

# Analisis Ketetapan Respon Chatbot Menggunakan Algoritma Boyer Moore

## *Analysis of Chatbot Response Constancy Using Boyer Moore Algorithm*

<sup>1</sup>Aldis Gandi Mitra Sanjung, <sup>2</sup>Norhikmah\*,

<sup>1</sup>Informatika, Ilmu Komputer, Universitas AMIKOM Yogyakarta

<sup>2</sup>Sistem Informasi, Ilmu Komputer, Universitas AMIKOM Yogyakarta  
Jln. Ring Road Utara, Condong Catur, Sleman, Yogyakarta, Indonesia

\*e-mail: [hikmah@amikom.ac.id](mailto:hikmah@amikom.ac.id)

(*received*: 23 November 2021, *revised*: 25 November 2021, *accepted*: 28 November 2021)

### Abstrak

*Amikom Computer Club* atau biasa disebut dengan nama AMCC adalah salah satu unit kegiatan mahasiswa bidang keilmuan di Universitas Amikom Yogyakarta. AMCC terdiri dari pengurus dan *member*, Seiring bertambahnya jumlah *member* AMCC disetiap tahun, maka pengurus mengalami kesulitan dalam memberikan pelayanan dalam menjawab pertanyaan *member* dengan cepat sesuai butuhan *member*. Maka dari itu dibutuhkan sebuah sistem Chatbot yang dibangun untuk membantu pengurus dalam menjawab berbagai pertanyaan dari *member* menggunakan Algoritma *Boyer Moore*, dengan cara kerja algoritma yang bergerak membandingkan karakter dari kanan ke kiri atau disebut dengan *string matching*, sehingga mempersingkat waktu pencarian informasi. Hasil penelitian ini adalah bahwa sistem chatbot bisa merespon dengan baik pertanyaan *member*, dan hasil diuji menggunakan *User Acceptance Test*, chatbot hanya gagal menjawab 4 pertanyaan dari total 50 pertanyaan, serta mendapatkan akurasi 70% dengan pengujian menggunakan metode confusion matrix.

**Kata kunci:** *Chatbot, Boyer Moore, String Matching*

### Abstract

*Amikom Computer Club* or commonly referred to by the name AMCC is one of the scientific student activity units at Amikom University Yogyakarta. AMCC consists of administrators and members. As the number of AMCC members increases every year, the management has difficulty providing services in answering member questions quickly according to members' needs. Therefore we need a Chatbot system that is built to assist administrators in answering various questions from members using the Boyer Moore Algorithm, with the workings of the algorithm that moves to compare characters from right to left or called string matching, thus shortening the information search time. The results of this study are that the chatbot system can respond well to member questions, and the results are tested using the User Acceptance Test, the chatbot only fails to answer 4 questions out of a total of 50 questions, and obtains 70% accuracy by testing using the confusion matrix method.

**Keywords:** *Chatbot, Boyer Moore, String Matching*

## 1 Pendahuluan

Setiap perguruan tinggi memiliki UKM atau dikenal dengan nama unit kegiatan mahasiswa yang dimana menjadi salah satu wadah mahasiswa untuk berkarya dan kreasi. salah satu UKM yang dimiliki Universitas AMIKOM adalah AMCC atau kepanjangan dari *Amikom Computer Club*, yang merupakan organisasi bidang keilmuan yang memiliki 5 divisi yaitu, *web programming, mobile programming, computer network, desktop programming*, dan *hardware software*. AMCC rutin dalam melaksanakan pelatihan kepada members-nya dalam tiap minggu. tidak hanya pelatihan mingguan, AMCC juga

<http://sistemasi.ftik.unisi.ac.id>

mengadakan pelatihan tahunan untuk mahasiswa baru di Universitas Amikom Yogyakarta yang biasa disebut *Introduction to Computer* sebelum memasuki perkuliahan. dalam penyebaran informasi AMCC memiliki group chat setiap masing-masing divisinya. Di dalam group chat tersebut pengurus ataupun member AMCC dapat berinteraksi guna saling bertukar informasi perihal AMCC ataupun diluar AMCC, memecahkan masalah, hingga berdiskusi mengenai permasalahan teknis program yang dihadapi, tetapi sering sekali pengurus terlambat dalam menanggapi pertanyaan yang diajukan oleh *members*, sehingga informasinya lambat didapat oleh member dan menyebabkan member jarang berinteraksi dengan pengurus maupun sesama member di group chat tersebut.

Maka dari itu dibutuhkannya sebuah sistem *chatbot* yang berfungsi sebagai *Virtual Assistant* untuk mewakili pengurus AMCC dalam merespon pertanyaan *member*. *String searching* merupakan suatu proses yang umum dilakukan dalam proses-proses yang dilakukan komputer karena teks merupakan bentuk utama penyimpanan data. Terdapat beberapa macam cara yang dapat dilakukan untuk mencari sebuah *string* pada kumpulan *string* lain yang lebih besar. Beberapa diantaranya adalah *algoritma Boyer-Moore*[1]. Algoritma *boyer-moore* bekerja dengan cara membandingkan pola mulai dari kanan ke kiri, sehingga algoritma ini menjadi solusi pencarian yang efisien. Jika terjadi ketidakcocokkan pola *string* maka pola akan bergerak ke arah kiri, gerakan ini akan memberikan informasi berapa banyak pergeseran untuk mencocokkan karakter yang sesuai dengan pola awal. Algoritma *boyer-moore* memiliki keunggulan dalam waktu menemukan pola dalam ukuran *file* yang lebih besar. Dari hasil pengujian *verifikasi*, pengujian *validitas*, dan pengujian *prototipe* yang dilakukan pada sistem *chatbot* dapat berjalan dengan baik sesuai dengan perencanaan[2]. Implementasi Algoritma *Boyer Moore*, *Turbo Boyer Moore*, *Tuned Boyer Moore* dalam Pencarian *Strng*” dijelaskan bahwa algoritma *Boyer-Moore* memiliki waktu pencarian tercepat dari tiga varian *Boyer-Moore* yang telah ditentukan Algoritma *Turbo Boyer-Moore* merupakan algoritma tercepat kedua dan yang menempati urutan ketiga adalah *Tuned Boyer-Moore*[1]. Implementasi Algoritma *Boyer-Moore* pada Aplikasi Kamus Kedokteran Berbasis Android” yang menerapkan Algoritma *Boyer Moore* menyatakan bahwa dengan algoritma *Boyer Moore* berhasil dilakukan dengan presentase sebesar 100% dari responden menyatakan bahwa proses pencarian istilah pada aplikasi Kamus Kedokteran dapat memberikan penjelasan atau hasil yang sesuai dengan yang di harapkan[3].Perbedaan dengan penelitian ini adalah menganalisis ketepatan respon aplikasi chatbot yang dibangun menggunakan algoritma *boyer moore* dan pengujian menggunakan UAT (*User Acceptence Test*) dan *confusion matrix*.

## 2. Tinjauan Literatur

*Chatbot* sebagai pengganti peran *customer service*, yang dimana *chatbot* dapat merespon pertanyaan pelanggan dengan cepat dan dapat bekerja hingga 24 jam penuh[4].Sistem *Chatbot* akan digunakan untuk pertanyaan sederhana[5]. *chatterbot* salah satu produk *intelligence* buatan yang dapat melakukan percakapan dalam bentuk teks (*chat*) dengan manusia dan hal ini dapat dicapai dengan menerapkan algoritma tertentu agar chatterbot dapat mencari jawaban dari beberapa kata (*keyword*) yang diberikan oleh penanya[6].Chatbot Interaktif dimana user bisa berinteraksi dengan chatbot yang terhubung ke database. Database *chatbot* ini menyediakan kata kunci serta kunci jawaban yang diatur oleh admin[7]. *chatbot* yang dapat membantu pengguna mendapatkan informasi[8]

Algoritma *Boyer-Moore* melakukan perbandingan dimulai dari kanan ke kiri, tetapi pergeseran *window* tetap dari kiri ke kanan[3]. Algoritma *Boyer-Moore* dianggap sebagai salah satu dari algoritma pencarian string yang dikenal efisien kinerjanya dalam melakukan pencocokan pola tunggal. Algoritma *Boyer Moore* telah dibuktikan sebagai salah satu algoritma yang paling efisien dalam aplikasi pencarian *string* dengan menggunakan *natural language* (bukan *binary language*). Algoritma ini telah sering diimplementasikan untuk fungsi “*Search*” dan “*Subtitute*” pada *text editor*. Padasarnya cara kerja algoritma ini mirip dengan algoritma *Knuth Morris Pratt* (KMP) dimana kedua algoritma ini akan melakukan lompatan pengecekan dalam proses pencarian *string*. Namun berbeda dengan algoritma *Knuth Morris Pratt*, algoritma *Boyer Moore* ini melakukan perbandingan *pattern* mulai dari kanan ke kiri[9].Pada proses pencarian string algoritma *Boyer Moore* membaca karakter karakter dari pola (*pattern*) dari kanan ke kiri. Dalam kasus

<http://sistemasi.ftik.unisi.ac.id>

dimana jumlah karakter pada *pattern* lebih sedikit daripada jumlah karakter pada teks maka algoritma tersebut menggunakan 2 buah fungsi precomputed. Dua buah fungsi pengubah ini disebut *good suffix shift*. Aturan pada *good suffix shift* bertujuan untuk menangani kasus dimana terdapat pengulangan karakter pada *pattern*[10]. Algoritma *Boyer Moore* menggunakan dua fungsi *shift* yaitu *good-suffix shift* dan *bad-character shift* untuk mengambil langkah berikutnya setelah terjadi ketidakcocokan antara karakter *pattern* dan karakter teks yang dicocokkan [11]. Algoritma *Boyer Moore* mempunyai keunggulan dalam waktu menemukan *pattern* yang akan dicari dalam ukuran file yang lebih besar[12]. sebuah algoritma yang disajikan yang mencari lokasi, "i," dari kemunculan pertama karakter *string*, "pat," di *string* lain, "*string*." Selama operasi pencarian, karakter *pat*[13].

## 2 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah sebagai berikut: (1) Pengumpulan data yaitu dilakukan observasi dengan cara mempelajari pola pertanyaan yang diajukan member kepada pengurus melalui chatting group. Melakukan tanya jawab langsung dengan pengurus inti dan departemen humas sebagai tim yang menangani pertanyaan ataupun keluhan yang diajukan oleh member AMCC. (2) Pembuatan *Brain File* menggunakan format *Javascript Object Notion*(JSON), (3) Analisis kata kunci yaitu, sebelum di proses menggunakan algoritma *boyer moore* diperlukan sebuah proses untuk pengambilan kata kunci dari *brain file*. (4) Analisis *bot* program, yaitu analisis pertanyaan pengguna menggunakan *chatbot* (5) Analisis *Boyer Moore* yaitu menggunakan algoritma *Boyer Moore* untuk analisis ketepatan respon *chatbot* (6) Pengujian, dimana dalam penelitian ini menggunakan 2 metode pengujian yaitu: *User Acceptance Test* (UAT) memfokuskan pada hasil respon pertanyaan dari beberapa orang yang berbeda. Pengujian ini bertujuan untuk melihat kemampuan *chatbot* dalam merespon pertanyaan dari member AMCC. (2) *Confusion Matriks* yaitu metode yang digunakan untuk melakukan perhitungan akurasi yang bertujuan untuk menguji seberapa akurat *chatbot* dalam merespon.

## 3 Hasil dan Pembahasan

Pada hasil pembahasan pada penelitian ini melakukan pembahasan yang pertama adalah mengumpulkan data dengan cara observasi dan wawancara, yang kedua menganalisis *algoritma boyer moore* dan tahapan terakhir dilakukan pengujian dengan menggunakan *Acceptence Test*(UAT) dan *confusion matrix*

### 3.1 Data

Dari hasil pengumpulan data dari observasi dan wawancara didapatkan beberapa pertanyaan dan jawaban serta kata kunci yang akan digunakan dalam pencarian pada *chatboat*, data detail terdapat pada Tabel 1 di bawah ini:

Tabel 1. Contoh Data

Pertanyaan	Jawaban	Kata Kunci
kak tanya, cara pindah divisi gimana ya kak ?	Koordinasi dengan koor divisi yang bersangkutan ya	Divisi Apa
Kapan ini mulai pelatihan?	-	Pelatihan Rutin
Kak ijin bertanya untuk jadwal pelatihan AMCC dimulai bulan apa sama apakah online atau offline untuk kedepannya kegiatan tersebut ?	insyaallah dimulai bulan ini, untuk sementara masih online dulu, mengingat dari kampus juga belum mengizinkan untuk offline	Pelatihan Rutin
Ada pelatihan offline gak kak?	-	Pelatihan Rutin
Setiap hari apa aja kak?	setiap hari sabtu	Pelatihan berapa kali

kalo join pelatihan divisi lain boleh ngga kak?	Boleh kok, nanti bisa hubungi koor divisinya ya	Pindah Divisi
Mau pindah divisi masih bisa ga	bisa kak	Pindah Divisi
Kak saya angkatan 19 sebelumnya ikut di divisi desktop namun sekarang mw mencoba pindah ke divisi mobile programing blh gk?	boleh, langsung pc aku aja ya	Pindah Divisi
mau ikut divisi lain tapi tabrakan , gimana dong?	Kalau mau ikut divisi lain jangan dibarengin 1 sesi gan wkwk. Misal sesi 1 mobile sesi 2 ikut web	Jadwal Bentrok,Jadwal Tabrakan
daftar amcc bayar berapa?	60k gan, yoklah gas daftar	Bayar Berapa
daftar nya offline atau online nih?	cuma online sih, karena pandemi gak buka pendaftaran offline	daftar online dan daftar offline
cuy, pada daftar divisi apa nih? gw bingung njir wkwk	sekedar info sih, semua divisi di amcc terbaik sih, jadi daftar apa aja juga bagus wkwk	divisi yang bagus

### 3.2 Pembuatan Brain File

*Brain File* dibuat menggunakan format *Javascript Object Nation*(JSON). JSON merupakan format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat ulang oleh computer. JSON adalah format teks yang tidak bergantung pada bahasa pemrograman manapun karena menggunakan bahasa yang umum, seperti Gambar 1 di bawah ini:

```

1  [
2  {
3      "kategori" : "pelatihan",
4      "kata_kunci": "pelatihan rutin",
5      "jawaban": "Untuk pelatihan rutin setiap hari sabtu ya kak, dan
6                  pelatihannya juga dua sesi loh yaitu sesi 1 : 12.30-15.00 dan
7                  sesi 2 : 15.30-17.00, jadi bisa menyesuaikan waktu kakak"
8  },
9  {
10     "kategori" : "pelatihan",
11     "kata_kunci": "setiap hari apa",
12     "jawaban": "Untuk pelatihan rutin setiap hari sabtu ya kak, dan
13               pelatihannya juga dua sesi loh yaitu sesi 1 : 12.30-15.00 dan
14               sesi 2 : 15.30-17.00, jadi bisa menyesuaikan waktu kakak"
15  },
16  {
17     "kategori" : "pelatihan",
18     "kata_kunci": "pelatihan berapa kali",
19     "jawaban": "Pelatihan akan diadakan sebanyak 12x pertemuan setiap
20               hari sabtu yang dibagi menjadi 2 sesi, jadi jangan takut
21               ketinggalan pelatihan ya kak :D"
22  },
23  ]
    
```

**Gambar 1. Contoh Pembuatan Brain File Pada Chatbot**

*Brain file* diterapkan dalam *chatbot* berupa *knowledge base* yang menjadi bagian otak dari *chatbot*. Di dalam *brain file* disimpan semua kosakata, kepribadian, respon, dan pengetahuan dari *chatbot*. Untuk kosakata dan jawaban yang menjadi isi dari *brain file* berdasarkan pertanyaan seputar AMCC, Untuk *sample* tabel yang digunakan dalam *brain file* mengakomodasi setiap kosakata yang disediakan serta jawaban dari pertanyaan pada Tabel 2 sebagai berikut:

**Tabel 2. Data Sampel *Brain File***

Kategori	Kata Kunci	Jawaban
Pelatihan	Pelatihan Rutin	Untuk pelatihan rutin akan dilaksanakan hari sabtu ya kak :D bakal ada 2 sesi loh! Sesi 1 di jam 12.30-14.30 Sesi 2 di jam 15.00-16.30 Ditunggu kehadirannya ya!
Pelatihan	Materi Pelatihan	Materi pelatihan bisa akses di akun mediumnya AMCC ya kak, biar gampang kita kasih linknya nih jadi tinggal klik aja! <a href="https://medium.com/amcc-amikom">https://medium.com/amcc-amikom</a>
Event	Daftar Seminar Nasional	Halo kak, untuk <b>Seminar Nasional</b> masih Coming Soon ya so, tungguin info nya di sosmed kita :D
Sosial Media	Sosial Media	Gak mau ketinggalan info AMCC? Yuk ikutin Sosial Media kita Instagram : amccamikom Twitter : amcc_amikom Medium : AMCC Amikom Youtube : AMIKOM Computer Club
Kaderisasi	Jadi Pengurus	Gampang banget kok kak! caranya daftar kader, setelah daftar kader kakak bakal di wawancarain sama temen temen pengurus AMCC, setelah itu tungguin info keterima atau enggak. Jika kakak keterima kader maka akan ada beberapa kegiatan yang harus diikuti dan setelah itu bakal ada wawancara tahap berikutnya untuk menentukan kakak keterima sebagai pengurus atau enggak, gampang kan? :D
dst	dst	dst

### 3.3 Analisis Kata Kunci

Dalam Proses pengambilan kata kunci terdiri dari dua tahap yaitu pertama Proses analisa text, merupakan proses untuk mengetahui maksud dari kalimat berdasarkan kata kunci, ketika kata kunci tersebut ada di *brain file* maka bot akan merespon kalimat tersebut sesuai dengan yang ada di *brain file*. Jika kata kunci tidak ditemukan pada *brain file* maka respon dari bot tersebut seperti Gambar 2 di bawah ini:

```
public static function handle()
{
    $text = self::getText();
    foreach (Client::handle(self::CLIENT_URL) as $pattern) {
        if (BoyerMoore::handle($text[1], $pattern['kata_kunci'])) {
            self::setMessage($pattern['jawaban']);
        }
    }
    return self::sendMessage($text[0], self::getMessage());
}

private static $message = "Maaf kak, kami tidak menemukan jawaban yang cocok
untuk pertanyaan kakak!";
```

Gambar 2. Jawaban ketika tidak menemukan kata kunci

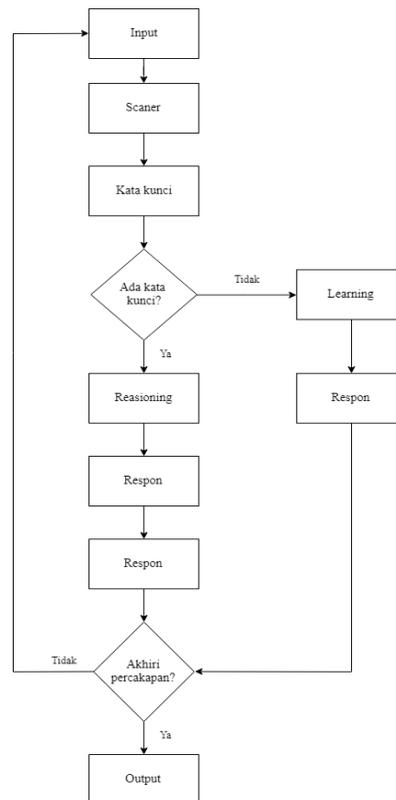
Kedua pengambilan kata kunci, yaitu sebelum di proses menggunakan algoritma *boyer moore* diperlukan sebuah proses untuk pengambilan kata kunci dari brain file dengan menggunakan fungsi seperti pada Gambar 3 di bawah ini:

```
5 class Client {
6
7     public static function handle(string $url)
8     {
9         return json_decode(file_get_contents($url), true);
10    }
```

Gambar 3. Proses Pengambilan Kata Kunci Dari *Brain File*

### 3.4 Analisis Bot Program

Program utama pada *chatbot* disebut bot program dan akan mengakses input pertanyaan dari pengguna, bot program sendiri terdiri dari beberapa komponen yaitu *scanner*, *reasoning*, dan *learning*.

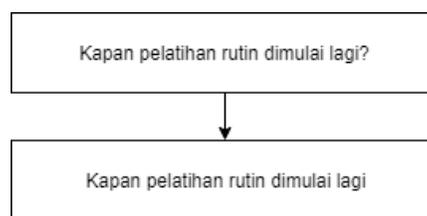


Gambar 4. Alur Bot Program

Gambar 4 menjelaskan alur dari proses yang ada pada bot program diawali dari pertanyaan yang diinputkan akan dilakukan *scanner* untuk menghilangkan simbol dan tanda baca serta mengubah setiap kata menjadi huruf kecil, kosakata yang telah melalui proses *scanner* akan dilakukan pengecekan untuk pencarian kata kunci, jika kata kunci di temukan maka akan di lakukan proses *reasoning* sebagai bentuk respon untuk jawaban dengan kata kunci yang ditemukan sedangkan kosakata selain dari kata kunci diabaikan dan jika kunci tidak ditemukan sistem akan memberikan respon berupa *learning* untuk menanggulangi beberapa pertanyaan yang tidak dapat dijawab oleh *chatbot*. Untuk implementasinya, proses pencocokan kata kunci digunakan Algoritma Boyer Moore.

### 3.4.1 Scanner

Inputan *scanner* berupa kalimat pertanyaan dari *user*. *Scanner* melakukan pemeriksaan karakter per karakter dari pertanyaan yang diinputkan. *Scanner* mengerjakan pengelompokan urutan-urutan karakter ke dalam komponen pokok yaitu: simbol-simbol operator, simbol tanda baca dan lambing benda contoh pada Gambar 5 di bawah ini:



**Gambar 5. Tahapan Scanner**

Berikut merupakan langkah-langkah umum untuk proses *scanner*, dalam contoh kalimat “Kapan pelatihan rutin dimulai lagi?” (1) Periksa seluruh karakter yang terdapat pada kalimat dan menghilangkan simbol-simbol operator dan tanda baca. (2) Mengubah setiap karakter mejadi huruf kecil.

### 3.4.2 Reasoning

Hasil *scanner* akan diproses untuk pencarian kata kunci di *brain file*, jika terdapat kata kunci yang sesuai akan menjadi kosakata yang terpilih untuk menjadi acuan respon untuk proses *reasoning* sedangkan kosakata selain dari kata kunci akan diabaikan.



**Gambar 6. Proses Reasoning**

<http://sistemasi.ftik.unisi.ac.id>

*Reasoning* dilakukan untuk mengembalikan respon yang sesuai dengan kata kunci berupa jawaban yang benar dari pertanyaan yang diinputkan *user*. Dimulai dari masuknya kosakata hasil *scanner* seperti pada Gambar 6 sampai dengan proses untuk menampilkan jawaban dari kosa kata yang terpilih merupakan alur *reasoning* menggunakan Algoritma Boyer Moore.

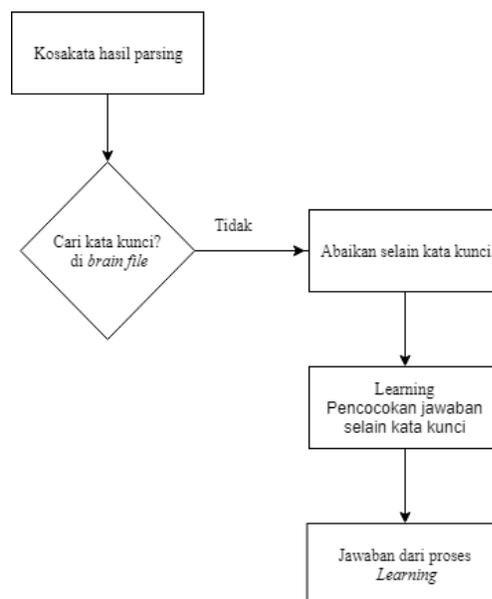
Berikut ini merupakan contoh proses pencocokan kata kunci dengan menggunakan algoritma Boyer Moore.

Teks pertanyaan: “kapan seminar nasional dimulai”

Kata kunci pertama pada *brain file* :”Seminar Umum”

### 3.4.3 Learning

*Learning* diterapkan oleh *chatbot* ketika tidak ada kecocokan antara kalimat yang di ajukan dengan kata kunci yang terdapat dalam *brain file*, maka perlu respon otomatis yang bisa memberikan penjelasan bahwa kalimat yang di ajukan tidak mengandung kata kunci yang ada di *knowledge base*. Hal ini diperlukan bagi *chatbot* untuk memberikan respon dalam berbagai kondisi inputan kalimat atau pertanyaan, dengan penjelasan detail terkait proses *learning*, seperti Gambar 7 di bawah ini.



Gambar 7. Proses *Learning*

## 3.5 Analisis Boyer Moore

*Good Suffix* adalah cara lain untuk mengatakan berapa banyak karakter yang kita dapat lewati jika tidak ada suatu ketidakcocokan[14]. *Bad Character Shift* menunjukkan seberapa banyak pergeseran karakter dapat melompat kedepan dalam teks setelah ketidakcocokan[14]

### Cara Kerja Algoritma Moore

Cara kerja algoritma Boyer Moore menurut (Edgewick,dkk, 2011) adalah sebagai berikut[15],[16]:

1. Buat table pergeseran string yang dicari (S) dengan pendekatan *Match Heuristic* (MH) dan *Occurrence Heuristic* (OH), untuk menentukan jumlah pergeseran yang akan dilakukan jika mendapat karakter tidak cocok pada proses pencocokan dengan *string* (T).

<http://sistemasi.ftik.unisi.ac.id>

2. Jika dalam proses pembagian terjadi ketidakcocokan antara pasangan karakter pada S dan karakter T, pergeseran dilakukan dengan memilih salah satu nilai pergeseran dari dua table analisa *string*, yang memiliki nilai pergeseran paling besar.
3. Dua kemungkinan penyelesaian dalam melakukan pergeseran S, jika sebelumnya belum ada karakter yang cocok adalah dengan melihat nilai pergeseran hanya pada table *occurrence heuristic*: Jika karakter yang tidak cocok tidak ada pada S maka pergeseran adalah sebanyak jumlah karakter pada S, dan jika karakter yang tidak cocok ada pada S, maka banyaknya pergeseran bergantung dari nilai pada table.
4. Jika karakter pada teks yang sedang dibandingkan cocok dengan karakter pada S maka posisi karakter pada S dan T diturunkan sebanyak 1 posisi, kemudian lanjutkan dengan pencocokan pada posisi tersebut dan seterusnya. Jika kemudian terjadi ketidakcocokan karakter S dan T, maka pilih nilai pergeseran terbesar dari dua table analisa pola yaitu nilai dari table *match heuristic* dan nilai table *occurrence heuristic* dikurangi dengan jumlah karakter yang telah cocok.
5. Jika semua karakter telah cocok, artinya S telah ditemukan di dalam T, selanjutnya geser pola sebesar 1 karakter.
6. Kemudian lanjutkan sampai karakter terakhir pada string T.

**Contoh Implementasi Algoritma Boyer Moore untuk pencocokan kata kunci:**

1. Menentukan nilai *Match heuristic* (MH) dan *Occurrence Heuristic*(OH), untuk menentukan jumlah pergeseran yang akan dilakukan jika mendapat karakter tidak cocok pada proses pencocokan dengan teks pertanyaan, seperti Tabel 3 dan Tabel 4 di bawah ini.

**Tabel 3. Tabel Pencacahan Kata Kunci**

Posisi	1	2	3	4	5	6	7	8	9	10	11	12
Kata Kunci	s	e	m	i	n	a	r		u	m	u	m
Pencacahan	11	10	9	8	7	6	5	4	3	2	1	0

**Tabel 4. Occurrence Heuristic(OH)**

Karakter	Shift/Pergeseran(OH)	Pencacahan
m	0	0
u	1	1
m	0	2
u	1	3
	4	4
r	5	5
a	6	6
n	7	7
i	8	8
m	0	9
e	10	10
s	11	11

Pencacahan dimulai dari posisi terakhir string sampai ke posisi awal, dimulai dengan 0. Karakter yang sudah ditemukan (missal karakter “m” dengan nilai 0) jika karakter tersebut ditemukan kembali maka nilainya sama dengan nilai pencacahannya. Jika karakter belum pernah ditemukan, maka nilai pergeserannya sama dengan nilai pencacah, penjelasan detail seperti Tabel 5 di bawah ini.

**Table 5. Match Heuristic (MH)**

Posisi String	Karakter	Shift/Pergeseran(MH)
0	m	1
1	um	9
2	mum	2
3	umum	12
4	_umum	12
5	r umum	12
6	ar umum	12
7	nar umum	12
8	inar umum	12
9	minar umum	12
10	eminar umum	12
11	seminar umum	12

Tabel 6 di bawah ini, memperlihatkan pola yang akan digeser kekiri untuk memperoleh hasil yang cocok (karakter yang tidak harus cocok ditulis dengan huruf besar. Contoh, pada pola “Tin” karakter “in” telah dilakukan pengecekan sebelum”T”). begitu seterusnya.

**Tabel 6. Nilai OH dan MH**

Posisi	1	2	3	4	5	6	7	8	9	10	11	12
Kata Kunci	s	e	m	i	n	a	r		u	m	u	m
Pencacahan	11	10	9	8	7	6	5	4	3	2	1	0
OH	11	10	0	8	7	6	5	4	1	0	1	0
MH	12	12	12	12	12	12	12	12	12	2	9	1

- Setelah nilai OH dan MH didapat selanjutnya melakukan pencocokan kata kunci dengan melakukan pergeseran yang didapat dari hasil perbandingan antara nilai OH dan MH

Langkah 1:

1	2	3	4	5	6	7	8	9	10	11	12	13	14
k	a	p	a	n		s	e	m	i	n	a	r	
s	e	m	i	n	a	r		u	m	u	m		

15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
n	a	s	i	o	n	a	l		d	i	m	u	l	a	i

Pencocokan pertama: ‘a’ dan ‘m’ = tidak cocok

OH = karakter ‘a’ ada dalam kata kunci, maka nilai pergeseran 6

MH = ketidakcocokan karakter ‘m’ pada posisi 12 memiliki nilai pergeseran 1

Dipilih nilai pergeseran OH dan MH yang paling besar yaitu 6

Langkah 2:

1	2	3	4	5	6	7	8	9	10	11	12	13	14
k	a	p	a	n		s	e	m	i	n	a	r	
						s	e	m	i	n	a	r	

15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
n	a	s	i	o	n	a	l		d	i	m	u	l	a	i
u	m	u	m												

Pencocokan pertama : ‘i’ dan ‘m’ = tidak cocok

OH = karakter ‘i’ ada dalam kata kunci, maka nilai pergeseran 8

MH = ketidakcocokan karakter ‘m’ pada posisi 18 memiliki nilai pergeseran 1

Dipilih nilai pergeseran OH dan MH yang paling besar yaitu 8.

Langkah 3:

1	2	3	4	5	6	7	8	9	10	11	12	13	14
k	a	p	a	n		s	e	m	i	n	a	r	

15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
n	a	s	i	o	n	a	l		d	i	m	u	l	a	i
s	e	m	i	n	a	r		u	m	u	m				

Pencocokan pertama : ‘m’ dan ‘m’ = cocok

Pencocokan kedua : ‘i’ dan ‘u’ = tidak cocok

OH = karakter ‘i’ ada dalam kata kunci, maka nilai pergeseran 8

MH = ketidakcocokan karakter ‘u’ pada posisi 25 memiliki nilai pergeseran 9

Dipilih nilai pergeseran OH dan MH yang paling besar yaitu 9.

3. Penelusuran selesai pada posisi karakter ke 26, tapi kata kunci tidak ditemukan dalam *brain file* maka proses selanjutnya adalah membandingkan kembali pertanyaan dengan kata kunci selanjutnya yang terdapat pada *brain file*.

Teks pertanyaan :”kapan seminar nasional dimulai”

Kata kunci kedua pada *brain file* adalah : “seminar nasional”

### 3.6 Proses Algoritma Boyer Moore

Prosedur pencarian *boyer moore* menjadikan nilai OH dan MH sebagai acuan seberapa banyak karakter akan digeser jika mengalami ketidakcocokan. Ditunjukkan pada gambar 8 seperti berikut:

```
1 k?php
2
3 namespace Vendor;
4
5 class BoyerMoore {
6
7     private static function makeCharTable(string $pattern)
8     {
9         $len = strlen($pattern);
10        $table = array();
11
12        for ($i = 0; $i < $len; $i++) {
13            $table[$pattern[$i]] = $len - $i - 1;
14        }
15
16        return $table;
17    }
18
19    private static function search(string $text, string $pattern)
20    {
21        $patlen = strlen($pattern);
22        $textlen = strlen($text);
23
24        $table = self::makeCharTable($pattern);
25
26        for ($i = $patlen - 1; $i < $textlen; ) {
27            $t = $i;
28
29            for ($j = $patlen - 1; $pattern[$j] == $text[$i]; $j--, $i--) {
30                if ($j == 0)
31                    return $i;
32            }
33            $i = $t;
34
35            if (array_key_exists($text[$i], $table)) {
36                $i = $i + max($table[$text[$i]], 1);
37            } else {
38                $i += $patlen;
39            }
40        }
41        return -1;
42    }
43
44    public static function handle(string $text, string $pattern)
45    {
46        return self::search($text, $pattern) == -1 ? false : true;
47    }
48 }
```

Gambar 8. Proses Algoritma Boyer Moore

### 3.7 Pengujian

Pada Penelitian ini menggunakan 2 metode pengujian yaitu UAT (*User Acceptance Test*) dan *confusion matrix* sebagai berikut:

Pengujian *chatbot* dilakukan dengan menggunakan pengujian *User Acceptance Test* (UAT) terhadap pertanyaan dari orang yang berbeda. Pengujian ini bertujuan untuk melihat respon dari *chatbot* ketika menghadapi pertanyaan yang berbeda setiap orangnya. Pengujian ini dilakukan kepada 10 orang yang berbeda dengan masing masing 5 pertanyaan, dengan contoh tabel pengujian UAT responden pada Tabel 7 di bawah ini:

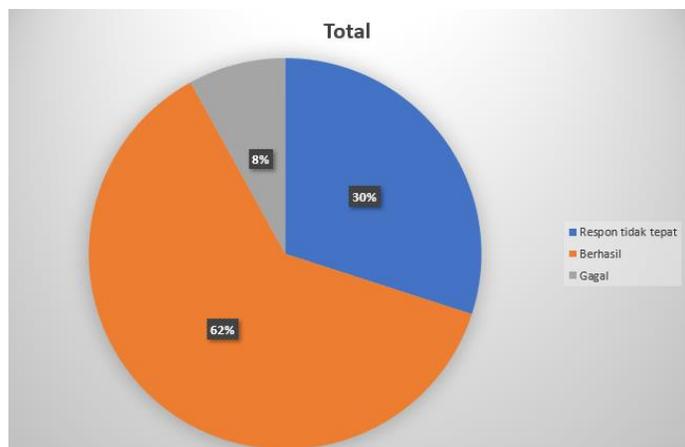
**Tabel 7. Contoh Tabel pengujian UAT Responden**

No	Pertanyaan	Kata Kunci	Good-Shuffix Shif	Respon Bot	Status
			Bad-Character Shif		
1	Ada divisi apa aja?	Divisi apa aja	Posisi= 1 2 3 4 5 6 7 8 9 10 11 12 13 14  String = divisi apa aja Pergeseran =2 2 4 4 8 8 8 8 10 10 12 12 13 1   String = divisi apa aja Pencacah = 13 12 11 10 9 8 7 6 5 4 3 2 1 0  Pergeseran= 13 8 11 8 9 8 3 0 5 0 3 0 1 0	AMCC memiliki 5 divisi, yaitu:_1.Web Programming_2.Desktop Programming_3.Mobile Programming_4.Hardware Software_5.Computer Network	Berhasil
2	Hardware Software belajar apa?	Hardware software	Posisi = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17  String = hardware software Pergeseran= 2 2 3 5 5 6 7 8 16 16 16 16 16 16 16 16 1   String = hardware software Pencacah = 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0  Pergeseran = 16 2 1 13 3 2 1 0 8 7 6 5 4 3 2 1 0	Mempelajari Internet of Things	Berhasil
3	Cara daftarnya gimana?	Cara daftar	Posisi= 1 2 3 4 5 6 7 8 9 10 11  String = cara daftar Pergeseran =2 2 3 4 7 7 7 10 10 10 1   String = cara daftar Pencacah = 10 9 8 7 6 5 4 3 2 1 0  Pergeseran= 10 1 0 1 6 5 1 3 2 1 0	Pendaftaran member dapat dilakukan secara online di join.amcc.or.id	Berhasil
4	Bisa daftar offline gak?	Daftar offline	Posisi= 1 2 3 4 5 6 7 8 9 10 11 12 13  String = daftar offline Pergeseran =3 3 9 9 9 9 9 9 9 12 12 12 1	Mohon maaf kak, karena keadaan pandemi seperti sekarang ini kita tidak membuka pendaftaran offline	Berhasil

No	Pertanyaan	Kata Kunci	Good-Shuffix Shif	Respon Bot	Status
			Bad-Character Shif		
			String = daftar offline Pencacah = 12 11 10 9 8 7 6 5 4 3 2 1 0 Pergeseran= 12 8 10 9 8 7 6 5 1 3 2 1 0		
5	Bayar berapa ?	Bayar berapa	Posisi= 1 2 3 4 5 6 7 8 9 10 11 12  String = Bayar berapa Pergeseran =1 2 4 4 5 10 10 10 10 10 10 11 1	Biaya pendaftaran cuma 60K, yuk segera daftar di join.amcc.or.id	Berhasil
			String = Bayar berapa Pencacah = 11 10 9 8 7 6 5 4 3 2 1 0 Pergeseran=5 0 9 0 3 6 5 4 3 0 1 0		

Dari hasil pengujian *User Acceptence Test* yang dilakukan terhadap *chatbot* dapat diambil kesimpulan:

1. Dari 10 responden pengujian dihasilkan 50 pertanyaan yang berbeda
2. Dari 50 pertanyaan yang diajukan oleh pengujian ke *chatbot*. Terdapat 15 pertanyaan yang tidak terjawab dengan baik, 4 gagal atau tidak terjawab oleh *chatbot*, dan 31 pertanyaan terjawab dengan baik
3. Dari 15 pertanyaan yang tidak terjawab dengan baik dikarenakan tidak adanya kata kunci pada *brain file* yang sesuai dengan kalimat yang diajukan. Ketidacocokan tersebut terjadi karena kata kunci yang terpisah oleh penghubung, salah ketik oleh pengujian, dan kurang banyaknya data di *brain file* sehingga menyebabkan respon *chatbot* tidak sesuai dengan maksud pengujian, pada gambar menunjukkan hasil persentase pengujian menggunakan metode UAT, yang dimana sebesar 62% berhasil menjawab pertanyaan dengan benar, seperti Gambar 9 grafik di bawah ini :



**Gambar 9. Grafik hasil pengujian UAT**

Pengukuran performa dari aplikasi *Chatbot* yang dibuat pada penelitian ini berdasarkan hasil *User Acceptance Test* dengan menghitung tingkat *accuracy*, *precision* dan *recall*. dengan hasil perhitungan pada Tabel 8 di bawah ini

**Tabel 8. Nilai Confusion Matriks**

	Positif	Negatif
True	31	5
False	10	4

Akurasi =  $(31+5) / (31+5+10+4) * 100\% = 36/50 * 100\% = 72\%$

Presisi =  $31/(10+31)*100\% = 31/41 * 100\% = 75.6\%$

Recall =  $31/(4+31) * 100\% = 31/35 * 100\% = 88.6\%$

Dapat ditarik kesimpulan berdasarkan pengujian *chatbot* melalui *confusion matriks* cukup baik karena akurasi, presisi, dan *recall* yang didapat diatas 70%.

#### 4 Kesimpulan

Kesimpulan penelitian ini adalah bahwa sistem chatbot bisa merespon dengan baik pertanyaan member dan hasil pengujian yang dilakukan terhadap respon chatbot dalam menangani berbagai macam pertanyaan dengan *User Acceptance Test* didapatkan kesimpulan, yaitu: a) Dapat disimpulkan respon dari *chatbot* adalah baik, karena hanya gagal merespon 4 pertanyaan dari 50 total pertanyaan yang diajukan dan menggunakan metode *confusion matrix* didapatkan tingkat akurasinya sebesar 70%. b) Kegagalan yang terjadi karena chatbot tidak bisa mengabaikan semua kata yang diterima.

#### Referensi

- [1] V. Sagita and M. I. Prasetyowati, 'Studi Perbandingan Implementasi Algoritma Boyer-Moore, Turbo Boyer-Moore, dan Tuned Boyer-Moore dalam Pencarian String', *J. Ultim.*, vol. 5, no. 1, pp. 31–37, 2013, doi: 10.31937/ti.v5i1.311.
- [2] T. Salisah, B. P. Sari, Y. Yulianto, and A. D. Hartanto, 'Implementasi Algoritma Boyer-Moore Pada Chatbot Wisata Yogyakarta', *Technomedia J.*, vol. 5, no. 1, pp. 54–66, 2020, doi: 10.33050/tmj.v5i1.1189.
- [3] K. W. Argakusumah and S. Hansun, 'Implementasi Algoritma Boyer-Moore pada Aplikasi Kamus Kedokteran Berbasis Android', *J. Ultim.*, vol. 6, no. 2, pp. 70–78, 2014, doi: 10.31937/ti.v6i2.340.
- [4] R. N. Astuti and M. Fatchan, 'View of PERANCANGAN APLIKASI TEKNOLOGI CHATBOT UNTUK INDUSTRI KOMERSIAL 4.0.pdf', in *Seminar Nasional Teknologi dan Sain (SNasTeks)*, pp. 339–348.
- [5] L. Hakim, S. Gustina, S. F. Putri, and S. U. Faudiah, 'Perancangan Chatbot di Universitas Proklamasi 45', *Edumatic J. Pendidik. Inform.*, vol. 4, no. 1, pp. 91–100, 2020, doi: 10.29408/edumatic.v4i1.2157.
- [6] R. T. Shita and L. L. Hin, 'Aplikasi Chatterbot Memanfaatkan Algoritma Boyer Moore Pada Pendaftaran Siswa Baru', *J. BIT, Issn 1693-9166*, vol. 14, no. 2, pp. 30–37, 2017.
- [7] N. Cahya and A. Triayudi, 'Implementasi Framework Codeigniter Pada Perancangan Chatbot Interaktif Menerapkan Metode Waterfall', *J. Media Inform. Budidarma*, vol. 5, no. 1, pp. 273–279, 2021, doi: 10.30865/mib.v5i1.2623.
- [8] M. Wijaya, Junaedy, and H. Arfandy, 'Perancangan Chatbot Untuk Informasi Penerimaan Mahasiswa Baru Pada Stmik Kharisma Makassar', *J. Ilmu Komput.*, vol. 1, pp. 1–11, 2017.
- [9] A. , Prayitno, A. Johar, and Y. Setiawan, 'Implementasi Algoritma Turbo Boyer Moore pada Aplikasi Kamus Istilah Biologi Berbasis Android', *J. rekursif*, vol. 6, no. 1, pp. 13–23, 2018, <http://sistemasi.ftik.unisi.ac.id>

- [Online]. Available: <http://ejournal.unib.ac.id/index.php/rekursif>.
- [10] A. P. Siahaan, I. Pendahuluan, A. S. Matching, and B. Kamus, 'Implementasi Algoritma Boyer Moore pada Aplikasi Kamus Nama Bayi Beserta Maknanya Berbasis Android', *Pelita Inform. Budi Darma*, vol. 17, no. 1, pp. 97–101, 2018.
- [11] R. I. Borman, 'Penerapan String Matching Dengan Algoritma Boyer Moore Pada Aplikasi Font Italic Untuk Deteksi Kata Asing', *J. Teknoinfo*, vol. 10, no. 2, p. 39, 2016, doi: 10.33365/jti.v10i2.9.
- [12] E. Rahmanita, 'Pencarian String Menggunakan Algoritma Boyer Moore Pada Dokumen', *J. Ilm. NERO*, vol. 1, no. 1, pp. 15–26, 2014.
- [13] R. s Boyer, S. R. Instutu, j strother Moore, and X. P. A. R. Center, 'A Fast String Searching Algorithm', *Communication of ACM* vol. 20, no. October, pp. 65–133, 1998, doi: 10.1007/978-1-4615-5475-2\_3.
- [14] A. Fau, Mesran, and G. L. Ginting, 'Analisa Perbandingan Boyer Moore Dan Knuth Morris Pratt Dalam Pencarian Judul Buku Menerapkan Metode Perbandingan Eksponensial ( Studi Kasus : Perpustakaan STMIK Budi Darma )', *J. Times (Technology Informatics Comput. Syst.*, vol. 6, no. 1, pp. 12–22, 2017.
- [15] Edgewick, Robert, Wayne, and Kevin, *Algorithm Fourth Edition*. Addison Wesley, Binding: Hardcover 4th Edition, 2011.
- [16] Firman, 'Boyer Moore String Matching'. 2013, [Online]. Available: <https://fnsfind16.wordpress.com/2015/01/03/boyer-moore-string-matching/amp/>.