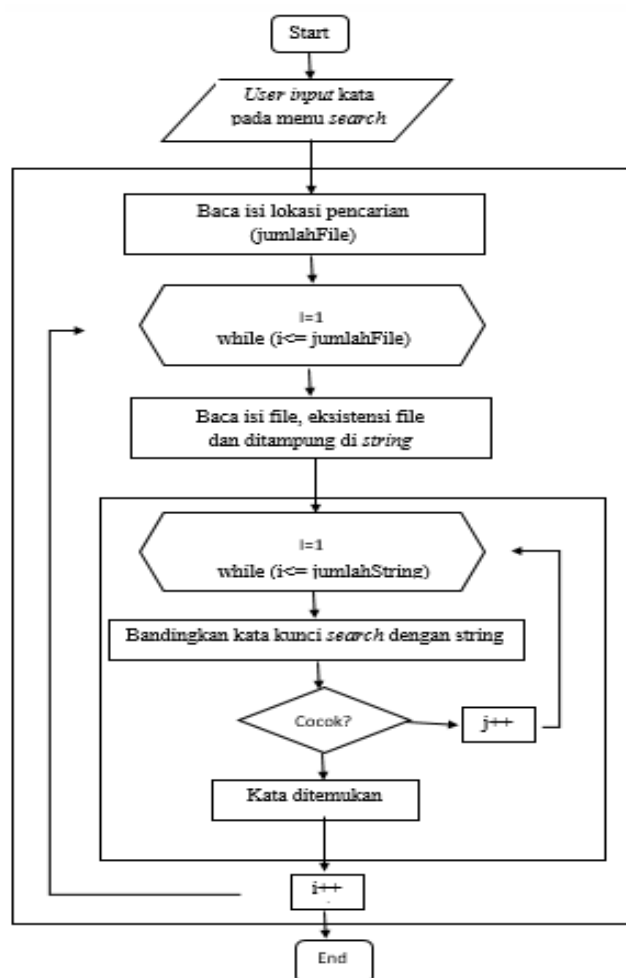
Gambar 2. Usecase pada *website Calonku*.

Usecase diagram pada Calonku terdapat 2 jenis aktor yaitu admin dan user. Admin bertanggungjawab untuk mengelola sistem sedangkan user adalah pihak yang mengunjungi *website* Calonku. *Usecase* pada Calonku terdapat 7 *usecase* dengan 4 *tool include*. Seperti yang terlihat pada Gambar 2, admin harus melakukan *login* sebelum melakukan akses ke Calonku.

2.1.4. Construction (Code & Test)

Tahapan ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk bahasa yang dapat dibaca oleh mesin [5]. Gambar 3 menggambarkan *flowchart* proses pencarian pada Calonku. Seperti terlihat pada *flowchart* Gambar 3, proses pencarian dimulai dengan memasukkan kata inputan. Proses pencocokan *string* dilakukan menggunakan algoritma *brute force*, algoritma *brute force* sendiri adalah metode yang menggunakan arah dari kanan ke kiri dalam pencocokan *pattern* [4]. Algoritma *brute force* melakukan pencocokan *pattern* dari awal teks, Algoritma *brute force* melakukan pergeseran pencocokan dari kiri ke kanan hingga karakter ditemukan, penggeseran *pattern* terus dilakukan sampai berada di ujung teks [6]. Pada Gambar 3 terlihat proses perbandingan dilakukan dengan membandingkan setiap karakter pada *string* file dengan *string* yang dicari. Apabila terdapat 1 karakter saja yang tidak sesuai akan dilakukan perpindahan dengan menggeser ke kanan sejauh *n* karakter. Apabila ditemukan *string* yang cocok, aplikasi akan

memberikan informasi pesan ditemukan. Apabila input tidak ditemukan, proses pencarian akan dilanjutkan ke file lainnya. Pencarian pada file berikutnya diawali dengan pembacaan seluruh isi file yang dikonversi menjadi string dan dilanjutkan proses lainnya. Proses yang dilalui sama halnya pada file sebelumnya. Proses pembacaan semua file yang memiliki ekstensi sesuai dengan batasan yang ditetapkan dan proses pembacaan file dilakukan satu persatu sampai semua file diproses. File yang ditemukan berkemungkinan lebih dari satu karena proses pencarian dilakukan terhadap semua file yang ada.



Gambar 3. Flowchart Proses Search Brute Force.

Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki. Untuk sistem pengujian dilakukan dengan menghitung performance testing pada website Calonku, pada percobaannya dimasukkan input subjek nama Ahmad seperti dalam tabel 2 pada bab 4. Perhitungan *performance testing* dilakukan dengan menghitung nilai- nilai TP (*True Positive*), TN (*True Negative*), FP (*False Positive*), FN (*False Negative*) terlebih dahulu.

2.1.5. Deployment (Delivery, Support, Feedback)

Tahapan terakhir ini merupakan tahapan implementasi software ke customer, perbaikan software, evaluasi software, dan pengembangan software berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya[5].

2.2. Algoritma Brute Force

Algoritma *brute force* adalah metode yang menggunakan arah dari kanan ke kiri dalam pencocokan *pattern* [3]. Algoritma *brute force* bergerak dari kiri ke kanan, memiliki kesederhanaan dalam penyelesaian suatu permasalahan dan tidak membutuhkan suatu pemikiran yang lama. Algoritma *brute force* dinilai metode yang paling baik dalam pencarian pola sehingga menghasilkan waktu pencarian yang lebih cepat [7].

2.1.1 Pencocokan Brute Force

Algoritma *brute force* melakukan proses pencarian dengan langkah– langkah pencocokan string menurut Danuri (2016)[6] ialah sebagai berikut:

- a). Algoritma *brute force* mulai mencocokkan *pattern* dari awal teks.
- b). Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per-karakter *pattern* dengan karakter pada teks yang bersesuaian, sampai salah satu kondisi berikut terpenuhi:
 - Karakter di *pattern* dan di teks yang dibandingkan tidak cocok.
 - Semua karakter di *pattern* cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
- c). Algoritma kemudian terus menggeser *pattern* sebesar satu ke kanan, dan mengulangi langkah ke-2 sampai *pattern* berada di ujung teks.

2.3. Sistem Temu Kembali Informasi

Information Retrieval System (Sistem Temu Kembali Informasi) merupakan sistem yang digunakan untuk menemukan informasi yang relevan dengan kebutuhan dari penggunaanya [8]. Adapun pengertian lain dari Sistem Temu Kembali Informasi menyatakan bahwa Sistem Temu Kembali Informasi ialah proses yang berhubungan dengan representasi, penyimpanan, pencarian dengan pemanggilan informasi yang relevan dengan kebutuhan informasi yang diinginkan pengguna [9]

a. Recall

Recall adalah proporsi jumlah dokumen yang dapat ditemukan-kembali oleh sebuah proses pencarian pada sistem IR. Perolehan (*Recall*) berhubungan dengan kemampuan sistem untuk memanggil dokumen yang relevan [9]. *Recall* dapat dihitung dengan Persamaan 1:

$$R = \frac{\text{Jumlah dokumen relevan yang ditemukan}}{\text{Jumlah semua dokumen yang ditemukan}} \quad (1)$$

b. Precision

Precision adalah jumlah kelompok dokumen relevan dari total jumlah dokumen yang ditemukan oleh sistem. Dalam hal ini, *precision* lebih mendefinisikan tingkat “gangguan” dari informasi yang ditampilkan [9]. *Precision* berkaitan dengan kemampuan sistem untuk tidak memanggil dokumen yang tidak relevan. Untuk menghitung nilai *precision* dapat menggunakan persamaan 2:

$$P = \frac{\text{Jumlah dokumen relevan yang ditemukan}}{\text{Jumlah dokumen yang ditemukan dalam pencarian}} \quad (2)$$

c. Accuracy

Accuracy didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai actual. *Accuracy* sangat bagus kita gunakan sebagai acuan performansi algoritma jika dataset kita memiliki jumlah data *false negatif* dan *false positif* yang sangat mendekati (*Symmetric*). Namun jika jumlahnya tidak mendekati, maka

sebaiknya gunakan F1 Score sebagai acuan [9]. Untuk menghitung nilai *accuracy* dapat menggunakan persamaan 3:

$$P = \frac{\text{Jumlah dokumen relevan yang ditemukan}}{\text{Jumlah seluruh dokumen dalam pencarian}} \quad (3)$$

d. Confusion Matrix

Confusion Matrix
melakukan

Tabel 1
Adapun Tabel 1

Actual Class

merupakan salah satu alat pengukur yang memiliki fungsi untuk pengukuran ketika melakukan analisis terhadap sebuah *classifier*. mengilustrasikan *confusion p* pada *accuracy*, *recall* dan *precision*. *Confusion Matrix* digambarkan sebagai berikut:

Tabel 1. Confusion Matrix

		Predicted Class			
		Yes	No	Total	
FP= False Positive Negative	Yes	TP	FN	P	TP = True Positive
	No	FP	TN	N	TP = True
	Total	P'	N'	P+N	= False Negative

Confusion Matrix dapat digunakan untuk mengukur sebuah evaluasi terhadap *accuracy*, *precision* dan *recall* [10]. Adapun Persamaan 4 *accuracy*, *recall*, dan *precision* menjadi :

$$A = \frac{TP+TN}{TP+TN+FP+FN} \quad R = \frac{TP}{TP+FN} \quad P = \frac{TP}{TP+FP} \quad (4)$$

3. HASIL DAN PEMBAHASAN

3.1. Hasil Penelitian

Setiap lima tahun sekali, Indonesia melakukan pemilihan umum untuk memilih para pemimpin baru baik dari calon eksekutif maupun calon legislatif. *website* “Calonku” sengaja dibuat untuk menghadapi tahun politik tersebut. *website* “Calonku” dibuat untuk mempermudah pemilih dalam menggali informasi tentang para kandidat pemimpin. *website* “Calonku” tidak hanya berisikan biodata calon namun juga riwayat pekerjaan, daftar prestasi serta visi dan misi apabila terpilih. Data yang ditampilkan dihimpun langsung dari laman resmi KPU sehingga data yang ditampilkan dapat dipertanggung jawabkan.

3.1. Implementasi

Implementasi pada penelitian ini ialah menggunakan algoritma *brute force* pada proses pencarian data di *website* “Calonku”. Penyaringan kata kunci hanya dilakukan pada nama calon pemimpin saja. Seperti halnya mesin pencari pada umumnya, sistem akan memanggil hasil kata kunci yang ditemukan ke baris pertama hasil pencarian. Berikut adalah tampilan halaman menu yang terdapat pada sistem serta penjelasannya.

3.1.1 Tampilan Halaman *Home*

Setelah admin melakukan *login*, selanjutnya akan muncul halaman utama dengan beberapa menu seperti kolom *search*, input data, perbandingan dan juga *change password* seperti yang terlihat pada Gambar 5.

CalonKu Home Input Calon Perbandingan Change Password Logout

Home : Pencarian Calon

Cari nama calon ...

Cari

Nama Calon : Drs. H. Almuzzammil Yusuf, MS.I

Tipe : DPRD Kab/kota

Nama : Drs. H. Almuzzammil Yusuf, MS.I
 TTL : Tanjung Karang, 06/06/1965
 Umur : 53 tahun
 Jenis Kelamin : laki laki
 Agama : Islam
 Alamat : Jl. Parkit/Musyawah, RT.004/RW.01, Kelurahan Sawah Lama, Ciputat, Kota Tangerang Selatan, Banten - RT / sebutan lain : 004 - RW / sebutan lain : 01 - Kelurahan : Sawah Lama - Kecamatan : Ciputat - Kabupaten / kota : Tangerang Selatan - Provinsi : Banten

Partai Pengusung : PKS
 Nomor Urut Parpol : 03
 Nomor Urut Bakal Calon : 01
 Keluarga, Istri : Nurul Hidayati K. Ubaya - Jumlah Anak : 6

Gambar 5. Tampilan Halaman Utama

3.1.1. Tampilan Menu *Search*

Pada menu ini, *user* bisa melakukan *searching* dengan memasukkan nama yang diinginkan. Apabila nama cocok dengan data yang tersimpan, maka sistem akan mengindikasikan keberhasilan proses *search* dengan memberi tanda merah pada huruf, seperti yang terlihat pada Gambar 6.

Home : Pencarian Calon

siti

Cari

Nama Calon : Dr. Ir. **Siti** Nurbaya Bakar M.sc

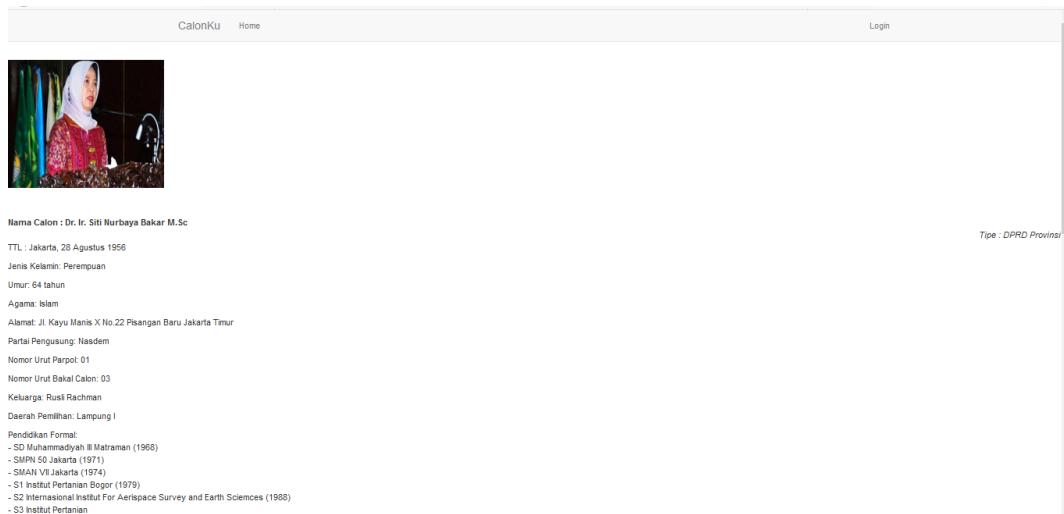
Tipe : DPRD Provinsi

Nama : Dr. Ir. **Siti** Nurbaya Bakar M.sc
 TTL : Jakarta, 28 Agustus 1956
 Umur : 64 tahun
 Jenis Kelamin : Perempuan
 Agama : Islam
 Alamat : Jl. Kayu Manis X No.22 Pisangan Baru Jakarta Timur
 Partai Pengusung : Nasdem
 Nomor Urut Parpol : 01
[View More >>>](#)

Gambar 6. Tampilan Menu *Search*

3.2.1. Tampilan Halaman Calon

Pada menu ini, *user* dapat menemukan nama calon yang diinginkan. Menu *view more* digunakan untuk melihat secara lengkap data seorang kandidat calon pada paragraf kolom nama calon tersebut. Menu *view more* tersebut kemudian menampilkan halaman berisi foto dan deskripsi lengkap calon tersebut seperti yang terlihat pada Gambar 7.



Gambar 7. Tampilan Halaman Calon

3.2. Hasil Pengujian

3.3.1. Performance Test

Dalam mengukur efektivitas OPAC “Calonku” maka dilakukan penghitungan *accuracy*, *recall* dan *precision* dengan menggunakan pendekatan subjek nama Ahmad. Tabel 2 Menunjukkan hasil penelusuran dengan penghitungan *accuracy*, *recall* dan *precision* dengan menggunakan pendekatan subjek nama Ahmad. Di bawah ini adalah tabel *Predicted Value* nama Ahmad

Tabel 2. *Predicted Value* dengan Pendekatan Subjek Nama Ahmad

No	Nama	TP	TN	FP	FN
1	Ahmad	11	2	2	1
2	Ahma	14	1	1	0
3	Ahmat	1	0	0	0
4	Ahmadin	1	0	0	0

Data-data pada tabel 2 apabila diproses dengan menggunakan rumus *accuracy*, *recall* dan *precision* perhitungan nama Ahmad menjadi:

$$A = \frac{TP+TN}{TP+TN+FP+FN} = \frac{27+3}{27+3+3+1} = \frac{30}{34} = 0,88$$

$$R = \frac{TP}{TP+FN} = \frac{27}{27+1} = 0,96$$

$$P = \frac{TP}{TP+FP} = \frac{27}{27+3} = 0,9$$

Berikut adalah Tabel *Predicted Value* dengan lebih banyak nama

Tabel 3. Hasil perhitungan *Predicted Value* 25 input yang berbeda

No	Nama	TP	TN	FP	FN
1.	Benny	6	2	1	2
2.	Chareta	4	1	1	2
3.	Drs.	7	1	1	-
4.	Dra	7	2	2	1
5.	Eko	6	2	1	1
6.	Eddy	5	1	2	1
7.	Frans	4	1	1	1
8.	F.X	3	-	-	-
9.	Gatot	5	1	2	1
10.	Heri	7	1	2	1
11.	H.	25	1	2	3
12.	Ir.	8	1	1	1
13.	Jannah	5	1	1	1
14.	Ketut	3	1	1	1
15.	Muhammad	5	1	1	1
16.	Nur	7	1	1	1
17.	Ophia	3	1	1	1
18.	Reza	4	1	1	1
19.	Rasyide	3	1	1	1
20.	Sri	4	1	1	1
21.	Tri	4	1	1	2
22.	Taufik	3	1	1	1
23.	Umar	5	1	2	1
24.	Yamin	6	1	1	1
25.	Zul	5	1	1	1
Total		144	28	29	26

Tabel 3 merupakan hasil perhitungan dari nilai TP (*True Positive*), TN (*True Negative*), FP (*False Positive*), FN (*False Negative*) dari masing masing nama pada tabel. Tabel 3 Menunjukkan jumlah TP=144, TN=28, FP=29, FN=26. Dengan nilai tersebut, dapat dihitung nilai *accuracy*, *recall* dan *precision* dengan rumus:

$$A = \frac{TP+TN}{TP+TN+FP+FN} = \frac{144+28}{144+28+29+26} = \frac{172}{227} = 0,76$$

$$R = \frac{TP}{TP+FN} = \frac{144}{144+26} = 0,85$$

$$P = \frac{TP}{TP+FP} = \frac{144}{144+29} = 0,83$$

4.3.2. Analisis Hasil Pengujian

Pencarian *basic search* untuk nama Ahmad mendapatkan nilai *accuracy* sebesar 0,88 *recall* sebesar 0.96 dan nilai *precision* sebesar 0,9. Angka ini didapatkan dari 3 kemungkinan similaritas dari nama Ahmad, yaitu: Ahma, Ahmat, dan Ahmadin. Pengujian juga dilakukan pada 26 nama lain dengan jumlah 1 suku kata nama seperti pada Tabel 2. Rata-rata didapatkan rata-rata *accuracy* sebesar 0,76 *recall* sebesar 0,85 dan rata-rata *precision* sebesar 0,83. Jika menganalisis pada Tabel 1 dan Tabel 2, terlihat bahwa nilai *recall* dan *precision* berbanding lurus. Namun besaran nilai *precision* tidak dipengaruhi signifikan oleh nilai *recall*. Secara interpretasi, nilai *recall* dan *precision* termasuk pada kategori tinggi yang artinya semakin tinggi *precision* semakin efektif sebuah sistem tersebut.

4. KESIMPULAN

Kesimpulan dalam penelitian ini adalah website “Calonku” dapat melakukan fungsi pencarian sebagai mana yang diharapkan. Berdasarkan perhitungan *performance testing*, nilai rata rata *accuracy* sebesar 0,76 *recall* sebesar 0.85 dan rata-rata *precision* sebesar 0,83. Nilai perhitungan tersebut diperoleh berdasarkan perhitungan dengan 25 input yang berbeda. Hal ini dipengaruhi oleh rendahnya nilai FN (*False Negative*) dan TN (*True Negative*) yang didapat oleh sistem “Calonku”.

DAFTAR PUSTAKA

- [1] F. R. Wutono, "Makalah IF2211Strategi Algoritma –Sem. IITahun 2013/2014Penerapan Algoritma Pattern Matching untuk Mengidentifikasi Musik Monophonic," Institut Teknologi Bandung, Bandung, 2014.
- [2] M. S. and R. M. , "Pencocokan String Berdasarkan Kemiripan Ucapan (Phonetic String Matching) dalam Bahasa Inggris," in *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, Yogyakarta, 2005.
- [3] F. S. M. A. and B. W. Santoso, "IMPLEMENTASI ALGORITMA BRUTE FORCE SEBAGAI MESIN PENCARI (SEARCH ENGINE) BERBASIS WEB PADA DATABASE," *Jurnal SISFOTEK Global*, vol. 6, no. 1, pp. 1-8, 2016.
- [4] H. R. Pratiwi, D. S. Sihabudin and A. W. , "Prototype Aplikasi SMS Content Filtering Menggunakan Metode String Matching," *Jurnal Teknik Informatika*, vol. 1, 2012.
- [5] R. S. Pressman, *Rakayasa Perangkat Lunak Pendekatan Praktisi Buku I*, Yogyakarta: Andi, 2015.
- [6] D. "Pencarian File Teks Berbasis Content dengan Pencocokan String Menggunakan Algoritma Brute force," *Scientific Journal of Informatics*, vol. 3, no. 1, 2016.
- [7] J. T. Ardiansyah, A. P. Wibawa and T. W. , "Penerapan Algoritma Start End Mid Untuk Mendeteksi Kesalahan Logika Structured Query Language Seminar Nasional Teknologi Informasi Dan Multimedia," in *STIMK AMIKOM*, Yogyakarta, 2016.
- [8] N. K. "Konsep Dasar Information Retrieval System / Sistem Temu Kembali Informasi. Materi Kuliah Departemen Ilmu Informasi Dan Perpustakaan Universitas Padjadjaran," Universitas Padjadjaran,

Bandung, 2014.

- [9] J. H. "Penggunaan Bahasa Ilmiah dan Kosa Kata Terkendali dalam Sistem Temu Balik Informasi Berbasis Teks," *Jurnal Studi Perpustakaan dan Informasi*, vol. 2, no. 2, 2006.
- [10] M. S. and G. L. , "A Systematic Analysis of Performance Measures for Classification Tasks," *Information Processing and Management*, vol. 45, no. 4, pp. 427-437, 2009.