

**IMPLEMENTASI ALGORITMA *BOYER MOORE* PADA PROSES  
PENCARIAN KATA STUDY KASUS APLIKASI *CHATBOT* POS  
KESEHATAN PERSANTREN DI MADRASAH BERTARAF  
INTERNASIONAL AMANATUL UMMAH**

**SKRIPSI**



**Oleh:  
RYAND ARIFRIANTONI  
NIM. 14650039**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2021**

**HALAMAN JUDUL**

**IMPLEMENTASI ALGORITMA *BOYER MOORE* PADA PROSES  
PENCARIAN KATA STUDY KASUS APLIKASI *CHATBOT* POS  
KESEHATAN PERSANTREN DI MADRASAH BERTARAF  
INTERNASIONAL AMANATUL UMMAH**

**SKRIPSI**

**Diajukan kepada:**

**Fakultas Sains dan Teknologi  
Universitas Islam Negeri (UIN)  
Maulana Malik Ibrahim Malang  
Untuk Memenuhi Salah Satu Persyaratan dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:**

**RYAND ARIFRIANTONI  
NIM. 14650039**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2021**

**HALAMAN PERSETUJUAN**

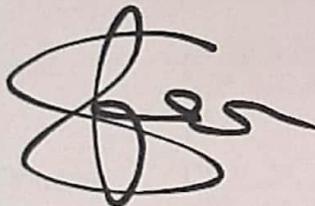
**IMPLEMENTASI ALGORITMA *BOYER MOORE* PADA PROSES  
PENCARIAN KATA STUDY KASUS APLIKASI *CHATBOT* POS  
KESEHATAN PERSANTREN DI MADRASAH BERTARAF  
INTERNASIONAL AMANATUL UMMAH**

**SKRIPSI**

**OLEH:  
RYAND ARIFRIANTONI  
NIM. 14650039**

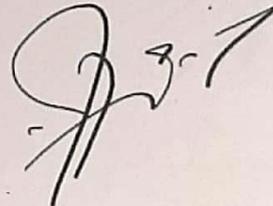
Telah Diperiksa dan Disetujui untuk Diuji  
Tanggal : Juni 2021

Pembimbing I



Dr. Mokhamad Amin Haryadi, MT  
NIP. 19670118 200501 1 001

Pembimbing II



Khadijah F.H. Holle, M.Kom  
NIDT. 19900626201608012077

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Cahyo Crysdiyan M.CS  
NIP. 19740424 200901 1 008

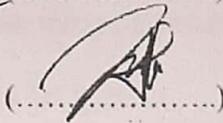
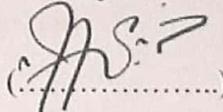
**HALAMAN PENGESAHAN**

**IMPLEMENTASI ALGORITMA *BOYER MOORE* PADA PROSES  
PENCARIAN KATA STUDY KASUS APLIKASI *CHATBOT* POS  
KESEHATAN PERSANTREN DI MADRASAH BERTARAF  
INTERNASIONAL AMANATUL UMMAH**

**SKRIPSI**

**OLEH:  
RYAND ARIFRIANTONI  
NIM. 14650039**

Telah Dipertahankan di Depan Dewan Penguji  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)  
Pada Tanggal Juni 2021

Penguji Utama	: <u>Dr. Cahyo Crysdian M.CS</u> NIP. 19740424 200901 1 008	
Ketua Penguji	: <u>Juniardi Nur Fadilah, MT</u> NIP. 19920605 201903 1 015	
Sekretaris Penguji	: <u>Dr. Ir. M. Amin Hariyadi, M.T</u> NIP. 19670118 200501 1 001	
Anggota Penguji	: <u>Khadijah F.H. Holle, M.Kom</u> NIDT. 19900626201608012077	

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Cahyo Crysdian M.CS  
NIP. 19740424 200901 1 008

**PERNYATAAN  
KEASLIAN TULISAN**

Saya yang bertanda tangan di bawah ini:

Nama : Ryand Arifriantoni  
NIM : 14650039  
Fakultas/Jurusan : Sains dan Teknologi/ Teknik Informatika  
Judul : Implementasi Algoritma Boyer Moore Pada Pencarian Kata  
Study Kasus Aplikasi Chatbot Pos Kesehatan Persantren  
Madrasah Bertaraf Internasional Amanatul Ummah

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar Pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan maka saya bersedia menerima sanksi atas perbuatan tersebut

Malang, Juli 2021  
Yang Membuat Pernyataan



Ryand Arifriantoni  
NIM. 14650039

## MOTTO

---

“JANGAN TERJEBAK PADA RUTINITAS, HARUS DINAMIS”

## PERSEMBAHAN

---



*Alhamdulillahillobbil'amin*

Sujud syukurku kusembahkan kepada Allah SWT Tuhan yang Maha Esa dan Maha Segalanya, atas takdirmu dan kehadiranNya telah menjadikan aku manusia yang senantiasa berpikir, berilmu, beriman dan bersabar dalam menjalani kehidupan ini. Semoga keberhasilan ini menjadi satu langkah awal untuk saya meraih cita-cita besar.

Kupersembahkan karya sederhana ini kepada orang tua ibunda, ibunda, ibunda, bapak, dan Keluargaku Tercinta Sebagai tanda bakti, hormat, dan rasa terima kasih yang tiada terhingga yang telah memberikan kasih sayang, segala dukungan, dan cinta kasih. Untuk Ibu, bapak yang selalu memotivasi yang senantiasa mendoakan dan restunya kepadaku dalam menuntut ilmu.

Tak lupa terima kasih kepada sahabat-sahabatku yang tiada hentinya selalu memberikan dorongan, motivasi dan telah menyempatkan memberikan waktu luangnya untuk membantu dalam proses pengerjaan penelitian ini.

**Ryand Arifriantoni**

## KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Syukur alhamdulillah atas kehadiran Allah SWT yang telah melimpahkan Rahmat dan Hidayah-Nya, sehingga penulis dapat menyelesaikan studi di Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus menyelesaikan Skripsi dengan maksimal.

Selanjutnya saya ucapkan terima kasih seiring do'a dan harapan jazakumullah ahsanal jaza' kepada semua pihak atas kebijaksanaannya dalam membantun penyelesaian Skripsi ini. Ucapan terima kasih saya sampaikan kepada:

1. Prof. DR. Abdul Haris, M.Ag, selaku rektor UIN Maulana Malik Ibrahim Malang, yang telah banyak memberikan pengetahuan dan pengalaman yang berharga.
2. Dr. Sri Harini, M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crysdiyan, selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Bapak Dr. M. Amin Hariyadi, M.T dan Ibu Khadijah F.H. Holle, M.Kom selaku dosen pembimbing Skripsi, yang telah banyak memberikan pengarahan dan pengalaman yang sangat berharga.
5. Bapak Dr. M. Amin Hariyadi, M.T selaku dosen wali saya ucapkan sangat sangat terima kasih atas kebijaksanaannya dalam memberi bimbingan, masukan, arahan dan pelajaran yang telah di berikan.
6. Segenap sivitas akademika Jurusan Teknik Informatika, terutama seluruh dosen, terima kasih atas segenap ilmu dan bimbingannya

## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERSETUJUAN.....	ii
HALAMAN PENGESAHAN.....	iii
PERNYATAAN.....	iv
KEASLIAN TULISAN.....	iv
MOTTO.....	v
PERSEMBAHAN.....	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xi
ABSTRAK.....	xii
ABSTRACT.....	xiii
المخلص.....	xiv
BAB I.....	1
PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Pernyataan Masalah.....	5
1.3. Tujuan Penelitian.....	6
1.4. Manfaat Penelitian.....	6
1.5. Batasan Masalah.....	6
BAB II.....	7
TINJAUAN PUSTAKA.....	7
2.1. Penelitian Terkait.....	7
2.2. <i>Chatbot</i> .....	9
2.3. <i>Preprocessing</i> .....	11
2.4. Pencarian.....	12
2.4.1. Algoritma Pencarian <i>String</i> .....	12
2.4.2. Algoritma <i>Brute Force</i> .....	13
2.4.3. Algoritma Knut Morris Pratt.....	14
2.4.4. Algoritma <i>Boyer Moore</i> .....	15
BAB III.....	23
METODOLOGI PENELITIAN.....	23
3.1. Prosedur Penelitian.....	23
3.2. Desain Aplikasi.....	24

3.3.	Desain Proses.....	25
3.3.1.	Input Kalimat .....	25
3.3.2.	<i>Preprocessing</i> .....	25
3.3.3.	Implementasi <i>Boyer Moore</i> pada aplikasi <i>Chatbot</i> .....	27
3.3.4.	<i>Output</i> .....	36
3.4.	Desain Pengukuran Akurasi .....	36
BAB VI .....		37
UJI COBA DAN PEMBAHASAN.....		37
4.1.	Uji Coba Sistem.....	37
4.1.1.	Uji Coba Perangkat Keras.....	37
4.1.2.	Uji Coba Perangkat Lunak.....	37
4.1.3.	Uji Coba Algoritma <i>Boyer Moore</i> .....	38
4.2.	Pengujian Sistem .....	41
4.2.1.	Pengujian Black Box .....	41
4.2.2.	Pengujian User.....	49
4.2.3.	Pengujian Algoritma Boyer Moore.....	57
BAB V.....		64
KESIMPULAN DAN SARAN.....		64
5.1.	Kesimpulan.....	64
5.2.	Saran.....	64
Daftar Pustaka .....		65

## DAFTAR GAMBAR

Gambar 2. 1 Contoh <i>Case Folding</i> .....	11
Gambar 2. 2 Contoh <i>Tokenizing</i> .....	11
Gambar 2. 3 Contoh <i>Stemming</i> .....	12
Gambar 2. 4 <i>Good-suffix shift</i> , <i>u</i> terjadi lagi didahului karakter <i>c</i> berbeda dari <i>a</i> 18	
Gambar 2. 5 <i>Good-suffix shift</i> , hanya <i>suffix</i> dari <i>u</i> yang terjadi lagi di <i>pattern x</i> . 18	
Gambar 2. 6 <i>Bad-character shift</i> , <i>b</i> terdapat di <i>pattern x</i> .....	18
Gambar 2. 7 <i>Bad-character shift</i> , <i>b</i> tidak ada di <i>pattern x</i> .....	19
Gambar 3. 1 Prosedur Penelitian.....	23
Gambar 3. 2 Desain Proses .....	25
Gambar 3. 3 Hasil <i>Case Folding</i> .....	25
Gambar 3. 4 Hasil <i>Tokenizing</i> .....	26
Gambar 3. 5 Hasil <i>Stemming</i> .....	26
Gambar 3. 6 <i>Flowchart Boyer Moore</i> .....	28
Gambar 3. 7 Penyelesaian contoh kasus Prosedur preBmBc.....	30
Gambar 3. 8 Penyelesaian contoh kasus Prosedur preBmGs.....	33
Gambar 3. 9 Penyelesaian contoh kasus Prosedur BM.....	35
Gambar 4. 1 Form input keluhan/gejala.....	38
Gambar 4. 2 Contoh Input Keluhan .....	38
Gambar 4. 3 Hasil Proses Keluhan .....	39
Gambar 4. 4 <i>Source Code Tokenizing</i> .....	40
Gambar 4. 5 Form Login.....	42
Gambar 4. 6 Login Gagal.....	42
Gambar 4. 7 <i>Form</i> Tambah Gejala .....	43
Gambar 4. 8 Data Gejala.....	44
Gambar 4. 9 <i>Alert</i> Data Tidak Boleh Kosong <i>Form</i> Gejala.....	44
Gambar 4. 10 <i>Form</i> Tambah Penyakit.....	45
Gambar 4. 11 Data Penyakit .....	46
Gambar 4. 12 <i>Alert</i> Data Tidak Boleh Kosong <i>Form</i> Penyakit.....	46
Gambar 4. 13 <i>Form</i> Data Penyakit dan gejala .....	47
Gambar 4. 14 Data Penyakit dan Gejala .....	48
Gambar 4. 15 <i>Alert</i> Data Tidak Boleh Kosong <i>Form</i> Penyakit dan Gejala .....	48

## DAFTAR TABEL

Tabel 3. 1 Contoh Data Gejala dan Penyakit .....	27
Tabel 3. 2 Daftar pencacahan <i>prefix</i> dan <i>suffix</i> dari <i>pattern</i> .....	32
Tabel 3. 3 Contoh hasil pencarian.....	35
Tabel 3. 4 Tabel Kontingensi ROC.....	36
Tabel 4. 1 Perangkat Keras Yang Digunakan .....	37
Tabel 4. 2 Perangkat Lunak Yang Digunakan .....	37
Tabel 4. 3 Uji Coba Form Login.....	41
Tabel 4. 4 Uji Coba <i>Form</i> Tambah Gejala.....	43
Tabel 4. 5 Uji Coba <i>Form</i> Tambah Penyakit .....	45
Tabel 4. 6 Uji Coba <i>Form</i> Tambah Data Gejala dan Penyakit .....	47
Tabel 4. 7 Bobot Nilai.....	49
Tabel 4. 8 Keterangan Nilai Kuesioner.....	49
Tabel 4. 9 Hasil Pengujian Kuesioner Pertanyaan Nomor 1.....	50
Tabel 4. 10 Hasil Pengujian Kuesioner Pertanyaan Nomor 2.....	50
Tabel 4. 11 Hasil Pengujian Kuesioner Pertanyaan Nomor 3.....	51
Tabel 4. 12 Hasil Pengujian Kuesioner Pertanyaan Nomor 4.....	51
Tabel 4. 13 Hasil Pengujian Kuesioner Pertanyaan Nomor 5.....	52
Tabel 4. 14 Hasil Pengujian Kuesioner Pertanyaan Nomor 6.....	52
Tabel 4. 15 Hasil Pengujian Kuesioner Pertanyaan Nomor 7.....	53
Tabel 4. 16 Hasil Pengujian Kuesioner Pertanyaan Nomor 8.....	53
Tabel 4. 17 Hasil Pengujian Kuesioner Pertanyaan Nomor 9.....	54
Tabel 4. 18 Hasil Pengujian Kuesioner Pertanyaan Nomor 10.....	54
Tabel 4. 19 Hasil Pengujian Kuesioner Pertanyaan Nomor 11.....	55
Tabel 4. 20 Hasil Pengujian Kuesioner Pertanyaan Nomor 12.....	55
Tabel 4. 21 Hasil Pengujian Kuesioner Pertanyaan Nomor 13.....	56
Tabel 4. 22 Hasil Pengujian Kuesioner Pertanyaan Nomor 14.....	56
Tabel 4. 23 Hasil Pengujian Kuesioner Pertanyaan Nomor 15.....	57
Tabel 4. 24 Jumlah Data Yang Digunakan .....	57
Tabel 4. 25 Data Nama Penyakit dan Gejala dari Pihak Klinik.....	58
Tabel 4. 26 Data Nama Penyakit dan Gejala dari Pihak Klinik (Lanjutan).....	59
Tabel 4. 27 Data Nama Penyakit dan Gejala dari Pihak Klinik (Lanjutan).....	60
Tabel 4. 28 Hasil Uji Coba Proses Algoritma.....	61
Tabel 4. 29 Hasil Uji Coba Proses Algoritma (Lanjutan).....	62

## ABSTRAK

Arifriantoni, Ryand. 2021. **Implementasi Algoritma Boyer Moore Pada Proses Pencarian Kata Study Kasus Aplikasi Chatbot Pos Kesehatan Persantren Di Madrasah Bertaraf Internasional Amanatul Ummah**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. M. Amin Hariyadi, M.T. (II) Khadijah F.H. Holle, M. Kom.

Kata Kunci : Pencarian, *Chatbot*, Algoritma *Boyer Moore*

Proses pencarian merupakan salah satu proses penting dalam pemrosesan data. Klinik Pesantren MBI Amanatul Ummah dalam memberi layanan konsultasi pasien masih di lakukan dengan cara bertatap muka dan merespon satu persatu gejala yang di alami pasien. Hal ini mengakibatkan butuh waktu lama dalam menangani pasien. Pembangunan Aplikasi Chatbot dengan implementasi algoritma Boyer Moore pada pencarian data ditujukan untuk menjadi asisten online dokter dan membantu mencari tindakan sementara yang bisa di lakukan pasien. Proses pencocokan string Algoritma Boyer Moore di mulai dari karakter paling kanan ke kiri dengan tiga tahapan proses. Pertama memperoleh nilai Occurence Heuristic (OH) yang didapat dari prosedur preBmBc, kemudian mendapatkan nilai Match Heuristic (MH) dari proses preBmGs, kemudian nilai maksimal OH dan MH digunakan untuk pergeseran pada persamaan karakter BM. Hasil uji coba sebanyak 26 kali pada proses pencarian hasil jawaban menghasilkan nilai rata-rata akurasi sebesar 100% dengan rata rata waktu 0.0037 detik.

## ABSTRACT

Arifriantoni, Ryand. 2021. **Implementation of Boyer Moore Algorithm on String Search Process Case Study Chatbot Application for Islamic School Health Posts International Standard School of Amanatul Ummah.** Undergraduate Theses. Informatics Engineering Department. Faculty of Sains and Technology. State Islamic University of Maulana Malik Ibrahim Malang. Advisor: (I) Dr. M. Amin Hariyadi, M.T. (II) Khadijah F.H. Holle, M. Kom.

Keyword : Searching, Chatbot, Boyer Moore Algorithm

The search process is one of the main processes in data processing. To give consultation service for the patients, the clinic of islamic boarding school MBI Amanatul Ummah was still performing face-to-face method to respond the symptoms which were complained by the patients. This method resulted to take longer time to handle the patients. The construction of Chatbot application with the implementation of Boyer Moore algorithm in data search was intended to be doctors' online assistant which helped the doctor to look for temporary action that able to be done for the patients. The Boyer Moore algorithm's calibration process was started from the rightmost character to the leftmost character with three steps; first, obtained the Occurrence Heuristic (OH) score that produced from preBmBc procedure. Then, received the Match Heuristic (MH) from preBmGs process. Last, the maximum score of OH and MH would be used for stepping friction on BM's character calibration. The calibration result which took 26 times in search process generated average accuracy by 100% during 0.0037 seconds.

## المخلص

عاريڤيانتوني ، ريباند. 2021. تنفيذ خوارزمية Boyer Moore في دراسة حالة عملية البحث عن الكلمات تطبيق Chatbot للوظائف الصحية للمدارس الإسلامية في المدرسة على المستوى الدولي أمانة الأمة. أطروحة. قسم هندسة المعلوماتية بكلية العلوم والتكنولوجيا مولانا مالك إبراهيم الدولة الإسلامية جامعة مالانج. المشرف: (الأول) محمد امين هاربيادي الماجستير (الثاني) خديجة ف. هول ، م. كوم .  
الكلمات المفتاحية : Search, Chatbot, Boyer Moore Algorithm

عملية البحث هي إحدى العمليات المهمة في معالجة البيانات. لا تزال عبادة مدرسة أمانة الأمة على المستوى الدولي تقدم خدماتها الاستشارية للمرضى وجهاً لوجه وتستجيب واحداً تلو الآخر للأعراض التي يعاني منها المرضى. ينتج عن هذا أخذ وقت طويل لعلاج المرضى. يهدف تطوير تطبيق Chatbot مع تنفيذ خوارزمية Boyer Moore في البحث عن البيانات إلى أن يكون مساعداً عبر الإنترنت للأطباء ويساعد في العثور على الإجراءات المؤقتة التي يمكن للمرضى اتخاذها. تبدأ عملية مطابقة السلسلة لخوارزمية Boyer Moore من الحرف الموجود في أقصى اليمين إلى اليسار بثلاث مراحل من العملية. أولاً ، حصل على قيمة استدلال الأحداث (OH) التي تم إنشاؤها من إجراء preBmBc ، ثم حصل على قيمة Match Heuristic (MH) من عملية preBmGs ، ثم سيتم استخدام أقصى قيم OH و MH للقفز في التحول في مطابقة أحرف BM. نتج عن نتائج الاختبار 26 مرة لعملية البحث متوسط دقة 100٪ مع مدة زمنية 0.0037 ثانية.

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Pondok pesantren Ammanatul Ummah Program MBI (Madrasah Bertaraf Internasional) dengan jumlah siswa di tahun 2020 sebanyak 1135 dengan jumlah pegawai aktif 196. MBI Amanatul Ummah adalah lembaga pendidikan pesantren yang mewajibkan setiap peserta didik bermukim di pesantren dan menjalankan aktivitas yang terstruktur setiap harinya. Sehingga lembaga juga harus menyediakan beberapa fasilitas utama untuk memenuhi kebutuhan peserta didik terutama fasilitas kesehatan.

MBI Amanatul Ummah menyediakan klinik pesantren atau POSKESTREN (Pos Kesehatan Pesantren) di dalamnya terdapat tenaga kesehatan yang terdiri dari 2 Dokter, 3 Perawat, dan 1 Apoteker yang dibagi sesuai jadwal setiap harinya. Informasi kesehatan adalah bagian sangat penting dalam lembaga pendidikan pesantren, dimana informasi kesehatan tersebut diperoleh dengan cara konsultasi secara langsung dengan dokter. Berdasarkan survei yang dilakukan terhadap beberapa pasien di klinik pasien memperoleh informasi kesehatan dengan cara datang ke klinik dan berkonsultasi langsung terhadap dokter. Penyampaian informasi kesehatan tersebut dirasa kurang karena keterbatasan waktu yang ada, sedangkan pasien sangat membutuhkan informasi kesehatan dan konsultasi kesehatan. Dengan permasalahan tersebut banyak pasien yang kurang tepat mengdiagnosa dan mengambil tindakan lanjut. Untuk menyampaikan informasi kesehatan atau tindakan lanjutan yang tidak terikat oleh waktu dan tempat, POSKESTREN memerlukan suatu alat media layanan informasi kesehatan yang

dapat merespon pertanyaan pasien dan menentukan tindakan ringan sementara tanpa ada keterbatasan waktu dan jumlah dokter.

Al-Qur'an telah menjelaskan masalah waktu dalam Surat Al-Baqarah [2] ayat 164 yang berbunyi:

إِنَّ فِي خَلْقِ السَّمَوَاتِ وَالْأَرْضِ وَاخْتِلَافِ اللَّيْلِ وَالنَّهَارِ وَالْفُلُوكِ الَّتِي تَجْرِي فِي الْبَحْرِ بِمَا يَنْفَعُ النَّاسَ وَمَا أَنْزَلَ اللَّهُ مِنَ السَّمَاءِ مِنْ مَّاءٍ فَأَحْيَا بِهِ الْأَرْضَ بَعْدَ مَوْتِهَا وَبَثَّ فِيهَا مِنْ كُلِّ دَابَّةٍ وَتَصْرِيفِ الرِّيْحِ وَالسَّحَابِ الْمُسَخَّرِ بَيْنَ السَّمَاءِ وَالْأَرْضِ لَآيَاتٍ لِّقَوْمٍ يَعْقِلُونَ

*Artinya : “Sesungguhnya dalam penciptaan langit dan bumi, silih bergantinya malam dan siang, bahtera yang berlayar dilaut membawa apa yang berguna bagi manusia, dan apa yang Allah turunkan dari langit berupa air, lalu dengan air itu dihidupkan-Nya bumi sesudah mati (kering), dan Dia sebarkan di bumi itu segala jenis hewan, dan pengisaran angin dan awan yang dikendalikan antara langit dan bumi, sungguh (terdapat) tanda-tanda (kebesaan dan kebesaran Allah) bagi kaum yang memikirkannya” (Q.S Al-Baqarah: 164)*

Berdasarkan Kitab Tafsir Ibnu Katsir (2015) Sesungguhnya dalam penciptaan langit dan bumi yang kita lihat tentang ketinggianannya, keindahannya, keluasannya, bintang-bintang yang beredar tetap serta perputaran falaknya. Dan bumi ini yang dengan penuh kepadatannya lembah-lembah, gunung-gunung, lautan serta segala sesuatu yang ada padanya berupa keramaian serta segala sesuatu yang ada padanya berupa berbagai manfaat. Pergantian malam dan siang datang lalu pergi kemudian digantikan dengan lainnya secara silih berganti tanpa adanya keterlambatan sedikitpun.

Ayat diatas menjelaskan bahwa Allah SWT telah menciptaka segala sesuatu yang bermanfaat bagi makhluk-Nya, dan Allah SWT menciptakan langit dan bumi, silih bergantinya siang dan malam sebagai tanda-tanda kebesaran-Nya bagi manusia yang mau berfikir untuk memanfaatkannya. Dimana manfaat itu bisa dicari oleh manusia misalnya dalam mengambil waktu yang bermanfaat antara siang dan malam. Mengingat di pondok pesantren yang mengajarkan nilai - nilai keperdulian

yang tinggi terhadap sesama termasuk persoalan antri atau nomor urutan santri dalam suatu kegiatan, sehingga ketika santri mendapatkan antrian di nomor belakang maka santri di ajarkan untuk sabar menunggu giliran.

Layanan Kesehatan MBI Amanatul Ummah mempunyai beberapa komponen yang harus ditata dengan seimbang sehingga pasien mendapatkan layanan yang maksimal dan tidak terjadi kesenjangan antara pasien yang mendapat urutan pertama dan terakhir karena dapat terlayani di hari yang sama. Konsep keseimbangan yang terjadi di alam semesta telah dijelaskan oleh Allah SWT dalam Al-Qur'an surat Al-Mulk [67] ayat 3, yang berbunyi:

الَّذِي خَلَقَ سَبْعَ سَمَاوَاتٍ طِبَاقًا مَّا تَرَىٰ فِي خَلْقِ الرَّحْمَنِ مِن تَفْوُتٍ ۗ فَارْجِعِ الْبَصَرَ ۖ هَلْ تَرَىٰ مِن فُطُورٍ

*Artinya: "Yang telah menciptakan tujuh langit berlapis-lapis. Kamu sekali-kali tidak melihat pada ciptaan Tuhan yang Maha Pemurah sesuatu yang tidak seimbang. Maka Lihatlah berulang-ulang. Adakah kamu lihat sesuatu yang tidak seimbang?"*

Ayat diatas menjelaskan bahwa Allah SWT telah menciptakan segala sesuatu secara seimbang. Hal ini sesuai dengan Al-Mahalli dan As-Suyuti (2008) yang mengatakan bahwa tidak ada satu pun yang diciptakan oleh Allah SWT di alam semesta ini dalam keadaan yang tidak seimbang.

Dari permasalahan yang telah di uraikan solusi yang di usulkan untuk mempermudah konsultasi yaitu menggunakan layanan *Chatbot* dengan pendekatan *Natural Language Processing (NPL)* dan dapat di akses calon pasien secara online sebelum menuju POSKESTREN.

*Chatbot* merupakan suatu teknologi buatan yang digunakan untuk memudahkan pekerjaan manusia dalam bidang bisnis maupun bidang Pendidikan, *chatbot* bekerja sebagai asisten untuk menjawab pertanyaan yang telah diajukan

oleh manusia. Hal ini sesuai dengan penjelasan bahwa *chatbot* merupakan inovasi dari teknologi yang digunakan untuk mempermudah menjawab pertanyaan-pertanyaan yang diajukan sesuai konteks yang ditentukan (Nugraha, 2020).

Teknologi chatbot merupakan salah satu bentuk aplikasi *Natural Language Processing*, *NLP* merupakan salah satu bidang ilmu Kecerdasan Buatan (*Artificial Intelligence*) yang mempelajari komunikasi antara manusia dengan komputer melalui bahasa alami. Model komputasi seperti ini berguna untuk memudahkan komunikasi antara manusia dengan komputer dalam hal pencarian informasi, sehingga dapat terjadi suatu interaksi antara keduanya dengan menggunakan bahasa alami. *Chatbot* dibangun untuk mempermudah pasien dalam mencari informasi kesehatan dan tindakan sementara. Pasien memasukkan pertanyaan seputar kesehatan atau gejala kesehatan berupa teks, kemudian *chatbot* akan memproses masukan dari pasien dan merespon berupa jawaban atau kesimpulan atas pertanyaan pasien sesuai dengan data yang ada di data *sheet*. Proses menentukan jawaban atau kesimpulan di butuhkan proses persamaan data antara masukan dan data *sheet*, dengan pendekatan *Natural Language Processing* kalimat atau kata masukan akan di olah dan akan di lanjutkan dengan proses pencocokan *string*. Sehingga memudahkan pasien dalam konsultasi kesehatan dan memberikan informasi terhadap pertolongan pertama pada suatu penyakit dan gejala gejalanya.

Proses pencocokan *string* merupakan proses pencarian sebuah string yang terdiri dari pattern terhadap karakter pada teks yang dicari (Charras & Lecroq, 2004). Dalam pencocokan string terdapat dua pendekatan yaitu *exact string matching* dan *inexact string matching*. Pendekatan yang digunakan adalah *exact string matching*, karena pendekatan ini digunakan untuk mencocokkan kata dengan

tepat yang melibatkan struktur karakter teks yang dicocokkan memiliki kesamaan dari jumlah urutan dan struktur katanya (Borman & Pratama, 2016). Salah satu pendekatan *inexact string matching* yang efisien dalam melakukan perbandingan pattern adalah *Boyer Moore*. Algoritma *Boyer Moore* memiliki dasar yaitu mencocokkan pola string dari kanan ke kiri mulai dari karakter paling kanan. Dalam beberapa penelitian algoritma *Boyer Moore* menunjukkan kinerja yang baik dalam pencocokan *string*. Pada penelitian dengan membandingkan *Boyer Moore* dan *Knuth Morris Pratt* (KMP) untuk melakukan pencarian judul buku, menunjukkan bahwa algoritma *Boyer Moore* lebih cepat dalam melakukan pencarian dan pencocokan *string* dibandingkan dengan algoritma KMP (Fau, Mesran, & Ginting, 2017). Selain itu, terdapat penelitian tentang pengembangan software kamus untuk istilah kesehatan dengan menerapkan *Boyer Moore*, menunjukkan bahwa algoritma *Boyer Moore* mampu menampilkan hasil pencarian istilah dengan cepat (Darmawan, Setianingrum, & Arini, 2018).

## **1.2. Pernyataan Masalah**

Berdasarkan latar belakang yang sudah dijelaskan, dapat diambil pertanyaan penelitian sebagai berikut:

- a. Seberapa tinggi tingkat kepuasan *user* terhadap aplikasi *Chatbot* pos kesehatan pesantren dalam memudahkan proses observasi pasien dan mengolah data riwayat pasien?
- b. Seberapa akurat aplikasi chatbot dalam menentukan hasil jawaban dengan Algoritma Boyer Moore?

### 1.3. Tujuan Penelitian

Dari pertanyaan rumusan masalah yang sudah disebutkan sebelumnya maka tujuan penelitian sebagai berikut:

- a. Mengetahui kegunaan aplikasi *chatbot* kesehatan pesantren untuk kebutuhan user.
- b. Mengetahui implementasi algoritma *boyer moore* pada proses pencarian jawaban pada Aplikasi *Chatbot* Pos Kesehatan Pesantren di Madrasah Bertaraf Internasional Amanatul Ummah.

### 1.4. Manfaat Penelitian

Dengan di lakukannya penelitian ini dapat memberikan manfaat sebagai berikut:

1. Mempermudah pasien untuk melakukan konsultasi keluhan.
2. Mempercepat proses observasi dan analisa dokter terhadap pasien dari hasil percakapan antara pasien dan *Chabot*.

### 1.5. Batasan Masalah

Dalam penelitian di buat batasan masalah yang digunakan untuk menghindari pembahasan yang tidak sesuai dengan topik penelitian. Adapun batasan-batasan tersebut sebagai berikut:

1. Objek penelitian dilakukan di MBI Amanatul Ummah dan sesuai dengan kebutuhan POSKESTREN.
2. Data yang digunakan pada penelitian di ambil dari pihak MBI Amanatul Ummah devisi kesehatan.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Penelitian Terkait**

Implementasi Algoritma *Boyer Moore* Pada Aplikasi Kamus Kedokteran Berbasis Android. Penelitian ini mengimplementasikan algoritma *Boyer Moore* untuk pencarian kata pada kamus istilah kedokteran, keakuratan kata dengan hasil dari proses pencarian menggunakan algoritma *Boyer Moore* menunjukkan persentase 100% dari responden yang menyatakan bahwa proses pencarian istilah pada aplikasi kamus kedokteran dapat memberikan penjelasan yang sesuai dengan yang diharapkan (Argakusumah & Hansun, 2014).

Penerapan Algoritma *Boyer Moore* Pada Aplikasi Pengajuan Judul Skripsi Berbasis Web. Penelitian ini melakukan pencocokan judul skripsi yang diajukan dengan judul skripsi yang telah diterima yang menghasilkan karakter *string* kemudian menggunakan algoritma *Boyer Moore* untuk pencocokan *string*. Untuk menentukan judul diterima ditentukan dari hasil presentase kesamaan *string*, hal tersebut tidak terlalu efektif dikarenakan judul dapat memiliki kesamaan *string* yang besar tapi tidak memiliki arti yang sama atau sebaliknya. Algoritma *Boyer Moore* cocok dilakukan untuk pencarian *string* tapi tidak cocok digunakan untuk pengambilan informasi dari kecocokan *string* berdasarkan satu kata atau lebih (Ginting, 2014).

Implementasi Boyer Moore pada Aplikasi Pencarian Rumus Matematika dan Fisika. Penelitian ini meneliti bagaimana cara melakukan pendeteksian kata kunci yang terdapat pada soal matematika dan fisika yang dicocokkan dengan menggunakan algoritma *Boyer Moore*. Kata kunci pada basis data akan dicocokkan dengan dengan soal yang dimasukkan. Pengujian dengan algoritma *Boyer Moore*

pada soal fisika dan matematika membuktikan bahwa aplikasi dapat menampilkan daftar rumus yang sesuai dalam melakukan pencocokan pola kata kunci disoal matematika atau fisika tingkat SMA (Agung & Yogyakarta, 2016).

Analisa Perbandingan *Boyer Moore* Dan *Knuth Morris Pratt* Dalam Pencarian Judul Buku Menerapkan Metode Perbandingan Eksponensial. Penelitian dilakukan untuk mencari judul buku di perpustakaan STMIK Budidarma Medan dimana proses pencarian dilakukan dengan menggunakan algoritma *Boyer Moore* dimana proses pencarian *string* dimulai dari kanan *pattern* sementara algoritma *Knuth Morris Pratt* memulai proses pencarian *string* dengan memanfaatkan pergeseran *pattern* dalam *text* dari sebelah kiri. Perbandingan algoritma *Boyer Moore* dan *Knuth Morris Pratt* dengan menggunakan Metode perbandingan *Eksponensial (MPE)* menunjukkan bahwa algoritma *Boyer Moore* lebih cepat dalam melakukan pencarian (Fau, Mesran, & Ginting, 2017).

Implementasi Algoritma *Boyer Moore* Pada Aplikasi Kamus Istilah Kebidanan Berbasis Web. Penelitian ini menguji kata kata dengan beberapa tahap, pada hasil dan pembahasan, implementasi algoritma *Boyer Moore* berhasil dilakukan dari beberapa kata masukan dan hasil keluaran yang sesuai, dengan cara melakukan kata masukan pada teks, dari kata *abortus* dari rentang total waktu terendah 126 dengan kata masukan (a, b) sampai yang tertinggi 165 dengan masukkan (a, b, o, r, t, u, s), cari dalam algoritma *Boyer Moore* dan berhasil, perbedaan jumlah data dalam databse tidak begitu banyak. Metode ini menghasilkan waktu pengiriman hasil kata dengan 0 ms/sec (Darmawan, Setianingrum, & Arini, 2018).

## 2.2. *Chatbot*

*Chatbot* berasal dari kata “Chat” dan “Bot”. Chat berarti percakapan sedangkan “bot” diambil kata robot, dari dua kata tersebut sehingga dapat di artikan bahwa *Chatbot* merupakan robot virtual yang bisa digunakan untuk membantu percakapan melalui sebuah media elektronik. Menurut (Maskur, 2016) *Chatbot* merupakan salah satu sistem kecerdasan yang berasal dari pemrosesan bahasa alami yang berasal dari salah satu cabang kecerdasan buatan. (Dwi, 2018) menjelaskan bahwa hal yang dapat membedakan *Chatbot* dengan sistem pemrosesan bahasa alami yaitu kesederhanaan dalam algoritma yang digunakan

*Chatbot* merupakan sebuah program yang berasal dari computer yang digunakan untuk simulasi berkomunikasi atau melakukan percakapan untuk membahas suatu hal. Tujuan dari pembuatan *Chatbot* yaitu untuk mengetahui apakah *Chatbot* mampu menipu user agar user mengira bahwa mereka sedang melakukan komunikasi dengan manusia (Vijayarani, 2017). (Santoso, 2011) menjelaskan bahwa *Chatbot* dirancang untuk menstimulasikan percakapan intelektual dengan satu atau lebih manusia baik secara audio maupun text. Program *Chatbot* diuji terlebih dahulu dengan *turing test* yaitu dengan merahasiakan identitas nya sebagai mesin.

*Chatbot* di bangun sesuai dengan topik yang dimodelkan untuk bisnis pengetahuan yang digunakan unrtuk menyelesaikan masalah perorangan atau masalah bisnis yang menyangkut banyak orang, contoh yang sering menggunakan *Chatbot* yaitu konsultasi kesehatan atau lebih dikenal dengan aplikasi hallo dokter. Di dalam *Chatbot* tersebut terdapat model pengetahuan untuk menjawab

pertanyaan-pertanyaan yang diajukan sesuai dengan konteks yang telah disusun (Nugraha, 2020).

Andriyani.2004 juga menjelaskan bahwa *Chatbot* adalah program kecerdasan yang dibuat oleh manusia untuk mempermudah kegiatan manusia, dimana *Chatbot* ini dirancang untuk dapat berkomunikasi langsung dengan manusia sebagai penggunanya. *Chatbot* mampu menjawab pertanyaan yang diajukan oleh manusia dengan cara mengetik. *Chatbot* mampu memberikan jawaban sesuai dengan input pertanyaan yang diajukan atau keluhan dari pengguna. Benedictus, 2017 menjelaskan bahwa *Chatbot* harus mengenali dan merespon kata yang diberikan oleh user. Kemampuan *Chatbot* dalam mengenali dan memberikan respon ditentukan oleh dua ruang lingkup pengetahuan.

*Chatbot* terdiri dari 3 kombinasi, diantaranya: (Guzman, 2016)

1. User interface

Merupakan jembatan atau penghubung antar *Chatbot* dan user sehingga bisa berinteraksi

2. Artificial intelligence

Artificial intelligence dalam *Chatbot* digunakan untuk memahami setiap interaksi yang terjadi antara *Chatbot* dengan user, sehingga mampu menjawab setiap pertanyaan yang diajukan oleh user. *Chatbot* menangani masalah melalui aturan yang telah ditentukan.

3. Integrasi

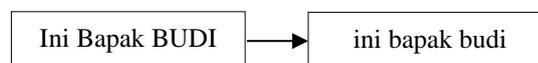
Integrasi dengan sistem lainnya akan menambah fitur dalam *Chatbot*, sehingga *Chatbot* mampu memberikan informasi tambahan yang lebih bervariasi

### 2.3. Preprocessing

*Preprocessing* dilakukan untuk penyiapan dokumen mentah menjadi dokumen atau representatif dokumen yang siap diproses untuk langkah selanjutnya (Zaman & Winarko, 2011). Pada proses ini yang dilakukan antara lain *case folding*, tokenizing, dan stemming.

#### 1. Case Folding

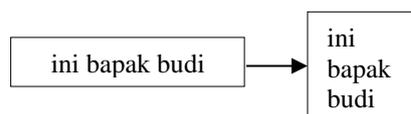
*Case folding* adalah mengubah huruf dalam dokumen menjadi huruf kecil (*lowercase*). Hanya merubah karakter huruf ‘a’ sampai dengan ‘z’ yang diterima. Karakter selain huruf akan dihilangkan dan dianggap sebagai *delimiter*. Berikut contoh dari case folding.



**Gambar 2. 1** Contoh *Case Folding*

#### 2. Tokenizing

Pada dasarnya proses *tokenizing* merupakan proses pemisahan setiap kata yang menyusun suatu dokumen. Umumnya setiap kata teridentifikasi atau terpisahkan oleh karakter spasi, sehingga proses tokenizing mengandalkan karakter spasi pada sebuah dokumen untuk melakukan pemisahan kata. Berikut contoh *tokenizing*.



**Gambar 2. 2** Contoh *Tokenizing*

#### 3. Stemming

*Stemming* merupakan suatu proses merubah kata menjadi kata dasar. *Stemming* teks berbahasa Indonesia memiliki permasalahan khusus, masalah merupakan perbedaan tipe – tipe dari imbuhan (*affixes*). Imbuhan

(*affixes*) dapat berubah sesuai dengan huruf kata dasar, ketika dalam suatu kata terdapat lebih dari satu imbuhan (*affixes*) yang melekat pada suatu kata maka urutan dalam menghilangkan kata imbuhan tersebut menjadi sangat penting, apabila urutan yang dilakukan salah maka kata dasar yang benar tidak akan ditemukan. Untuk menghilangkan imbuhan yang berada diakhir (*suffixes*) terlebih dahulu untuk menghilangkan imbuhan yang berada diawal (*prefix*). Berikut contoh dari stemming.



**Gambar 2. 3** Contoh *Stemming*

## 2.4. Pencarian

Aktivitas yang berkaitan dengan pengolahan data sering didahului dengan proses pencarian. Pencarian (*searching*) merupakan proses yang fundamental dalam pengolahan data. Proses pencarian adalah menemukan nilai tertentu didalam sekumpulan data. Aktivitas yang sering berkaitan dengan pengolahan data sering didahului dengan proses pencarian (Renaldi & Lidya, 2001).

### 2.4.1. Algoritma Pencarian *String*

*String Matching Algorithm* atau sering disebut juga pencocokan *string* adalah algoritma untuk melakukan pencarian semua kemunculan *string* pendek *pattern*  $[0 \dots n - 1]$ , yang disebut *pattern* di *string* yang lebih panjang *teks*  $[0 \dots m - 1]$  yang disebut teks (Charras & Lecroq, 2004).

Menurut arah pencariannya algoritma pencarian *string* dapat diklasifikasikan menjadi 3 bagian yaitu:

1. Dari arah kiri ke kanan:
  - Algoritma *Brute Force*

- Algoritma *Knuth Moriss Pratt*
2. Dari arah kanan ke kiri:
    - Algoritma *Boyer Moore*
  3. Dari arah yang di tentukan secara spesifik:
    - Algoritma *Colussi*
    - Algoritma *Crochemore-Perrin*

#### 2.4.2. Algoritma *Brute Force*

Algoritma *Brute Force* merupakan algoritma pencocokan *string* yang ditulis tanpa memikirkan peningkatan performa. Algoritma ini sangat jarang dipakai dalam praktik, namun berguna dalam studi perbandingan dan studi studi lainnya.

Cara kerja

Secara sistematis, langkah langkah yang dilakukan algoritma *Brute Force* pada saat pencocokan *string* adalah:

- a) Algoritma *Brute Force* mulai mencocokkan *pattern* awal teks.
- b) Dari kiri ke kanan, Algoritma ini mencocokkan karakter per karakter *pattern* dengan karakter teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:
  1. Karakter di *pattern* dan teks yang dibandingkan tidak cocok (*mismatch*).
  2. 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-b) sampai *pattern* berada di ujung teks.

Algoritma *brute force string match* adalah algoritma yang paling sederhana untuk memecahkan masalah *string match*. Cara kerja algoritma ini adalah dengan mencoba setiap posisi *pattern* terhadap teks, kemudian dilakukan proses pencocokan karakter dan teks pada posisi tersebut

### 2.4.3. Algoritma Knut Morris Pratt

Algoritma *Knuth Morris Pratt* adalah salah satu algoritma pencarian *string*, dikembangkan secara terpisah oleh Donal E. Knuth pada tahun 1967 dan James H. Morris bersama Vaughan R. Pratt pada tahun 1966, namun keduanya mempublikasikan secara bersama pada tahun 1977.

Jika melihat algoritma *Brute Force* lebih mendalam, kita mengetahui bahwa dengan mengingat beberapa perbandingan yang dilakukan sebelumnya kita dapat meningkatkan besar pergeseran yang dilakukan. Hal ini akan menghemat perbandingan, yang selanjutnya akan meningkatkan kecepatan pencarian (Charras & Lecroq, 2004).

Cara kerja:

Perhitungan pergeseran pada algoritma ini adalah sebagai berikut, bila terjadi ketidakcocokan pada saat *pattern* sejajar dengan  $teks[i \dots i + n - 1]$ , kita bisa menganggap ketidakcocokan pertama terjadi di antara  $teks[i+j]$  dan  $pattern[j]$ , dengan  $0 < j < n$ . Berarti,  $teks[i \dots i+j-1] = pattern[0 \dots j-1]$  dan  $a = teks[i+j]$  tidak sama dengan  $b = pattern[j]$ . Ketika kita menggeser, sangat beralasan bila ada sebuah  $v$  dari *pattern* akan sama dengan akhiran  $u$  dari sebagian teks. Sehingga bisa menggeser *pattern* agar awalan  $v$  tersebut sejajar dengan akhiran  $u$ .

Dengan kata lain, pencocokan *string* akan berjalan secara efisien bila kita mempunyai table yang menentukan seberapa panjang seharusnya kita menggeser

seandainya terdeteksi ketidakcocokan di karakter ke  $j$  dari *pattern*. Tabel itu harusnya memuat  $next[j]$  yang merupakan posisi karakter  $pattern[j]$  setelah digeser, sehingga kita bisa menggeser *pattern* sebesar  $j - next[j]$  relative terhadap teks (Knuth, Morris, & Pratt, 1997).

Secara sistematis, langkah langkah yang dilakukan algoritma *Knuth Morris Pratt* pada saat mencocokkan *string*:

- a) Algoritma *Knuth Morris Pratt* mulai mencocokkan *pattern* pada awal teks.
- b) Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter *pattern* dengan karakter di teks yang bersesuaian, samapai salah satu kondisi berikut dipenuhi:
  1. Karakter di *pattern* dan teks yang dibandingkan tidak cocok (*mismatch*).
  2. Semua karakter di *pattern* cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.

Algoritma kemudian menggeser *pattern* berdasarkan table *next*, lalu mengulang langkah 2 samapai *pattern* berada di ujung teks

#### **2.4.4. Algoritma Boyer Moore**

Algoritma *Boyer Moore* pertama kali diperkenalkan oleh Bob Boyer dan J.S Moore pada tahun 1977. Algoritma ini dikenal karena banyak digunakan pada algoritma pencocokan untuk banyak string (Imam Sulisty, 2008). Algoritma *Boyer Moore* merupakan salah satu contoh algoritma yang menggunakan arah dari kanan ke kiri, dan hal ini dianggap memiliki hasil yang paling baik dalam prakteknya yang bergerak dalam mencocokkan string (Sagita, 2013).

Menurut (Argakusumah & Hansun, 2014) Algoritma *Boyer Moore* adalah salah satu algoritma yang digunakan untuk mencari suatu string dalam teks. Dengan menggunakan algoritma *Boyer Moore* secara rata-rata proses pencarian menjadi lebih cepat jika dibandingkan dengan metode algoritma lainnya. (Rahmanita, 2014) juga mengatakan bahwa algoritma *Boyer Moore* termasuk dari salah satu algoritma pattern matching yang menggunakan beberapa kasus pengecekan teks dengan pattern. Cara kerjanya yaitu dengan cara membandingkan sebuah huruf dengan huruf yang ada di pattern yang dicari dan menggeser pattern tersebut sehingga posisinya sama dengan teks yang akan dicari dan membandingkan dengan kata tersebut atau biasa juga disebut dengan Character jump.

Algoritma *Boyer Moore* menggunakan pengetahuan teks pencarian untuk meningkatkan kecepatan pencarian secara signifikan. menggunakan langkah preprocessing untuk membuat fungsi kejadian dan fungsi pergeseran yang masing-masing digunakan untuk menjalankan heuristik karakter buruk dan heuristik karakter yang baik (Prabhakar, 2012). Menurut (Syauqi, 2014) Pergeseran karakter dalam algoritma *Boyer Moore* dapat dibuat dengan pendekatan *Match Heuristic* (MH) dan *Occurrence Heuristic* (OH). Nilai yang dihasilkan oleh MH disebut dengan *Good shifting shift*. MH digunakan untuk menangani kasus pencocokan yang di dalamnya terdapat pengulangan karakter. Sedangkan OH disebut dengan *Bad Character shift* yang digunakan untuk menghindari pengulangan yang gagal dari karakter dalam teks.

Karakter utama dalam algoritma *Boyer Moore* adalah dengan mencocokkan string mulai dari kanan ke kiri. Dengan karakter tersebut jika ada ketidakcocokan saat terjadi perbandingan maka string akan membuat pergerakan pattern melompat

lebih jauh untuk menghindari perbandingan karakter pada string yang diperkirakan gagal (Rahmanita, 2014). Prinsip dasar dari algoritma *Boyer Moore* yaitu: (Fau, Mesran, & Ginting, 2017)

1. Pembacaan karakter dilakukan dari kanan ke kiri
2. *Preprocessing* dimana terdapat kondisi *bad character preprocessing* dan *good suffix preprocessing*

Algoritma *Boyer Moore* melakukan perbandingan dari kanan ke kiri, tetapi dalam pergeseran window tetap dari kiri ke kanan. Jika terjadi ke tidak cocokan maka dilakukan perbandingan karakter teks dan karakter pola yang sebelumnya, yaitu dengan cara mengurangi indeks teks dan pola masing-masing satu (Kumara, 2009). Algoritma *Boyer Moore* memiliki dua fase. Fase yang pertama yaitu fase *preprocessing* dan fase yang kedua yaitu fase pencarian. Fase *preprocessing* memiliki dua fungsi yaitu *good suffix shift* dan *bad character shift* (Minandar, 2010).

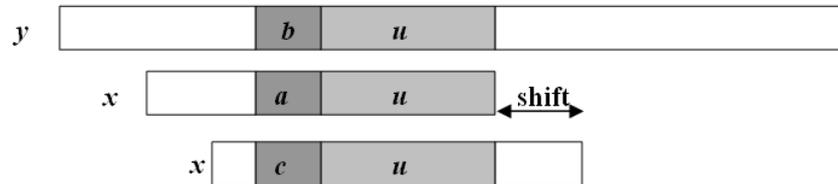
#### **2.3.4.1. Deskripsi Kerja Algoritma *Boyer Moore***

Perlu adanya contoh kasus untuk penjelasan *good-suffix shift* dan *bad-character shift* seperti kasus ketidak cocokan ditengah pencocokan karakter pada teks dan *pattern*. Karakter *pattern*  $x[i]=a$  tidak cocok dengan karakter teks  $y[i+j]=b$  saat pencocokan pada posisi  $j$ . Maka  $x[i+l .. m-1]=y[i+j+1 .. j+m-1]=u$  dan  $x[i] \neq y[i+j]$ .

##### **A. *Good-suffix shift***

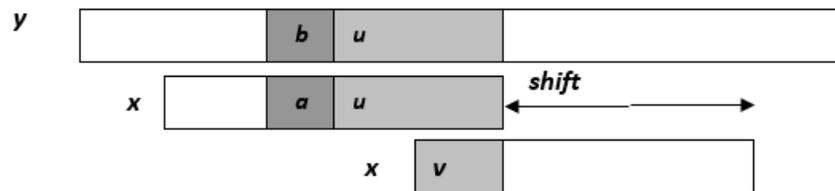
Konsep dari fungsi *good-suffix shift* adalah sebagai berikut:

1. *Good-suffix shift* adalah pergeseran yang dibutuhkan dari  $x[i]=a$  ke karakter lain yang terletak lebih kiri dari  $x[i]$  dan terletak di sebelah kiri segmen  $u$ .



**Gambar 2. 4** *Good-suffix shift*,  $u$  terjadi lagi didahului karakter  $c$  berbeda dari  $a$

2. Jika segmen tidak ada yang sama dengan  $u$ , maka dicari  $u$  yang merupakan *suffix* terpanjang  $u$ .



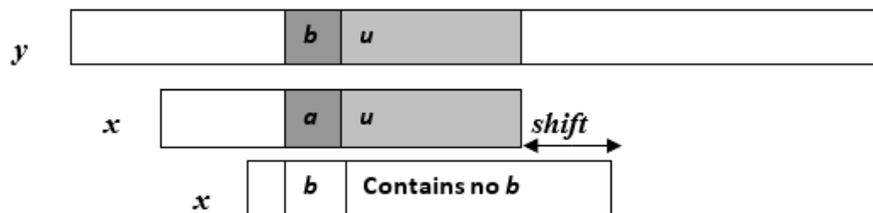
**Gambar 2. 5** *Good-suffix shift*, hanya *suffix* dari  $u$  yang terjadi lagi di *pattern*  $x$

### B. *Bad-character shift*

Berdasarkan contoh kasus, *bad-character* adalah karakter pada teks yaitu  $y[i+j]$  yang tidak cocok dengan karakter pada *pattern*.

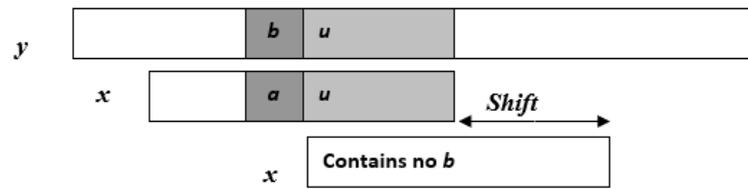
Berikut fungsi konsep *bad-character shift*:

1. Jika *bad-character*  $y[i+j]$  terdapat pada *pattern* di posisi terkanan  $k$  yang lebih kiri dari  $x[i]$  maka *pattern* digeser ke kanan sejauh  $i-k$ .



**Gambar 2. 6** *Bad-character shift*,  $b$  terdapat di *pattern*  $x$

2. Jika *bad-character*  $y[i+j]$  tidak ada pada *pattern* sama sekali, maka *pattern* digeser ke kanan sejauh  $i$ .



**Gambar 2. 7** *Bad-character shift*,  $b$  tidak ada di *pattern*  $x$

3. Jika *bad-character*  $y[i+j]$  terdapat pada *pattern* terkanan  $k$  yang lebih kanan dari  $x[i]$  maka *pattern* seharusnya digeser sejauh  $i-k$  yang hasilnya negatif (*pattern* digeser kembali ke kiri). Maka akan diabaikan.

Pada kasus ketidakcocokan di atas, algoritma akan membandingkan langkah yang diambil oleh fungsi *good-suffix shift* dan *bad-character shift* di mana langkah yang paling besar yang akan digunakan

#### 2.3.4.2. Cara kerja algoritma *Boyer Moore*

1. Menjalankan prosedur *preBmBc* dan *preBmGs* untuk memperoleh nilai banding pergeseran.
  - a. Prosedur *preBmBc* menentukan berapa besar pergeseran yang dibutuhkan untuk mencapai karakter tertentu pada *pattern* dari karakter *pattern* terakhir/terkanan. Hasil dari prosedur *preBmBc* disimpan pada tabel *BmBc* yang berisi nilai *Occurance Heuristic* (OH)
  - b. Prosedur *preBmGs* dilakukan setelah melakukan prosedur *suffix* pada *pattern*. Fungsi prosedur *suffix* adalah memeriksa kecocokan karakter yang dimulai dari karakter terakhir/terkanan dengan karakter yang dimulai dari setiap karakter yang lebih kiri dari karakter terkanan tadi.

Hasil dari prosedur *suffix* disimpan pada tabel *suff*. *Suff[i]* mencatat panjang dari *suffix* yang cocok dengan segmen dari *pattern* yang diakhiri karakter ke-*i*. Dengan prosedur *preBmGs*, dapat diketahui berapa banyak langkah pada *pattern* dari sebuah segmen ke segmen lain, yang letaknya sama lebih kiri dengan karakter di sebelah kiri segmen yang berbeda. Prosedur *preBmGs* menggunakan tabel *suff* untuk mengetahui semua pasangan segmen yang sama. Hasil dari prosedur *preBmGs* disimpan pada tabel *BmGs* yang berisi nilai *Match Heuristic* (MH).

2. Dilakukan proses pencarian *string* dengan menggunakan hasil dari prosedur *preBmBc* dan *preBmGs* yaitu tabel *BmBc* dan *BmGs*.

Contoh untuk menjelaskan proses inialisasi dari algoritma *Boyer Moore* dengan *pattern* **gcagagag** yang akan dicari pada *string* **gcatcgcagagagtatacagtacg**.

Dengan prosedur *preBmBc*, didapatkan jumlah pergeseran pada *pattern* yang dibutuhkan untuk mencapai karakter a, c, g, t dari posisi terkanan. Berdasarkan contoh diketahui untuk mencapai masing-masing karakter tadi dibutuhkan pergeseran sebanyak 1, 6, 2 dan 8.

Dengan prosedur *preBmGs*, dijalankan prosedur *suffix* terlebih dulu.

Dengan prosedur *suffix* akan diketahui:

- $\text{suff}[0] = 1$ , 1 karakter g posisi 7 cocok dengan 1 karakter g posisi 0.
- $\text{suff}[1] = 0$ , karakter g posisi 7 tidak cocok dengan karakter c posisi 1.
- $\text{suff}[2] = 0$ , karakter g posisi 7 tidak cocok dengan karakter a posisi 2.

- $\text{suff}[3] = 2$ , 2 karakter dimulai dari karakter g posisi 7 cocok dengan 2 karakter dimulai dari karakter g posisi 3, yang artinya karakter a,g posisi 6,7 cocok dengan karakter a,g posisi 2,3.
- $\text{suff}[4] = 0$ , karakter g posisi 7 tidak cocok dengan karakter a posisi 4.
- $\text{suff}[5] = 4$ , 4 karakter dimulai dari karakter g posisi 7 cocok dengan 4 karakter dimulai dari karakter 5, artinya karakter a,g,a,g posisi 4,5,6,7 cocok dengan karakter a,g,a,g posisi 2,3,4,5.
- $\text{suff}[6] = 0$ , karakter g posisi 7 tidak cocok dengan karakter a posisi 6.
- $\text{suff}[7] = 8$ , 8 karakter g,c,a,g,a,g,a,g posisi 0,1,2,3,4,5,6,7 cocok dengan 8 karakter g,c,a,g,a,g,a,g posisi 0,1,2,3,4,5,6,7.

Dengan prosedur BmGs akan didapatkan:

0 1 2 3 4 5 6 7

g c a g a g a g

- $\text{bmGs}[0] = 7$ , karakter ke-0 g adalah karakter sebelah kiri segmen cagagag. Tidak ada segmen cagagag lain dengan karakter sebelah kiri bukan g maka digeser 7 langkah.
- $\text{bmGs}[1] = 7$ , karakter ke-1 c adalah karakter sebelah kiri segmen agagag. Tidak ada segmen agagag lain dengan karakter sebelah kiri bukan c maka digeser 7 langkah.
- $\text{bmGs}[2] = 7$ , karakter ke-2 a adalah karakter sebelah kiri segmen gagag. Tidak ada segmen gagag lain dengan karakter sebelah kiri bukan a maka digeser 7 langkah.

- $bmGs[3]= 2$ . karakter ke-3 g adalah karakter sebelah kiri segmen agag. Karena ada segmen agag posisi 2,3,4,5 dengan karakter sebelah kiri bukan g yaitu c posisi 1 maka digeser 2 langkah.
- $bmGs[4]= 7$ , karakter ke-4 a adalah karakter sebelah kiri segmen gag. Karena tidak ada seamen gag lain dengan karakter sebelah kiri bukan a maka digeser 7 langkah.
- $bmGs[5]= 4$ . karakter ke-5 g adalah karakter sebelah kiri seamen ag. Karena ada segmen ag posisi 2,3 dengan karakter sebelah kiri bukan g yaitu c posisi 1 maka digeser 4 langkah.
- $bmGs[6]= 7$ , karakter ke-6 a adalah karakter sebelah kiri segmen yaitu a posisi 7. Karena tidak ada segmen g dengan karakter sebelah kirinya bukan a maka digeser 7 langkah.
- 1.  $bmGs[7]= 1$ , karakter ke-7 g adalah karakter sebelah kiri segmen dan karena segmen tidak ada maka digeser 1 langkah

#### 2.3.4.3. Kelebihan dan kekurangan algoritma *Boyer Moore*

Beberapa kelebihan yang terdapat pada algoritma *Boyer Moore*:

1. Digunakan sebagai mesin pencari
2. Membandingkan dari kanan ke kekiri
3. Lebih cepat menemukan pattern yang di cari, sehingga mampu menghemat waktu yang ditempuh.

Beberapa kekurangan yang terdapat pada algoritma *Boyer Moore*:

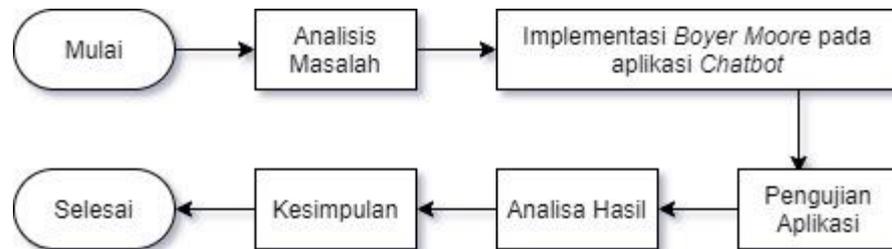
1. Pencocokan hanya dilakukan dari kanan ke kiri, karena tidak semua karakter memiliki kesamaan dan hanya karakter terakhir atau yang paling kiri, sehingga waktu yang diperlukan sedikit lama.

## BAB III METODOLOGI PENELITIAN

Pada bab metodologi penelitian di paparkan secara sistematis proses hasil analisa dan langkah yang di gunakan untuk menyelesaikan permasalahan sesuai tujuan penelitian sehingga mendapatkan hasil yang sesuai.

### 3.1. Prosedur Penelitian

Pada prosedur penelitian, akan dijelaskan mulai dari penentuan masalah, analisis penentuan solusi, dan penentuan metode *Boyer Moore* sebagai proses *string matching*. Prosedur penelitian akan di tampilkan pada diagram berikut:



**Gambar 3. 1** Prosedur Penelitian

Secara garis besar terdapat lima proses yang dilakukan untuk menyelesaikan penelitian ini. Penelitian dimulai dengan studi terhadap objek penelitian untuk menganalisa masalah dan menentukan usulan solusi dari data atau informasi yang terkumpul, usulan solusi yang di buat berupa chatbot dengan pendekatan *Natural Language Processing* dan *Boyer Moore* sebagai Algoritma *String Matching*. Proses selanjutnya dilakukan pembuatan sistem dengan implementasi usulan solusi yang telah di rancang dan selanjutnya diimplementasikan pada sistem yang sedang berjalan. Tahap berikutnya adalah melakukan pengujian sistem sehingga dapat dilakukan analisa dari hasil pengujian yang nantinya akan menjadi kesimpulan.

### 3.2. Desain Aplikasi

Pada subbab ini dibahas mengenai aplikasi yang dibangun. Aplikasi yang dibangun merupakan aplikasi *chatbot* yang membahas tentang kesehatan.

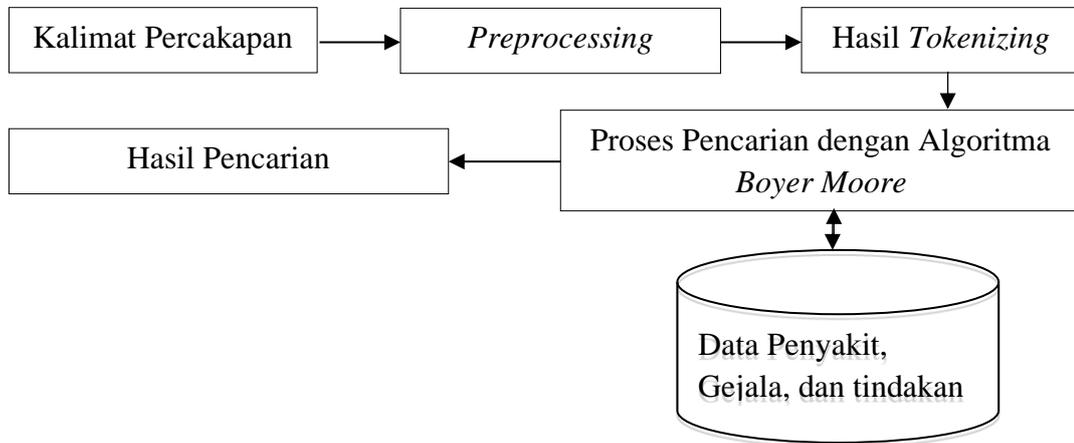
Aplikasi berbasis web, sehingga *user* dapat mengakses dimanapun dan kapanpun. Aplikasi dirancang dan dibangun secara bertahap, tahapan pertama yaitu input yang dimasukan *user*. Kemudian setelah user memberi masukan pertanyaan, kalimat tersebut akan memasuki tahap *preprocessing*. *Preprocessing* merupakan tahapan awal dalam mengelola data input sebelum memasuki proses utamanya. *Preprocessing* dilakukan dengan bertujuan untuk penyeragaman dan memudahkan pembacaan ketika telah memasuki proses utamanya. *Preprocessing* sendiri terdiri dari *case folding*, *tokenizing*, dan *stemming*.

1. *Case folding* yaitu proses mengubah huruf menjadi huruf kecil.
2. *Tokenizing* merupakan pemecahan kalimat dengan menghilangkan tanda baca dan simbol lainnya, pemecahan kalimat dilakukan berdasarkan spasi dari tiap kata hingga didapatkan potongan kata.
3. *Stemming* digunakan untuk menghilangkan kata imbuhan dan menjadikannya menjadi kata dasar.

Ketika proses *preprocessing* sudah selesai, data di tampung dan lanjut menjalankan prosedur Algoritma *Boyer Moore*.

### 3.3. Desain Proses

Berdasarkan desain aplikasi dapat di gambarkan alr proses dari pembuatan *chatbot*, berikut alur proses:



Gambar 3. 2 Desain Proses

#### 3.3.1. Input Kalimat

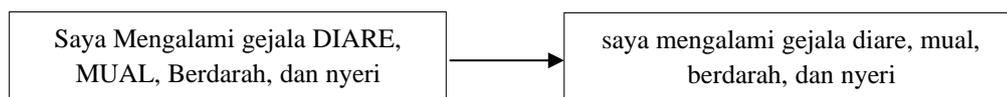
Percakapan menggunakan format teks bahasa Indonesia sehingga kalimat yang di proses pada proses *preprocessing* mendapat hasil yang maksimal dan mendapat jawaban yang sesuai.

#### 3.3.2. Preprocessing

Dari masukan pertanyaan user langkah selanjutnya yaitu proses *preprocessing*, yang terbagi menjadi *case folding*, *tokenizing* dan *stemming*.

- Proses *case folding* merupakan proses merubah semua karakter huruf menjadi huruf kecil (*lowercase*).

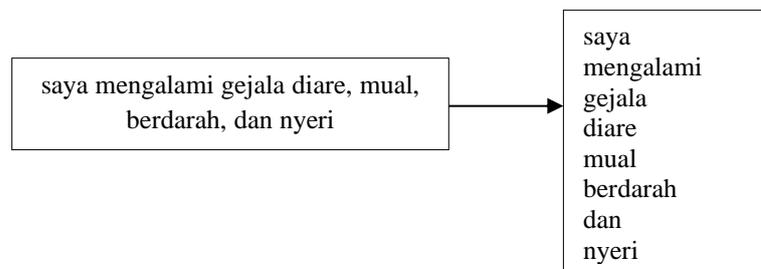
Hasil *output* terlihat seperti gambar dibawah ini



Gambar 3. 3 Hasil Case Folding

- Proses *tokenizing* merupakan proses pemisahan kata berdasarkan spasi dan menghapus karakter selain huruf yang kemudian dimasukkan kedalam sebuah tabel untuk proses selanjutnya.

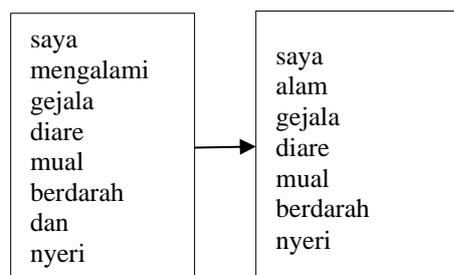
Hasil *output* terlihat seperti pada gambar dibawah ini.



**Gambar 3. 4** Hasil *Tokenizing*

- Proses *stemming* adalah proses menghilangkan imbuhan kata baik kata imbuhan yang berada didepan, ditengah, dibelakang, dan juga menghilangkan kata yang di anggap tidak berperan bagi sistem dalam menentukan jawaban sesuai dengan data yang telah di tentukan dalam *database*.

Setelah proses selesai, hasil output seperti gambar dibawah ini.



**Gambar 3. 5** Hasil *Stemming*

### 3.3.3. Implementasi *Boyer Moore* pada aplikasi *Chatbot*

Proses pencarian jawaban pada aplikasi *Chatbot* menggunakan algoritma *Boyer Moore*, yang mana setelah tahapan memecahan kalimat menjadi kata (*stemming*), sehingga menjadi beberapa kata dasar. Hasil dari *stemming* akan di cocokkan dengan data kata kunci menggunakan algoritma *Boyer Moore*, sehingga dari hasil pencocokan dapat diseleksi lagi untuk selanjutnya menampilkan jawaban berdasarkan *query sql* yang sesuai.

#### Contoh Implementasi Algoritma *Boyer Moore*

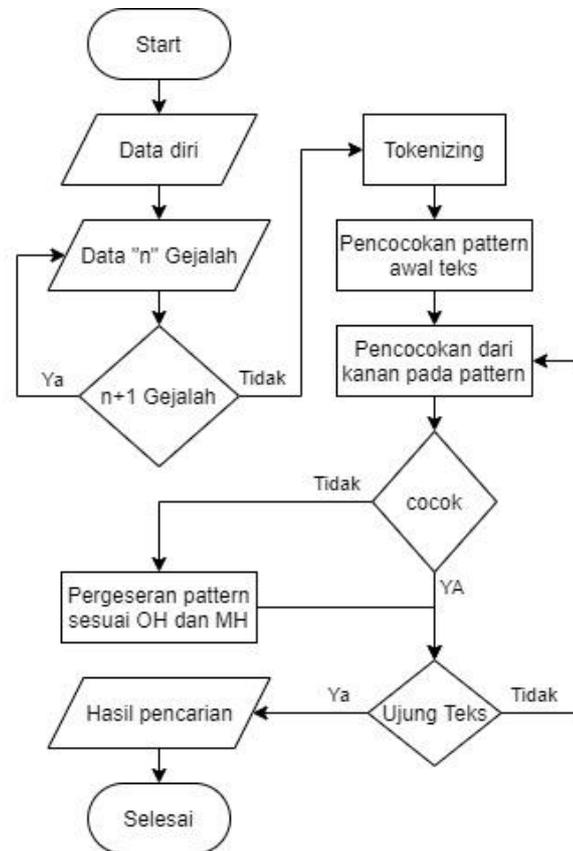
Pencarian di lakukan pada tabel data Gejala Penyakit dan Tindakan yang di lakukan.

**Tabel 3. 1** Contoh Data Gejala dan Penyakit

NO	NAMA PENYAKIT	GEJALA	TINDAKAN
1	VERTIGO	{PUSING, BERPUTAR, MUAL, MUNTAH, DEMAM}	{HISTIGO, DIMENHYDRINATE}
2	DIARE AKUT	{DIARE, CAIR, LAMA, LENDIR, BERDARAH, NYERI, MUAL}	{WAJIB TINDAKAN LANJUT, LANSUNG RAWAT INAP}
3	DIARE	{DIARE, MUAL, NYERI}	{DIAPECTA, ZINC, ANTASIDA, OBAT MUAL}

Pada Tabel 3.1 merupakan contoh data Penyakit, Gejala, dan Tindakan atau obat apa yang akan di berikan selanjutnya. Ketika user membuka aplikasi user akan mendapatkan pesan pembukanaan dan beberapa pertanyaan tentang gejala apa yang di rasakan, pertanyaan gejala akan di tanyakan secara bertahap dari gejala pertama sampai gejala terakhir. Setelah jawaban gejala terkumpul jawaban akan di satukan menjadi satu kalimat yang kemudian akan di lakukan proses tokenizi. Sehingga hasil tokenizing akan di cocokkan dengan data yang ada pada database

menggunakan algoritma *Boyer Moore*. Berikut diagram unruk proses pencocokan kata menggunakan algoritma *Boyer Moore*.



**Gambar 3. 6** Flowchart Boyer Moore

Seperti pada studi pustaka secara umum, terdapat tiga tahapan pada teknik penerapan algoritma *Boyer Moore*.

#### 1. Prosedur preBmB

Prosedur preBmBc memiliki tiga nilai penting:

- a. *Pattern*, sebagai subjek inialisasi terhadap teks.
- b. Karakter, sebagai karakter-karakter yang terdapat pada *pattern*.
- c. *Occurence Heuristic* (OH), sebagai nilai pergeseran yang diperoleh ketika menemukan ketidakcocokan karakter.
- d. Pergeseran, sebagai nilai yang dicapai ketika melakukan pergeseran dari kanan ke kiri *pattern*.

Langkah-langkah pelaksanaan prosedur:

- 1) *Create* atau berikan nilai kosong ke stack BmBc.
- 2) Lakukan perhitungan terhadap panjang *pattern*. Jika panjang tidak lebih dari satu, maka hentikan proses dengan menambahkan langsung nilai OH dan karakter ke dalam stack BmBc. Jika tidak, maka lanjutkan ke proses selanjutnya.
- 3) Cacahan karakter *pattern* mulai dari karakter ke-2 paling kanan.
- 4) Bandingkan setiap karakter yang dicacah terhadap stack BmBc, jika karakter yang dicacah tidak ditemukan di dalam stack, maka tambahkan karakter tersebut ke dalam stack dimana OH sama dengan jumlah pergeseran karakter yang telah dilakukan.
- 5) Lakukan langkah-4 kembali, dengan melakukan perpindahan 1 karakter ke kiri hingga mencapai karakter paling kiri secara terus menerus.
- 6) Jika telah mencapai karakter paling kiri, cacah karakter paling kanan lalu kembali ke langkah-4.



## 2. Prosedur preBmGs

Prosedur preBmGs memiliki enam nilai penting, diantaranya:

- a. *Pattern*, sebagai subjek pencocokan terhadap teks.
- b. *Match Heuristic* (MH), sebagai nilai pergeseran yang diperoleh ketika menemukan kecocokan suffix.
- c. *Compare*, sebagai akhiran atau sejumlah karakter sebelah kanan dari sebuah karakter *pattern* yang diperoleh dari pergeseran kanan ke kiri.
- d. *Prefix*, sebagai awalan atau karakter *pattern* yang diperoleh dari pergeseran dari kiri ke kanan.
- e. *Suffix*, sebagai akhiran sebelah kanan *prefix*.
- f. *Pergeseran*, sebagai nilai yang dicapai ketika melakukan pergeseran dari *compare*.

Langkah-langkah pelaksanaan prosedur:

- 1) *Create* atau berikan nilai kosong ke stack iBmGs dan Compare.
- 2) Lakukan aksi yang sama dengan preBmBc jika panjang *pattern* tidak lebih dari satu.
- 3) Cacah *compare* terhadap *pattern*. Lalu simpan hasil pencacah dengan menambahkan ke dalam stack compare.
- 4) Cacah *prefix* dan *suffix* terhadap *pattern*. Lalu simpan hasil pencacahan *prefix* ke dalam stack BmGs, dan bandingkan *suffix* terhadap compare dengan melihat kecocokan antara masing-masing *prefix*.
- 5) Jika panjang kedua *suffix* tidak sama besar, cacah salah satu *suffix* yang memiliki panjang terbesar ke arah kanan sepanjang *suffix* dengan panjang terkecil. Jika ditemukan kecocokan antar *suffix*, bandingkan *prefix* kedua

*suffix*. Jika *prefix* yang diperbandingkan adalah sama, maka kecocokan tidak dapat diterima.

- 6) Untuk *compare*, *prefix* didapat ketika terjadi pencacahan pada *compare*, *prefix* akan bernilai sama dengan karakter paling kanan dari hasil pencacahan terhadap *compare* sebelum terjadinya pencacahan. Jika tidak terjadi pemotongan, maka *prefix-compare* bernilai null.
- 7) Indeks *prefix* paling kanan, nilai MH harus selalu bernilai 1. Sedangkan nilai MH pada *prefix* lainnya, diberikan ketika ditemukan ketidakcocokan antara *prefix-suffix* terhadap *prefix-compare* dan ditemukannya kecocokan di antara kedua *suffix* sebesar nilai pergeseran terbentuknya *compare*
- 8) Ulangi langkah 5 hingga *compare* yang terakhir.
- 9) Ulangi langkah 5 hingga *suffix* yang terakhir.

### Contoh Kasus

*Pattern*: BERDARAH

Penyelesaian:

**Tabel 3. 2** Daftar pencacahan *prefix* dan *suffix* dari *pattern*

Suffix (kanan-kiri)	Prefix	Suffix(kiri-kanan)
Null	B	ERDARAH
B	E	RDARAH
BE	R	DARAH
BER	D	ARAH
BERD	A	RAH
BERDA	R	AH
BERDAR	A	H
BERDARA	H	Null

Dari tabel di atas, diasumsikan bahwa telah terjadi pemotongan pada kedua sisi *suffix* terhadap *prefix* pada *pattern*. Terlihat bahwa masing-masing *suffix* dipersatukan oleh *prefix* hingga menjadi satu kesatuan *pattern* utuh. Sedangkan *null*

diartikan kosong, yakni semua kondisi pencocokan akan bernilai benar jika dilakukannya perbandingan terhadap karakter tersebut.

Stack Compare

Compare	BERDARA	BERDAR	BERDA	BERD	BER	BE	B	Null
Pergeseran	1	2	3	4	5	6	7	8

Pattern	B	E	R	D	A	R	A	H
---------	---	---	---	---	---	---	---	---

Stack BmGs

Karakter	B	E	R	D	A	R	A	H
Nilai MH	-	-	-	-	-	-	-	1

MH selalu bernilai 1

Pattern	B	E	R	D	A	R	A	H
---------	---	---	---	---	---	---	---	---

Stack BmGs

Karakter	B	E	R	D	A	R	A	H
Nilai MH	-	-	-	-	-	-	8	1

Suffix compare Null

Pergeseran Suffix Comparator ke 8

Prefix	A
Suffix	H

Stack BmGs

Karakter	B	E	R	D	A	R	A	H
Nilai MH	-	-	-	-	-	8	8	1

Suffix compare Null

Pergeseran Suffix Comparator ke 8

Pattern	B	E	R	D	A	R	A	H
---------	---	---	---	---	---	---	---	---

Prefix	R
Suffix	AH

Stack BmGs

Karakter	B	E	R	D	A	R	A	H
Nilai MH	-	-	-	-	8	8	8	1

Suffix compare Null

Pergeseran Suffix Comparator ke 8

Pattern	B	E	R	D	A	R	A	H
---------	---	---	---	---	---	---	---	---

Prefix	A
Suffix	RAH

Stack BmGs

Karakter	B	E	R	D	A	R	A	H
Nilai MH	-	-	-	8	8	8	8	1

Suffix compare Null

Pergeseran Suffix Comparator ke 8

Pattern	B	E	R	D	A	R	A	H
---------	---	---	---	---	---	---	---	---

Prefix	D
Suffix	ARAH

Stack BmGs

Karakter	B	E	R	D	A	R	A	H
Nilai MH	-	-	8	8	8	8	8	1

Suffix compare Null

Pergeseran Suffix Comparator ke 8

Pattern	B	E	R	D	A	R	A	H
---------	---	---	---	---	---	---	---	---

Prefix	R
Suffix	DARAH

Stack BmGs

Karakter	B	E	R	D	A	R	A	H
Nilai MH	-	8	8	8	8	8	8	1

Suffix compare Null

Pergeseran Suffix Comparator ke 8

Pattern	B	E	R	D	A	R	A	H
---------	---	---	---	---	---	---	---	---

Prefix	E
Suffix	RDARAH

Stack BmGs

Karakter	B	E	R	D	A	R	A	H
Nilai MH	8	8	8	8	8	8	8	1

Suffix compare Null

Pergeseran Suffix Comparator ke 8

Pattern	B	E	R	D	A	R	A	H
---------	---	---	---	---	---	---	---	---

Prefix	B
Suffix	ERDARAH

Gambar 3. 8 Penyelesaian contoh kasus Prosedur preBmGs

Pada gambar 3.5 terlihat bahwa pergeseran pada *compare* memiliki peran utama dalam penentuan nilai MH. Berbeda dengan nilai OH, nilai MH selalu diberikan pada karakter-karakter yang ada pada *pattern*. Penyusunan stack BmGs pun berdasarkan urutan indeks posisi karakter pada *pattern*.

### 3. Prosedur BM

Prosedur BM memiliki empat nilai penting, diantaranya:

- a. *Pattern*, merupakan subjek pencocokan terhadap teks.
- b. *Teks*, sebagai objek pencocokan.
- c. *Stack BmBc*, sebagai pembanding ketika ditemukan ketidakcocokan.
- d. *Stack BmGs*, sebagai pembanding ketika ditemukan kecocokan.

Langkah-langkah pelaksanaan prosedur:

- 1) Pencocokan dimulai dari indeks teks terkecil atau dari karakter terkiri.
- 2) Pencocokan per karakter dimulai dari karakter paling kanan *pattern*.
- 3) Setiap kali ditemukan ketidakcocokan karakter, maka ambil nilai OH pada stack BmBc dengan karakter yang berkesesuaian terhadap teks, lalu nilai OH dikurangi dengan jumlah kecocokan yang telah terjadi. Ambil nilai MH pada stack BmGs dengan indeks karakter yang ditemukan ketidakcocokan, lalu bandingkan dengan nilai OH yang telah dioperasikan. Dengan nilai terbesar, geser *pattern* ke arah kiri teks dan ulangi pencocokan.
- 4) Jika nilai kecocokan sama dengan panjang *pattern* maka *pattern* telah ditemukan pada teks, geser *pattern* sepanjang jumlah karakter *pattern* ke arah kiri teks untuk melanjutkan pencocokan selanjutnya.

**Contoh kasus:**

Pattern: BERDARAH

Teks: TESBERDARAH

Penyelesaian:

Stack BmBc								
Karakter	B	E	R	D	A	R	A	H
Nilai OH	7	6	2	4	1	2	1	0

Stack BmGs								
Karakter	B	E	R	D	A	R	A	H
Nilai MH	8	8	8	8	8	8	8	1

Index	0	1	2	3	4	5	6	7	8	9	10
Text	T	E	S	B	E	R	D	A	R	A	H
Pattern	B	E	R	D	A	R	A	H			

Max BMBC[A] = 1 banding BMGS[H]= 1

Shift = 1 pergeseran

Text	T	E	S	B	E	R	D	A	R	A	H
Pattern		B	E	R	D	A	R	A	H		

Max BMBC[R] = 2 banding BMGS[H]= 1

Shift = 2 pergeseran

Text	T	E	S	B	E	R	D	A	R	A	H
Pattern				B	E	R	D	A	R	A	H

**Gambar 3. 9** Penyelesaian contoh kasus Prosedur BM

Pada proses pencarian diatas memberikan hasil bahwa kata BERDARAH ditemukan pada data gejala yang ada. Sehingga data gejala yang dimaksud akan dimunculkan dalam bentuk tabel.

**Tabel 3. 3** Contoh hasil pencarian

NO	NAMA PENYAKIT	GEJALA	TINDAKAN
1	DIARE AKUT	{DIARE, CAIR, LAMA, LENDIR, BERDARAH, NYERI, MUAL}	{WAJIB TINDAKAN LANJUT, LANSUNG RAWAT INAP}

Pada Gambar 3.5 pergeseran pada pattern selalu mengikuti keputusan dari hasil perbandingan OH dan MH. Pada umumnya, pencocokan standar akan mencocokkan tiap karakter, ketika ditemukan ketidakcocokan maka pattern akan

digeser satu langkah ke arah selanjutnya. Namun tidak demikian pada algoritma *Boyer Moore*. Inilah yang menjadi salah satu keunggulan dan alasan penerapan algoritma *Boyer Moore* sebagai algoritma pencarian data pada aplikasi *Chatbot*.

### 3.3.4. Output

*Output* merupakan hasil jawaban dari masukan *user* yang telah melewati tahap *preprocessing* dan proses pencocokan kata kunci pada *database*. Sehingga akan menampilkan hasil pada kolom tindakan sesuai dengan kata kunci pada kolom gejala.

### 3.4. Desain Pengukuran Akurasi

Desain akurasi pada penelitian menggunakan rumus pengukuran akurasi ROC (*Receiver Operating Character*) dengan pengukuran klasifikasi dalam bentuk dua dimensi yang di formulasikan dalam table kontingensi 2 X 2 untuk menganalisa.

**Tabel 3. 4 Tabel Kontingensi ROC**

		Kelas Sebenarnya	
		<i>True</i>	<i>False</i>
Kelas Prediksi	<i>Positive</i>	<i>True Positive</i>	<i>False Positive</i>
	<i>Negative</i>	<i>True Negative</i>	<i>False Negative</i>

Kreteria ROC:

- True Positive Rate* disebut juga *Sensitivity* (TPR)= $TP/(TP+FN)$
- True Negative Rate* disebut juga *Specifity* (TNR)=  $TN/(TN+FP)$
- Accuracy* =  $(TP+TN)/(TP+FP+TN+FN)$

Keterangan:

- TP = *True Positive* (Nilai Benar)  
 FN = *False Negative* (Kesalahan 2)  
 FP = *False Positive* (Kesalahan 1)  
 TN = *True Negative* (Nilai Salah)

**BAB VI**  
**UJI COBA DAN PEMBAHASAN**

**4.1. Uji Coba Sistem**

Uji coba sistem merupakan tahap lanjut dari perancangan sistem. Hasil dari uji coba adalah sebuah sistem yang siap untuk digunakan dan di terapkan.

**4.1.1. Uji Coba Perangkat Keras**

Spesifikasi perangkat keras (*hardware*) yang digunakan untuk uji coba aplikasi dapat dilihat pada tabel 4.1

**Tabel 4. 1 Perangkat Keras Yang Digunakan**

<b>Perangkat Keras</b>	<b>Keterangan</b>
<i>Processor</i>	Intel Core i5 3.7 GHz
<i>Memory</i>	DDR3L 8GB
<i>Harddisk</i>	500GB HDD
VGA	Intel HD Graphics 530
<i>Monitor</i>	21" HD (1366 x 768)

**4.1.2. Uji Coba Perangkat Lunak**

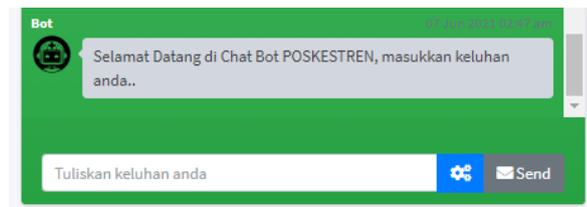
Spesifikasi perangkat lunak (*software*) yang digunakan untuk uji coba Sistem dapat dilihat pada tabel 4.2.

**Tabel 4. 2 Perangkat Lunak Yang Digunakan**

<b>Perangkat Lunak</b>	<b>Keterangan</b>
Sistem Operasi	Windows 10
IDE	Visual Studio Code 1.56.2
Web Server	XAMPP version 3.2.4
Database Server	MySQL
Web Browser	Google Chrome 91.0.4472.77

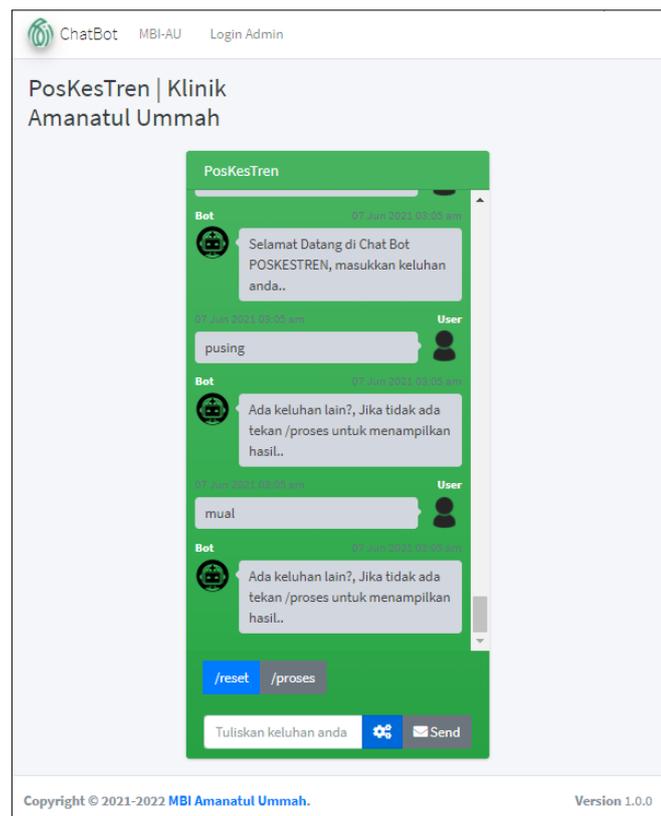
### 4.1.3. Uji Coba Algoritma *Boyer Moore*

Algoritma *Boyer Moore* di terapkan pada proses pencarian jawaban dari gejala-gejala yang di masukkan pengguna pada aplikasi. Berikut tampilan uji coba proses pencarian jawaban.



**Gambar 4. 1 Form input keluhan/gejala**

Pada Gambar 4.1 merupakan form input keluhan untuk pasien dimana form bisa di beri inputan berupa teks yang akan diberi jawaban langsung oleh aplikasi secara otomatis setelah menekan tombol enter atau klik tombol “Send”.



**Gambar 4. 2 Contoh Input Keluhan**

Pada Gambar 4.2 merupakan hasil pesan yang tampil ketika pasien selesai memberi inputan dan secara langsung di jawab oleh aplikasi.

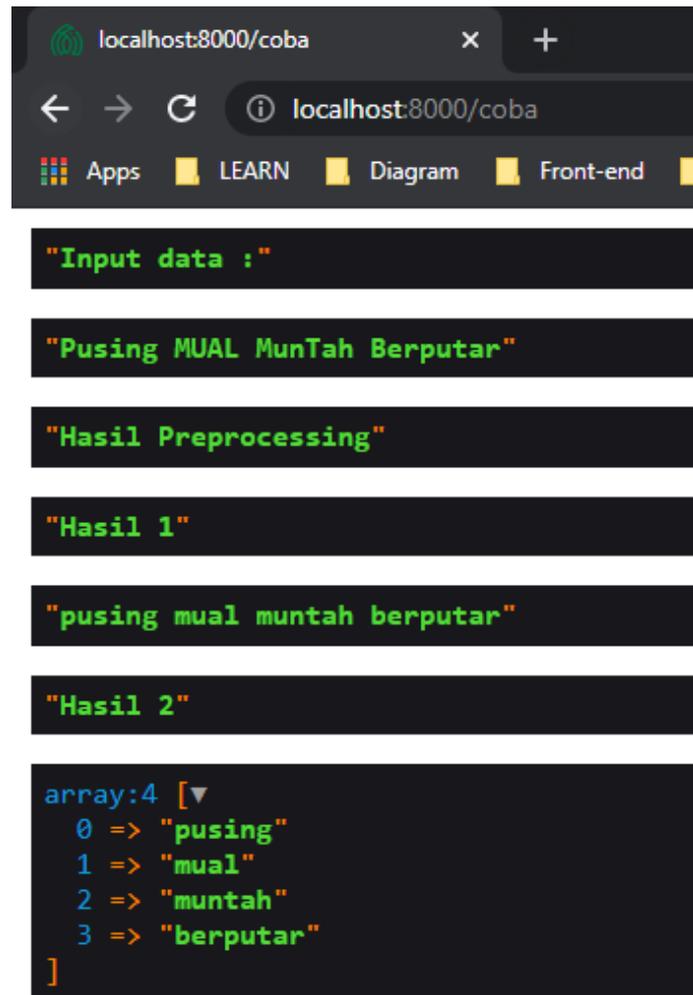


**Gambar 4. 3 Hasil Proses Keluhan**

Pada Gambar 4.3 merupakan proses pencarian dan menampilkan hasil jawaban dari seluruh keluhan yang telah di masukkan oleh pasien yang selanjutnya aplikasi akan menampilkan hasil atau tindakan pertama yang bisa di lakukan oleh pasien sebelum datang menuju klinik dengan membawa *printout* hasil konsultasi.

Proses pencarian diawali dengan *preprocessing* dimana teks akan diformat sesuai dengan kebutuhan proses algoritma.

*Preprocessing* dilakukan pada semua masukan. Berikut hasil dari proses fungsi *preprocessing* jika di tampilkan.



```
localhost:8000/coba
localhost:8000/coba
Apps LEARN Diagram Front-end
"Input data : "
"Pusing MUJAL MunTah Berputar"
"Hasil Preprocessing"
"Hasil 1"
"pusing mual muntah berputar"
"Hasil 2"
array:4 [
  0 => "pusing"
  1 => "mual"
  2 => "muntah"
  3 => "berputar"
]
```

**Gambar 4. 4 Hasil *Preprocessing***

Pada Gambar 4.4 *Input* data dari *user* di proses menjadi *lowercase* terlebih dahulu dengan fungsi “*strtolower*”, selanjutnya data di simpan setelah melalui proses pemisahan tiap kata berdasarkan karakter khusus (spasi, *underscore* “\_”) sehingga menjadi array data.

## 4.2. Pengujian Sistem

Pengujian sistem bertujuan untuk menemukan kesalahan dan kekurangan pada sistem dan bermaksud untuk mengetahui sistem sudah memenuhi kriteria sesuai dengan tujuan perancangan.

### 4.2.1. Pengujian Black Box

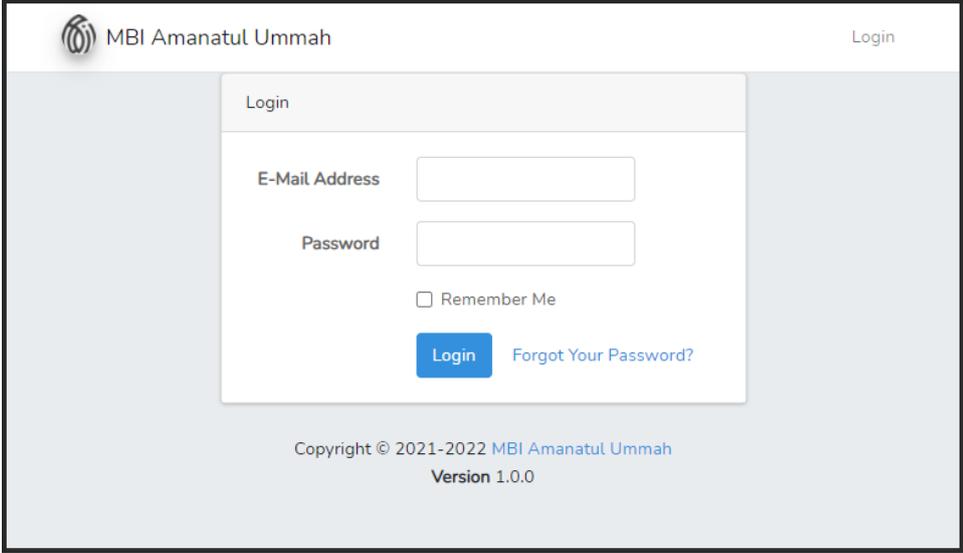
Pengujian fungsional yang digunakan untuk menguji sistem yang baru adalah metode pengujian alpha. Metode yang digunakan dalam pengujian ini adalah pengujian *black box* yang berfokus pada persyaratan fungsional dari sistem yang dibangun.

#### a. Form Login

Menu Login berfungsi untuk melakukan verifikasi dan membatasi hak – hak penggunaan aplikasi yang dimiliki oleh user.

**Tabel 4. 3 Uji Coba Form Login**

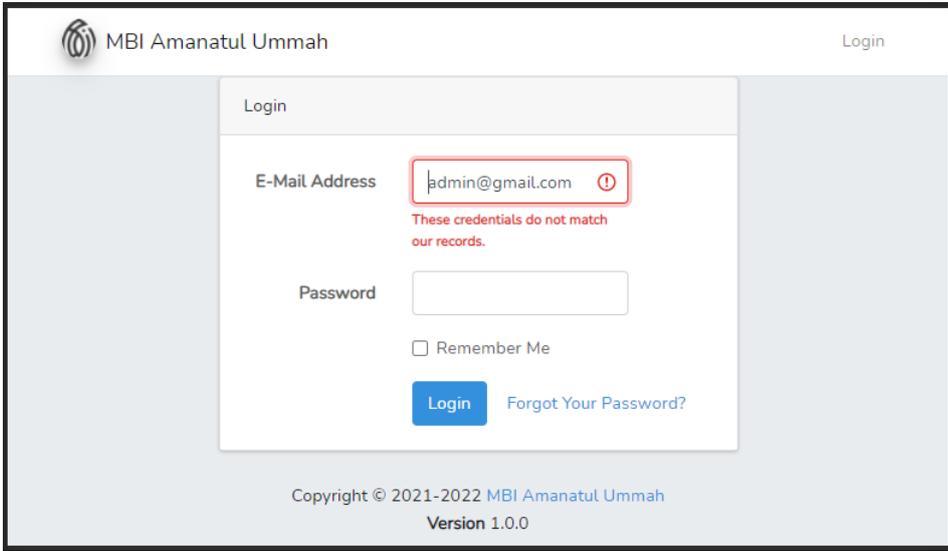
No	Tujuan	Input	Output Diharapkan	Output Sistem
1	Deskripsi <i>username</i> dan <i>password</i> valid	- <i>username</i> : admin@gmail.com - <i>password</i> : admin	Halaman login tertutup, halaman admin aktif	1. Sukses 2. Login Tampil <i>Dashboard admin</i>
2	Deskripsi <i>username</i> dan <i>password</i> tidak valid	- <i>username</i> : admin@gmail.com - <i>password</i> : test	Tampil <i>alert</i> <i>username</i> dan <i>password</i> salah	1. Sukses 2. Login tidak berhasil 3. Muncul pesan seperti yang diharapkan



The screenshot shows the login page for MBI Amanatul Ummah. At the top left is the logo and the text "MBI Amanatul Ummah". At the top right is the word "Login". The main content is a white box with a light gray border, titled "Login". Inside this box, there are two input fields: "E-Mail Address" and "Password". Below the "Password" field is a checkbox labeled "Remember Me". At the bottom of the box are two buttons: a blue "Login" button and a link "Forgot Your Password?". Below the white box, centered, is the text "Copyright © 2021-2022 MBI Amanatul Ummah" and "Version 1.0.0".

**Gambar 4. 5 Form Login**

Gambar 4.8 Merupakan tampilan halaman login dimana pada halaman terdapat *input textfield Email* dan *password* sebagai validasi untuk masuk ke halaman admin.



The screenshot shows the same login page as Gambar 4.5, but with a failed login attempt. The "E-Mail Address" field now contains the text "admin@gmail.com" and has a red border with a red warning icon (an exclamation mark inside a circle) to its right. Below the "E-Mail Address" field, there is a red error message: "These credentials do not match our records." The "Password" field is empty. The "Remember Me" checkbox, "Login" button, and "Forgot Your Password?" link are still present. The footer text "Copyright © 2021-2022 MBI Amanatul Ummah" and "Version 1.0.0" is also visible.

**Gambar 4. 6 Login Gagal**

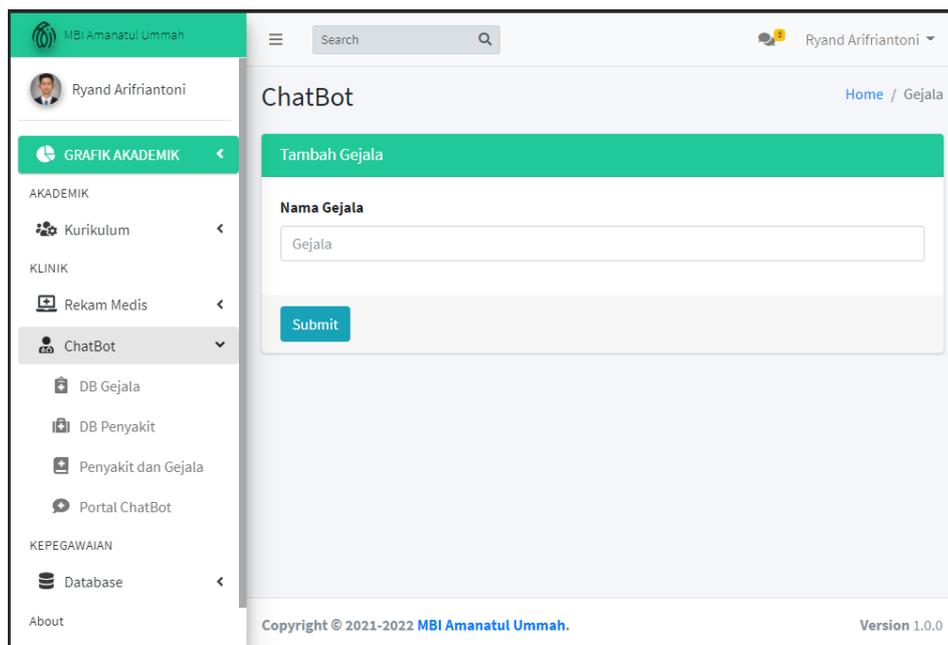
Gambar 4.9 halaman login ketika kombinasi email dan password tidak ada kecocokan dengan data yang terdapat di database dan memunculkan *alert*.

b. *Form* tambah gejala

*Form* tambah gejala berfungsi untuk menambahkan data gejala yang belum ada pada *database*.

**Tabel 4. 4 Uji Coba *Form* Tambah Gejala**

No	Tujuan	Input	Output Diharapkan	Output Sistem
1	Menambahkan data baru.	Memasukkan data kemudian menekan tombol submit.	Data berhasil disimpan	1. Sukses 2. Data tersimpan di database
2	Menghindari data input kosong	Tidak memasukkan data kemudian tekan tombol simpan	Tampil <i>alert</i> data tidak boleh kosong	1. Sukses 2. Muncul <i>Alert</i> data tidak boleh kosong.



**Gambar 4. 7 *Form* Tambah Gejala**

Gambar 4.10 Merupakan tampilan halaman tambah gejala dimana pada halaman terdapat input *textfield* nama gejala sebagai input data gejala baru.

Aksi	Nomor	ID Gejala	Nama Gejala
<input checked="" type="checkbox"/>	1	1	kepala_panas
<input checked="" type="checkbox"/>	2	3	badan_lemas
<input checked="" type="checkbox"/>	3	4	meriang
<input checked="" type="checkbox"/>	4	6	diare
<input checked="" type="checkbox"/>	5	7	mual

**Gambar 4. 8 Data Gejala**

Gambar 4.11 halaman database gejala yang tampil setelah menginputkan data gejala pada *form* tambah gejala.

**Gambar 4. 9 Alert Data Tidak Boleh Kosong Form Gejala**

Gambar 4.12 *alert* akan muncul jika *textfield* di kosongi dan melakukan klik tombol submit.

c. *Form* tambah penyakit

*Form* tambah penyakit berfungsi untuk menambahkan data penyakit meliputi nama penyakit dan tindakan pertama yang harus dilakukan.

Tabel 4. 5 Uji Coba *Form Tambah Penyakit*

No	Tujuan	Input	Output Diharapkan	Output Sistem
1	Menambahkan data baru.	Memasukkan data kemudian menekan tombol submit.	Data berhasil disimpan	1. Sukses 2. Data tersimpan di database
2	Menghindari data input kosong	Tidak memasukkan data kemudian tekan tombol simpan	Tampil <i>alert</i> data tidak boleh kosong	3. Sukses 4. Muncul <i>Alert</i> data tidak boleh kosong.

Gambar 4. 10 *Form Tambah Penyakit*

Gambar 4.13 Merupakan tampilan halaman tambah data penyakit dimana pada halaman terdapat dua input *textfield* nama penyakit sebagai input data penyakit baru dan tindakan pertama pasien.

Aksi	Nomor	ID Penyakit	Nama Penyakit	Tindakan Sementara
<input checked="" type="checkbox"/>	1	1	vertigo	Segera Menuju Klinik Di Dampingi Teman/Ustadzah (HISTIGO, DIMENHYDRINATE)
<input checked="" type="checkbox"/>	2	2	diare akut	Wajib Tindakan Lanjut, Lansung Rawat Inap
<input checked="" type="checkbox"/>	3	3	diare	Petugas Apoteker (Diapecta, Zinc, Antasida, Obat Mual)
<input checked="" type="checkbox"/>	4	4	scabies	Butuh Observasi Lanjutan Dari Dokter
<input type="checkbox"/>	5	5	demam typhoid	Wajib Tindakan Lanjut, Lanseune Rawat Inap

**Gambar 4. 11 Data Penyakit**

Gambar 4.14 tampilan halaman *database* penyakit setelah menginputkan data penyakit pada *form* tambah penyakit.

**Gambar 4. 12 Alert Data Tidak Boleh Kosong Form Penyakit**

Gambar 4.15 alert akan muncul jika *textfield* di kosongi dan melakukan klik tombol *submit*

d. *Form* tambah data gejala dan penyakit

*Form* gejala dan penyakit berfungsi untuk menambahkan data kombinasi antara data penyakit dan beberapa data gejala yang di timbulkan sesuai acuan data yang tersimpan di database.

**Tabel 4. 6 Uji Coba *Form* Tambah Data Gejala dan Penyakit**

No	Tujuan	Input	Output Diharapkan	Output Sistem
1	Menambahkan data baru.	Memasukkan data kemudian menekan tombol submit.	Data berhasil disimpan	1. Sukses 2. Data tersimpan di database
2	Menghindari data input kosong	Tidak memasukkan data kemudian tekan tombol simpan	Tampil <i>alert</i> data tidak boleh kosong	5. Sukses 6. Muncul <i>Alert</i> data tidak boleh kosong.

**Gambar 4. 13 *Form* Data Penyakit dan gejala**

Pada Gambar 4.15 merupakan halaman tambah data penyakit dan gejala, dimana pada input *textselected* nama penyakit terdapat list penyakit yang telah di inputkan di halaman tambah penyakit sebagaimana juga input *textselected* pilih gejala hanya saja pada data gejala bisa menginputkan beberapa gejala sesuai acuan dokter.

Database Penyakit Dan Gejala

Aksi	Nomor	ID	Nama Penyakit	Nama Gejala
<input checked="" type="checkbox"/>	1	1	vertigo	pusing
<input checked="" type="checkbox"/>	2	2	vertigo	berputar
<input checked="" type="checkbox"/>	3	3	vertigo	mual
<input checked="" type="checkbox"/>	4	4	vertigo	muntah
<input checked="" type="checkbox"/>	5	5	vertigo	demam

Copyright © 2021-2022 MBI Amanatul Ummah. Version 1.0.0

**Gambar 4. 14 Data Penyakit dan Gejala**

Pada gambar 4.17 tampilan halaman database penyakit dan gejala setelah menginputkan data pada form tambah penyakis dan gejala.

Tambahkan Dataa Penyakit dan Gejala

**Pilih Nama Penyakit**

vertigo

**Pilih Gejala**

Pilih Gejala

Please fill out this field.

Submit

**Gambar 4. 15 Alert Data Tidak Boleh Kosong Form Penyakit dan Gejala**

Gambar 4.18 *alert* akan muncul jika data di kosongi dan melakukan klik tombol *submit*.

#### 4.2.2. Pengujian User

Tujuan dilakukan pengujian adalah untuk mengetahui kualitas dari perangkat lunak, apakah sesuai dengan rancangan dan harapan atau belum. Maka dilakukan pengujian lanjut dengan cara memberikan kuesioner pada calon pengguna perangkat lunak. Contoh kuesioner bisa dilihat pada lampiran 1. Adapun metode penelitian yang digunakan adalah metode kuantitatif.

Kuesioner diberikan kepada 10 sample calon pengguna perangkat lunak. Kuesioner terdiri dari 20 pertanyaan dan saran. Kuisisioner dibuat menggunakan skala *likert* dengan skala 1 sampai 5. Tabel bobot nilai bisa dilihat pada Tabel 4.7

**Tabel 4. 7 Bobot Nilai**

Huruf	Nilai
A	1
B	2
C	3
D	4
E	5

**Tabel 4. 8 Keterangan Nilai Kuesioner**

Jawaban	Keterangan
0% - 19.99%	Sangat (Tidak Setuju, Buruk atau Kurang Sekali)
20% - 39.99%	Tidak Setuju atau Kurang Baik
40% - 59.99%	Cukup atau Netral
60% - 79.99%	Setuju, Baik atau Suka
80% - 100%	Sangat (Setuju, Baik, Suka)

Untuk mendapatkan hasil interpretasi, harus diketahui dulu skor tertinggi

(X) dan angka terendah (Y) untuk item penilaian dengan rumus sebagai berikut:

$Y = \text{Skor tertinggi likert} \times \text{jumlah responden}$

$X = \text{Skor terendah likert} \times \text{jumlah responden}$

**Rumus Index % = Total Skor / Y x 100**

Jumlah skor tertinggi untuk item sangat setuju ialah  $5 \times 10 = 50$ , sedangkan item sangat tidak setuju ialah  $1 \times 10 = 10$

Berikut ini adalah hasil prosentase masing-masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus diatas.

1. Apakah informasi yang disediakan oleh aplikasi mudah dimengerti?

**Tabel 4. 9 Hasil Pengujian Kuesioner Pertanyaan Nomor 1**

Bobot Soal	Keterangan	Responden	Skor	Hasil
5	Sangat Setuju	3	15	86%
4	Setuju	7	28	
3	Netral			
2	Tidak Setuju			
1	Sangat Tidak Setuju			
Total		10	43	

Dari hasil di atas maka dapat disimpulkan responden **sangat setuju** bahwa aplikasi mudah untuk dipahami.

2. Secara keseluruhan apakah penggunaan aplikasi ini memuaskan?

**Tabel 4. 10 Hasil Pengujian Kuesioner Pertanyaan Nomor 2**

Bobot Soal	Keterangan	Responden	Skor	Hasil
5	Sangat Setuju			78%
4	Setuju	9	36	
3	Netral	1	3	
2	Tidak Setuju			
1	Sangat Tidak Setuju			
Total		10	39	

Dari hasil di atas maka dapat disimpulkan responden **setuju** bahwa penggunaan aplikasi ini memuaskan.

3. Apakah aplikasi ini sesuai dengan kebutuhan??

**Tabel 4. 11 Hasil Pengujian Kuesioner Pertanyaan Nomor 3**

Bobot Soal	Keterangan	Responden	Skor	Hasil
5	Sangat Setuju	2	10	84%
4	Setuju	8	32	
3	Netral			
2	Tidak Setuju			
1	Sangat Tidak Setuju			
Total		10	42	

Dari hasil di atas maka dapat disimpulkan responden **sangat setuju** bahwa aplikasi ini sesuai dengan kebutuhan.

4. Apakah dapat dengan mudah menghindari kesalahan dalam menggunakan aplikasi?

**Tabel 4. 12 Hasil Pengujian Kuesioner Pertanyaan Nomor 4**

Bobot Soal	Keterangan	Responden	Skor	Hasil
5	Sangat Setuju	4	20	88%
4	Setuju	6	24	
3	Netral			
2	Tidak Setuju			
1	Sangat Tidak Setuju			
Total		10	44	

Dari hasil di atas maka dapat disimpulkan responden **sangat setuju** bahwa responden dapat dengan mudah menghindari kesalahan dalam penggunaan aplikasi.

5. Apakah aplikasi bermanfaat bagi pengguna?

**Tabel 4. 13 Hasil Pengujian Kuesioner Pertanyaan Nomor 5**

Bobot Soal	Keterangan	Responden	Skor	Hasil
5	Sangat Setuju	3	15	86%
4	Setuju	7	28	
3	Netral			
2	Tidak Setuju			
1	Sangat Tidak Setuju			
Total		10	43	

Dari hasil di atas maka dapat disimpulkan responden **sangat setuju** bahwa aplikasi bermanfaat bagi pengguna.

6. Apakah tampilan menu dalam aplikasi mudah untuk dikenali?

**Tabel 4. 14 Hasil Pengujian Kuesioner Pertanyaan Nomor 6**

Bobot Soal	Keterangan	Responden	Skor	Hasil
5	Sangat Setuju	4	20	86%
4	Setuju	5	20	
3	Netral	1	3	
2	Tidak Setuju			
1	Sangat Tidak Setuju			
Total		10	43	

Dari hasil di atas maka dapat disimpulkan responden **sangat setuju** bahwa menu di dalam aplikasi mudah untuk di kenali.

7. Apakah aplikasi mempunyai kemampuan dan fungsi sesuai yang diharapkan??

**Tabel 4. 15 Hasil Pengujian Kuesioner Pertanyaan Nomor 7**

Bobot Soal	Keterangan	Responden	Skor	Hasil
5	Sangat Setuju	4	20	80%
4	Setuju	2	8	
3	Netral	4	12	
2	Tidak Setuju			
1	Sangat Tidak Setuju			
Total		10	40	

Dari hasil di atas maka dapat disimpulkan responden setuju bahwa aplikasi mudah untuk dioperasikan.

8. Apakah dengan adanya aplikasi ini membantu pengelolaan data menjadi lebih mudah?

**Tabel 4. 16 Hasil Pengujian Kuesioner Pertanyaan Nomor 8**

Bobot Soal	Keterangan	Responden	Skor	Hasil
5	Sangat Setuju	3	15	82%
4	Setuju	5	20	
3	Netral	2	6	
2	Tidak Setuju			
1	Sangat Tidak Setuju			
Total		10	41	

Dari hasil di atas maka dapat disimpulkan responden **sangat setuju** bahwa aplikasi dapat membantu pengolahan data lebih mudah.

9. Apakah dengan adanya aplikasi ini memberikan informasi data secara akurat?

**Tabel 4. 17 Hasil Pengujian Kuesioner Pertanyaan Nomor 9**

Bobot Soal	Keterangan	Responden	Skor	Hasil
5	Sangat Setuju	1	5	74%
4	Setuju	5	20	
3	Netral	4	12	
2	Tidak Setuju			
1	Sangat Tidak Setuju			
Total		10	37	

Dari hasil di atas maka dapat disimpulkan responden **setuju** bahwa aplikasi dapat memberikan informasi data secara akurat.

10. Apakah tampilan website menarik dan mudah dipahami (user friendly)?

**Tabel 4. 18 Hasil Pengujian Kuesioner Pertanyaan Nomor 10**

Bobot Soal	Keterangan	Responden	Skor	Hasil
5	Sangat Setuju	5	25	90%
4	Setuju	5	20	
3	Netral			
2	Tidak Setuju			
1	Sangat Tidak Setuju			
Total		10	45	

Dari hasil di atas maka dapat disimpulkan responden **sangat setuju** bahwa tampilan aplikasi menarik dan mudah dipahami.

11. Apakah dengan adanya aplikasi ini dapat menghasilkan laporan seperti yang dibutuhkan?

**Tabel 4. 19 Hasil Pengujian Kuesioner Pertanyaan Nomor 11**

Bobot Soal	Keterangan	Responden	Skor	Hasil
5	Sangat Setuju	3	15	86%
4	Setuju	7	28	
3	Netral			
2	Tidak Setuju			
1	Sangat Tidak Setuju			
Total		10	43	

Dari hasil di atas maka dapat disimpulkan responden **sangat setuju** bahwa aplikasi dapat menghasilkan laporan seperti yang di butuhkan.

12. Apakah dengan adanya aplikasi ini dapat memperoleh informasi yang dibutuhkan dengan tepat waktu?

**Tabel 4. 20 Hasil Pengujian Kuesioner Pertanyaan Nomor 12**

Bobot Soal	Keterangan	Responden	Skor	Hasil
5	Sangat Setuju	4	20	88%
4	Setuju	6	24	
3	Netral			
2	Tidak Setuju			
1	Sangat Tidak Setuju			
Total		7	44	

Dari hasil di atas maka dapat disimpulkan responden **sangat setuju** bahwa aplikasi dapat mempercepat proses perolehan informasi bagi pengguna.

13. Apakah aplikasi ini mampu memberikan informasi sesuai dengan format yang dibutuhkan?

**Tabel 4. 21 Hasil Pengujian Kuesioner Pertanyaan Nomor 13**

Bobot Soal	Keterangan	Responden	Skor	Hasil
5	Sangat Setuju	3	15	74%
4	Setuju	1	4	
3	Netral	6	18	
2	Tidak Setuju			
1	Sangat Tidak Setuju			
Total		10	37	

Dari hasil di atas maka dapat disimpulkan responden **setuju** bahwa aplikasi dapat memberikan informasi sesuai dengan format yang dibutuhkan.

14. Apakah aplikasi ini mampu menghasilkan informasi yang dapat dipahami secara jelas?

**Tabel 4. 22 Hasil Pengujian Kuesioner Pertanyaan Nomor 14**

Bobot Soal	Keterangan	Responden	Skor	Hasil
5	Sangat Setuju	3	15	78%
4	Setuju	3	12	
3	Netral	4	12	
2	Tidak Setuju			
1	Sangat Tidak Setuju			
Total		10	39	

Dari hasil di atas maka dapat disimpulkan responden **setuju** bahwa aplikasi dapat memberikan informasi yang dapat dipahami secara jelas.

15. Apakah aplikasi mudah dipelajari oleh orang yang baru pertama kali digunakan?

**Tabel 4. 23 Hasil Pengujian Kuesioner Pertanyaan Nomor 15**

Bobot Soal	Keterangan	Responden	Skor	Hasil
5	Sangat Setuju	6	30	92%
4	Setuju	4	16	
3	Netral			
2	Tidak Setuju			
1	Sangat Tidak Setuju			
Total		10	46	

Dari hasil di atas maka dapat disimpulkan responden **sangat setuju** bahwa aplikasi mudah dipelajari oleh orang yang baru pertama kali menggunakan.

#### 4.2.3. Pengujian Algoritma Boyer Moore

Algoritma *Boyer Moore* pada aplikasi *Chatbot* Klinik Amanatul Ummah di implementasikan pada proses pencarian penyakit dan penentuan tindakan sementara berdasarkan pengumpulan data masukan dari pengguna. Pengujian algoritma di lakukan dengan menilai keakuratan sistem dalam menampilkan jawaban dari inputan pengguna. Jumlah data yang di uji saat penelitian ini berlangsung bisa di lihat pada tabel berikut.

**Tabel 4. 24 Jumlah Data Yang Digunakan**

No	Nama Data	Jumlah Data
1	Data Gejala	106
2	Data Penyakit	26
3	Data Penyakit dan Gejala	26
4	Data Riwayat Chat	20
5	Data Sesi	
Total		178

**Tabel 4. 25 Data Nama Penyakit dan Gejala dari Pihak Klinik**

No	Nama Penyakit	Nama Gejala	Tindakan Sementara
1	Vertigo	pusing, kepala panas, demam, muntah, mual, berputar	Segera Menuju Klinik Di Dampingi Teman/Ustadzah (HISTIGO, DIMENHYDRINATE)
2	Diare Akut	mencret, mual, nyeri, darah, lendir, cair, diare	Wajib Tindakan Lanjut, Lansung Rawat Inap
3	Diare	mencret, diare, mual	Petugas Apoteker (Diapecta, Zinc, Antasida, Obat Mual)
4	Scabies	gatal, nanah, nyeri	Butuh Observasi Lanjutan Dari Dokter
5	Demam Typhoid	muntah, pusing, mual, demam, nyeri	Wajib Tindakan Lanjut, Langsung Rawat Inap
6	Demam	demam, meriang	Menjaga pola makan dan istirahat yang cukup, jika gejala sakit bertambah segera menuju klinik dengan membaya printout riwayat chat
7	Cantengan	bernanah, cantengan, canteng, nanah	Menuju klinik dengan print atau riwayat pesan ini
8	Mata Ikan	nanah, bernanah, mata ikan	menuju ke klinik kesehatan dan meminta perawatan medis
9	Sakit Gigi	gusi membengkak, gigi, sakit gigi, bengkak	segera periksa ke klinik kesehatan untuk diperiksa lebih lanjut
10	Mimisan	mimisan, darah	duduk tegak dan jangan berbaring, condongkan tubuh ke depan, encet hidung selama 10-15 detik, kompres pangkal hidung
11	Maag	maag, perut kembung	Makan porsi sedikit tapi sering, jangan langsung rebahan setelah makan, batasi konsumsi makanan pedas, asam, dan berlemak, konsumsi makanan yang mengandung probiotik, Minum teh herbal, Istirahat yang cukup, Kurangi kafein
12	Biduran	bentol, panas, gatal, biduran	jauhi faktor pemicu biduran, komprs dingin, jaga tubuh dari suasana dingin, gunakan bedak gatal untuk megurangi efek gatal

**Tabel 4. 26 Data Nama Penyakit dan Gejala dari Pihak Klinik (Lanjutan)**

No	Nama Penyakit	Nama Gejala	Tindakan Sementara
13	Alergi	alergi, bentol, gatal, bintik	observasi lanjut dan meminta perawatan dokter
14	Sariawan	sariawan, perih, bau mulut	larutan kumur, salep, dan obat-obatan, jika diperlukan
15	pilek/flu	flu, pilek, meriang	minum air putih yang banyak, hindari udara dingin
16	batuk kering	batuk kering, batuk	minum air hangat, hindari makanan berMSG untuk mengurangi efek iritasi di tenggorokan
17	batuk dahak	batuk, batuk berdahak	minum air hangat, hindari makanan berlemak berminyak, selalu pakai masker . buang dahak disalurkan air mengalir. lebih dari 3 hari segera ke poskestren/klinik
18	radang (faringitis, laringitis, tonsilitis)	radang	minum air hangat, hindari makanan berlemak berminyak, lebih dari 3 hari segera ke poskestren/klinik
19	Asma	sesak, asma	Jangan panik. Jika punya riwayat asma sebelumnya segera minum obat yang dipunya, jika tidak punya atau tidak berriwayat asma segera ke poskestren/klinik
20	konstipasi (susah BAB)	susah bab	minum air putih atau susu yang banyak, makan sayur sayuran dan atau buah buahan (semangka, melon,etc)
21	dermatitis	kulit gatal, kulit kering, kulit lecet, kulit melepuh, kulit menebal, kulit merah, kulit pecah pecah, kulit bersisik, bernanah, bengkak	jangan digaruk, segera ke poskestren/klinik
22	furunkel (bisul)	bengkak, bernanah, bisul	jangan melakukan tindakan operasi sembarangan, jangan sering disentuh. segera ke poskstren/klnik untuk tindakan lebih lanjut

**Tabel 4. 27 Data Nama Penyakit dan Gejala dari Pihak Klinik (Lanjutan)**

No	Nama Penyakit	Nama Gejala	Tindakan Sementara
23	hordeolum (bintitan)	bintik, bintilan, gatal	jangan sering dipegang, segera kompres air hangat. jika keluhan disertai pada saat sujud dan atau keluar kotoran mata lebih banyak dari biasanya segera ke poskestren atau klinik
24	otitis media (radang diaera telinga)	radang telinga	hentikan melakukan tindakan memasukkan kasa telinga (cotton bud). segera ke poskestren/klinik
25	Insomnia	susah tidur, sulit fokus, mudah lelah, mengantuk siang hari, insomnia, cepat lelah	Priksa lebih lanjut, menuju klinik untuk observasi lebih lanjut
26	Flu like symptoms (Masuk Angin)	sendawa, sakit kepala, pilek, perut kembung, nafsu makan menurun, muntah, mual, menggigil, lemas, keringat dingin, kembung, demam, buang gas, sering sendawa	Menjaga pola makan (Makan makanan dan minuman bergizi), jika gejala tidak kunjung mereda segera menuju klinik

Data yang di uji merupakan data penyakit dan gejala dengan jumlah 26 data, uji coba di lakukan sebanyak 26 kali percobaan dengan menggunakan masukkan dari user yang sering di gunakan dengan mereferensi dari arahan pihak klinik secara langsung. Perhitungan akurasi menggunakan perhitungan ROC (*Reciever Operating Characterestic*).

$$Akurasi = \frac{\Sigma True Positive (TP) + \Sigma True Negative (TN)}{\Sigma Total Population}$$

Seingga di peroleh hasil pengujian seperti pada table berikut:

**Tabel 4. 28 Hasil Uji Coba Proses Algoritma**

No	Input Gejala User	Jumlah Data	Hasil Pencarian yang benar (TP)	Data Seharusnya yang benar	Hasil Pencarian Salah	Data yang tidak terseleksi dengan benar (TN)	Waktu	Ket	Akurasi
1	Pusing, Berputar, Mual, Muntah	26	5	5	0	21	0,0051	Tepat	100%
2	Pusing, Panas, Demam	26	5	5	0	21	0,0052	Tepat	100%
3	Diare, Mual	26	1	1	0	25	0,0021	Tepat	100%
4	Diare, Lendir, Cair, Nyeri, Lemas	26	2	2	0	24	0,0053	Tepat	100%
5	Muntah, Pusing, Mual, Demam, Nyeri	26	5	5	0	21	0,0057	Tepat	100%
6	Demam, Meriang, Pusing	26	3	3	0	23	0,0027	Tepat	100%
7	Gatal, Bernanah, Luka, Gatal, Nyeri	26	2	2	0	24	0,0024	Tepat	100%
8	Cantengan, Bengkak, Bernanah	26	2	2	0	24	0,0021	Tepat	100%
9	Mata Ikan, Nyeri, Bengkak	26	3	2	1	23	0,0034	Tepat	100%
10	Sakit Gigi, Nyeri, Ngilu, Gigi Goyang	26	1	1	0	25	0,0049	Tepat	100%
11	Mimisan, Pusing	26	1	1	0	25	0,0021	Tepat	100%
12	Maag, Perut Kembung, Nyeri Perut	26	1	1	0	25	0,0022	Tepat	100%
13	Biduran, Gatal, Bentol Bentol	26	1	1	0	25	0,0055	Tepat	100%
14	Gatal, Alergi	26	2	2	0	24	0,0021	Tepat	100%
15	Sariawan, Bibir Perih, Mulut Bau	26	1	1	0	25	0,0057	Tepat	100%
16	Flu, Pilek, Ingus, meriang	26	2	2	0	24	0,005	Tepat	100%
17	Batuk Kering	26	3	3	0	23	0,0059	Tepat	100%
18	Batuk Berdahak	26	3	3	0	23	0,0049	Tepat	100%

**Tabel 4. 29 Hasil Uji Coba Proses Algoritma (Lanjutan)**

No	Input Gejala User	Jumlah Data	Hasil Pencarian yang benar (TP)	Data Seharusnya yang benar	Hasil Pencarian Salah	Data yang tidak terseleksi dengan benar (TN)	Waktu	Ket	Akurasi
18	Batuk Berdahak	26	3	3	0	23	0,0049	Tepat	100%
19	Batuk	26	3	3	0	23	0,0021	Tepat	100%
20	Radang Tenggorokan	26	1	1	0	25	0,0021	Tepat	100%
21	Asma, Sesak	26	1	1	0	25	0,0021	Tepat	100%
22	Sembelit, Susah BAB	26	1	1	0	25	0,0041	Tepat	100%
23	Bisul, Bernanah, Bengkak, Panas	26	2	2	0	24	0,0034	Tepat	100%
24	Kotoran Mata, Belekan, Gatal	26	2	2	0	24	0,0043	Tepat	100%
25	Radang Telinga, Panas	26	2	2	0	24	0,0045	Tepat	100%
26	Susah Tidur, Insomnia	26	1	1	0	25	0,0041	Tepat	100%
<b>Rata-Rata Hasil</b>							0,0037		<b>100%</b>

Berdasarkan hasil uji coba di dapatkan hasil akurasi 100% pada 26 kali percobaan menjalankan aplikasi. Rata-rata hasil pencarian memberikan *output* yang sesuai. karena algoritma ini bersifat *Match Case*. sehingga data akan mudah di temukan dari proses pencocokan dengan kata kunci yang tersedia di database, dengan kata lain data kata kunci pada database menjadi hal yang penting dalam menentukan keakuratan hasil uji coba. Waktu pencarian yang di butuhkan dalam proses pencarian jawaban mempunyai nilai rata-rata 0.0037 detik.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1. Kesimpulan

Dari hasil pembahasan penelitian yang telah dilakukan maka dapat di ambil kesimpulan sebagaimana berikut:

- a. Aplikasi *Chatbot* Pos Kesehatan Pesantren MBI Amanatul Ummah berjalan dengan baik dan sesuai kebutuhan pengguna dengan rata rata percobaan user (*survei* kuesioner) menggunakan skala *likert* sebesar 83.5% *responden* setuju bahwa sistem memudahkan user dalam proses observasi pasien dan mengolah data riwayat pasien.
- b. Penerapan Algoritma *boyer moore* pada Aplikasi *Chatbot* Pos Kesehatan Pesantren MBI Amanatul Ummah memberikan hasil akurasi sebesar 100% dari 26 kali percobaan dengan rata rata waktu 0.0037 detik dalam menentukan jawaban.

#### 5.2. Saran

Setelah proses dan penerapan penelitian di lakukan terdapat beberapa saran sebagai pertimbangan untuk penelitian selanjutnya:

- a. Menambahkan beberapa fitur lain pada Aplikasi termasuk menambahkan fitur rekam medis dan respon interaktif lain pada *Chatbot*.
- b. Untuk meningkatkan akurasi dan validasi hasil dapat di bandingkan dengan algoritma lain dan memperkaya sample database sebagai data acuan *Chatbot*

## Daftar Pustaka

- Agung, H., & Yogyakarta. (2016). Implementasi Boyer Moore Pada Aplikasi Pencarian Rumus Matematika Dan Fisika. *Jurnal Ilmiah Teknologi dan Informasi Terapan*, 74-85.
- Argakusumah, K. W., & Hansun, S. (2014). Implementasi Algoritma Boyer Moore Pada Aplikasi Kamus Kedokteran Berbasis Android. *Ultimatic*, 70-78.
- Borman, & Pratama. (2016). Penerapan String Matching Dengan Algoritma Boyer Moore Pada Aplikasi Font Italic Untuk Deteksi Kata Asing. *J. Teknoinfo*, 1-5.
- Budiman, H. (2017). Peran Teknologi informasi dan komunikasi dalam Pendidikan. *Jurnal Pendidikan islam. Vol 8 No 1 E-ISSN*, 2528-2476.
- Charras, C., & Lecroq, T. (2004). *Handbook of Exact String-Matching Algorithms*. London: King's College Publication.
- Daeli, M. m. (2017). *Perancangan aplikasi pencarian kata dengan kombinasi algoritma knuth morris pratt dan algoritma Boyer Moore*.
- Darmawan, R. I., Setianingrum, A. H., & Arini. (2018). Implementasi Algoritma Booyer Moore Pada Aplikasi Kamus Istilah Kebidanan Berbasis Web. *Jurnal Sistem Informasi*, 53-62.
- Dwi, a. (2018). Aplikasi Chatbot (milki bot) yang terntegrasi dengan web cms untuk customer service pada ukm minsu. *Jurnal cendikia Vol XVI E-ISSN*, 0216-9436.
- Fau, A., Mesran, & Ginting, G. L. (2017). Analisa Perbandingan Boyer Moore Dan Knuth Morris Pratt Dalam Pencarian Judul Buku Menerapkan Metode Perbandingan Eksponensial. *Jurnal Time: Tecnology Informatic & Computer System*, 12-22.
- Ginting, G. L. (2014). Penerapan Algoritma Boyer Moore Pada Aplikasi Pengajuan Judul Skripsi Berbasis Web. *Informasi dan Teknologi Ilmiah*, 123-132.
- Guzman, I. (2016). *Accenture Chatbot customer service*.
- Imam Sulisty, A. P. (2008). *Algoritma Boyer Moore Dalam Pencarian String*.
- Jon Orwant, J. (1999). *Matering Algorithms*. California: O'Reilly United State.

- Knuth, D. E., Morris, J. H., & Pratt, V. R. (1997). Fast Pattern Matching in String. *SIAM Journal Of Computing*, 6, 323-350.
- Kumara, G. H. (2009). *Visualisasi beberapa algoritma pencocokkan string dengan java*. Bandung: Jurnal jurusan Teknik informatika Fakultas Elektro Informatika Institute Teknologi Bandung.
- Maskur, M. (2016). Perancangan Chatbot pusat informasi mahasiswa menggunakan aiml sebagai virtual assistant berbasis web. *Jurnal kinetik*, 1-3.
- Minandar. (2010). *Aplikasi algoritmapencarian string boyer moor pada pencocokan DNA*. Bandung: Laboratorium Ilmu dan Rekayasa Komputasi. Departemen Teknik Informatika, Institut Teknologi Bandung.
- Nugraha, A. T. (2020). *Aplikasi pemesanan travel menggunakan Chatbot dengan machine learning*. Malang: Skripsi Universitas islam negeri maulana malik Ibrahim malang.
- Prabhakar, G. (2012). *Desight and analysis of Algorithms*. New Delhi: PHI Learning Private Limited.
- Rahmanita, e. (2014). Pencarian string menggunakan algoritma Boyer Moore pada dokumen. *Jurnal Ilmiah NERO*, 1-1.
- Renaldi, M., & Lidya, L. (2001). *Algoritma dan Pemrograman dalam Bahasa Pascal dan C*. Bandung: Penerbit Informatika.
- Sagita, V. (2013). Studi Perbandingan Implementasi algoritma Boyer Moore, turbo Boyer Moore, dan tuned Boyer Moore dalam pencarian string. *Jurnal ultimatics*, 04-01.
- Santoso, R. B. (2011). *Rancangan bangun prototype Chatbot customer service system berbasis web*. . Pekanbaru: Jurusan Informatika. Fakultas Sains dan Teknologi Universitas Islam Negri Sultan Syarif Kasim Riau.
- Susanto. (2014). Perancangan aplikasi kamus istilah latin hewan dan tumbuhan dengan menerapkan algoritma Boyer Moore berbasis metode mobile. *Jurnal Informasi dan teknologi ilmiah (INTI)* 4 (3), 88-90.
- Syauqi, A. F. (2014). *Implementasi Semantic Search untuk mesin pencarian pada ensiklopedia masjid bersejarah di indonesia*.
- Vijayarani, M. J. (2017). Preprocessing techniques for text mining-an Overview. *International journal of computer science& communication Network*, 5.

Zaman, B., & Winarko, E. (2011). Analisis Fitur Kalimat untuk Peringkasan Teks otomatis pada Bahasa Indonesia. *Indonesian Journal of Computing and Cybernetics Systems*, 5.

## LAMPIRAN 1

### KUESIONER PENELITIAN UJI COBA APLIKASI CHATBOT POSKESTREN MBI AMANATUL UMMAH

Nama :

No Identitas :

Status / Jabatan :

Berikan tanda centang (V) pada nilai yang di anggap sesuai

1. Sangat Tidak Setuju
2. Tidak Setuju
3. Netral
4. Setuju
5. Sangat Setuju

No	Pertanyaan	Nilai				
		1	2	3	4	5
1	Apakah informasi yang disediakan oleh aplikasi mudah dimengerti?					
2	Secara keseluruhan apakah penggunaan aplikasi ini memuaskan?					
3	Apakah aplikasi ini sesuai dengan kebutuhan?					
4	Apakah dapat dengan mudah menghindari kesalahan dalam menggunakan aplikasi?					
5	Apakah aplikasi bermanfaat bagi pengguna?					
6	Apakah tampilan menu dalam aplikasi mudah untuk dikenali?					
7	Apakah aplikasi mempunyai kemampuan dan fungsi sesuai yang diharapkan?					
8	Apakah dengan adanya aplikasi ini membantu pengelolaan data menjadi lebih mudah?					
9	Apakah dengan adanya aplikasi ini memberikan informasi data secara akurat?					
10	Apakah tampilan website cukup menarik dan mudah dipahami (user friendly)?					
11	Apakah dengan adanya aplikasi ini dapat menghasilkan laporan seperti yang dibutuhkan?					
12	Apakah dengan adanya aplikasi ini dapat memperoleh informasi yang dibutuhkan dengan tepat waktu?					
13	Apakah aplikasi ini mampu memberikan informasi sesuai dengan format yang dibutuhkan?					
14	Apakah aplikasi ini mampu menghasilkan informasi yang dapat dipahami secara jelas?					
15	Apakah aplikasi mudah dipelajari oleh orang yang baru pertama kali digunakan?					