

**SISTEM KLASIFIKASI PENYAKIT DENGAN GEJALA
DEMAM PADA ANAK MENGGUNAKAN METODE
*FUZZY K-NEAREST NEIGHBOR (FK-NN)***



Disusun Oleh:

N a m a : Rizky Karunia Putra

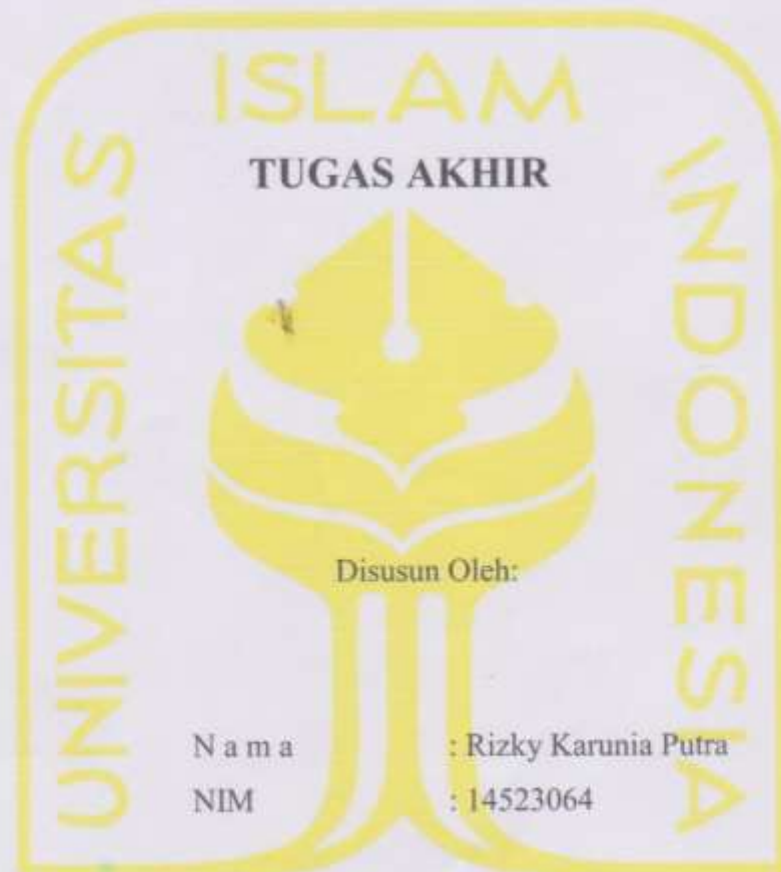
NIM : 14523064

**PROGRAM STUDI TEKNIK INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2018

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**SISTEM KLASIFIKASI PENYAKIT DENGAN GEJALA
DEMAM PADA ANAK MENGGUNAKAN METODE
*FUZZY K-NEAREST NEIGHBOR (FK-NN)***



Yogyakarta, 14 September 2018

Pembimbing,

(Sri Mulyati, S.Kom., M.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**SISTEM KLASIFIKASI PENYAKIT DENGAN GEJALA
DEMAM PADA ANAK MENGGUNAKAN METODE
FUZZY K-NEAREST NEIGHBOR (FK-NN)**

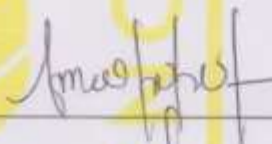
TUGAS AKHIR

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Teknik Informatika di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 03 Oktober 2018

Tim Penguji

Sri Mulyati, S.Kom., M.Kom.



Anggota 1

Elyza Gustri Wahyuni, S.T., M.Cs.



Anggota 2

Nur Wijyaning Rahayu, S.Kom., M.Cs.



Mengetahui,

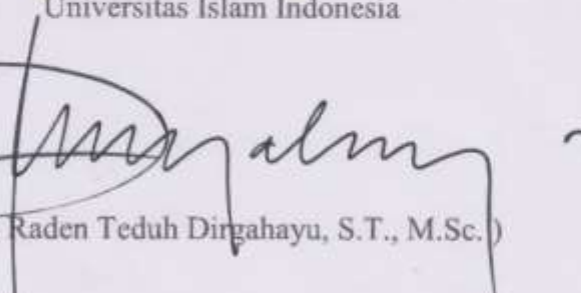
Ketua Program Studi Teknik Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)



HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Rizky Karunia Putra

NIM : 14523064

Tugas akhir dengan judul:

**SISTEM KLASIFIKASI PENYAKIT DENGAN GEJALA
DEMAM PADA ANAK MENGGUNAKAN METODE
*FUZZY K-NEAREST NEIGHBOR (FK-NN)***

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 14 September 2018



(Rizky Karunia Putra)

HALAMAN PERSEMBAHAN

Alhamdulillah Robbil 'Alamin puji syukur atas segala nikmat dan karunia yang Allah SWT berikan kepada saya sehingga tugas akhir ini dapat terselesaikan dengan baik. Atas semua dukungan dan bantuan yang telah diberikan, saya persembahkan tugas akhir ini untuk,

Kedua orang tua saya yang tercinta, Bapak Muh. Ridwan. Dan Ibu Mujinem ,
Kakakku, Muhammad Ilham Fitri Adzani, S.Stat,

Sahabat-sahabat saya, SFGH,
Seluruh teman-teman Magnifico 2014 yang menjadi teman seperjuangan di bangku perkuliahan,
Dan seluruh teman-teman yang tidak bisa disebutkan satu per satu.

HALAMAN MOTO

“Maka sesungguhnya beserta kesulitan itu ada kemudahan. Sesungguhnya beserta kesulitan itu ada kemudahan.”

(QS. Al-Insyirah : 5-6)

“Allah akan meninggikan orang-orang yang beriman di antara dan orang-orang yang diberi ilmu pengetahuan beberapa derajat.”

(QS. Al-Mujadalah : 11)

“Mencari Ilmu itu adalah wajib bagi setiap muslim laki-laki maupun muslim perempuan.”

(HR. Ibnu Abdil Barr)

“Barang siapa yang keluar untuk mencari ilmu maka ia berada di jalan Allah hingga ia pulang.”

(HR. Tirmidzi)

“Barang siapa yang menempuh jalan untuk mencari suatu ilmu, Niscaya Allah Memudahkannya ke jalan menuju surga.”

(HR. Muslim)

“Menuntut ilmu itu diwajibkan bagi setiap orang islam.”

(HR. Ibnu majah, Al-Baihaqi, Ibnu Abdil Barr, dan Ibnu Adi, dari Anas bin malik)

KATA PENGANTAR



Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah, penulis ucapkan puji syukur kepada Allah SWT yang telah memberikan rahmat, hidayah, serta karunia-Nya, sehingga laporan Tugas Akhir dapat penulis selesaikan. Tidak lupa shalawat serta salam kami ucapkan kepada junjungan nabi Allah Nabi Muhammad SAW, beserta para keluarga dan sahabatnya yang telah membawa kita dari zaman kegelapan menuju zaman terang benderang.

Tugas Akhir ini dibuat sebagai salah satu syarat yang harus dipenuhi untuk memperoleh gelar sarjana di Jurusan Teknik Informatika Universitas Islam Indonesia. Adapun Tugas Akhir kami mengenai “Sistem Klasifikasi Penyakit Dengan Gejala Demam Pada Anak Menggunakan Metode *Fuzzy K-Nearest Neighbor*.”

Pelaksanaan Tugas Akhir ini merupakan salah satu mata kuliah wajib dari jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia dan juga merupakan sarana bagi penulis untuk menambah wawasan serta pengalaman dalam menerapkan keilmuan, sesuai dengan yang dibambil di bangku perkuliahan.

Oleh, Karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada:

1. Orang tua dan keluarga penulis atas segala doa dan dukungan selama penulis melakuakn Tugas Akhir.
2. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku Ketua Program Studi Teknik Informatika – Program Sarjana Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Ibu Sri Mulyati, S.Kom., M.Kom., selaku Dosen Pembimbing 1 Tugas Akhir di Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Segenap keluarga besar teman-teman di Fakultas Teknologi Industri terutama dari Jurusan Teknik Informatika 2014 Universitas Islam Indonesia yang telah memberikan bantuan dan dukungannya.
5. Teman-teman KKN Unit 419 (Aida, Anes, Aidil, Aong, Arif, Fety, Nadia, dan Winda) yang telah memberikan bantuan, dukungan dan doanya.
6. Semua pihak yang telah banyak membantu kami dalam pelaksanaan Tugas Akhir yang tidak dapat kami sebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih belum sempurna, karena keterbatasan kemampuan dan pengalaan di lapangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun demi kesempurnaan Laporan Tugas Akhir ini. Akhir kata, penulis berharap agar laporan ini dapat bermanfaat bagi semua pihak.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Yogyakarta, 14 September 2018

(Rizky Karunia Putra)

SARI

Rekam medis merupakan berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain yang telah diberikan kepada pasien. Rekam medis yang dimiliki instansi Rumah Sakit Islam Banjarmasin banyak didapati keluhan pasien anak dengan gejala demam. Hal ini juga seiring dengan 8 jenis penyakit yang direkomendasikan pihak instansi yang sama-sama mempunyai gejala demam. Namun semakin banyaknya volume data rekam medis yang tersimpan, tidak sebanding dengan pemanfaatan terhadap rekam medis tersebut. Apabila tidak dimanfaatkan, rekam medis tersebut akan sia-sia dan hanya akan memenuhi kapasitas penyimpanan.

Untuk menjawab permasalahan tersebut, dibuatlah sistem klasifikasi penyakit dengan gejala demam pada anak yang mampu melakukan klasifikasi penyakit berdasarkan data rekam medis yang berbentuk teks sehingga mampu membantu tenaga medis khususnya dokter umum dan dokter koas dalam mendiagnosis penyakit. Dalam pembuatan sistem klasifikasi ini menggunakan metode *Fuzzy K-Nearest Neighbor* dimana metode ini digunakan untuk mencari jarak tetangga terdekat dan menghasilkan himpunan fuzzy yang berguna untuk menghilangkan bias. Sistem ini dibangun menggunakan bahasa pemrograman Python dan *framework* Django untuk menampilkan ke dalam bentuk web. Pengujian dilakukan dengan memasukkan data gejala penyakit pada pasien.

Hasil pengujian sistem klasifikasi menunjukkan akurasi 83.3% pada data penyakit demam berdarah dengue dan pneumonia dengan perbandingan training dan tes sebanyak 80 : 20, nilai K sebesar 10, dan nilai M sebesar 2. Hal tersebut dapat disimpulkan sistem klasifikasi menggunakan metode *Fuzzy K-Nearest Neighbor* dapat digunakan secara baik sebagai solusi untuk klasifikasi penyakit anak dengan gejala demam.

Kata kunci: *fuzzy k-nearest neighbor, klasifikasi penyakit anak dengan gejala demam, klasifikasi, rekam medis*

GLOSARIUM

<i>Use Case</i>	Sebuah cara yang menggambarkan kebutuhan fungsional dari sebuah sistem yang akan dibuat.
<i>Activity Diagram</i>	Diagram atau aktifitas yang berjalan pada sistem.
<i>User</i>	Pelaku atau subyek yang terlibat dalam penggunaan sistem.
<i>Tokenizing</i>	Proses pemisahan kalimat menjadi token/ <i>term</i> /kata.
Nilai Keanggotaan	Sebuah nilai pada kelas yang memiliki interval antara 0 sampai 1.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR.....	vii
SARI.....	ix
GLOSARIUM	x
DAFTAR ISI	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	2
1.6 Metode Penelitian	3
1.7 Sistematika Penulisan	3
BAB II LANDASAN TEORI	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori.....	8
2.2.1 Demam	8
2.2.2 <i>Machine Learning</i>	8
2.2.3 Klasifikasi.....	9
2.2.4 <i>Text Mining</i>	9
2.2.5 TF-IDF.....	10
2.2.6 Algoritma <i>Fuzzy K-Nearest Neighbor</i>	10
2.2.7 Django	11
2.2.8 Confusion Matrik	12
BAB III METODOLOGI	14

3.1	Pengumpulan Data	14
3.1.1	Studi Pustaka	14
3.1.2	Kajian Dokumen.....	14
3.2	Analisis Kebutuhan	15
3.2.1	Analisis kebutuhan fungsi	16
3.2.2	Analisis Input	16
3.2.3	Analisis Output.....	16
3.2.4	Analisis Kebutuhan Perangkat Lunak	16
3.2.5	Analisis Kebutuhan Perangkat Keras	17
3.3	Perancangan Sistem	17
3.3.1	<i>Use Case Diagram</i>	17
3.3.2	<i>Activity Diagram</i>	18
3.3.3	Perancangan Antarmuka.....	23
3.3.4	<i>Flowchart</i>	28
3.4	Implementasi	42
3.5	Pengujian Sistem.....	42
BAB IV HASIL DAN PEMBAHASAN.....		43
4.1	Implementasi	43
4.2	Pembahasan.....	47
4.2.1	Data Pengujian	47
4.2.2	Validasi Nilai.....	48
4.2.3	Pengujian Akurasi Sistem	56
4.3	Analisis Sistem.....	58
4.3.1	Kelebihan.....	58
4.3.2	Kekurangan	58
BAB V KESIMPULAN DAN SARAN.....		59
5.1	Kesimpulan	59
5.2	Saran.....	59
DAFTAR PUSTAKA.....		60
LAMPIRAN		62

DAFTAR TABEL

Tabel 2. 1 Review Penelitian Sejenis.....	6
Tabel 2. 2 Tabel Confusion Matrik.....	12
Tabel 2. 3 Skala AUC	13
Tabel 4. 1 Daftar Penyakit	47
Tabel 4. 2 Data Uji Diagnosis.....	48
Tabel 4. 3 Hasil Klasifikasi Data Testing	57
Tabel 4. 4 Confusion Matrik.....	57
Tabel 4. 5 Tabel Precision Dan Recall.....	57

DAFTAR GAMBAR

Gambar 2. 1 Teknik dalam <i>Machine Learning</i>	9
Gambar 3. 1 Foto rekam medis Rumah Sakit Islam Banjarmasin	15
Gambar 3. 2 <i>Use Case Diagram</i>	18
Gambar 3. 3 Perancangan <i>Activity Diagram</i> Login	19
Gambar 3. 4 Perancangan <i>Activity Diagram</i> Tambah Dokter	20
Gambar 3. 5 Perancangan <i>Activity Diagram</i> Training Data	21
Gambar 3. 6 Perancangan <i>Activity Diagram</i> Ganti Parameter	22
Gambar 3. 7 Perancangan <i>Activity Diagram</i> Klasifikasi	23
Gambar 3. 8 Rancangan Halaman Login	24
Gambar 3. 9 Rancangan Halaman Tambah Dokter	24
Gambar 3. 10 Rancangan Halaman Training Data	25
Gambar 3. 11 Rancangan Halaman Hasil Training	26
Gambar 3. 12 Rancangan Halaman Ganti Parameter	26
Gambar 3. 13 Rancangan Halaman Diagnosis	27
Gambar 3. 14 Rancangan Halaman Hasil Diagnosis	28
Gambar 3. 15 <i>Flowchart</i> Sistem Klasifikasi	29
Gambar 3. 16 <i>Flowchart</i> Membaca Dataset	30
Gambar 3. 17 <i>Flowchart Text Preprocessing</i>	31
Gambar 3. 18 <i>Flowchart Tokenization</i>	32
Gambar 3. 19 <i>Flowchart</i> TF-IDF	32
Gambar 3. 20 <i>Flowchart</i> Matrik TF	33
Gambar 3. 21 <i>Flowchart</i> IDF	34
Gambar 3. 22 <i>Flowchart</i> Matrik TF-IDF	35
Gambar 3. 23 <i>Flowchart</i> Klasifikasi FK-NN	36
Gambar 3. 24 <i>Flowchart</i> Hitung Jarak	37
Gambar 3. 25 <i>Flowchart</i> Hitung Voting Penyakit	38
Gambar 3. 26 <i>Flowchart</i> Hitung Nilai Keanggotaan	39
Gambar 3. 27 Hitung Nilai Bobot	40
Gambar 3. 28 <i>Flowchart</i> Hitung Himpunan Fuzzy	41
Gambar 3. 29 <i>Flowchart</i> Prediksi	42
Gambar 4. 1 Halaman Login	43

Gambar 4. 2 Halaman Tambah Dokter	44
Gambar 4. 3 Notifikasi Berhasil Buat Dokter Baru	44
Gambar 4. 4 Halaman Training Data	45
Gambar 4. 5 Halaman Hasil Training	45
Gambar 4. 6 Halaman Ganti Parameter	46
Gambar 4. 7 Halaman Diagnosis	46
Gambar 4. 8 Halaman Hasil Diagnosis	47
Gambar 4. 9 Kode Membaca Data.....	49
Gambar 4. 10 Hasil Membaca Data.....	49
Gambar 4. 11 Kode <i>Tokenization</i>	50
Gambar 4. 12 Hasil <i>Tokenization</i>	50
Gambar 4. 13 Kode Matrik TF	51
Gambar 4. 14 Hasil Matrik TF.....	51
Gambar 4. 15 Kode IDF.....	51
Gambar 4. 16 Hasil IDF.....	52
Gambar 4. 17 Kode Matrik TF-IDF.....	52
Gambar 4. 18 Hasil Matrik TF-IDF.....	52
Gambar 4. 19 Kode Perkalian TF-IDF	53
Gambar 4. 20 Hasil Perkalian TF-IDF.....	53
Gambar 4. 21 Kode Panjang Vektor	53
Gambar 4. 22 Hasil Panjang Vektor	53
Gambar 4. 23 Kode Menghitung Jarak	54
Gambar 4. 24 Hasil Menghitung Jarak	54
Gambar 4. 25 Kode Mencari Tetangga K.....	54
Gambar 4. 26 Hasil Mencari Tetangga K	54
Gambar 4. 27 Kode Menghitung Nilai Keanggotaan	55
Gambar 4. 28 Hasil Menghitung Nilai Keanggotaan.....	55
Gambar 4. 29 Kode Menghitung Bobot.....	55
Gambar 4. 30 Hasil Menghitung Bobot.....	55
Gambar 4. 31 Kode Menghitung Himpunan Fuzzy.....	56
Gambar 4. 32 Hasil Himpunan Fuzzy.....	56
Gambar 4. 33 Kode Prediksi.....	56
Gambar 4. 34 Hasil Prediksi	56

BAB I

PENDAHULUAN

1.1 Latar Belakang

Rumah Sakit Islam Banjarmasin merupakan rumah sakit yang beroperasi di wilayah Banjarmasin Kalimantan Selatan. Rumah sakit ini memiliki lima poliklinik yang beroperasi dan ditangani dokter spesialis yang handal yaitu anak, penyakit dalam, bedah, psikologi, dan gigi. Pasien yang paling banyak berobat di Rumah Sakit Islam Banjarmasin adalah anak-anak. Hal ini dikarenakan awal mula berdirinya Rumah Sakit Islam Banjarmasin pada tahun 1972 yang dikhususkan untuk ibu dan anak.

Sejak berdirinya Rumah Sakit Islam Banjarmasin, tenaga medis yang bekerja selalu menuliskan hasil pemeriksaan ke dalam rekam medis. Hal ini sejalan dengan peraturan menteri yang berbunyi bahwa setiap dokter wajib membuat rekam medis dalam menjalankan praktik kedokteran (Indonesia, 2008). Rekam medis sendiri merupakan berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain yang telah diberikan kepada pasien (Indonesia, 2008). Rekam medis yang tersimpan dalam ruangan rekam medis di Rumah Sakit Islam Banjarmasin ditulis dalam dokumen kertas yang bertipe teks.

Dari rekam medis yang terdapat di Rumah Sakit Islam Banjarmasin, didapati banyaknya keluhan pasien anak yang memiliki gejala demam. Hal tersebut juga seiring dengan delapan jenis penyakit yang telah direkomendasikan dari pihak rumah sakit yang semuanya mengandung gejala demam. Dengan volume data rekam medis yang semakin besar apabila tidak dimanfaatkan maka rekam medis menjadi sia-sia dan hanya akan memenuhi kapasitas penyimpanan. Maka dari itu muncul lah *Machine Learning* yang dapat bekerja, berfikir dan mengambil keputusan seperti layaknya para pakar (MathWorks, 2016).

Text Mining menjadi salah satu teknik *Machine Learning* dalam mengolah data rekam medis yang berbentuk teks. Didalam *Text Mining* terdapat teknik klasifikasi yaitu *Fuzzy K Nearest Neighbor* yang bekerja dengan mencari jarak terdekat dan menghasilkan himpunan fuzzy yang berfungsi untuk menghilangkan bias.

Atas dasar tersebut, maka diusulkanlah pembuatan sistem yang dapat mengklasifikasikan penyakit dengan gejala demam pada anak menggunakan metode *Fuzzy K Nearest Neighbor* dengan berdasarkan pengetahuan yang didapat dari dokumen rekam medis. Sistem ini dibangun dengan tujuan agar dapat menghasilkan diagnosis sementara dengan himpunan fuzzy yang

berguna untuk membantu dokter umum ataupun dokter yang sedang koas dalam menangani pasien anak apabila dokter spesialis berhalangan hadir. Dengan begitu proses penanganan terhadap pasien menjadi lebih cepat.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas maka dapat dirumuskan permasalahannya yaitu bagaimana membuat sistem yang dapat melakukan klasifikasi penyakit pada anak dengan gejala demam dari dokumen rekam medis berbentuk teks menggunakan metode *Fuzzy K-Nearest Neighbor* (FK-NN).

1.3 Batasan Masalah

Di dalam penelitian ini dilakukan suatu batasan pekerjaan, yaitu:

- A. Sistem ini hanya melakukan klasifikasi untuk penyakit dengan gejala utama demam pada anak-anak.
- B. Umur anak-anak berkisar dari antara umur 1 bulan sampai 18 tahun.
- C. *User* yang menggunakan sistem adalah admin dan dokter umum/koas/asisten.
- D. Sistem hanya menerima masukan dalam Bahasa Indonesia.
- E. Sistem hanya bisa menerima maksimal 3 pilihan penyakit.
- F. Isi dari atribut merupakan gabungan yang dibuat secara urut dari frekuensi nadi, suhu, frekuensi nafas, nyeri, berat badan, tinggi badan, dan riwayat penyakit positif yang dipisahkan dengan karakter “_”.
- G. Nilai K dan M bebas namun diberikan sebuah rekomendasi nilai K sebesar 10 dan M sebesar 2.

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dari penelitian ini adalah membantu dokter umum atau koas dalam melakukan diagnosis penyakit anak dengan cara membangun sebuah sistem yang dapat mengklasifikasikan penyakit pada anak yang diawali dengan gejala demam berdasarkan gejala-gejala yang diberikan oleh pasien dalam bentuk teks. Klasifikasi ini mampu untuk mengetahui diagnosa sementara dan nilai himpunan fuzzy dari penyakit.

1.5 Manfaat Penelitian

Manfaat yang dihasilkan oleh penelitian ini adalah berupa suatu sistem klasifikasi penyakit pada anak dengan gejala demam menggunakan metode *Fuzzy K-Nearest Neighbor* yang dapat

membantu dokter umum dan dokter koas dalam memberikan diagnosis sementara dari dokumen rekam medis secara cepat dan akurat.

1.6 Metode Penelitian

Metodologi yang digunakan dalam penelitian ini meliputi pengumpulan data, perancangan sistem, implementasi, dan pengujian sistem.

A. Pengumpulan Data

Metode ini dilakukan dengan cara mengumpulkan data rekam medis pasien yang telah terbukti secara klinis terdeteksi mengidap penyakit dengan gejala demam.

B. Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengetahui kebutuhan yang harus dimiliki dalam proses pengembangan sistem.

C. Perancangan Sistem

Perancangan dibuat untuk mengetahui masukan, proses dan keluaran dari sistem yang akan dibangun. Perancangan sistem meliputi pembuatan *flowchart*, *Activity Diagram*, *UseCase Diagram*, dan antarmuka sistem.

D. Implementasi

Tahapan implementasi merupakan tahap mengimplementasikan metode *Fuzzy K-Nearest Neighbor* ke dalam sistem menggunakan bahasa pemrograman Python dengan menggunakan *framework Django*

E. Pengujian Sistem

Tahapan ini dilakukan untuk mengetahui tingkat akurasi dari sistem yang dibangun.

1.7 Sistematika Penulisan

Sistematika penulisan penelitian ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Sistematika penelitian ini adalah sebagai berikut:

BAB I PENDAHULUAN

Pada bagian ini berisi tentang latar belakang permasalahan, identifikasi masalah, menentukan batasan masalah yang akan dibahas, tujuan dan manfaat dari penelitian, asumsi metodologi penelitian, dan sistematika penulisan

BAB II LANDASAN TEORI

Pada bagian ini menjelaskan teori-teori yang digunakan di dalam penelitian. Setiap teori yang digunakan di dalam penelitian akan dijabarkan di bagian ini.

BAB III METODOLOGI PENELITIAN

Pada bagian ini memuat uraian tentang langkah-langkah yang harus ditempuh untuk dalam melakukan penelitian.

BAB IV HASIL DAN PEMBAHASAN

Pada bagian ini membahas tentang implementasi sistem klasifikasi penyakit dengan gejala demam pada anak yang telah dibuat dan pengujian terhadap sistem.

BAB V KESIMPULAN DAN SARAN

Pada bagian ini akan dijabarkan kesimpulan yang didapatkan setelah menyelesaikan penelitian. Selain itu juga diberikan saran yang memberikan poin-poin yang dapat ditingkatkan untuk penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam penelitian ini diambil beberapa penelitian sejenis sebagai bahan perbandingan. Berikut adalah beberapa penelitian yang berkaitan dengan penelitian kali ini yang membahas dari klasifikasi penyakit dengan gejala demam dan teknik klasifikasi *Fuzzy K-Nearest Neighbor*.

Mulyati et al. (2012) dalam penelitiannya mengusulkan dan mencoba untuk membuat model keputusan untuk diagnosis penyakit anak dengan gejala demam. Penelitian ini menggunakan 10 jenis penyakit dan mendapatkan 205 sampel data yang diambil dari rekam medis. Basis pengetahuan dalam penelitian ini berbasis IF-THEN *rules* dengan mekanisme inferensi forward chaining dan Naïve Bayesian Classification yang digunakan untuk mendapatkan tingkat kepastian dengan pendekatan probabilitas.

Wafiyah et al. (2017) dalam penelitiannya mengusulkan dan mencoba untuk mengimplementasikan *Modified K-Nearest Neighbor* untuk klasifikasi penyakit demam. Penyakit yang digunakan dalam penelitian ini hanya tiga yaitu demam berdarah, tifoid, dan malaria. Data yang digunakan sebanyak 133 data pasien yang diambil dari Rumah Sakit Umum Daerah Selasih, Kota Riau. Penelitian ini menggunakan 15 gejala penyakit beserta bobot untuk masing masing gejala dalam beberapa tingkatan. Hasil akurasi yang didapat dalam penelitian ini bervariasi antara lain: pengujian pengaruh nilai K terhadap akurasi sebesar 88.55%, nilai rata-rata akurasi berdasarkan pengujian pengaruh variasi jumlah data latih sebesar 92.42%, pengujian pengaruh komposisi data latih terhadap akurasi sebesar 87.89%, dan pengujian pengaruh komposisi data latih dan data uji terhadap akurasi sebesar 96.35%.

Nugraha et al. (2017) dalam penelitiannya mengusulkan dan mencoba untuk mengimplementasikan *Fuzzy K-Nearest Neighbor* dalam menentukan status gizi balita. Data yang dipakai sebanyak 192 data dengan pembagian 160 untuk data latih dan 32 untuk data uji. Hasil akurasi yang dihasilkan dalam penelitian ini sebesar 84.37%. Penelitian juga membandingkan metode FK-NN dan K-NN yang menghasilkan kesimpulan bahwa FK-NN lebih baik dibandingkan dengan K-NN dengan rata-rata akurasi sebesar 82% dan 71%.

Keller et al. (1985) dalam penelitiannya mengusulkan pembuatan teori baru yang berjudul *Fuzzy K-Nearest Neighbor*. Penelitian ini memberikan 3 macam metode untuk memberikan nilai keanggotaan *fuzzy*. Kesimpulan yang penelitian ini berikan yaitu FK-NN ini dapat melawan *error rate* yang rendah, serta dapat bersaing baik dengan algoritma lainnya.

Siringoringo & Perangin-angin (2017) dalam penelitiannya mengusulkan untuk melakukan hibridisasi FK-NN dengan metode Modified Particle Swarm Optimization agar mendapatkan nilai K dan M yang terbaik. Penerapan MPSO dalam penelitian ini dapat mengeliminasi aspek subjektivitas dalam penentuan parameter K dan M. Hasil akurasi yang didapatkan dalam penelitian ini yaitu sebesar 84%. Penelitian ini juga membandingkan antara FK-NN biasa dengan FK-NN+MPSO yang menghasilkan perbandingan sebesar 81% dan 84%. Gambaran perbandingan penelitian diatas dapat dilihat pada Tabel 2. 1

Tabel 2. 1 Review Penelitian Sejenis

NO	Penelitian	Fokus Penelitian	Metode	Hasil Kesimpulan
1	(Mulyati et al., 2012)	Model keputusan untuk diagnosis penyakit anak dengan gejala demam	Naïve Bayesian Classification	Pembuatan model keputusan dengan menggunakan basis pengetahuan berbasis IF-THEN rules dapat diaplikasikan untuk diagnosis penyakit dengan gejala demam. Dengan penggabungan menggunakan Naïve Bayesian Classification dapat mendapatkan tingkat kepastian dengan pendekatan probabilitas.
2	(Wafiyah et al., 2017)	Klasifikasi penyakit demam	Modified K-Nearest Neighbor	Dengan menggunakan data sampel sebanyak 133 data dan 15 gejala, penelitian ini berhasil melakukan klasifikasi dengan tingkat akurasi yang bervariasi yaitu: 88.55% untuk pengujian pengaruh nilai K, 92.42% untuk pengujian pengaruh variasi jumlah data latih, 87.89% untuk pengujian pengaruh komposisi data latih, dan 96.35% untuk pengujian

				pengaruh komposisi data latih dan data uji.
3	(Nugraha et al., 2017)	Menentukan status gizi balita	Fuzzy K-Nearest Neighbor	Pada penelitian ini berhasil melakukan klasifikasi untuk menentukan status gizi balita dengan menggunakan 192 sampel data yang menghasilkan tingkat akurasi sebesar 84.37%.
4	(Keller et al., 1985)	Pembuatan teori FK-NN	Fuzzy K-Nearest Neighbor	Penelitian ini memberikan sebuah teori baru bernama FK-NN yang berguna untuk melawan error rate yang rendah, serta dapat bersaing baik dengan algoritma lainnya
5	(Siringoringo & Perangin-angin, 2017)	Hibridisasi FK-NN	Fuzzy K-Nearest Neighbor + Modified Particle Swarm Optimization	Pada penelitian ini menghasilkan tingkat akurasi sebesar 84% yang membuktikan bahwa penerapan MPSO dapat memberikan peningkatan performa klasifikasi. Hal ini dibuktikan dengan cara membandingkan FK-NN dengan FK-NN + MPSO yang mana hasil FK-NN + MPSO memberikan performa yang lebih unggul.

Berdasarkan hasil review penelitian di atas, dapat disimpulkan belum ada penelitian yang melakukan klasifikasi penyakit dengan gejala demam pada anak menggunakan *Fuzzy K Nearest Neighbor*. Dalam penelitian ini akan membuat sebuah sistem klasifikasi dengan menggunakan

Python dan *framework* Django serta mengambil data rekam medis dengan rentang tahun 2017 sampai 2018.

2.2 Dasar Teori

2.2.1 Demam

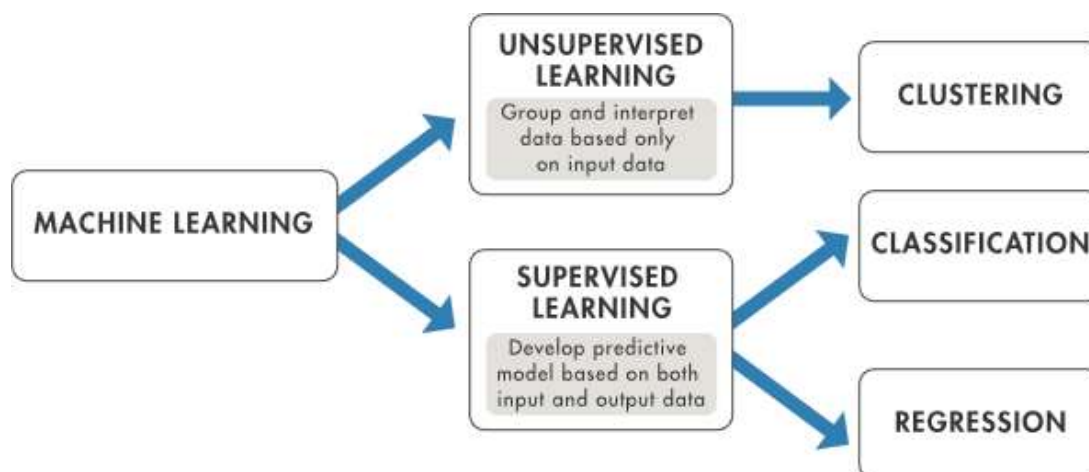
Demam atau dalam dunia klinis disebut dengan pireksia merupakan kondisi meningkatnya suhu tubuh. Secara klinis, demam adalah kondisi suhu tubuh yang meningkat 1 atau lebih derajat Celsius dari rata-rata suhu tubuh pada tempat pemeriksaan suhu tubuh dilakukan (El-Radhi et al., 2009). Berdasarkan penelitian yang telah dilakukan dari Billroth pada tahun 1868 hingga Gery dan Waksman, demam dapat didefinisikan sebagai keadaan suhu tubuh di atas normal akibat peningkatan pusat pengatur suhu di hipotalamus yang dipengaruhi oleh IL-1 (Soedarmo et al., 2010).

Demam diakibatkan oleh zat bernama pirogen. Terdapat dua pirogen yaitu pirogen eksogen dan endogen. Pirogen endogen yang berperan sebagai reseptor pada hipotalamus (Dall & Stanford, 1990). Interleukin-1 (IL-1) merupakan salah satu dari tiga jenis pirogen endogen. IL-1 memiliki fungsi primer untuk menginduksi demam pada hipotalamus (Soedarmo et al., 2010).

Termometer merupakan alat yang dapat digunakan untuk mengetahui suhu tubuh. Suhu tubuh dapat dilakukan pemeriksaan pada empat bagian yaitu mulut, ketiak, dubur, dan telinga. Masing-masing bagian tubuh tersebut memiliki perbedaan batas ambang untuk dapat dikatakan teridentifikasi demam. Pada mulut dan telinga, thermometer harus berada pada 37.6 derajat Celsius atau lebih untuk bisa dikatakan demam. Sedangkan apabila diperiksa pada ketiak, suhu tubuh sebesar 37.4 derajat Celsius sudah bisa dikatakan demam. Berbeda dengan pemeriksaan pada bagian dubur yang mencapai angka 38 derajat Celcius.

2.2.2 Machine Learning

Machine Learning memberikan sistem komputer kemampuan yang lebih dan baru. *Machine Learning* merupakan algoritma yang diberikan komputer untuk dapat belajar dari data dan mampu mengolah data tersebut untuk dapat memberikan prediksi dan keputusan. Dalam *Machine Learning* terdapat dua tipe yang dapat digunakan yaitu *Supervised Learning* dan *Unsupervised Learning*.



Gambar 2. 1 Teknik dalam *Machine Learning*

Sumber: MathWorks(2016)

Seperti yang dapat dilihat pada Gambar 2. 1 masing-masing tipe memiliki teknik tersendiri. *Supervised Learning* merupakan tipe algoritma yang menggunakan data latih untuk membuat prediksi (MathWorks, 2013a). *Supervised Learning* memiliki dua algoritma yaitu klasifikasi dan regresi. Berbeda dengan *Supervised Learning* yang mewajibkan memiliki data latih, *Unsupervised Learning* tidak memiliki data latih sehingga untuk menarik kesimpulan hanya menggunakan kumpulan dari data masukan (MathWorks, 2013b). *Unsupervised Learning* hanya memiliki satu algoritma untuk menarik kesimpulan yaitu klustering.

2.2.3 Klasifikasi

Klasifikasi adalah suatu proses pengelompokan data dengan didasarkan pada ciri-ciri tertentu ke dalam kelas-kelas yang telah ditentukan. Algoritma yang sering digunakan untuk mengklasifikasikan antara lain seperti: *Support Vector Machine*, *Naïve Baiyes*, *Artificial Neuron Network*, dan *K-Nearest Neighbor*.

Untuk dapat mendapatkan hasil klasifikasi, proses yang paling utama yaitu membuat sebuah model. Model berfungsi untuk menjawab pertanyaan yang diajukan. Model sendiri dibuat melalui proses latihan. Proses latihan merupakan proses untuk melatih data yang telah dikumpulkan. Data yang sebelumnya telah terkumpul dibagi dua dengan perbandingan lebih banyak data digunakan untuk data latih dan sisanya untuk data uji.

2.2.4 Text Mining

Text mining merupakan sebuah proses untuk menghasilkan informasi baru berdasarkan kumpulan data dari sumber tertulis yang secara khusus menggunakan perangkat lunak pada komputer (Prato, 2013). Sebelum suatu data teks dapat diproses, *Text Mining* memerlukan beberapa tahapan *Text Preprocessing* yang tujuannya untuk membuat teks menjadi lebih

terstruktur (Informatikalogi, 2017). Tahapan *Text Preprocessing* yang digunakan dalam penelitian ini *Tokenizing*.

2.2.5 TF-IDF

TF-IDF (*Term Frequency times Inverse Document Frequency*) merupakan sebuah teknik transformasi yang digunakan untuk menghitung bobot dari dokumen (Massaron & Mueller, 2015). Bobot ini digunakan dalam proses statistika untuk mengukur seberapa pentingnya kata dalam dokumen. TF-IDF ini sering dijumpai pada pembuatan mesin pencarian.

TF-IDF tersusun dari dua buah kata yaitu TF dan IDF. TF merupakan sebuah pengukuran berapa banyak kata muncul pada suatu dokumen. Sedangkan IDF merupakan pengukuran seberapa penting kah kata. IDF dapat dihitung menggunakan persamaan (2.1). Setelah nilai TF dan IDF telah didapatkan, maka untuk menghitung nilai TF-IDF dapat menggunakan persamaan (2.2).

$$IDF(t) = \log(\text{total dokumen} / \text{total kata } t \text{ pada semua dokumen}) \quad (2.1)$$

$$TFIDF(t) = TF(t) \cdot IDF(t) \quad (2.2)$$

2.2.6 Algoritma Fuzzy K-Nearest Neighbor

K-Nearest Neighbor merupakan salah satu algoritma klasifikasi yang paling sering digunakan. Algoritma ini melakukan klasifikasi berdasarkan jarak kedekatan antara data yang akan dievaluasi dengan K tetangga dalam data latih. Perhitungan jarak kedekatan antara data terdapat berbagai macam cara salah satunya adalah *Cosine Similarity*. *Cosine Similarity* sangat cocok digunakan untuk data yang menggunakan teks. Perhitungan *Cosine Similarity* dapat menggunakan rumus pada persamaan (2.3).

$$\cos(\Theta_{ij}) = \frac{\sum_k d_{ik} d_{jk}}{\sqrt{\sum_k d_{ik}^2} \sqrt{\sum_k d_{jk}^2}} \quad (2.3)$$

Dimana:

d_{ik} = merupakan bobot kata pada data ke- i

d_{jk} = merupakan bobot kata pada data ke- j

Cosine Similarity memerlukan nilai bobot per kata dari sebuah teks pada data. *Cosine Similarity* dapat dikombinasikan dengan TF-IDF untuk mendapatkan bobot. TF-IDF merupakan kombinasi dari *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) untuk pemberian bobot *term* dengan menyertakan frekuensi kemunculan *term* pada dokumen.

Fuzzy K-Nearest Neighbor merupakan gabungan dari dua algoritma yaitu *Fuzzy* dan *K-Nearest Neighbor*. Konsep dari algoritma ini adalah dengan menambahkan operasi *Fuzzy* untuk menentukan kelas yang akan ditempati oleh data uji. Dengan adanya penambahan ini, hubungan antara data dengan kelas menjadi tidak kaku karena setiap kelas memiliki hubungan keanggotaan atau *membership* dengan tingkatan tertentu (Siringoringo & Perangin-angin, 2017). Algoritma dari *Fuzzy K-Nearest Neighbor* adalah sebagai berikut (Keller et al., 1985):

1. Tentukan parameter K
2. Hitung jarak antara data yang akan dievaluasi dengan semua data pelatihan.
3. Urutkan jarak yang terbentuk (urut naik) dan tentukan jarak terdekat sampai urutan ke-K.
4. Hitung U_i (pemberian nilai keanggotaan semua kelas pada x) menggunakan rumus pada persamaan (2.4).

$$U_i(x) = \frac{\sum_{j=1}^k U_{ij} \left(1/\|x - x_j\|^{2/(m-1)}\right)}{\sum_{j=1}^k \left(1/\|x - x_j\|^{2/(m-1)}\right)} \quad (2.4)$$

Dimana:

U_{ij} = Nilai keanggotaan *fuzzy*

k = Nilai tetangga terdekat

j = variable data keanggotaan data uji

m = kekuatan *fuzzy* yang besarnya $m > 1$

Untuk menghitung nilai keanggotaan pada *Fuzzy K-NN*, terlebih dahulu dilakukan proses perhitungan nilai keanggotaan menggunakan persamaan berikut (Nugraha et al., 2017):

$$u_{ij} = \begin{cases} 0.51 + \left(\frac{n_j}{n}\right) * 0.49, & \text{jika } j = 1 \\ \left(\frac{n_j}{n}\right) * 0.49, & \text{jika } j \neq 1 \end{cases} \quad (2.5)$$

Dimana:

n_j = Jumlah anggota kelas j pada suatu data latih n

n = Jumlah data latih yang digunakan

j = Kelas data

2.2.7 Django

Django adalah sebuah *framework* untuk *high-level language* yaitu *Python* yang mendukung pembuatan dan pengembangan website secara cepat (Mozilla, 2018). Django merupakan web *framework* yang dikeluarkan sejak tahun 2005 sebagai *framework* yang gratis dan *open-source*.

Seperti *framework* pada umumnya, Django juga memiliki pola MVC (Model-View-Controller) dalam proses pengembangan sebuah website. Namun ada yang berbeda dari *framework* Django ini yaitu istilah MVC berubah menjadi MTV (Model-Template-View). *Model* merupakan tempat untuk pengolahan data, fungsi-fungsi yang ada pada sistem, dan *interface* dengan basis data. *Template* adalah bagian dalam *framework* yang tugasnya untuk menampilkan sesuatu kepada *User*. *View* adalah jembatan antara *model* dan *template* yang menangani logika proses dalam menerima respon dan permintaan sistem.

2.2.8 Confusion Matrik

Confusion Matrik merupakan sebuah tabel yang digunakan untuk mendeskripsikan performa dari model klasifikasi (Markhan, 2014). Tabel dari Confusion Matrik terdiri dari N baris dan kolom. Confusion Matrik berisi nilai prediksi benar dan salah yang dibandingkan dengan data testing yang sebenarnya. Contoh tabel dari Confusion Matrik dapat dilihat pada Tabel 2. 2

Tabel 2. 2 Tabel Confusion Matrik

	<i>Predicted 1</i>	<i>Predicted 2</i>
<i>Actual 1</i>	<i>True Positive</i>	<i>False Positive</i>
<i>Actual 2</i>	<i>False Negative</i>	<i>True Negative</i>
Total	P	N

Confusion Matrik sangat berguna untuk mengukur tingkat Recall, Precision, dan Akurasi. Precision dapat dihitung dengan menggunakan persamaan (2.6). Recall dapat dihitung dengan persamaan (2.7). Sedangkan Akurasi dapat dihitung menggunakan persamaan (2.8).

$$Precision = \frac{TP}{TP + FP} \quad (2.6)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.7)$$

$$Akurasi = \frac{TP + FN}{P + N} \quad (2.8)$$

2.2.9 Area Under Curve

AUC merupakan sebuah model perbandingan dari suatu model klasifikasi. AUC memiliki nilai di rentang antara nilai 0 sampai 1 (Gorunescu, 2011). AUC mengukur tingkat akurasi sistem kedalam tabel AUC. Semakin besar nilai AUC menunjukkan performa dari klasifikasi. Tabel AUC dapat dilihat pada Tabel 2. 3.

Tabel 2. 3 Skala AUC

0.5 – 0.6	Gagal
0.6 – 0.7	Buruk
0.7 – 0.8	Cukup
0.8 – 0.9	Baik
0.9 – 1.0	Sangat Baik

BAB III

METODOLOGI

3.1 Pengumpulan Data

Data yang digunakan dalam tugas akhir ini merupakan jenis data sekunder yang didapatkan dari Rumah Sakit Islam Banjarmasin. Pada pengumpulan data tugas akhir ini dilakukan beberapa tahapan untuk mendapatkan informasi yang diperlukan guna mendukung pembuatan sistem. Tahapan yang dilakukan terdiri dari studi pustaka dan kajian dokumen. Penjelasan dari setiap tahapan yang dilakukan dapat dilihat di bawah ini:

3.1.1 Studi Pustaka

Dilakukan dengan cara mengumpulkan data dari informasi yang terdapat di artikel, jurnal, buku, dan penelitian terdahulu sebagai referensi dan pedoman dalam pengembangan sistem serta penulisan laporan.

3.1.2 Kajian Dokumen

Kajian dokumen merupakan metode yang digunakan untuk menelusuri data historis (Bungin, 2007). Dokumen yang digunakan pada penelitian ini adalah dokumen rekam medis Rumah Sakit Islam Banjarmasin. Dokumen rekam medis yang didapatkan berbentuk teks yang ditulis oleh perawat Rumah Sakit Islam Banjarmasin. Didalam dokumen rekam medis terdapat dua form yang digunakan yaitu form formulir triase pasien instalasi gawat darurat dan form pengantar pasien rawat inap.

Dari proses kajian dokumen tersebut, terdapat dokumentasi berupa foto dari contoh rekam medis yang dapat dilihat pada Gambar 3. 1.

RUMAH SAKIT ISLAM
BANJARMASIN No. RM

2 7 53 10


**PENGANTAR PASIEN
RAWAT INAP**


NAMA: WAL/SUAMI/ISTERI
ALAMAT: dl. getah sekeloa

DIKIRIM KE UNIT: 08/09/2023 Jam • Ditulis oleh dokter poliklinik/dokter jaga bila penderita harus rawat inap di rumah sakit
surat pengantar dan dokter pengirim disertakan

Tanda Vital	Antropometri	Fungsional
1. Tekanan Darah : <u>110</u> mmHg 2. Frekuensi Nadi : <u>110</u> x/menit 3. Suhu : <u>37,5</u> °C 4. Frekuensi Nafas : <u>21</u> x/menit 5. Skor Nyeri : <input checked="" type="checkbox"/> Nyeri <input type="checkbox"/> Tidak	1. Berat Badan : <u>70</u> g/kg 2. Tinggi Badan : <u>174</u> m/cm 3. IMT (BB/TB) : <u>23</u> kg/M2	1. Alat Bantu : 2. Profesi : 3. Cacat Tubuh : 4. ADL : <input type="checkbox"/> Mandiri <input type="checkbox"/> Dibantu 5. Resiko Jatuh : <input type="checkbox"/> Ringan <input type="checkbox"/> Sedang <input type="checkbox"/> Berat
Khusus Pediatri 4. Lingkar Kepala :		Score Di isi oleh perawat : <u>7</u>

Berisikan catatan singkat penderita meliputi :

- Riwayat penyakit yang positif : pasien anak dengan keluhan demam 1 hari, nyeri perut, pusing, mual muntah (-) - Bakt (+) .
Riwayat opname .
- Pemeriksaan jasmani : Ku: lemah
tercek.
leata: ca-/- thorax: vesikuler +/+
abd: dupel, hu (+) hst (+) 
ekst: akral hangat
- Laboratorium :
- Diagnosa kerja : Obs. febris hr. I + Abdominal pain .
- Usul pengobatan di bangsal perawatan :
- Pengobatan yang sudah diberikan di poliklinik / emergency / kamar bersalin :
mp. RL 15 tpm .
inj. Ekanud 200 ug / jam . jam 22.37


dr. lea
Tanda tangan & nama lengkap

dr. H. Edy Hartono, Sp. A (K).

Gambar 3. 1 Foto rekam medis Rumah Sakit Islam Banjarmasin

3.2 Analisis Kebutuhan

Pada analisis kebutuhan ini dilakukan beberapa tahapan yang terdiri dari pengumpulan data, analisis fungsionalitas, analisis input, analisis output, analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras. Adapun penjelasan mengenai tahapan-tahapan yang ada dapat dilihat pada penjelasan di bawah ini:

3.2.1 Analisis kebutuhan fungsi

Analisis kebutuhan fungsi merupakan tahapan yang menjelaskan tentang fungsi yang dapat dilakukan oleh sistem sehingga mampu menjawab rumusan masalah. Adapun kebutuhan fungsi sistem ini sebagai berikut:

- a. Sistem dapat melakukan pembelajaran data.
- b. Sistem dapat menambahkan data dokter.
- c. Sistem dapat memberikan hasil akurasi dari model yang telah dibuat.
- d. Sistem dapat mengganti parameter nilai K dan M .
- e. Sistem dapat melakukan diagnosis penyakit.
- f. Sistem dapat memberikan hasil diagnosis penyakit.

3.2.2 Analisis Input

Data *input* merupakan data yang digunakan sebagai masukan pada Sistem Klasifikasi Penyakit Anak Dengan Gejala Demam Pada Anak Menggunakan Metode *Fuzzy K-Nearest Neighbor* yang dibangun. Data yang digunakan antara lain:

- a. Data *inputan user* adalah nama dan password yang digunakan untuk login.
- b. Pada admin, data *inputan* berupa frekuensi nadi, suhu, frekuensi nafas, nyeri, berat badan, tinggi badan, riwayat penyakit positif, nilai K , nilai M , dan nilai pembagian data.
- c. Pada admin, data *inputan* menambahkan dokter adalah nama, email, dan password.
- d. Pada admin, data *inputan* mengganti parameter adalah nilai K dan nilai M .
- e. Pada dokter, data *inputan* berupa frekuensi nadi, suhu, frekuensi nafas, nyeri, berat badan, tinggi badan, dan riwayat penyakit positif.

3.2.3 Analisis Output

Data *output* merupakan data hasil keluaran dari Sistem Klasifikasi Penyakit Anak Dengan Gejala Demam Pada Anak Menggunakan Metode *Fuzzy K-Nearest Neighbor* yang dibangun. Adapun *output* dari sistem adalah:

- a. Pada admin, terdapat *output* notifikasi berhasil menambahkan dokter, tabel klasifikasi, grafik donat dari akurasi, precision, dan recall.
- b. Pada dokter, terdapat *output* tabel diagnosis.

3.2.4 Analisis Kebutuhan Perangkat Lunak

Penelitian ini menggunakan beberapa perangkat lunak sebagai pendukung pengembangan Sistem Klasifikasi Penyakit Anak Dengan Gejala Demam Pada Anak Menggunakan Metode *Fuzzy K-Nearest Neighbor*. Perangkat lunak yang digunakan sebagai berikut:

- a. Windows 10 Pro.
- b. Python 3.5.2.

- c. Anaconda 4.5.2.
- d. Django 2.0.7.
- e. Bootstrap 3.3.7.
- f. JQuery 3.3.1.
- g. Chart.js 2.7.2.
- h. Sklearn 0.19.2.
- i. Sublime Text Build 3103.
- j. Google Chrome 69.0.3497.81.
- k. Inkscape 0.91.

3.2.5 Analisis Kebutuhan Perangkat Keras

Penelitian ini menggunakan beberapa perangkat keras sebagai pendukung pengembangan Sistem Klasifikasi Penyakit Anak Dengan Gejala Demam Pada Anak Menggunakan Metode *Fuzzy K-Nearest Neighbor*. Perangkat lunak yang digunakan sebagai berikut:

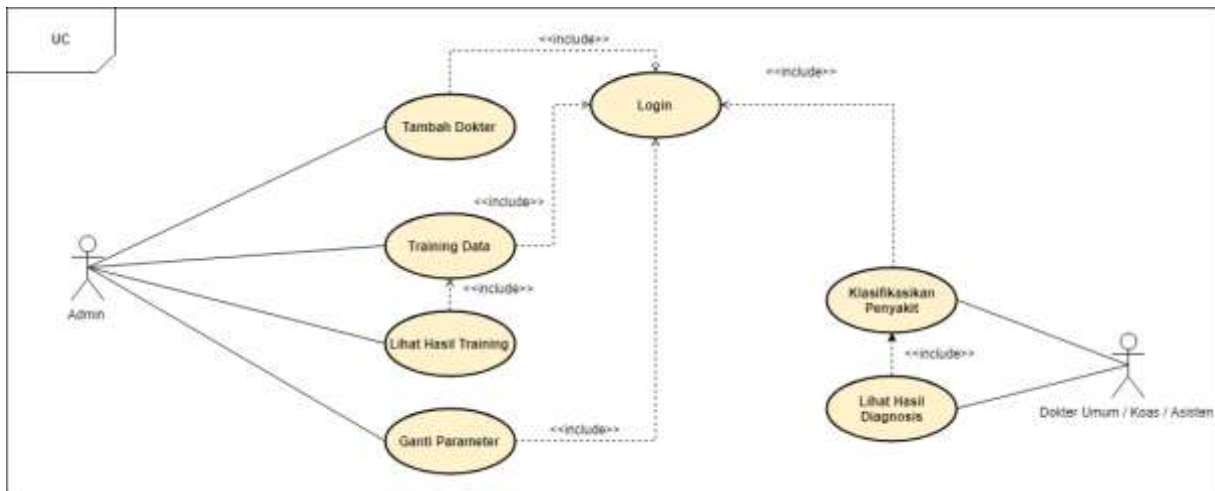
- a. Lenovo G405s
- b. RAM 4 GB
- c. Processor AMD A8-5550M

3.3 Perancangan Sistem

Pada perancangan sistem ini dilakukan beberapa langkah yang terdiri dari *Use Case diagram*, *Activity diagram*, rancangan antarmuka, dan *Flowchart*. Adapun penjelasan mengenai tahapan tersebut dapat dilihat pada penjelasan di bawah ini:

3.3.1 Use Case Diagram

Use Case Diagram adalah sebuah gambaran diagram dari fungsional yang dapat dilakukan sistem berdasarkan sudut pandang *User*. Semua proses fungsional dipresentasikan dengan urutan alur yang sederhana. Berikut ini merupakan rancangan *Use Case Diagram* yang dapat dilihat pada Gambar 3. 2.



Gambar 3. 2 Use Case Diagram

Untuk penjelasan dari hak akses *user* (admin, dokter umum / koas / asisten) terhadap proses didalam sistem dapat dilihat pada Tabel 3. 1.

Tabel 3. 1 Tabel Hak Akses User

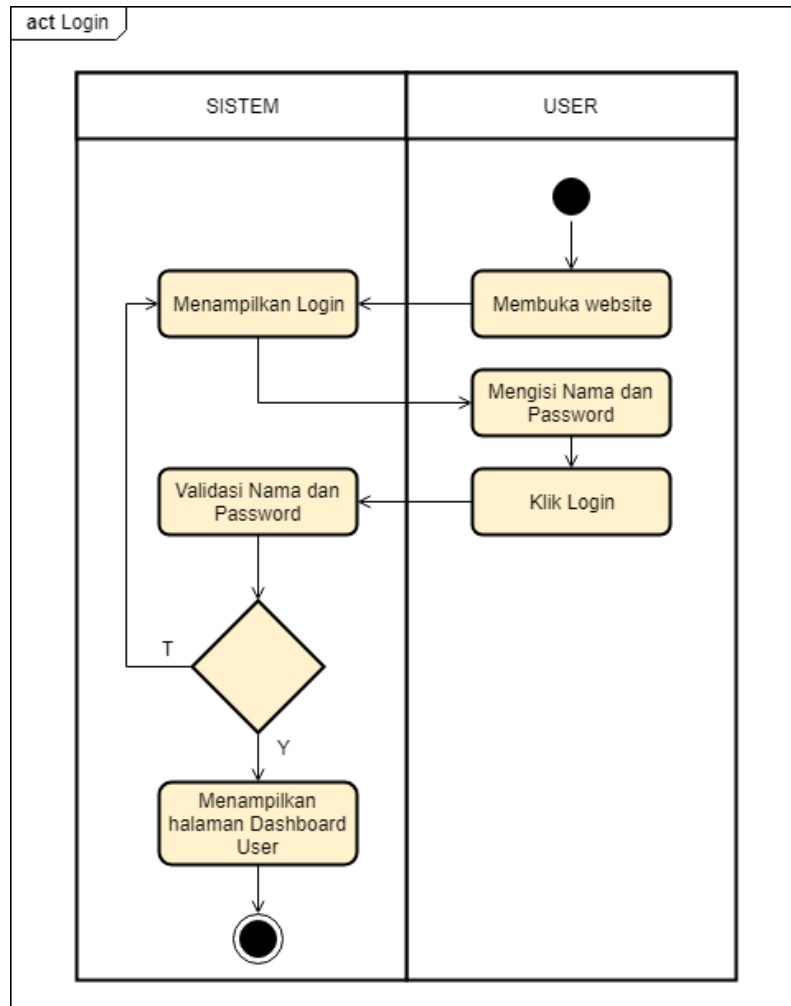
TABEL HAK AKSES			
No.	Nama User	Proses	Hak Akses
1.	Admin	Login	Isi Nama dan Password
		Tambah Dokter	Isi data dokter (Nama, Email, Password)
		Training Data	Isi jumlah pembagian data, nilai K, nilai M, pilihan penyakit
		Lihat Hasil Training	Lihat hasil klasifikasi dan akurasi
		Ganti Parameter	Ganti nilai K dan M
2.	Dokter Umum / Koas / Asisten	Login	Isi Nama dan Password
		Klasifikasi Penyakit	Isi data gejala pasien dan 2 – 3 kemungkinan penyakit
		Lihat Hasil Diagnosis	Lihat hasil diagnosis penyakit

3.3.2 Activity Diagram

A. Activity Diagram Login

Activity Diagram Login menjelaskan tentang aktivitas proses login *user* kedalam sistem. Tahapan Login dimulai ketika *user* membuka alamat website, maka sistem akan menampilkan halaman login yang didalamnya terdapat form login yang berisi *inputan* nama dan password. Kemudian *user* mengisi nama dan password lalu klik login. Sistem akan melakukan validasi

terhadap *inputan user*. Jika nama yang *diinputkan user* adalah admin, maka sistem akan mengarahkan *user* admin ke halaman dashboard admin. Sedangkan jika nama yang *diinputkan user* terdapat dalam data dokter, maka sistem akan mengarahkan *user* dokter ke halaman dashboard dokter. Penjelasan dari *activity diagram* Login dapat dilihat pada Gambar 3. 3.

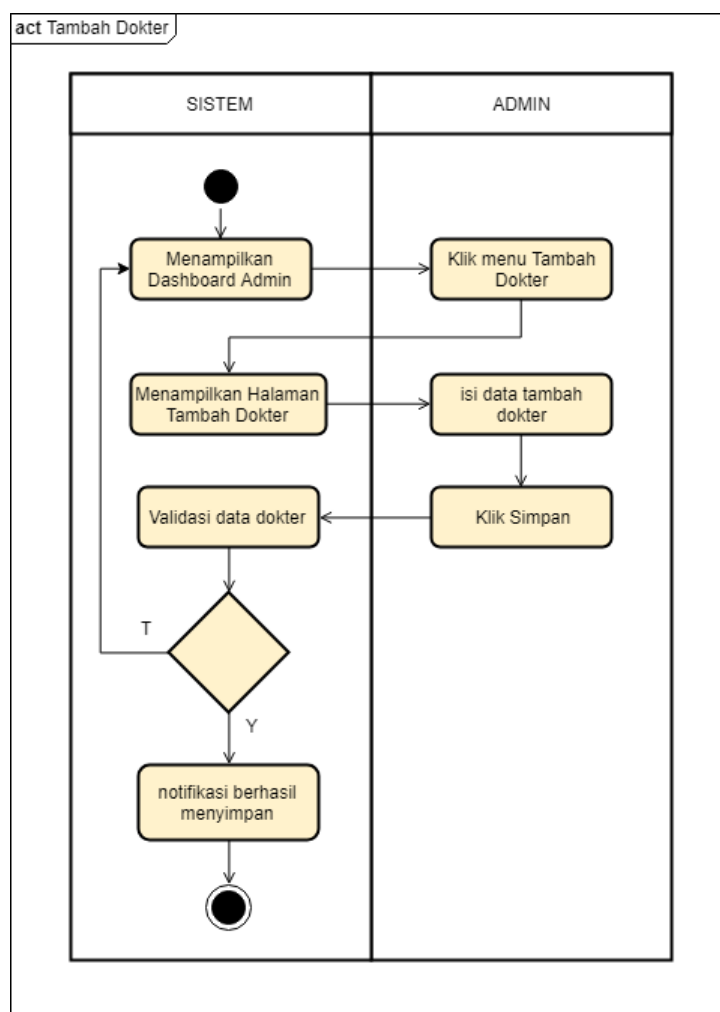


Gambar 3. 3 Perancangan *Activity Diagram* Login

B. *Activity Diagram* Tambah Dokter

Activity Diagram Tambah Dokter menjelaskan tentang aktivitas proses menambahkan *user* dokter baru kedalam sistem. Tahapan tambah dokter dimulai setelah *user* admin login, maka sistem akan menampilkan halaman dashboard admin. Kemudian admin memilih menu tambah dokter, maka sistem akan menampilkan halaman tambah dokter. Pada halaman tambah dokter ini, admin dapat mengisi nama dokter, email, dan password pada form yang telah disediakan. Kemudian admin memilih tombol simpan untuk membuat *user* dokter baru. Sistem akan melakukan validasi untuk mengecek apakah sudah ada atau belum. Jika belum ada nama dokter yang dimasukkan, maka sistem akan menyimpan *user* dokter yang dibuat dan memberikan

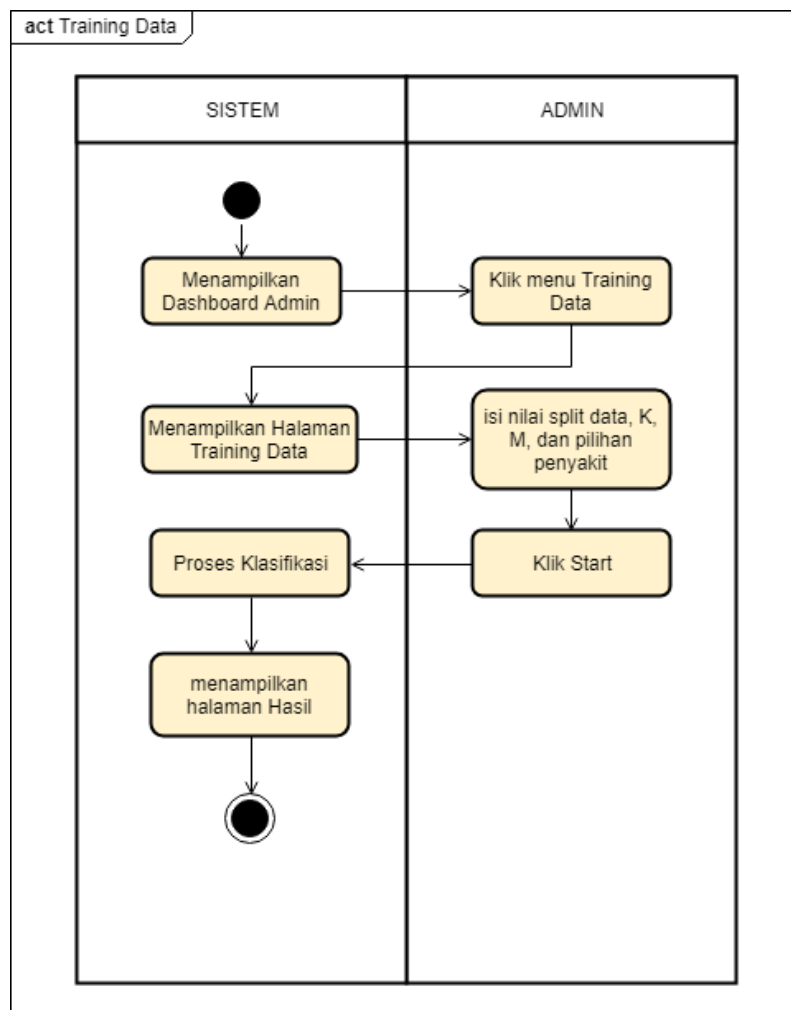
notifikasi berhasil menyimpan. Penjelasan dari *activity diagram* Tambah Dokter dapat dilihat pada Gambar 3. 4.



Gambar 3. 4 Perancangan *Activity Diagram* Tambah Dokter

C. *Activity Diagram* Training Data

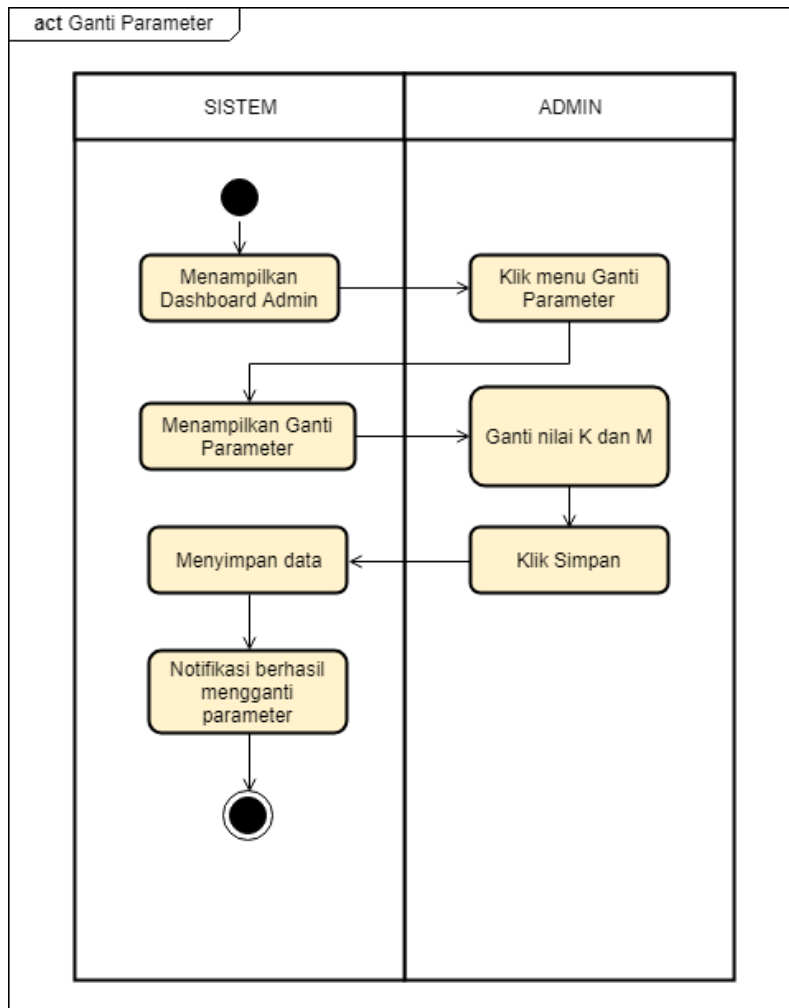
Activity Diagram Training Data menjelaskan tentang aktivitas proses admin dalam melakukan pembelajaran terhadap sistem guna mendapatkan hasil akurasi yang baik. Tahapan training data dimulai setelah *user* admin login, maka sistem akan menampilkan halaman dashboard admin. Kemudian admin memilih menu training data, maka sistem akan menampilkan halaman training data. Pada halaman training data ini, admin dapat mengisi nilai pembagian data, pilihan penyakit, nilai K, dan nilai M pada *form* yang telah disediakan. Kemudian admin memilih tombol start untuk memulai proses klasifikasi. Sistem akan menjalankan proses klasifikasi dan menampilkan hasil klasifikasi, akurasi, precision, dan recall setelah selesai proses klasifikasi. Penjelasan dari *activity diagram* Training Data dapat dilihat pada Gambar 3. 5.



Gambar 3. 5 Perancangan *Activity Diagram* Training Data

D. *Activity Diagram* Ganti Parameter

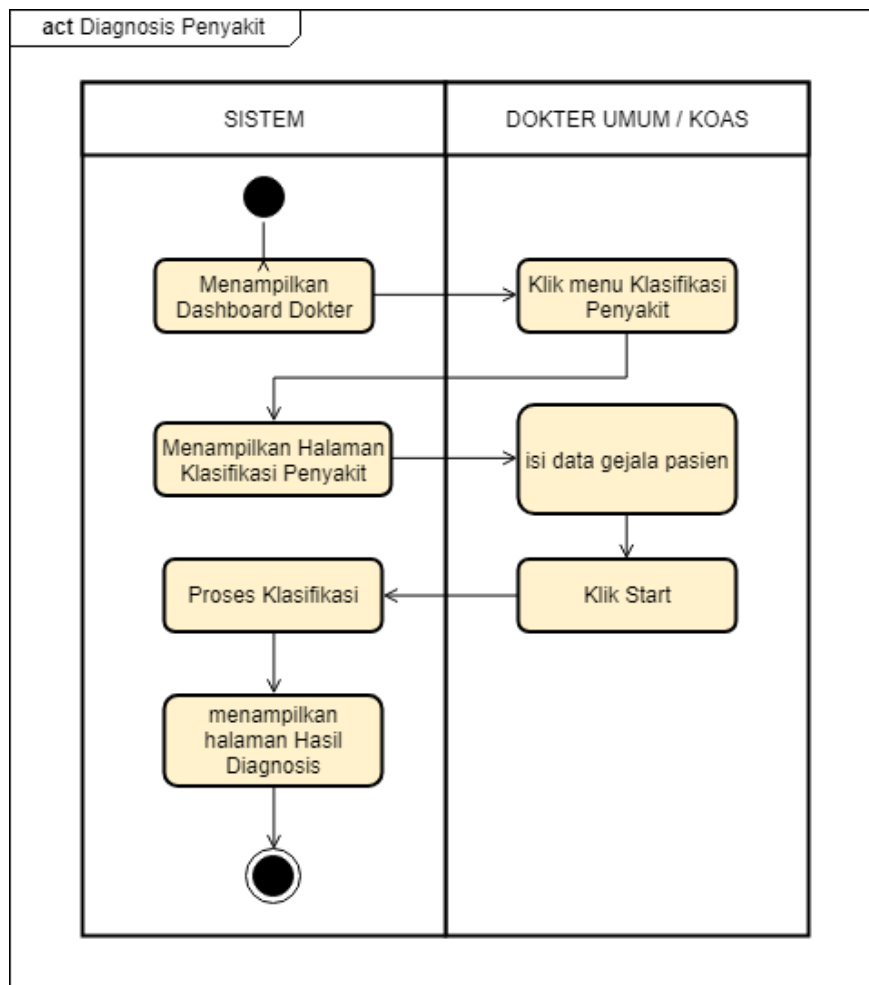
Activity Diagram Ganti Parameter menjelaskan tentang aktivitas proses admin dalam mengganti parameter K dan M pada model klasifikasi. Tahapan ganti parameter dimulai setelah *user* admin login, maka sistem akan menampilkan halaman dashboard admin. Kemudian admin memilih menu ganti parameter, maka sistem akan menampilkan halaman ganti parameter. Pada halaman training data ini, admin dapat mengisi nilai K dan M pada *form* yang telah disediakan. Kemudian admin memilih tombol simpan untuk menyimpan parameter. Sistem akan menyimpan dan memberikan notifikasi berhasil menyimpan. Penjelasan dari *activity diagram* Ganti Parameter dapat dilihat pada Gambar 3. 6.



Gambar 3. 6 Perancangan *Activity Diagram* Ganti Parameter

E. *Activity Diagram* Klasifikasi

Activity Diagram Klasifikasi menjelaskan tentang aktivitas proses *user* dokter dalam mengklasifikasi penyakit pasien. Tahapan klasifikasi dimulai setelah *user* dokter login, maka sistem akan menampilkan halaman dashboard dokter. Kemudian dokter memilih menu klasifikasi, maka sistem akan menampilkan halaman klasifikasi. Pada halaman klasifikasi ini, dokter dapat mengisi nilai frekuensi nadi, frekuensi nafas, suhu, nyeri, tinggi badan, berat badan, dan riwayat penyakit positif dari pasien pada form yang telah disediakan. Kemudian dokter memilih tombol start untuk memulai proses klasifikasi. Sistem melakukan proses klasifikasi dan akan menampilkan hasil klasifikasi pada halaman hasil diagnosis beserta nilai himpunan. Penjelasan dari *activity diagram* Klasifikasi dapat dilihat pada Gambar 3. 7.



Gambar 3. 7 Perancangan Activity Diagram Klasifikasi

3.3.3 Perancangan Antarmuka

A. Halaman Login

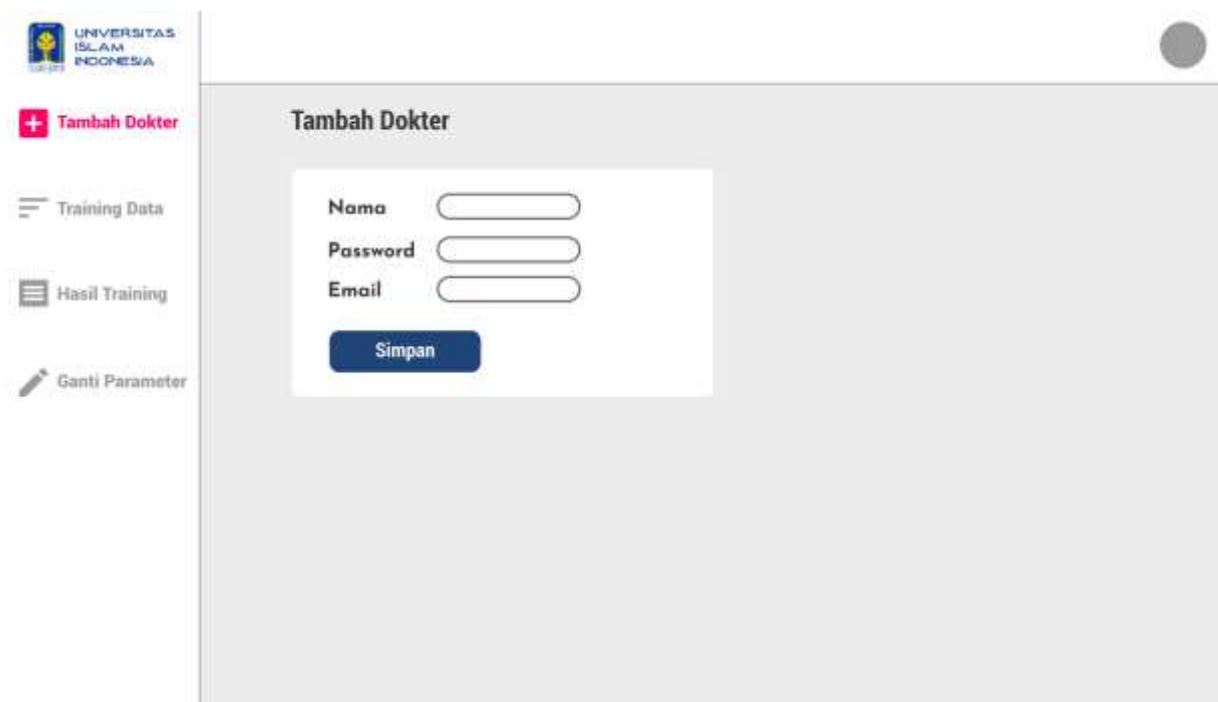
Halaman login akan ditampilkan pertama kali oleh sistem ketika *user* mengakses website. Fungsi dari halaman ini adalah untuk melakukan proses login ke dalam sistem. Pada halaman ini terdapat *form* login yang terdiri dari nama, password dan tombol login. Perancangan halaman ini dapat dilihat pada Gambar 3. 8.



Gambar 3. 8 Rancangan Halaman Login

B. Halaman Tambah Dokter

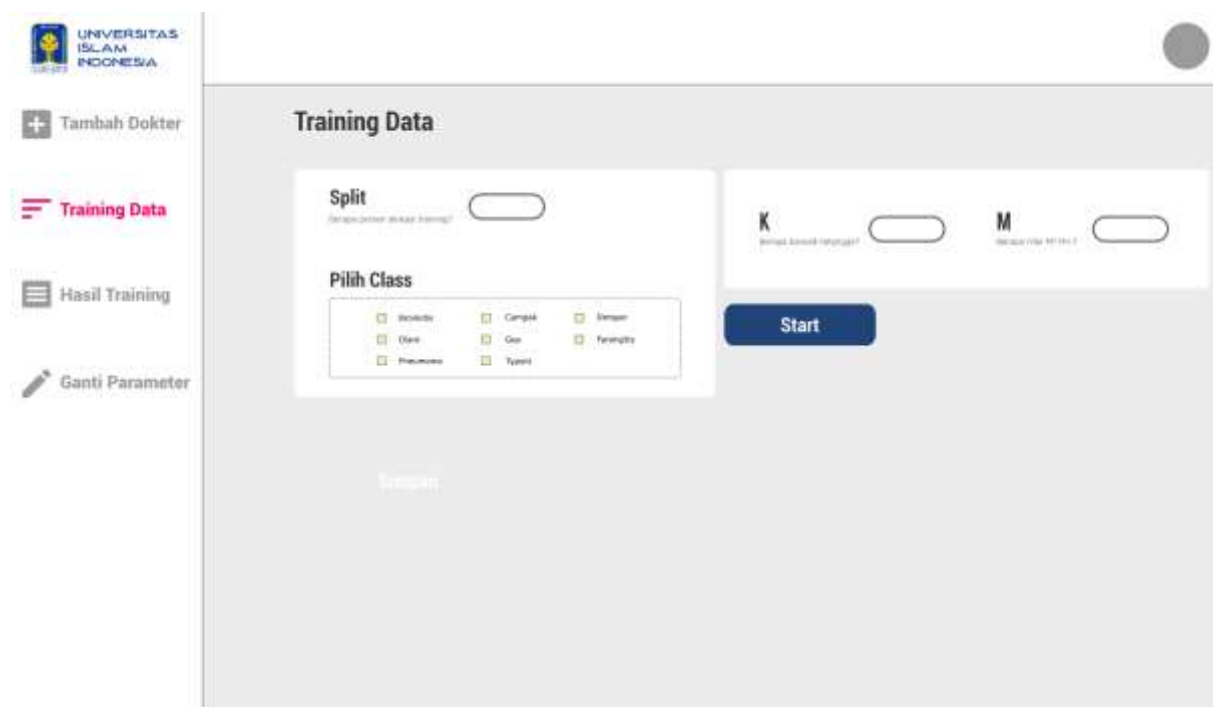
Halaman tambah dokter akan ditampilkan ketika *user* admin memilih menu tambah dokter setelah login ke dalam sistem. Fungsi dari halaman ini adalah untuk menambahkan *user* dokter. Pada halaman ini terdapat *form* tambah dokter yang terdiri dari nama, password, email dan tombol login. Perancangan halaman ini dapat dilihat pada Gambar 3. 9.



Gambar 3. 9 Rancangan Halaman Tambah Dokter

C. Halaman Training Data

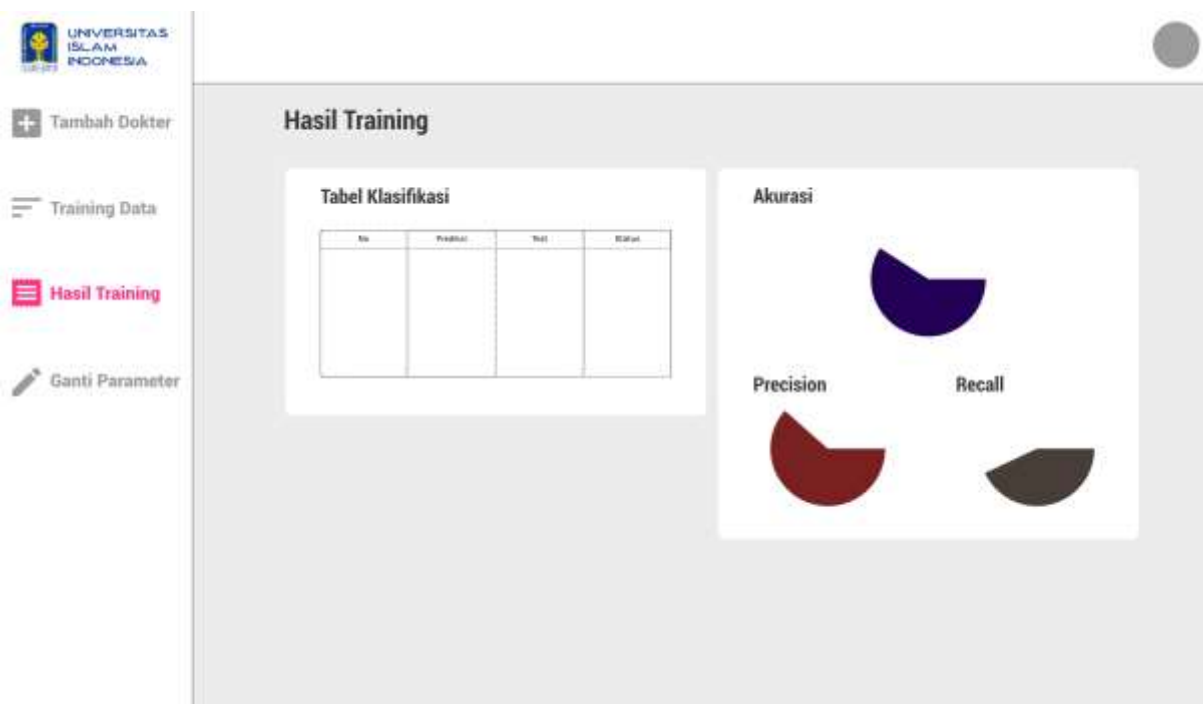
Halaman training data akan ditampilkan ketika *user* admin memilih menu training data setelah login ke dalam sistem. Fungsi dari halaman ini adalah untuk melakukan pembelajaran terhadap sistem guna mendapatkan hasil akurasi yang baik. Pada halaman ini terdapat *form* training data yang terdiri dari nilai pembagian data, nilai K, nilai M, pilihan penyakit / kelas dan tombol start. Perancangan halaman ini dapat dilihat pada Gambar 3. 10.



Gambar 3. 10 Rancangan Halaman Training Data

D. Halaman Hasil Training

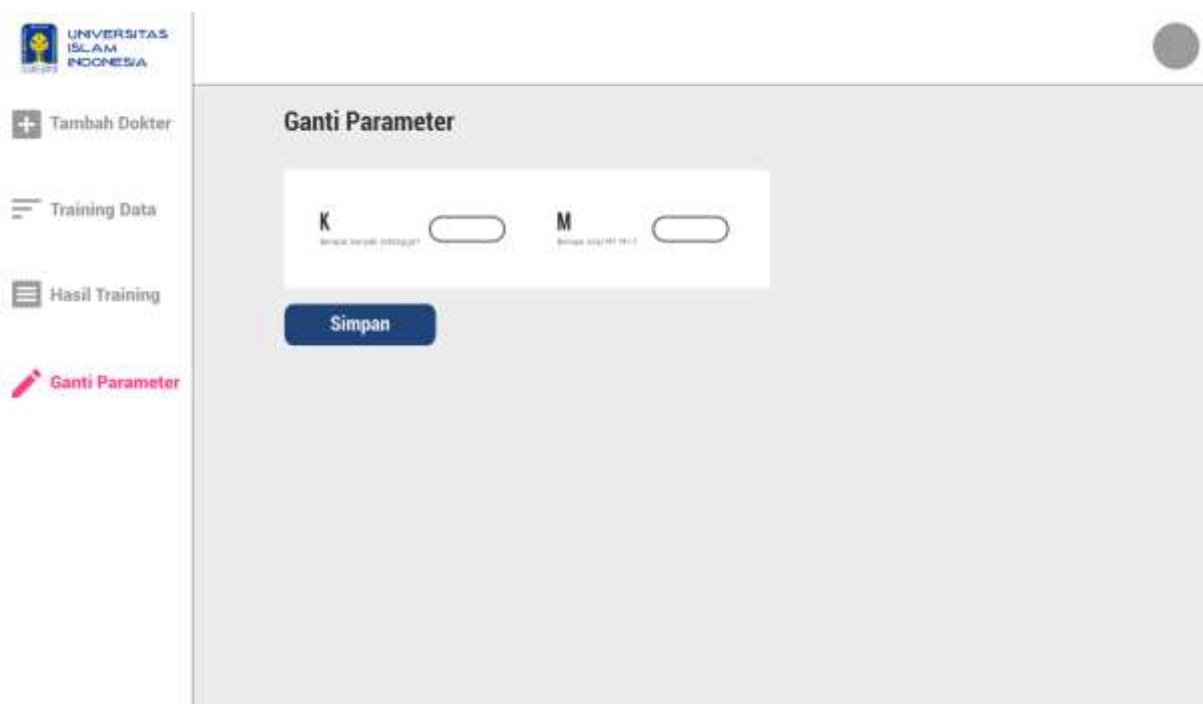
Halaman hasil training data akan ditampilkan ketika *user* admin selesai melakukan training data. Fungsi dari halaman ini adalah untuk menampilkan hasil klasifikasi, akurasi, precision, dan recall dari proses pada menu training data. Pada halaman ini terdapat tabel klasifikasi dan grafik dari akurasi, precision, dan recall. Perancangan halaman ini dapat dilihat pada Gambar 3. 11.



Gambar 3. 11 Rancangan Halaman Hasil Training

E. Halaman Ganti Parameter

Halaman ganti parameter akan ditampilkan ketika *user* admin memilih menu ganti parameter setelah login ke dalam sistem. Fungsi dari halaman ini adalah untuk melakukan pergantian nilai K dan M pada model klasifikasi. Pada halaman ini terdapat *form* ganti parameter yang terdiri dari nilai K, nilai M dan tombol simpan. Perancangan halaman ini dapat dilihat pada Gambar 3. 12.

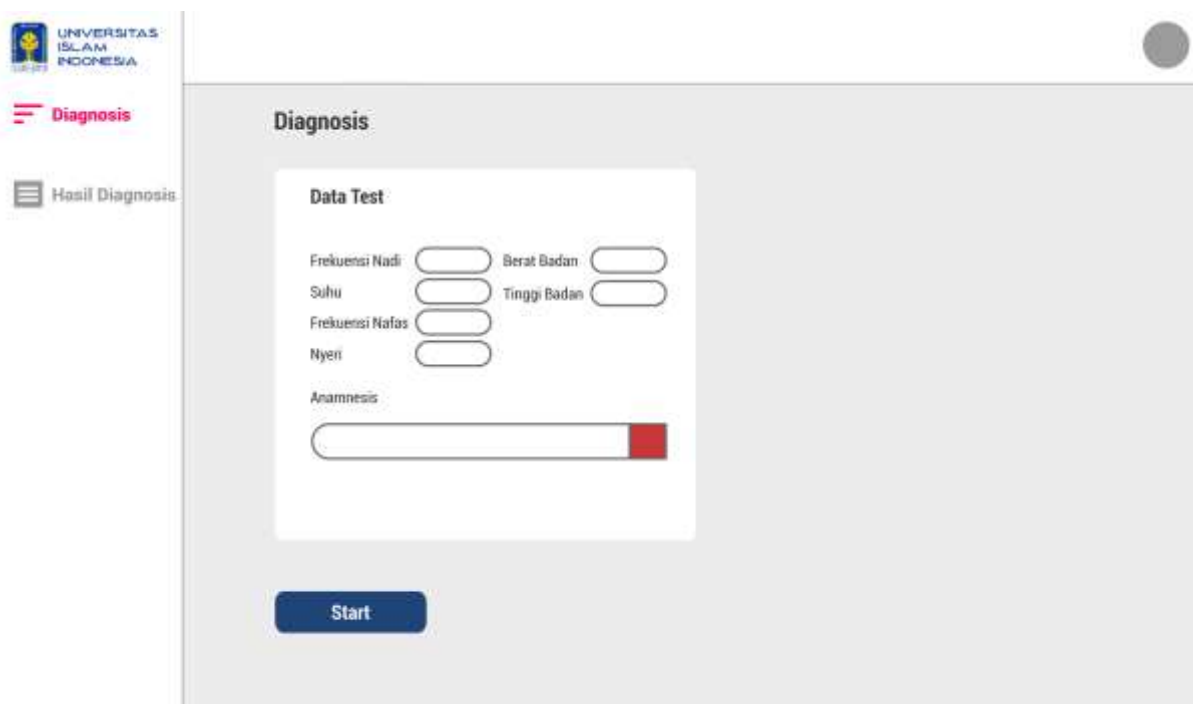


Gambar 3. 12 Rancangan Halaman Ganti Parameter

F. Halaman Diagnosis

Halaman diagnosis akan ditampilkan ketika *user* dokter berhasil login ke dalam sistem. Fungsi dari halaman ini adalah untuk melakukan diagnosis penyakit dari rekam medis pasien. Pada halaman ini terdapat *form* diagnosis yang terdiri dari frekuensi nadi, suhu, frekuensi nafas, nyeri, berat badan, tinggi badan, anamnesis, dan tombol start. Perancangan halaman ini dapat dilihat pada

Gambar 3. 13.

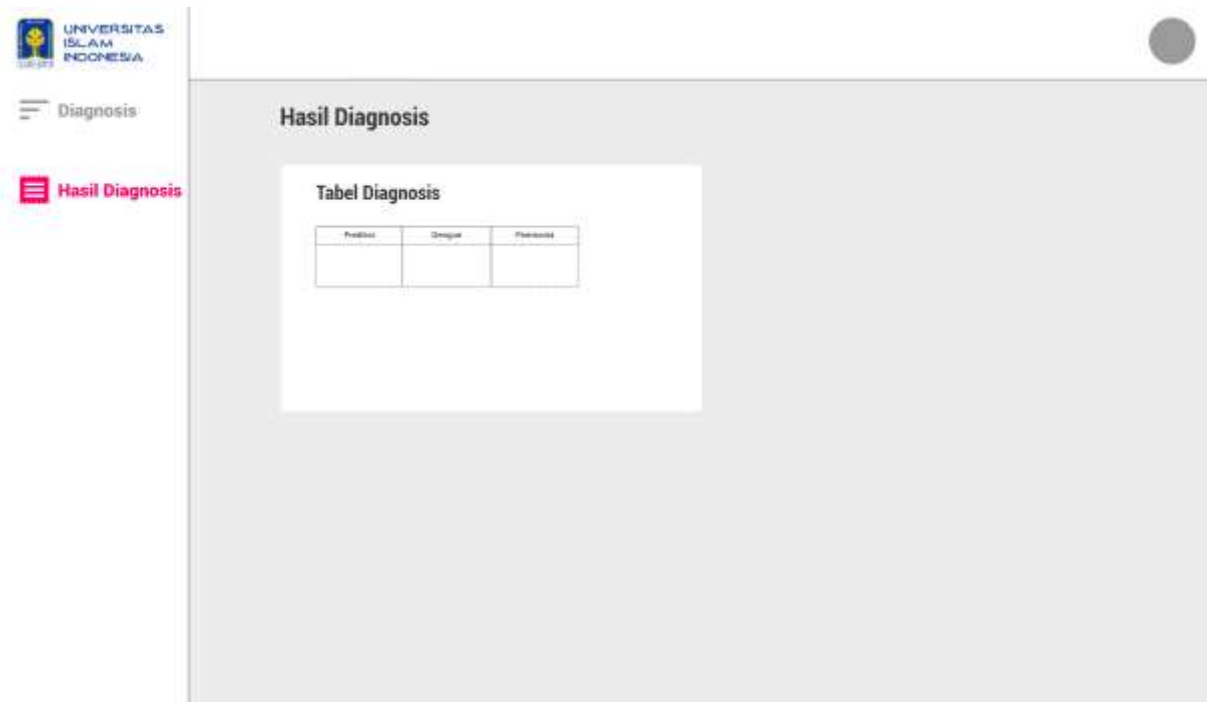


The image shows a web application interface for a medical diagnosis system. On the left is a sidebar with the logo of Universitas Islam Indonesia and two menu items: 'Diagnosis' (highlighted in red) and 'Hasil Diagnosis'. The main area is titled 'Diagnosis' and contains a 'Data Test' section with several input fields: 'Frekuensi Nadi', 'Suhu', 'Frekuensi Nafas', 'Nyeri', 'Berat Badan', and 'Tinggi Badan'. Below these is an 'Anamnesis' section with a text input field. At the bottom of the main area is a blue 'Start' button.

Gambar 3. 13 Rancangan Halaman Diagnosis

G. Halaman Hasil Diagnosis

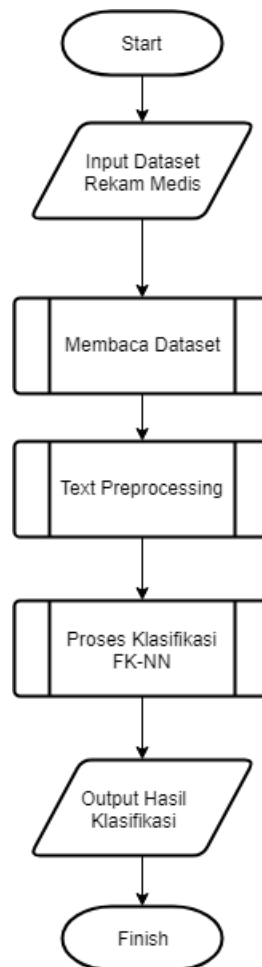
Halaman hasil diagnosis akan ditampilkan ketika *user* dokter selesai melakukan diagnosis. Fungsi dari halaman ini adalah untuk menampilkan hasil diagnosis dan nilai himpunan dari proses klasifikasi. Pada halaman ini terdapat tabel klasifikasi yang terdiri dari hasil diagnosis dan nilai himpunan. Perancangan halaman ini dapat dilihat pada Gambar 3. 14.



Gambar 3. 14 Rancangan Halaman Hasil Diagnosis

3.3.4 *Flowchart*

Model klasifikasi yang dipakai pada penelitian ini memiliki lima tahapan untuk mendapatkan hasil klasifikasi dari penyakit. Tahapan tersebut terdiri dari proses memasukkan data, tiga prosedur, dan proses mengeluarkan hasil klasifikasi. Tiga prosedur pada *flowchart* ini adalah prosedur membaca dataset, *text preprocessing*, dan klasifikasi FK-NN. Tahapan tersebut dapat dilihat pada Gambar 3. 15.



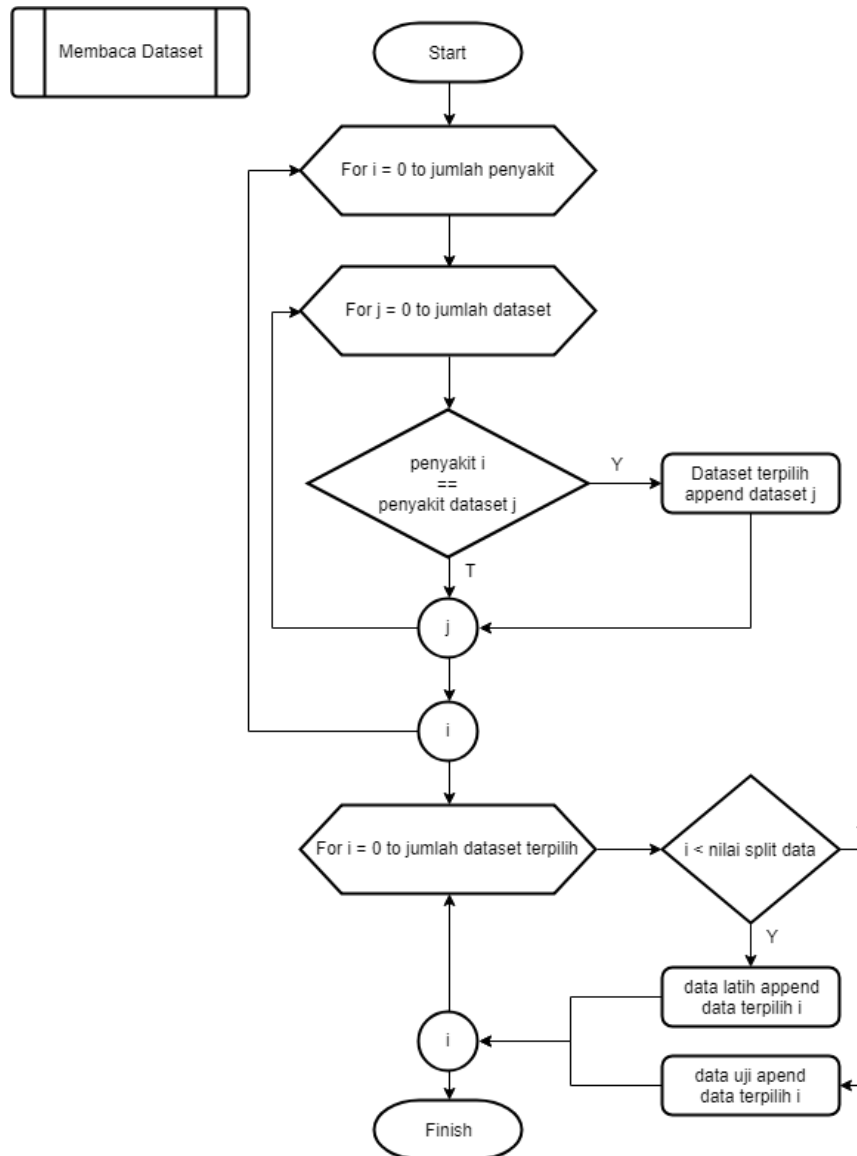
Gambar 3. 15 *Flowchart* Sistem Klasifikasi

A. Membaca Data

Pada *flowchart* membaca dataset ini merupakan gambaran dari prosedur membaca dataset. Tahapan ini bertujuan untuk membaca dataset yang sudah dimasukkan kemudian membagi dataset menjadi data uji dan data latih. Tahapan dari prosedur membaca dataset adalah sebagai berikut:

1. Lakukan perulangan For dari $i = 0$ sampai jumlah penyakit yang dipilih.
2. Didalam perulangan i , lakukan perulangan For dari $j = 0$ sampai jumlah dataset.
3. Didalam perulangan j , lakukan pengecekan apakah penyakit i sama dengan penyakit pada dataset j .
4. Jika sama maka dataset terpilih diisi dengan dataset j
5. Setelah dua perulangan selesai, lakukan perulangan For dari $i = 0$ sampai jumlah dataset terpilih
6. Didalam perulangan i , lakukan pengecekan apakah i kurang dari nilai split data.
7. Jika iya maka data latih diisi dengan dataset j
8. Jika tidak maka data uji diisi dengan dataset j

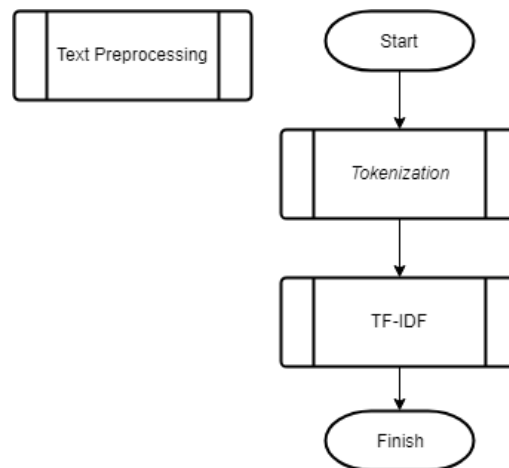
Untuk lebih jelasnya dapat melihat *flowchart* pada Gambar 3. 16.



Gambar 3. 16 *Flowchart* Membaca Dataset

B. Text Preprocessing

Pada *flowchart text preprocessing* ini merupakan gambaran dari prosedur *text preprocessing*. Tahapan ini bertujuan untuk membuat data menjadi terstruktur dan dapat diproses sistem. Tahapan tersebut terdiri dari dua prosedur yaitu prosedur *tokenization* dan TF-IDF. Tahapan tersebut dapat dilihat pada Gambar 3. 17.



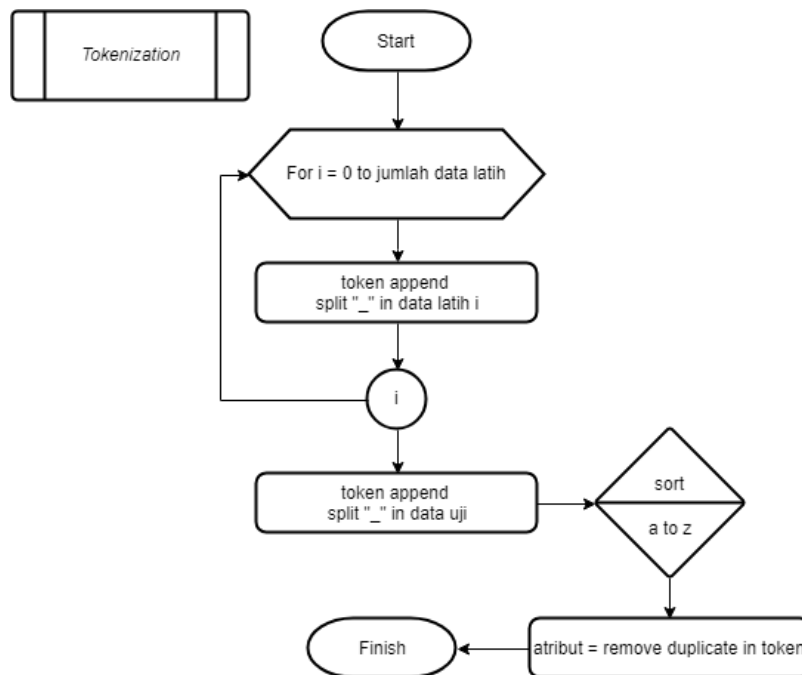
Gambar 3. 17 *Flowchart Text Preprocessing*

a. *Tokenization*

Pada *flowchart tokenization* ini merupakan gambaran dari prosedur *tokenization*. Tahapan ini bertujuan untuk membuat *token* dari dataset teks. *Tokenization* yang dipakai dalam penelitian ini adalah *sentence tokenization*. Tahapan dari prosedur *tokenization* adalah sebagai berikut:

1. Lakukan perulangan For dari $i = 0$ sampai jumlah data latih.
2. Didalam perulangan i , lakukan pembuatan token dengan memisahkan karakter “_” pada setiap data latih i .
3. Setelah perulangan selesai, lakukan pembuatan token dengan memisahkan karakter “_” pada data uji.
4. Lakukan pengurutan berdasarkan abjad A sampai Z.
5. Terakhir hilangkan duplikat dan masukkan ke dalam variabel atribut

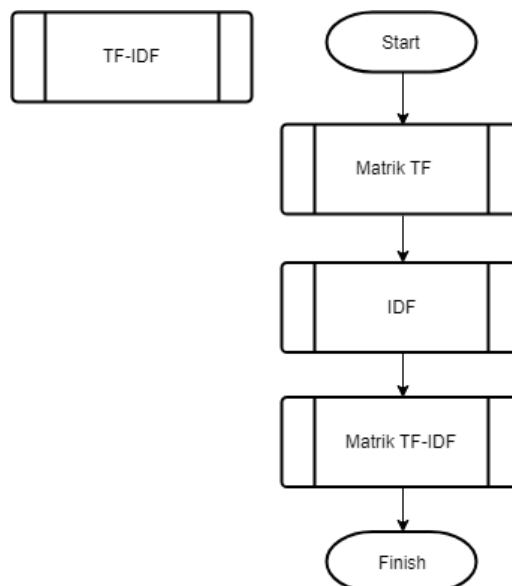
Untuk lebih jelasnya dapat melihat *flowchart* pada Gambar 3. 18.



Gambar 3. 18 *Flowchart Tokenization*

b. TF-IDF

Pada *flowchart* TF-IDF ini merupakan gambaran dari prosedur TF-IDF. Tahapan ini bertujuan untuk mentransformasi data teks yang telah dilakukan proses *tokenization* menjadi data angka sehingga dapat dihitung. Tahapan tersebut terdiri dari tiga prosedur yaitu prosedur matrik TF, IDF, dan matrik TF-IDF. Tahapan tersebut dapat dilihat pada Gambar 3. 19.



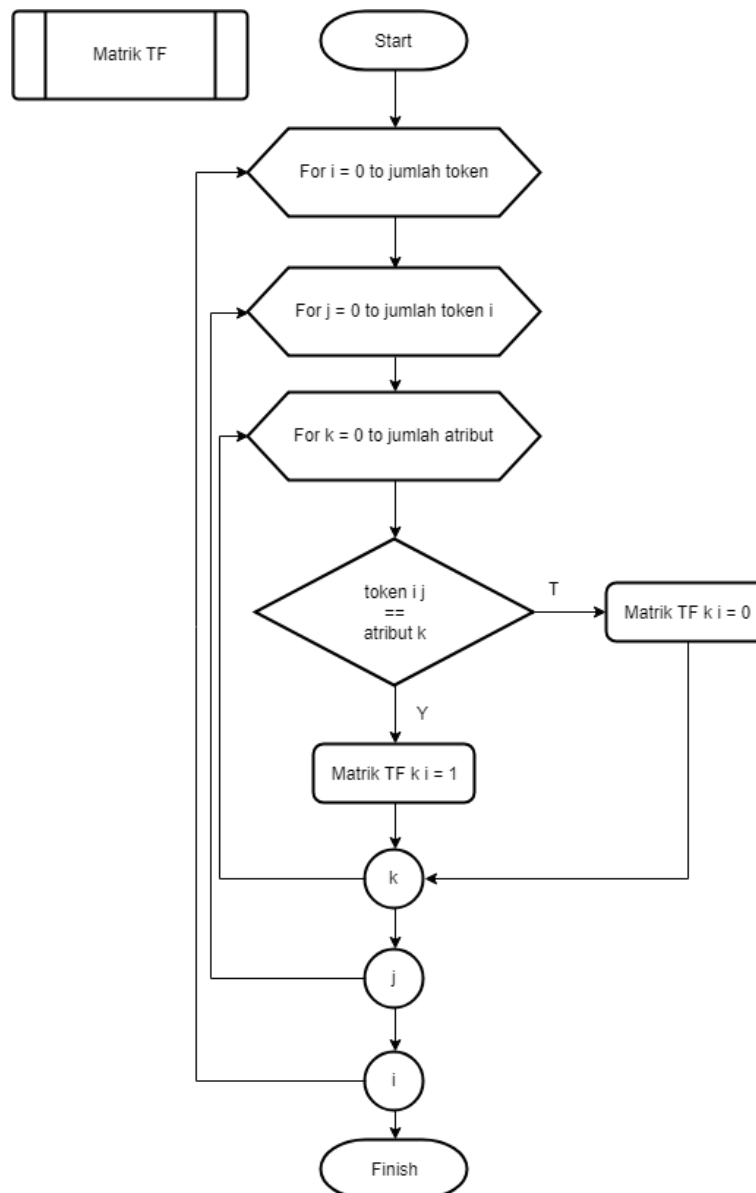
Gambar 3. 19 *Flowchart TF-IDF*

1. Matrik TF

Pada *flowchart* matrik TF ini merupakan gambaran dari prosedur matrik TF. Tahapan ini bertujuan untuk membuat matrik TF yang berisi nilai 1 dan 0 dari perbandingan antara *token* dan atribut. Tahapan dari prosedur matrik TF adalah sebagai berikut:

1. Lakukan perulangan For dari $i = 0$ sampai jumlah token.
2. Didalam perulangan i , lakukan perulangan For dari $j = 0$ sampai jumlah token i .
3. Didalam perulangan j , lakukan perulangan For dari $k = 0$ sampai jumlah atribut.
4. Didalam perulangan k , lakukan pengecekan apakah token i sama dengan atribut j .
5. Jika iya maka matrik TF k $i = 1$.
6. Jika tidak maka matrik TF k $i = 0$.

Untuk lebih jelasnya dapat melihat *flowchart* pada Gambar 3. 20.



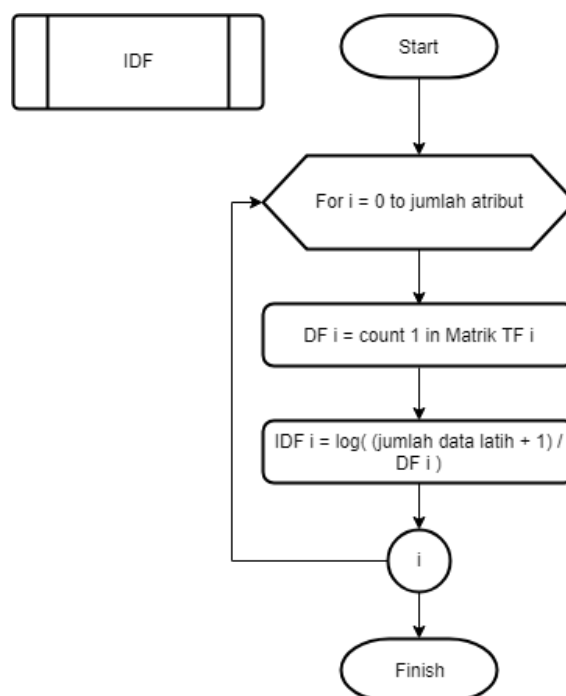
Gambar 3. 20 *Flowchart* Matrik TF

2. IDF

Pada *flowchart* matrik TF ini merupakan gambaran dari prosedur matrik TF. Tahapan ini bertujuan untuk membuat matrik TF yang berisikan nilai 1 dan 0 dari perbandingan antara *token* dan atribut. Tahapan dari prosedur matrik TF adalah sebagai berikut:

1. Lakukan perulangan For dari $i = 0$ sampai jumlah token.
2. Didalam perulangan i , lakukan perulangan For dari $j = 0$ sampai jumlah token i .
3. Didalam perulangan j , lakukan perulangan For dari $k = 0$ sampai jumlah atribut.
4. Didalam perulangan k , lakukan pengecekan apakah token i sama dengan atribut j .
5. Jika iya maka matrik TF k $i = 1$.
6. Jika tidak maka matrik TF k $i = 0$.

Untuk lebih jelasnya dapat melihat *flowchart* pada Gambar 3. 20.



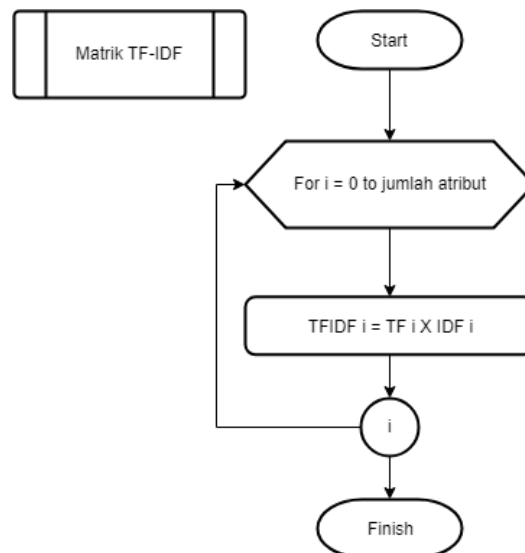
Gambar 3. 21 *Flowchart* IDF

3. Matrik TF-IDF

Pada *flowchart* matrik TF-IDF ini merupakan gambaran dari prosedur matrik TF-IDF. Tahapan ini bertujuan untuk membuat matrik TF-IDF yang berisikan nilai perkalian antara matrik TF dan IDF. Tahapan dari prosedur matrik TF-IDF adalah sebagai berikut:

1. Lakukan perulangan For dari $i = 0$ sampai jumlah atribut.
2. Didalam perulangan i , lakukan perkalian scalar antara matrik TF i dan IDF i .

Untuk lebih jelasnya dapat melihat *flowchart* pada Gambar 3. 22.



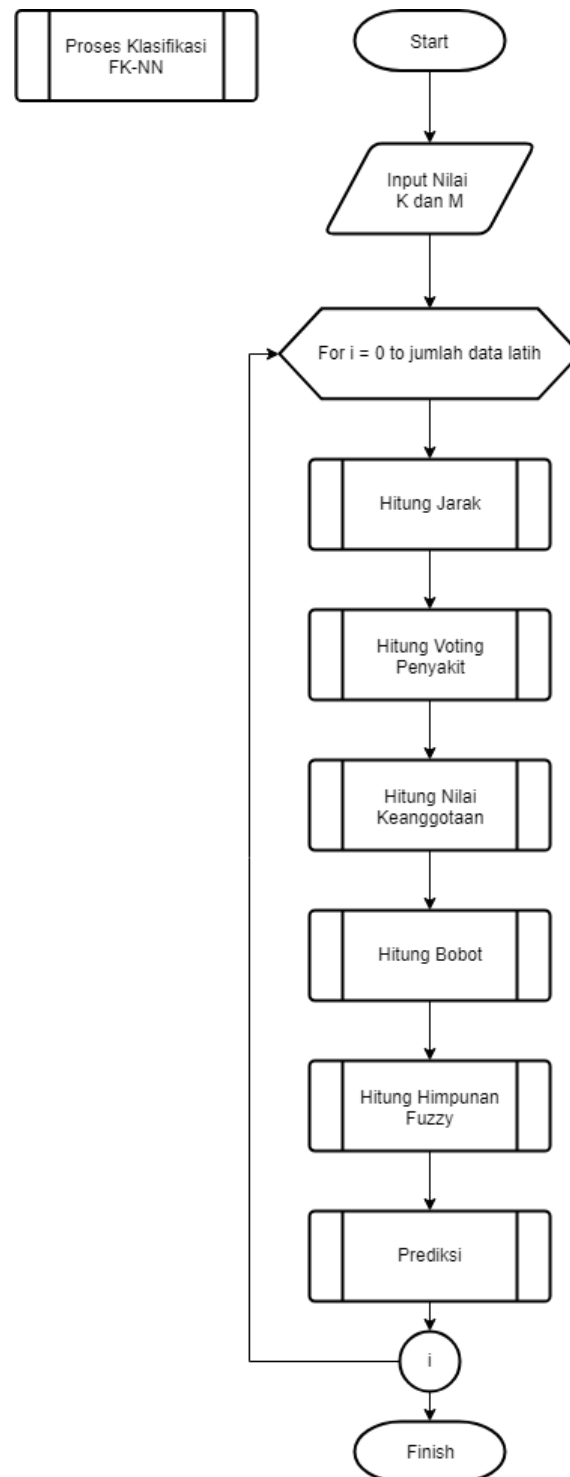
Gambar 3. 22 *Flowchart* Matrik TF-IDF

C. Klasifikasi FK-NN

Pada *flowchart* klasifikasi FK-NN ini merupakan gambaran dari prosedur klasifikasi FK-NN. Tahapan ini bertujuan untuk mengklasifikasikan penyakit pada rekam medis dengan FK-NN menggunakan hasil transformasi teks menjadi angka pada proses TF-IDF. Tahapan dari prosedur klasifikasi FK-NN adalah sebagai berikut:

1. Memasukkan nilai K dan M.
2. Lakukan perulangan For dari $i = 0$ sampai jumlah data latih.
3. Didalam perulangan i , lakukan prosedur hitung jarak.
4. Lakukan prosedur hitung voting setiap penyakit.
5. Lakukan prosedur hitung nilai keanggotaan.
6. Lakukan prosedur hitung nilai bobot.
7. Lakukan prosedur hitung himpunan fuzzy
8. Lakukan prosedur prediksi

Untuk lebih jelasnya dapat melihat *flowchart* pada Gambar 3. 23.



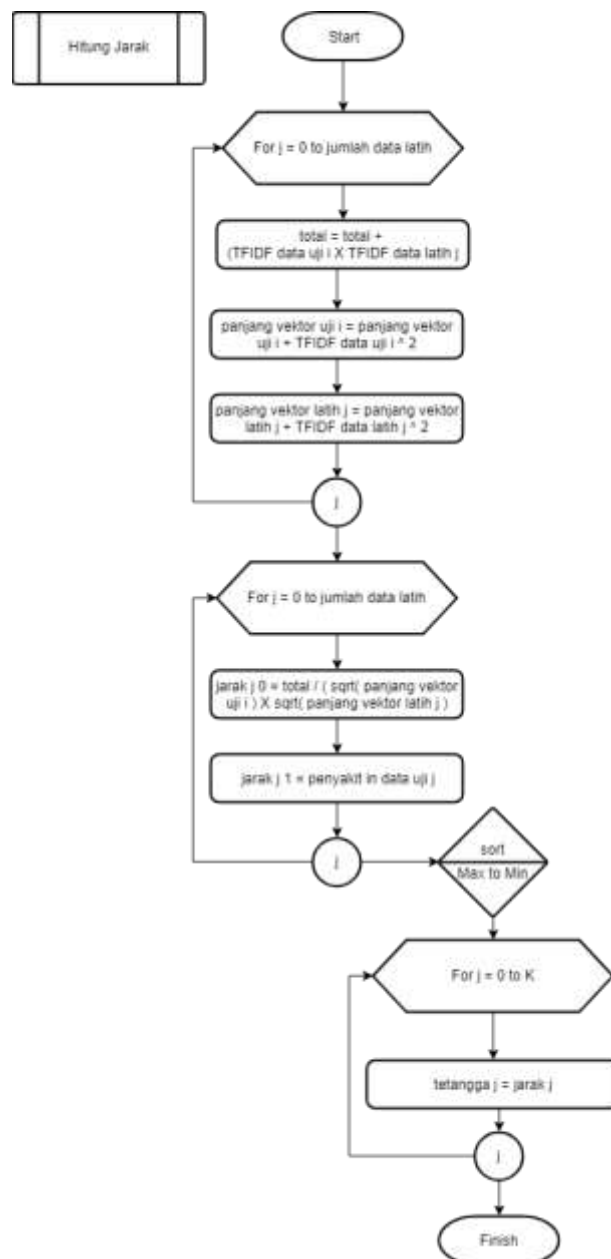
Gambar 3. 23 *Flowchart* Klasifikasi FK-NN

a. Hitung Jarak

Pada *flowchart* hitung jarak ini merupakan gambaran dari prosedur hitung jarak. Tahapan ini bertujuan untuk menghitung jarak tetangga antara data uji dengan semua data latih. Tahapan dari prosedur hitung jarak adalah sebagai berikut:

1. Lakukan perulangan For dari $j = 0$ sampai jumlah data latih.

2. Didalam perulangan j, lakukan perhitungan total dari perkalian TF-IDF data uji dengan data latih j.
 3. Lakukan perhitungan panjang vektor data uji dan data latih j.
 4. Setelah perulangan j selesai, lakukan perulangan For dari j = 0 sampai jumlah data latih.
 5. Didalam perulangan j, lakukan perhitungan jarak.
 6. Memasukkan penyakit data latih j kedalam nilai jarak j.
 7. Setelah perulangan j selesai, lakukan pengurutan dari besar ke kecil.
 8. Lakukan perulangan For dari j = 0 sampai K.
 9. Didalam perulangan j, lakukan pengambilan tetangga dari jarak 0 sampai K
- Untuk lebih jelasnya dapat melihat *flowchart* pada Gambar 3. 24.



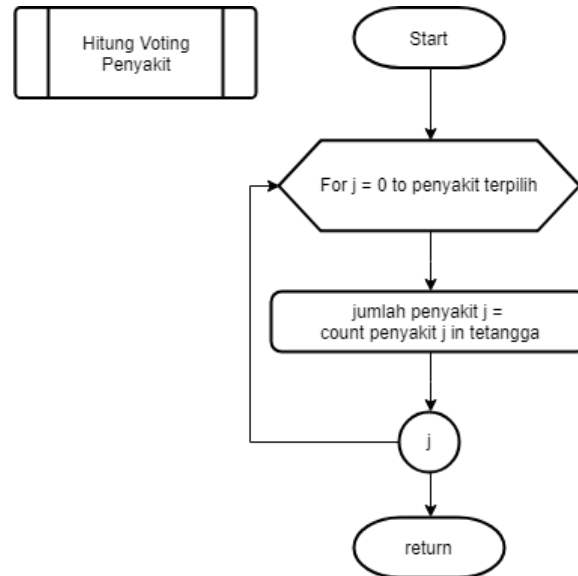
Gambar 3. 24 *Flowchart* Hitung Jarak

b. Hitung Voting Penyakit

Pada *flowchart* hitung voting penyakit ini merupakan gambaran dari prosedur hitung voting penyakit. Tahapan ini bertujuan untuk menghitung voting setiap penyakit pada tetangga terpilih. Tahapan dari prosedur hitung jarak adalah sebagai berikut:

1. Lakukan perulangan For dari $j = 0$ sampai jumlah penyakit terpilih.
2. Didalam perulangan j , lakukan perhitungan jumlah penyakit j pada tetangga.

Untuk lebih jelasnya dapat melihat *flowchart* pada Gambar 3. 25.



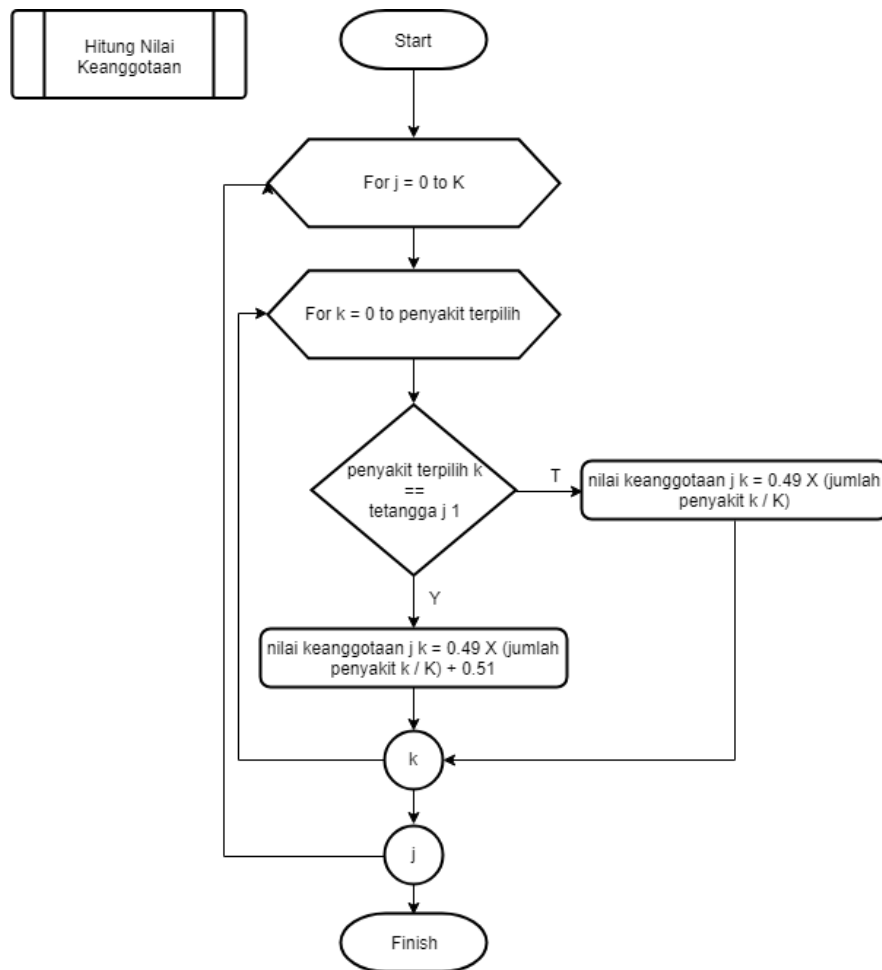
Gambar 3. 25 *Flowchart* Hitung Voting Penyakit

c. Hitung Nilai Keanggotaan

Pada *flowchart* hitung nilai keanggotaan ini merupakan gambaran dari prosedur hitung nilai keanggotaan. Tahapan ini bertujuan untuk menghitung nilai keanggotaan tetangga terpilih terhadap kelas penyakit. Tahapan dari prosedur hitung jarak adalah sebagai berikut:

1. Lakukan perulangan For dari $j = 0$ sampai K .
2. Didalam perulangan j , lakukan perulangan For dari $k = 0$ sampai jumlah penyakit terpilih.
3. Didalam perulangan k , lakukan pengecekan apakah penyakit terpilih k sama dengan penyakit pada tetangga j .
4. Jika iya maka nilai keanggotaan = $0.49 \times (\text{nilai voting penyakit } k / \text{nilai } K) + 0.51$
5. Jika tidak maka nilai keanggotaan = $0.49 \times (\text{nilai voting penyakit } k / \text{nilai } K)$

Untuk lebih jelasnya dapat melihat *flowchart* pada Gambar 3. 26.



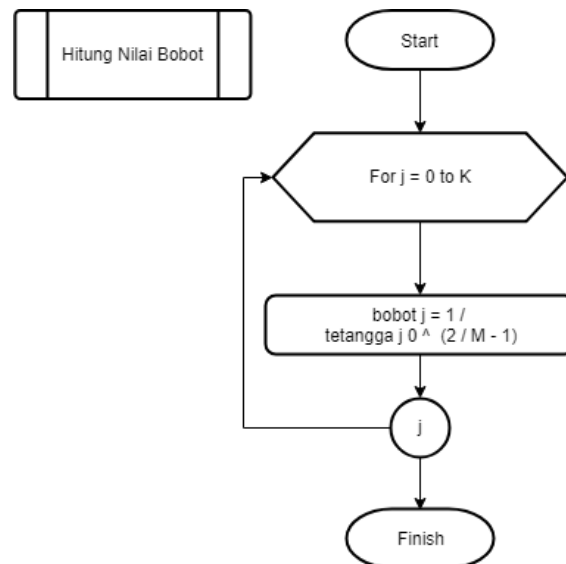
Gambar 3. 26 Flowchart Hitung Nilai Keanggotaan

d. Hitung Nilai Bobot

Pada *flowchart* hitung nilai bobot ini merupakan gambaran dari prosedur hitung nilai bobot. Tahapan ini bertujuan untuk menghitung bobot untuk setiap tetangga terpilih. Tahapan dari prosedur hitung nilai bobot adalah sebagai berikut:

1. Lakukan perulangan For dari $j = 0$ sampai K .
2. Didalam perulangan j , lakukan perhitungan bobot j dengan rumus $1 / \text{nilai tetangga } j^{(2 / M - 1)}$

Untuk lebih jelasnya dapat melihat *flowchart* pada Gambar 3. 27.



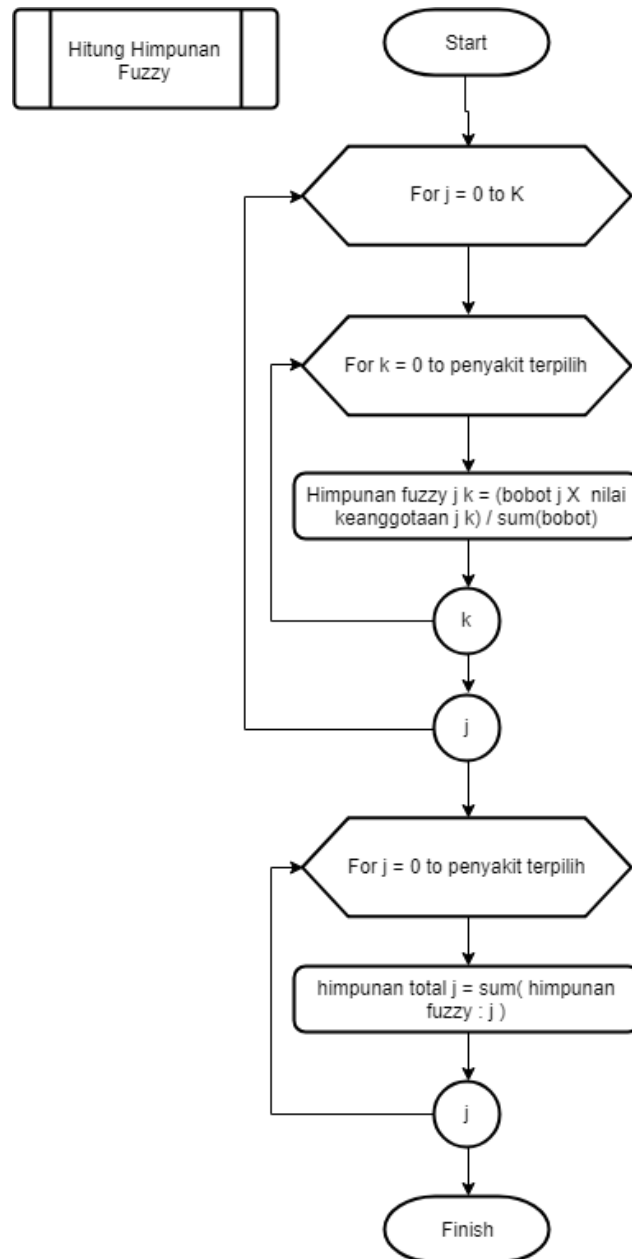
Gambar 3. 27 Hitung Nilai Bobot

e. Hitung Himpunan Fuzzy

Pada *flowchart* hitung himpunan fuzzy ini merupakan gambaran dari prosedur hitung himpunan fuzzy. Tahapan ini bertujuan untuk menghitung nilai himpunan fuzzy untuk tiap penyakit. Tahapan dari prosedur hitung himpunan fuzzy adalah sebagai berikut:

1. Lakukan perulangan For dari $j = 0$ sampai K .
2. Didalam perulangan j , lakukan perulangan For dari $k = 0$ sampai jumlah penyakit terpilih.
3. Didalam perulangan k , lakukan perhitungan fungsi fuzzy dengan rumus (bobot j X nilai keanggotaan $j k$) / total bobot
4. Setelah melakukan perulangan j , Lakukan perulangan For dari $j = 0$ sampai jumlah penyakit terpilih.
5. Didalam perulangan j , lakukan penjumlahan nilai himpunan fuzzy untuk setiap penyakit.

Untuk lebih jelasnya dapat melihat *flowchart* pada Gambar 3. 28.



Gambar 3. 28 *Flowchart* Hitung Himpunan Fuzzy

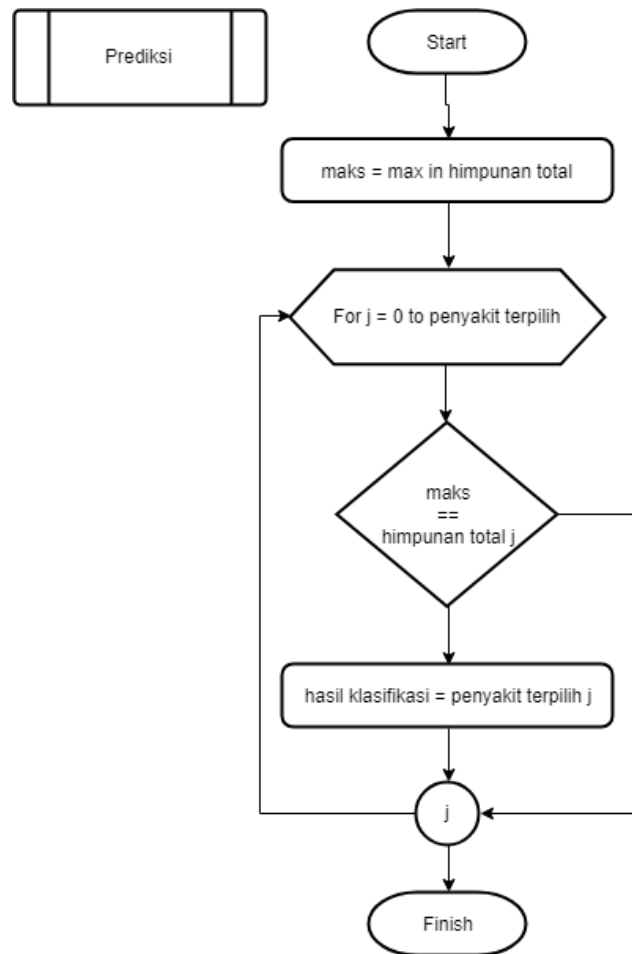
f. Hitung Prediksi

Pada *flowchart* prediksi ini merupakan gambaran dari prosedur prediksi. Tahapan ini bertujuan untuk menghasilkan hasil evaluasi berdasarkan himpunan fuzzy yang telah dihitung.

Tahapan dari prosedur prediksi adalah sebagai berikut:

1. Lakukan pencarian nilai maksimum dari himpunan fuzzy yang telah dijumlahkan.
2. Lakukan perulangan For dari $j = 0$ sampai jumlah penyakit terpilih.
3. Didalam perulangan j , lakukan pengecekan apakah nilai maksimum sama dengan nilai himpunan fuzzy yang telah dijumlahkan j .
4. Jika iya maka penyakit j diambil sebagai hasil evaluasi.

Untuk lebih jelasnya dapat melihat *flowchart* pada Gambar 3. 29.



Gambar 3. 29 *Flowchart* Prediksi

3.4 Implementasi

Dalam implementasi ini akan dilakukan pemrograman dan pengimplementasian modul kedalam framework Django.

3.5 Pengujian Sistem

Pengujian sistem akan dilakukan setelah sistem telah selesai dibuat. Pengujian akan menggunakan *Confusion Matrix*. *Confusion Matrix* sangat cocok digunakan untuk menguji performa dari model klasifikasi.

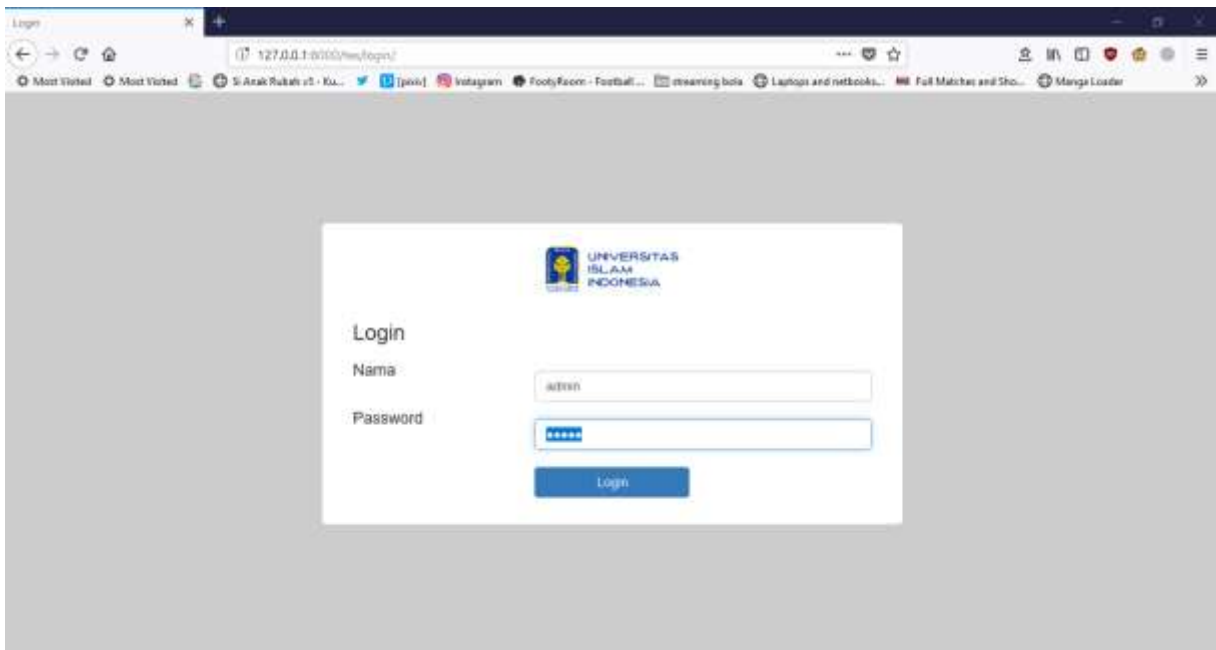
BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi

Sistem klasifikasi ini diimplementasikan ke dalam perangkat lunak berbasis web dengan berlandaskan kepada hasil perancangan antarmuka yang telah dibuat sebelumnya. Perangkat lunak dinilai baik apabila perangkat lunak yang dibangun bebas dari *error* dan memiliki tingkat akurasi yang baik sehingga mampu memberikan manfaat yang sesuai dengan tujuan dibuatnya perangkat lunak.

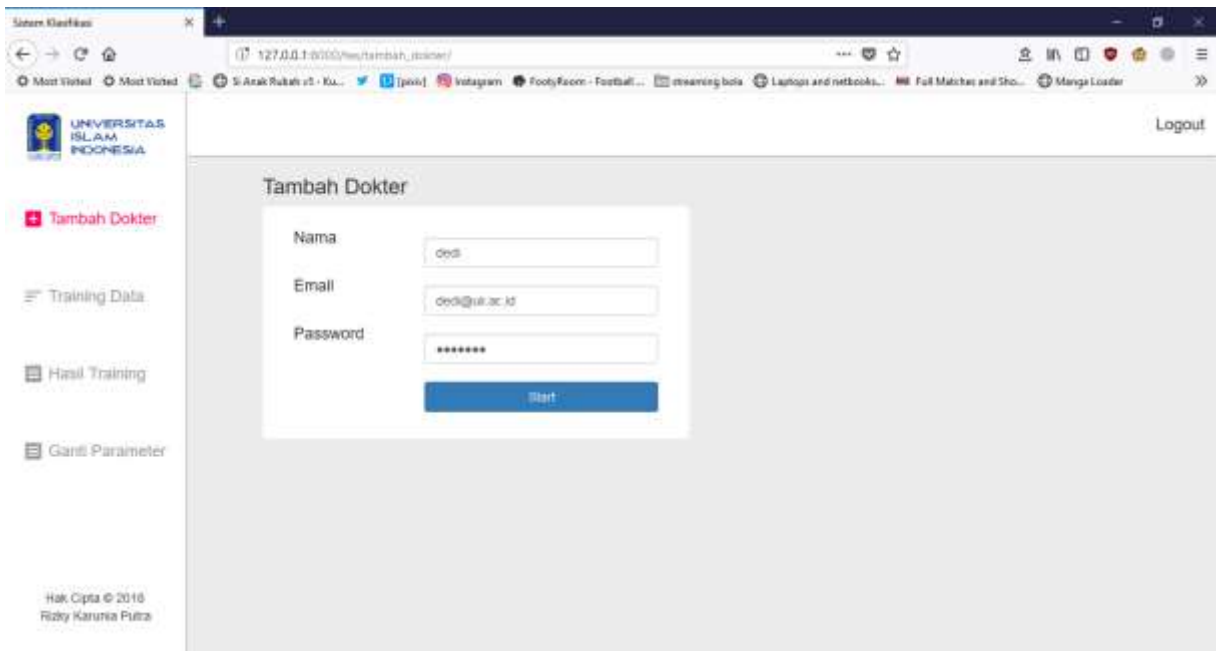
Perangkat lunak ini memiliki 2 *user* yang masing-masing memiliki fungsi sendiri. Setiap *user* wajib login terlebih dahulu agar sistem dapat mengenali antara *user* admin dan dokter. Pada halaman login terdapat *form login* yang berisi *inputan* nama dan password serta tombol untuk login. Tampilan dari halaman login dapat dilihat pada Gambar 4. 1.



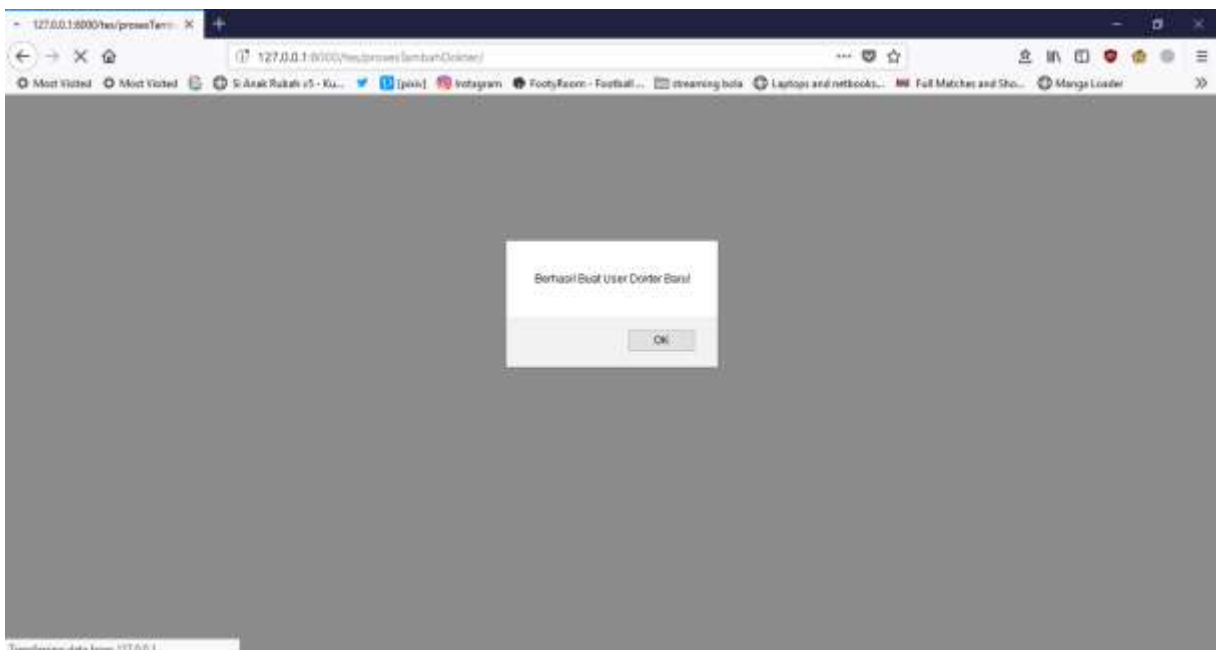
Gambar 4. 1 Halaman Login

Setelah *user* login, maka sistem akan mengarahkan *user* ke halaman dashboard nya masing-masing. Pada *user* admin, sistem akan mengarahkan ke halaman tambah dokter ketika admin berhasil login ke dalam sistem. Terdapat 4 halaman yang bisa admin lakukan terhadap sistem. Halaman pertama adalah halaman tambah dokter yang didalamnya terdapat *form* untuk menambahkan *user* dokter baru. *Form* ini terdiri dari *inputan* nama, email, dan password serta tombol start untuk menyimpan penambahan *user* dokter. Sistem akan memberikan notifikasi

telah berhasil menyimpan ketika tombol start dipilih. Tampilan dari halaman dan notifikasi dapat dilihat pada Gambar 4. 2 dan Gambar 4. 3.

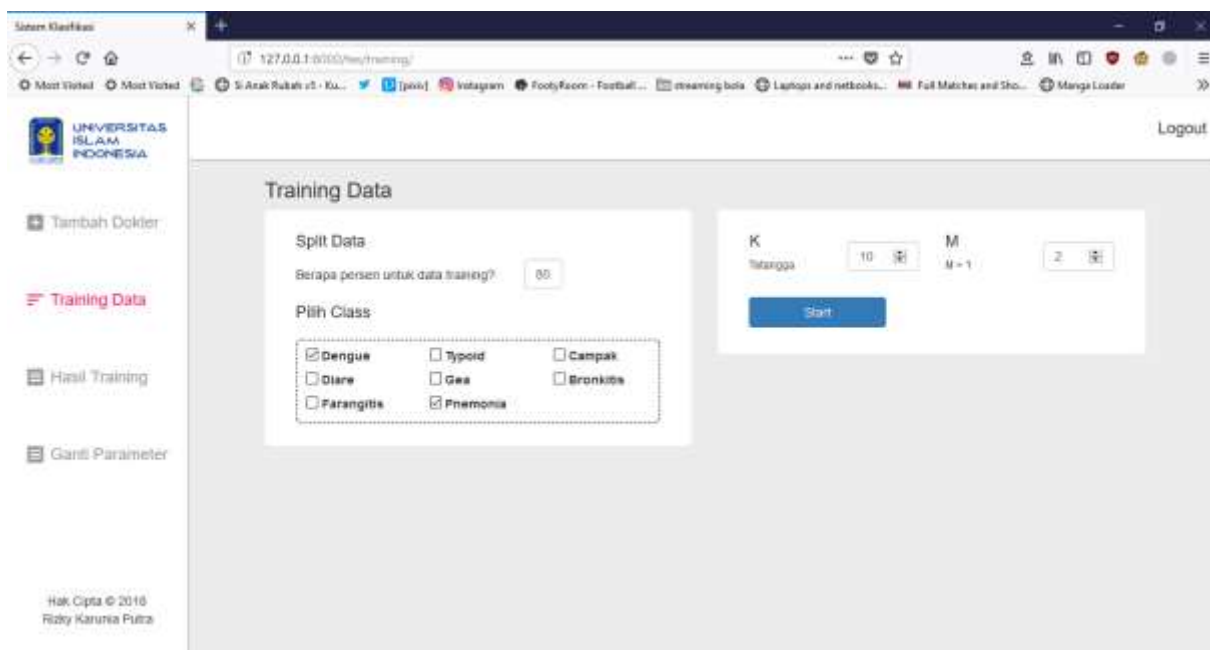


Gambar 4. 2 Halaman Tambah Dokter



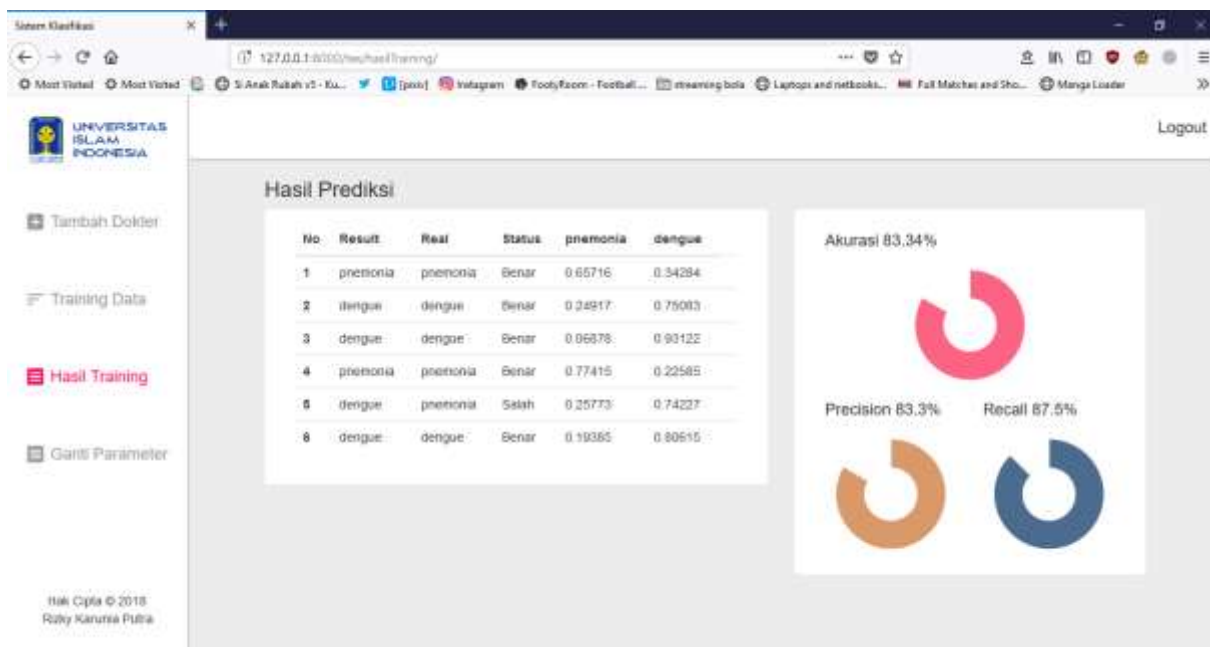
Gambar 4. 3 Notifikasi Berhasil Buat Dokter Baru

Halaman kedua adalah halaman training data yang berfungsi untuk menganalisis model klasifikasi dalam sistem. Halaman ini terdapat *form* yang berisikan *inputan* pembagian dataset, pilihan kelas penyakit, nilai K dan nilai M serta tombol start untuk memulai proses klasifikasi. Tampilan dari halaman ini dapat dilihat pada Gambar 4. 4.



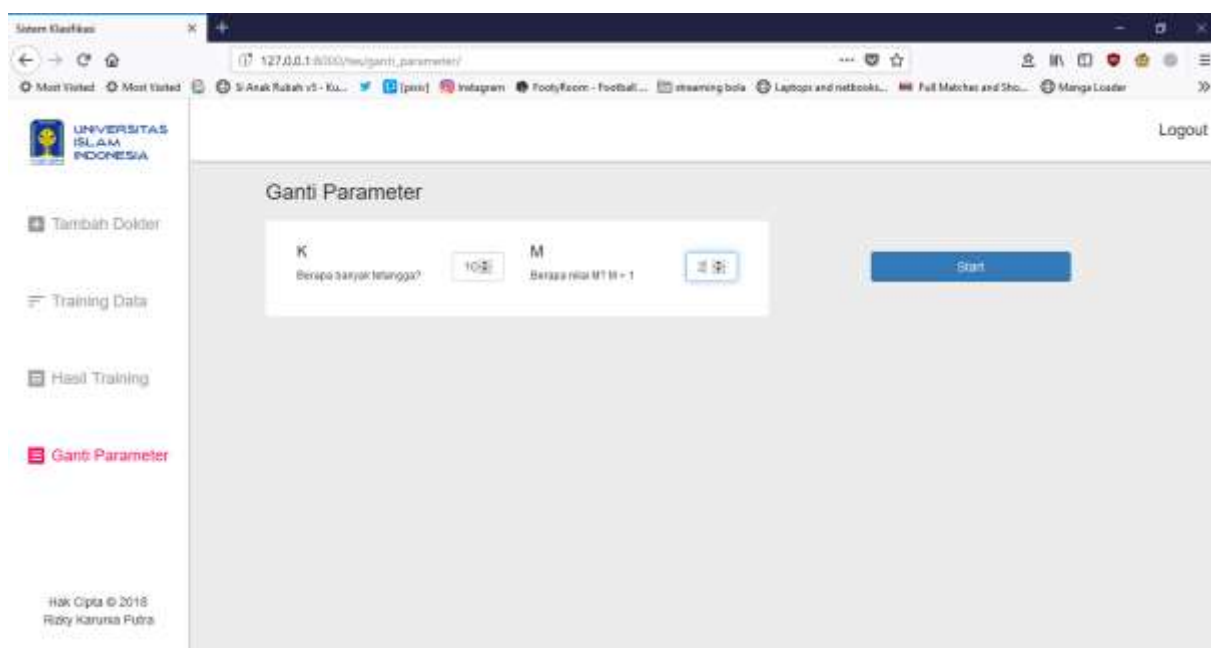
Gambar 4. 4 Halaman Training Data

Halaman ketiga adalah halaman hasil training yang berfungsi untuk melihat hasil klasifikasi dan akurasi dari model klasifikasi yang sistem ini gunakan. Halaman ini terdiri dari tabel klasifikasi beserta nilai himpunan fuzzy dan grafik dari akurasi, precision, dan recall. Tampilan dari halaman ini dapat dilihat pada Gambar 4. 5.



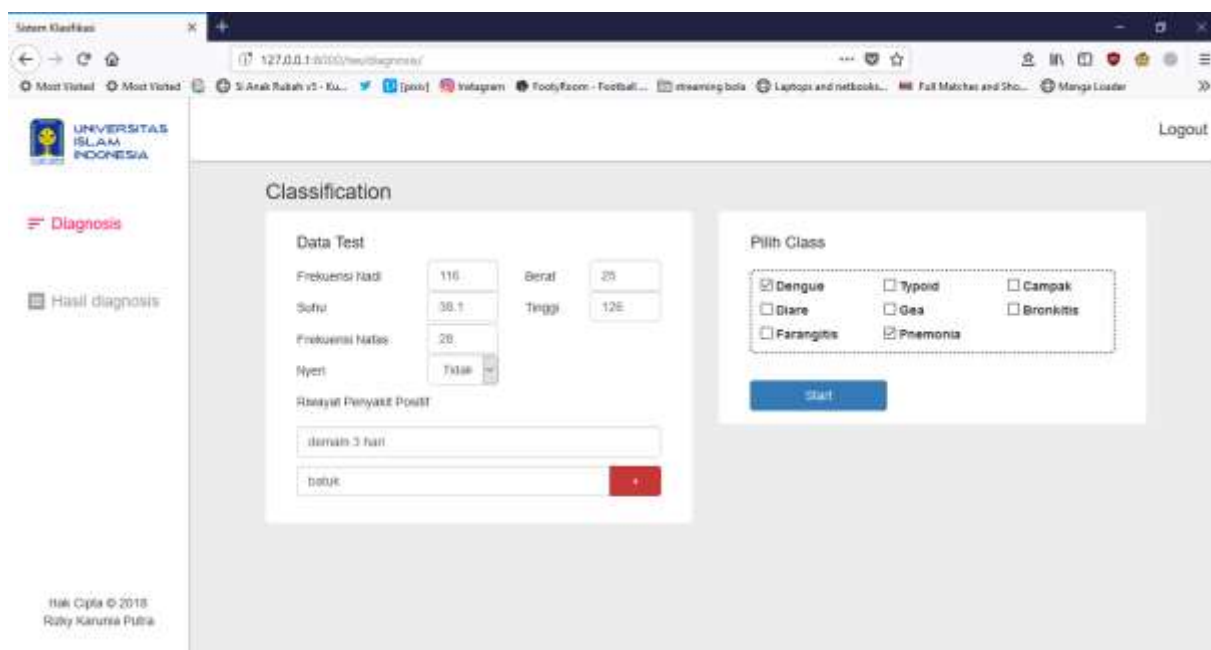
Gambar 4. 5 Halaman Hasil Training

Halaman terakhir adalah halaman ganti parameter yang berfungsi untuk mengganti nilai K dan M. Halaman ini terdapat *form* yang berisikan *inputan* nilai K dan nilai M serta tombol start untuk memulai ganti parameter. Tampilan dari halaman ini dapat dilihat pada Gambar 4. 6.



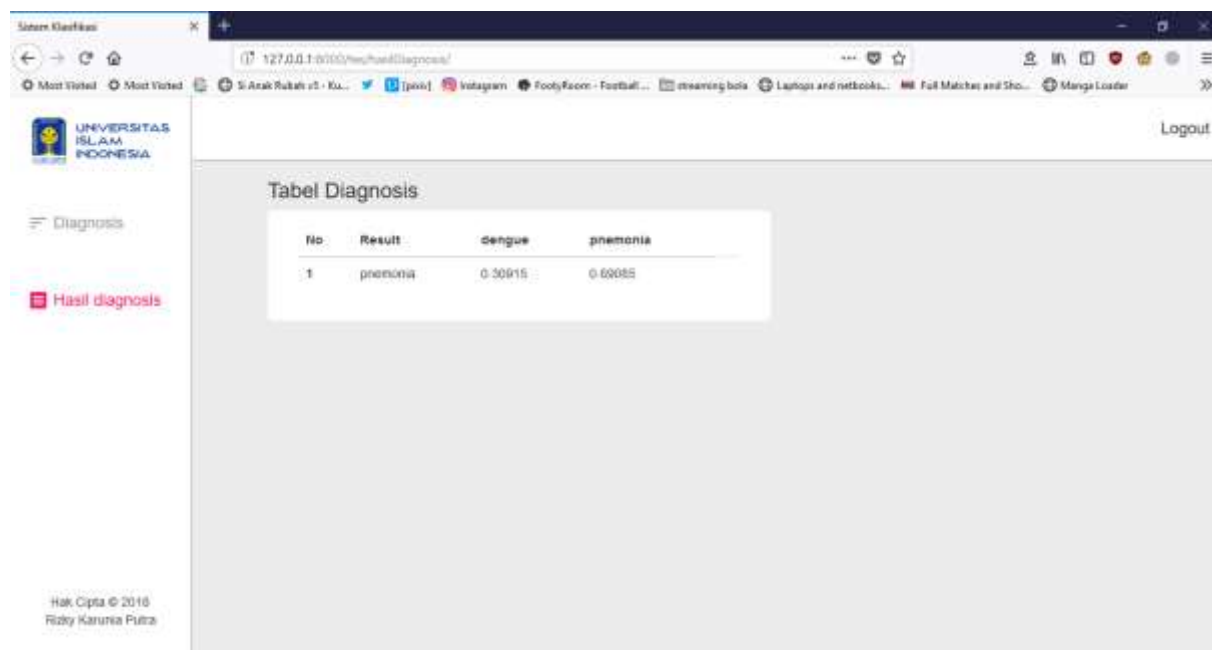
Gambar 4. 6 Halaman Ganti Parameter

Pada *user* dokter, sistem akan mengarahkan ke halaman diagnosis ketika dokter berhasil login ke dalam sistem. Terdapat 2 halaman yang bisa dokter lakukan terhadap sistem. Halaman pertama adalah halaman diagnosis yang mendiagnosis penyakit pasien berdasarkan gejala yang diberikan. Halaman ini terdapat *form* yang berisikan *inputan* frekuensi nadi, suhu, frekuensi nafas, nyeri, riwayat penyakit positif, pilihan kelas penyakit serta tombol start untuk memulai proses diagnosis penyakit. Tampilan dari halaman ini dapat dilihat pada Gambar 4. 7.



Gambar 4. 7 Halaman Diagnosis

Halaman kedua adalah halaman hasil diagnosis yang berfungsi untuk melihat hasil diagnosis penyakit dari gejala yang diberikan. Halaman ini terdiri dari tabel diagnosis beserta nilai himpunan fuzzy dari pilihan penyakit. Tampilan dari halaman ini dapat dilihat pada Gambar 4. 8.



Gambar 4. 8 Halaman Hasil Diagnosis

4.2 Pembahasan

4.2.1 Data Pengujian

Data yang digunakan adalah data rekam medis yang didapatkan dari instansi Rumah Sakit Islam Banjarmasin. Data rekam medis yang digunakan adalah data rekam medis dari 8 jenis penyakit anak yang merupakan hasil rekomendasi dari pihak instansi. Daftar penyakit dapat dilihat pada Tabel 4. 1.

Tabel 4. 1 Daftar Penyakit

No	Penyakit
1	Demam Berdarah
2	Demam Tifoid
3	Campak
4	Farangitis
5	Bronkitis
6	Pneumonia
7	Diare
8	Gastroenteritis

Dari data rekam medis Rumah Sakit Islam Banjarmasin, berhasil dihimpun sampel data sebanyak 106 data. Alokasi dari 106 data sampel adalah sebagai berikut:

1. Penyakit Bronkitis terkumpul sebanyak 9 dokumen.
2. Penyakit Campak terkumpul sebanyak 10 dokumen.
3. Penyakit Demam Berdarah Dengue sebanyak 18 dokumen.
4. Penyakit Demam Tifoid sebanyak 17 dokumen.
5. Penyakit Diare sebanyak 13 dokumen.
6. Penyakit Faringitis sebanyak 10 dokumen.
7. Penyakit Gastrosiritis sebanyak 17 dokumen.
8. Penyakit Pnemonia sebanyak 12 dokumen.

Dari dokumen rekam medis yang dilakukan pembersihan data dan pemilihan atribut yang dipakai kemudian dimasukkan kedalam file csv. Atribut yang dipakai adalah frekuensi nadi, frekuensi nafas, suhu, nyeri, berat badan, tinggi badan, dan riwayat penyakit positif. Atribut tersebut dipilih dikarenakan 106 sampel data memiliki nilai dari atribut tersebut. Semua dokumen dijadikan satu menjadi tabel dataset yang terurut dalam satu file csv. Daftar data dari dataset dapat dilihat pada LAMPIRAN A.

4.2.2 Validasi Nilai

Validasi dilakukan untuk mengetahui hasil pada Gambar 4. 8 sesuai dengan perhitungan manual. Data uji pada Gambar 4. 7 dapat dilihat pada Tabel 4. 2.

Tabel 4. 2 Data Uji Diagnosis

Frekuensi Nadi	Suhu	Frekuensi Nafas	Nyeri	Berat Badan	Tinggi Badan	Riwayat Penyakit Positif	Pilihan Kelas Penyakit
116	38.1	28	Tidak	25	126	Demam 3 hari, Batuk	Dengue, Pnemonia

Dari data uji pada Tabel 4. 2 dan data latih pada LAMPIRAN A, dilakukan pembacaan dataset menggunakan kode program pada Gambar 4. 9 menghasilkan data latih yang terfilter dan data uji seperti pada Gambar 4. 10.

```

with open(filename, "rt") as csvfile:
    lines = csv.reader(csvfile)
    dataset = list(lines)
    print("Banyaknya Dataset : ", len(dataset))
    temp = []
    for i in range(len(dataset)):
        for j in range(len(t)):
            if dataset[i][1] == t[j]:
                temp.append(dataset[i])
    random.shuffle(temp)
    for x in range(len(temp)):
        if x < len(temp)*split:
            trainingSet.append(temp[x])
        else:
            testSet.append(temp[x])

```

Gambar 4. 9 Kode Membaca Data

```

Banyaknya Dataset : 106
Banyaknya Dataset Yang Terfilter : 30
Banyaknya Data Latih : 24
Banyaknya Data Uji : 6
Data Latih :
[['110_38.7_24_tidak_42.0_160.0_demam 5 hari_pusing_nyeri kepala_mimisan_nyeri
perut_lemas', 'dengue'], ['120_36.5_32_tidak_11.0_98.0_sesak_nafas_batuk_pilek_demam',
'pnemonia'], ['100_36.7_24_tidak_19.0_123.0_demam 4 hari_mimisan_batuk_makan
sedikit_bintik_merah', 'dengue'], ['111_37.0_30_ya_55.0_155.0_batuk_batuk_darah_demam
naik_turun_demam 14 hari_muntah_pusing_sesak_nafas', 'pnemonia'],
['120_38.8_26_ya_14.0_101.0_demam naik_turun_demam 4 hari_makan
sedikit_lemas_mual_muntah_BAK_normal_mimisan', 'dengue'],
['132_35.6_26_tidak_9.0_70.0_demam 6 hari_demam naik_turun_lemas_makan
sedikit_muntah_BAB_cair_bintik_merah', 'dengue'],
['128_37.0_60_tidak_4.5_54.0_batuk_sesak_nafas_BAB_cair', 'pnemonia'],
['96_34.4_28_tidak_43.0_151.0_demam 4 hari_mual_batuk', 'dengue'],
['105_37.0_22_tidak_53.0_150.0_muntah_demam_nyeri_ulu_hati_pusing', 'dengue'],
['100_40.1_24_ya_83.0_178.0_demam 5 hari_mual_muntah_batuk_nyeri_tenggorokan_BAB
cair', 'dengue'], ['132_38.7_30_tidak_15.0_98.0_demam 5 hari_demam naik
turun_batuk_pilek_makan_sedikit_bibir_kering', 'pnemonia'],
['112_36.6_28_tidak_38.0_127.0_demam 4 hari_mimisan', 'dengue'],
['110_37.5_26_ya_15.0_98.0_demam 14 hari_batuk_pilek_mimisan_nyeri
tenggorokan_muntah', 'pnemonia'], ['112_38.1_28_tidak_15.0_99.0_demam 6 hari_demam
naik_turun_bintik_merah', 'dengue'], ['116_38.1_28_tidak_25.0_126.0_demam 3
hari_batuk_pilek_mimisan_menggigil', 'dengue'], ['100_39.7_34_ya_43.0_155.0_demam 3
hari_demam naik_turun_mual_pusing_BAK_normal_BAB_normal', 'dengue'],
['96_38.5_26_ya_73.0_165.0_demam naik_turun_demam 4 hari_muntah_nyeri_ulu
hati_pusing_mual_lemas', 'dengue'], ['108_37.3_26_tidak_34.0_135.0_demam 5 hari_demam
daik_turun_nyeri_perut', 'dengue'],
['146_37.4_30_tidak_4.1_50.0_batuk_pilek_demam_sesak_nafas', 'pnemonia'],
['160_38.3_34_ya_7.5_69.0_demam 2 hari_batuk_pilek_muntah', 'pnemonia'],
['122_36.0_24_tidak_19.0_107.0_demam 6 hari_mimisan', 'dengue'],
['120_38.4_36_tidak_61.0_165.0_sesak_nafas_batuk_demam_riwayat_tb', 'pnemonia'],
['120_37.0_36_tidak_7.3_69.0_demam 2 hari_batuk_pilek', 'pnemonia'],
['124_38.7_26_tidak_45.0_148.0_demam 2 hari_mual_muntah 4 hari_batuk_pilek_makan
sedikit_nyeri_perut', 'dengue'], ['132_39.6_40_tidak_8.0_74.0_sesak_nafas_batuk_demam
3 hari_demam naik_turun', 'pnemonia'], ['82_38.5_28_tidak_14.0_94.0_demam 1
hari_kejang', 'dengue'], ['120_37.0_24_tidak_6.0_69.0_batuk_pilek_sesak_nafas',
'pnemonia'], ['124_30.0_36_tidak_5.2_50.0_batuk_pilek_sakit_mata_BAB_cair_sejak_15
hari_sesak_nafas', 'pnemonia'], ['127_37.8_30_ya_25.0_127.0_demam naik_turun_demam 5
hari_nyeri_perut_muntah_makan_sedikit_BAB_cair', 'dengue'],
['84_38.0_24_tidak_16.0_99.0_demam 4 hari_bintik_merah', 'dengue']]
Data Uji :
[['116_38.1_28_tidak_25_126_demam 3 hari_batuk']]

```

Gambar 4. 10 Hasil Membaca Data

Kemudian dilakukan proses *Text Preprocessing* yaitu *tokenization* untuk membuat *token* dari data latih dan uji. *Tokenization* dilakukan dengan cara memisahkan karakter “_” menjadi

token. Proses *tokenization* pada Python dilakukan dengan kode seperti pada Gambar 4. 11 yang menghasilkan hasil *tokenization* pada Gambar 4. 12.

```
for i in range(len(trainingSet)):
    sentence.append(re.split(r'_', str(trainingSet[i][0])))
sentence.append(re.split(r'_', str(testingSet[index][0])))
```

Gambar 4. 11 Kode *Tokenization*

Kalimat *Tokenization* :

```
[[ '100', '39.7', '34', 'ya', '43.0', '155.0', 'demam 3 hari', 'demam naik turun',
'mual', 'pusing', 'BAK normal', 'BAB normal'], [ '108', '37.3', '26', 'tidak',
'34.0', '135.0', 'demam 5 hari', 'demam daik turun', 'nyeri perut'], [ '96',
'34.4', '28', 'tidak', '43.0', '151.0', 'demam 4 hari', 'mual', 'batuk'], [ '120',
'37.0', '24', 'tidak', '6.0', '69.0', 'batuk', 'pilek', 'sesak nafas'], [ '120',
'37.0', '36', 'tidak', '7.3', '69.0', 'demam 2 hari', 'batuk', 'pilek'], [ '82',
'38.5', '28', 'tidak', '14.0', '94.0', 'demam 1 hari', 'kejang'], [ '124', '38.7',
'26', 'tidak', '45.0', '148.0', 'demam 2 hari', 'mual', 'muntah 4 hari', 'batuk',
'pilek', 'makan sedikit', 'nyeri perut'], [ '120', '38.8', '26', 'ya', '14.0',
'101.0', 'demam naik turun', 'demam 4 hari', 'makan sedikit', 'lemas', 'mual',
'muntah', 'BAK normal', 'mimisan'], [ '84', '38.0', '24', 'tidak', '16.0', '99.0',
'demam 4 hari', 'bintik merah'], [ '96', '38.5', '26', 'ya', '73.0', '165.0',
'demam naik turun', 'demam 4 hari', 'muntah', 'nyeri ulu hati', 'pusing', 'mual',
'lemas'], [ '127', '37.8', '30', 'ya', '25.0', '127.0', 'demam naik turun', 'demam
5 hari', 'nyeri perut', 'muntah', 'makan sedikit', 'BAB cair'], [ '122', '36.0',
'24', 'tidak', '19.0', '107.0', 'demam 6 hari', 'mimisan'], [ '112', '38.1', '28',
'tidak', '15.0', '99.0', 'demam 6 hari', 'demam naik turun', 'bintik merah'],
[ '110', '38.7', '24', 'tidak', '42.0', '160.0', 'demam 5 hari', 'pusing', 'nyeri
kepala', 'mimisan', 'nyeri perut', 'lemas'], [ '120', '36.5', '32', 'tidak',
'11.0', '98.0', 'sesak nafas', 'batuk', 'pilek', 'demam'], [ '111', '37.0', '30',
'ya', '55.0', '155.0', 'batuk', 'batuk darah', 'demam naik turun', 'demam 14
hari', 'muntah', 'pusing', 'sesak nafas'], [ '132', '39.6', '40', 'tidak', '8.0',
'74.0', 'sesak nafas', 'batuk', 'demam 3 hari', 'demam naik turun'], [ '100',
'40.1', '24', 'ya', '83.0', '178.0', 'demam 5 hari', 'mual', 'muntah', 'batuk',
'nyeri tenggorokan', 'BAB cair'], [ '128', '37.0', '60', 'tidak', '4.5', '54.0',
'batuk', 'sesak nafas', 'BAB cair'], [ '105', '37.0', '22', 'tidak', '53.0',
'150.0', 'muntah', 'demam', 'nyeri ulu hati', 'pusing'], [ '116', '38.1', '28',
'tidak', '25.0', '126.0', 'demam 3 hari', 'batuk', 'pilek', 'mimisan',
'menggigil'], [ '112', '36.6', '28', 'tidak', '38.0', '127.0', 'demam 4 hari',
'mimisan'], [ '120', '38.4', '36', 'tidak', '61.0', '165.0', 'sesak nafas',
'batuk', 'demam', 'riwayat tb'], [ '110', '37.5', '26', 'ya', '15.0', '98.0',
'demam 14 hari', 'batuk', 'pilek', 'mimisan', 'nyeri tenggorokan', 'muntah'],
[ '132', '38.7', '30', 'tidak', '15.0', '98.0', 'demam 5 hari', 'demam naik turun',
'batuk', 'pilek', 'makan sedikit', 'bibir kering'], [ '160', '38.3', '34', 'ya',
'7.5', '69.0', 'demam 2 hari', 'batuk', 'pilek', 'muntah'], [ '100', '36.7', '24',
'tidak', '19.0', '123.0', 'demam 4 hari', 'mimisan', 'batuk', 'makan sedikit',
'bintik merah'], [ '132', '35.6', '26', 'tidak', '9.0', '70.0', 'demam 6 hari',
'demam naik turun', 'lemas', 'makan sedikit', 'muntah', 'BAB cair', 'bintik
merah'], [ '124', '30.0', '36', 'tidak', '5.2', '50.0', 'batuk', 'pilek', 'sakit
mata', 'BAB cair sejak 15 hari', 'sesak nafas'], [ '146', '37.4', '30', 'tidak',
'4.1', '50.0', 'batuk', 'pilek', 'demam', 'sesak nafas'], [ '116', '38.1', '28',
'tidak', '25', '126', 'demam 3 hari', 'batuk']]
```

Gambar 4. 12 Hasil *Tokenization*

Langkah selanjutnya adalah membuat matrik TF-IDF dari hasil *tokenization*. Dalam membuat matrik TF-IDF, perlu tau dahulu nilai matrik TF dan IDF. Pertama adalah membuat matrik TF menggunakan Python dapat dilakukan seperti pada Gambar 4. 13. Hasil dari kode

tersebut akan menghasilkan nilai frekuensi dari hasil *tokenization* terhadap atribut dari *token* yang sudah dihilangkan duplikasinya. Hasil tersebut dapat dilihat pada Gambar 4. 14.

```
matrix = numpy.zeros(shape=(len(attribute), len(sentence)))
for i in range(len(sentence)):
    for j in range(len(sentence[i])):
        for k in range(len(attribute)):
            if sentence[i][j] == attribute[k]:
                matrix[k][i] = 1
```

Gambar 4. 13 Kode Matrik TF

```
Matrix TF | :
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [1. 1. 1. ... 0. 1. 1.]
 [0. 0. 0. ... 1. 0. 0.]]
```

Gambar 4. 14 Hasil Matrik TF

Kemudian membuat nilai IDF setiap *token* dengan rumus $\log(\text{panjang } tokenization / \text{jumlah nilai } 1 \text{ pada tiap token})$. Kode program dan hasil dari proses ini dapat dilihat pada Gambar 4. 15 dan Gambar 4. 16. Selanjutnya melakukan perkalian skalar antara matrik TF dan IDF untuk mendapatkan nilai dari matrik TF-IDF. Kode program dan hasil dari proses ini dapat dilihat pada Gambar 4. 17 dan Gambar 4. 18.

```
for i in range(len(attribute)):
    df.append(matrix[i][:].tolist().count(1))
    try:
        idf.append(math.log10(len(sentence) /
(matrix[i][:].tolist().count(1))))
    except:
        idf.append(0)
```

Gambar 4. 15 Kode IDF

```

IDF :
[1.0142404391146103, 1.4913616938342726, 1.4913616938342726, 1.4913616938342726,
1.4913616938342726, 1.4913616938342726, 1.1903316981702914, 1.4913616938342726,
1.1903316981702914, 1.1903316981702914, 0.7923916894982539, 1.4913616938342726,
1.4913616938342726, 1.1903316981702914, 1.4913616938342726, 1.4913616938342726,
1.4913616938342726, 1.1903316981702914, 1.4913616938342726, 1.0142404391146103,
1.4913616938342726, 1.1903316981702914, 1.4913616938342726, 1.4913616938342726,
1.0142404391146103, 1.4913616938342726, 1.4913616938342726, 1.1903316981702914,
1.4913616938342726, 1.4913616938342726, 1.4913616938342726, 1.1903316981702914,
1.4913616938342726, 1.1903316981702914, 0.7132104434506291, 0.7132104434506291,
0.8893017025063102, 1.4913616938342726, 1.4913616938342726, 1.1903316981702914,
1.4913616938342726, 1.4913616938342726, 1.4913616938342726, 1.0142404391146103,
1.4913616938342726, 1.4913616938342726, 1.4913616938342726, 1.4913616938342726,
0.7923916894982539, 1.4913616938342726, 1.4913616938342726, 1.4913616938342726,
1.4913616938342726, 1.1903316981702914, 1.0142404391146103, 1.4913616938342726,
1.4913616938342726, 1.1903316981702914, 1.0142404391146103, 1.4913616938342726,
1.4913616938342726, 1.4913616938342726, 1.4913616938342726, 1.4913616938342726,
1.4913616938342726, 1.0142404391146103, 1.4913616938342726, 1.4913616938342726,
1.4913616938342726, 1.4913616938342726, 1.4913616938342726, 1.4913616938342726,
1.4913616938342726, 1.0142404391146103, 1.4913616938342726, 1.4913616938342726,
1.4913616938342726, 1.4913616938342726, 1.4913616938342726, 1.4913616938342726,
1.4913616938342726, 1.1903316981702914, 1.0142404391146103, 1.1903316981702914,
0.8893017025063102, 1.4913616938342726, 1.4913616938342726, 1.1903316981702914,
0.23608918873096663, 1.4913616938342726, 1.4913616938342726, 0.8893017025063102,
0.8893017025063102, 1.4913616938342726, 1.1903316981702914, 1.0142404391146103,
0.8893017025063102, 0.7132104434506291, 0.7923916894982539, 1.0142404391146103,
1.4913616938342726, 0.5371191843949479, 1.4913616938342726, 0.8893017025063102,
0.7132104434506291, 1.4913616938342726, 0.6462636538200158, 0.7132104434506291,
0.5371191843949479, 1.4913616938342726, 1.4913616938342726, 0.8893017025063102,
1.1903316981702914, 1.1903316981702914, 0.4913616938342727, 0.7923916894982539,
1.4913616938342726, 1.4913616938342726, 0.5882717068423291, 0.1296338578166798,
0.5882717068423291]

```

Gambar 4. 16 Hasil IDF

```

df = []
idf = []
for i in range(len(attribute)):
    df.append(matrix[i][:].tolist().count(1))
    idf.append(math.log10(len(sentence) /
(matrix[i][:].tolist().count(1))))

tfidf = []
for i in range(len(attribute)):
    tfidf.append((idf[i] * matrix[i][:]).tolist())
tfidf = numpy.array(tfidf)

```

Gambar 4. 17 Kode Matrik TF-IDF

```

Nilai TFIDF :
[[0.      0.      0.      ... 0.      0.      0.      ]
 [0.      0.      0.      ... 0.      0.      0.      ]
 [0.      0.      0.      ... 0.      0.      0.      ]
 ...
 [0.      0.      0.      ... 0.      0.      0.      ]
 [0.12963386 0.12963386 0.12963386 ... 0.      0.12963386 0.12963386]
 [0.      0.      0.      ... 0.58827171 0.      0.      ]]

```

Gambar 4. 18 Hasil Matrik TF-IDF

Setelah matrik TF-IDF didapatkan, maka selanjutnya menghitung jarak antara data uji dengan data latih menggunakan persamaan (2.3). Sebelum masuk ke persamaan (2.3), terlebih dahulu mencari nilai perkalian TF-IDF antara data uji dengan data latih dan nilai panjang vektor. Kode program dan hasil untuk proses mencari nilai perkalian TF-IDF antara data uji dengan data latih dapat dilihat pada Gambar 4. 19 dan Gambar 4. 20. Sedangkan untuk kode program dan hasil dari mencari nilai panjang vektor dapat dilihat pada Gambar 4. 21 dan Gambar 4. 22.

```
tr_te = []
for i in range(len(sentence)-1):
    x = 0
    for j in range(len(attribute)):
        x += tfidf[j][len(sentence)-1] * tfidf[j][i]
    tr_te.append(x)
```

Gambar 4. 19 Kode Perkalian TF-IDF

Perkalian TFIDF :

```
[0.07254304212808113, 0.5254740737394782, 0.07254304212808113, 3.8176429168601134,
0.016804937092435156, 0.016804937092435156, 0.863400560208703, 0.016804937092435156,
0.5254740737394782, 0.7908575180806219, 0.07254304212808113, 0.0, 0.0,
0.05573810503564598, 0.05573810503564598, 0.016804937092435156, 0.07254304212808113,
0.05573810503564598, 0.07254304212808113, 0.5812121787751242, 0.07254304212808113,
0.07254304212808113, 0.016804937092435156, 0.05573810503564598, 0.07254304212808113,
0.07254304212808113, 0.07254304212808113, 0.016804937092435156, 0.0,
1.5541577420748758]
```

Gambar 4. 20 Hasil Perkalian TF-IDF

```
p_vektor = []
for i in range(len(sentence)):
    x = 0
    for j in range(len(attribute)):
        x += pow(tfidf[j][i], 2)
    p_vektor.append(math.sqrt(x))
```

Gambar 4. 21 Kode Panjang Vektor

Panjang Vektor :

```
[3.574574139413809, 3.5008416108733535, 2.622967640347472, 3.2158894409817678,
3.614555946528861, 3.558856138087607, 3.379834856203081, 3.565607415756544,
2.8154268375289844, 3.702233534275435, 3.1272908145673886, 3.4226133911160024,
3.4750872663970993, 3.2233920920250125, 3.4992043996922484, 3.017797083217747,
2.3827137140994132, 3.3284337205911236, 3.663215054618717, 2.9800417713648932,
3.2142070018444455, 3.276276477681746, 3.5914714706975426, 3.588557405109643,
3.309894420729888, 3.1146359825949896, 2.978050032157325, 3.171936807299498,
3.4292163920757655, 2.7431111385449416, 2.8750586638419353]
```

Gambar 4. 22 Hasil Panjang Vektor

Setelah didapat nilai perkalian TF-IDF dan panjang vektor, maka langsung saja memasukkannya kedalam persamaan (2.3). Kode program dan hasil untuk proses mencari jarak menggunakan Cosine Similarity dapat dilihat pada Gambar 4. 23 dan Gambar 4. 24.

```

for i in range(len(sentence)-1):
    x = 0
    x = tr_te[i] / (p_vektor[len(sentence)-1] * p_vektor[i])
    cosine.append(x)
jarak=[]
for i in range(len(cosine)):
    nested = []
    nested.append(cosine[i])
    nested.append(trainingSet[i][1])
    jarak.append(nested)

jarak.sort(key=itemgetter(0), reverse=True)

```

Gambar 4. 23 Kode Menghitung Jarak

```

Nilai Cosine Similarity :
[[0.4129024728713262, 'dengue'], [0.19706295096941623, 'dengue'],
[0.08885259883967137, 'pnemonia'], [0.07429981400355548, 'dengue'],
[0.06783684537872257, 'dengue'], [0.06491728324973332, 'dengue'],
[0.05220740661705471, 'dengue'], [0.01058954228485517, 'pnemonia'],
[0.009619580218998627, 'pnemonia'], [0.008472607026646185, 'pnemonia'],
[0.008101058283908313, 'dengue'], [0.008068276704752346, 'pnemonia'],
[0.00785010038671466, 'pnemonia'], [0.007701379233419824, 'pnemonia'],
[0.007623157847613761, 'pnemonia'], [0.007058700321794918, 'dengue'],
[0.00688789690257109, 'pnemonia'], [0.006014400616483972, 'pnemonia'],
[0.005824592890496778, 'pnemonia'], [0.005540336936919102, 'dengue'],
[0.005402385749170554, 'pnemonia'], [0.0019368685440185063, 'dengue'],
[0.0018427467499554616, 'dengue'], [0.0016424030688287875, 'dengue'],
[0.0016392932707301588, 'dengue'], [0.0016274878668547548, 'dengue'],
[0.0016170938641379756, 'dengue'], [0.0, 'dengue'], [0.0, 'dengue'], [0.0, 'dengue']]

```

Gambar 4. 24 Hasil Menghitung Jarak

Kemudian memilih tetangga sebanyak K dari hasil perhitungan jarak. Kode program dan hasil dapat dilihat pada Gambar 4. 25 dan Gambar 4. 26.

```

for i in range(nilaiK):
    x = []
    x.append(jarak[i][0])
    x.append(jarak[i][1])
    neighbor.append(x)

```

Gambar 4. 25 Kode Mencari Tetangga K

```

Tetangga :
[[0.4129024728713262, 'dengue'], [0.19706295096941623, 'dengue'],
[0.08885259883967137, 'pnemonia'], [0.07429981400355548, 'dengue'],
[0.06783684537872257, 'dengue'], [0.06491728324973332, 'dengue'],
[0.05220740661705471, 'dengue'], [0.01058954228485517, 'pnemonia'],
[0.009619580218998627, 'pnemonia'], [0.008472607026646185, 'pnemonia']]

```

Gambar 4. 26 Hasil Mencari Tetangga K

Langkah selanjutnya mencari nilai keanggotaan berdasarkan hasil voting tiap penyakit pada K tetangga menggunakan persamaan (2.5). Kode program dan hasil dapat dilihat pada Gambar 4. 27 dan Gambar 4. 28.

```

for i in range(nilaiK):
    y = neighbor[i][1]
    membership = dict()
    for c in label:
        try:
            uci = 0.49 * (vote[c] / nilaiK)
            if c == y:
                uci += 0.51
            membership[c] = uci
        except:
            membership[c] = 0

memberships.append(membership)

```

Gambar 4. 27 Kode Menghitung Nilai Keanggotaan

```

Nilai Membership :
[{'dengue': 0.804, 'pnemonia': 0.196}, {'dengue': 0.804, 'pnemonia': 0.196},
{'dengue': 0.294, 'pnemonia': 0.706}, {'dengue': 0.804, 'pnemonia': 0.196}, {'dengue':
0.804, 'pnemonia': 0.196}, {'dengue': 0.804, 'pnemonia': 0.196}, {'dengue': 0.804,
'pnemonia': 0.196}, {'dengue': 0.294, 'pnemonia': 0.706}, {'dengue': 0.294,
'pnemonia': 0.706}, {'dengue': 0.294, 'pnemonia': 0.706}]

```

Gambar 4. 28 Hasil Menghitung Nilai Keanggotaan

Kemudian menghitung nilai bobot dari K tetangga menggunakan rumus $1 / \text{nilai tetangga pangkat } 2 / (M - 1)$. Kode program dan hasil dapat dilihat pada Gambar 4. 29 dan Gambar 4. 30.

```

pangkat = 2 / (m - 1)
for i in range(len(neighbor)):
    x = 0
    if pow(neighbor[i][0], pangkat) != 0:
        x = 1 / pow(neighbor[i][0], pangkat)
    w.append(x)

```

Gambar 4. 29 Kode Menghitung Bobot

```

Nilai Bobot : [5.86549990725677, 25.750759103976726, 126.66590490576125,
181.14424745857139, 217.30449635369416, 237.2899401897892, 366.88990250987547,
8917.551420867487, 10806.567207417896, 13930.473319819836]

```

Gambar 4. 30 Hasil Menghitung Bobot

Langkah selanjutnya adalah menghitung nilai himpunan fuzzy berdasarkan nilai keanggotaan dan nilai bobot menggunakan persamaan (2.4). Kode program dan hasil perhitungan himpunan fuzzy dapat dilihat pada Gambar 4. 31 dan Gambar 4. 32. Langkah terakhir adalah mencari nilai maksimum dalam himpunan fuzzy untuk mendapatkan hasil klasifikasi. Kode program dan hasil prediksi dapat dilihat pada Gambar 4. 33 dan Gambar 4. 34.

```

matrix = numpy.zeros(shape=(nilaiK, len(label)))
for i in range(nilaiK):
    m = list(memberships[i].values())
    keys = list(memberships[i].keys())
    for j in range(len(m)):
        if sum(w) != 0:
            matrix[i][j] = (w[i] * m[j]) / sum(w)
        else:
            matrix[i][j] = 0
tot = sum(matrix)

```

Gambar 4. 31 Kode Menghitung Himpunan Fuzzy

```

Matrix Fuzzy :
[[1.35452932e-04 3.30208641e-05]
 [5.94666419e-04 1.44968431e-04]
 [1.06963201e-03 2.56857210e-03]
 [4.18319322e-03 1.01978342e-03]
 [5.01824766e-03 1.22335391e-03]
 [5.47977473e-03 1.33586548e-03]
 [8.47264749e-03 2.06547128e-03]
 [7.53043878e-02 1.80832986e-01]
 [9.12562081e-02 2.19139058e-01]
 [1.17636077e-01 2.82486634e-01]]

```

```

Total Fuzzy :
[0.30915029 0.69084971]

```

Gambar 4. 32 Hasil Himpunan Fuzzy

```

maks = max(tot)
indeces = 0
for i in range(len(label)):
    if maks == tot[i]:
        indeces = i
        break
result = label[indeces]

```

Gambar 4. 33 Kode Prediksi

```

Nilai Maksimum Fuzzy :
0.6908497126184243

```

```

Hasil Klasifikasi : pnemonia

```

Gambar 4. 34 Hasil Prediksi

Dari hasil perhitungan diatas, dapat dilihat hasil diagnosis pada perangkat lunak dan hasil perhitungan secara manual sudah sesuai. Hal ini membuktikan bahwa tidak ada kesalahan dalam model yang dibangun.

4.2.3 Pengujian Akurasi Sistem

Pengujian akurasi sistem klasifikasi menggunakan Confusion Matrik. Pengujian akurasi didapatkan proses klasifikasi menggunakan data pada LAMPIRAN A yang dibagi dengan perbandingan 80:20 dimana 80% untuk data latih dan 20% untuk data uji. Perhitungan

Confusion Matrik dapat dilihat pada Tabel 2. 2. Hasil klasifikasi pada Gambar 4. 5 dapat dilihat pada Tabel 4. 3.

Tabel 4. 3 Hasil Klasifikasi Data Testing

Hasil Klasifikasi	Data Testing	Status	Pnemonia	Dengue
Pnemonia	Pnemonia	Benar	0.65716	0.34284
Dengue	Dengue	Benar	0.24917	0.75083
Dengue	Dengue	Benar	0.06878	0.93122
Pnemonia	Pnemonia	Benar	0.77415	0.22585
Dengue	Pnemonia	Salah	0.25773	0.74227
Dengue	Dengue	Benar	0.19385	0.80615

Berdasarkan hasil klasifikasi pada Tabel 4. 3, kita mendapatkan informasi yang diperlukan untuk mengukur tingkat akurasi. Informasi yang didapatkan sebagai berikut:

1. Data uji dengan kelas Dengue yang terklasifikasi Dengue sebanyak 3.
2. Data uji dengan kelas Dengue yang terklasifikasi Pnemonia sebanyak 1.
3. Data uji dengan kelas Pnemonia yang terklasifikasi Dengue sebanyak 0.
4. Data uji dengan kelas Pnemonia yang terklasifikasi Pnemonia sebanyak 2.

Berdasarkan informasi tersebut, maka Confusion Matrik dapat dibuat seperti pada Tabel 4. 4 berikut.

Tabel 4. 4 Confusion Matrik

Kelas	Terklasifikasi Dengue	Terklasifikasi Pnemonia
Dengue	3	1
Pnemonia	0	2

Setelah confusion matrix sudah didefinisikan, maka selanjutnya adalah menghitung nilai precision, recall, dan akurasi. Dengan menggunakan persamaan (2.6) dan (2.7), nilai precision dan recall dapat dihitung. Hasil perhitungan precision dan recall dapat dilihat pada Tabel 4. 5.

Tabel 4. 5 Tabel Precision Dan Recall

Kelas	Precision	Recall
Dengue	0.75	1
Campak	1	0.67
Rata-rata	0.875	0.833

Berdasarkan Tabel 4. 5, nilai rata-rata precision dari hasil klasifikasi adalah 0.875. Sedangkan untuk nilai rata-rata recall nya yaitu 0.833.

Sistem Klasifikasi Penyakit Anak Dengan Gejala Demam Pada Anak Menggunakan Metode *Fuzzy K-Nearest Neighbor* dengan memiliki pembagian data latih sebanyak 80%, nilai K sebesar 10, dan nilai M sebesar 2, serta pemilihan kelas ada 2 yaitu dengue dan pnemonia, menghasilkan tingkat akurasi sebesar 0.833. Dari hasil perhitungan akurasi, perlu dilakukan evaluasi menggunakan AUC ROC. ROC (*Receiver Operating Characteristic*) bertujuan untuk mengetahui seberapa baiknya model yang telah dibuat mampu membedakan antara penyakit satu dengan lainnya. Metode untuk proses evaluasi dengan cara membandingkan nilai akurasi terhadap nilai AUC (*area under curve*) dari ROC dengan skala seperti pada Tabel 2. 3.

Dari Tabel 2. 3, nilai akurasi Sistem Klasifikasi Penyakit Anak Dengan Gejala Demam Pada Anak Menggunakan Metode *Fuzzy K-Nearest Neighbor* sebesar 0.833 berada diantara nilai 0.8 – 0.9 yang artinya bahwa sistem ini memiliki akurasi yang baik.

4.3 Analisis Sistem

Pada tahap analisis sistem merupakan tahapan untuk menilai kelebihan dan kekurangan yang dimiliki sistem. Berikut merupakan kelebihan dan kekurangan dari sistem yang telah dibuat.

4.3.1 Kelebihan

Adapun kelebihan dari sistem yang telah dibuat sebagai berikut:

- a. Sistem dapat melakukan klasifikasi penyakit anak dengan gejala demam berdasarkan rekam medis berbentuk teks dengan akurasi baik.

4.3.2 Kekurangan

- a. Pengguna tidak dapat memasukkan gejala lain selain gejala yang sudah ditentukan.
- b. Akurasi yang baik hanya bisa terjadi ketika jumlah penyakit yang dipilih kurang dari sama dengan 3.
- c. Penulisan riwayat penyakit positif harus sama persis dengan dataset yang dimiliki sistem.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengembangan dan pengujian yang sudah dilakukan pada Sistem Klasifikasi Penyakit Anak Dengan Gejala Demam Pada Anak Menggunakan Metode *Fuzzy K-Nearest Neighbor*, maka didapatkan kesimpulan bahwa penelitian ini yaitu:

- a. Sistem ini dapat digunakan oleh dokter umum atau koas atau asisten dokter sebagai alat bantu untuk melakukan diagnosis sementara pada penyakit anak dengan gejala demam.
- b. Akurasi yang dihasilkan sistem ini adalah 0.833 yang mana nilai akurasi ini sudah berada rentang 0.8 – 0.9 yang dapat ditarik kesimpulan bahwa sistem klasifikasi ini baik.

5.2 Saran

Setelah sebelumnya telah dilakukan analisis terhadap sistem, maka masih terdapat kekurangan dari sistem ini. Maka dari itu dapat diberikan saran kepada penelitian selanjutnya untuk dapat menghilangkan atau menutupi kekurangan yang dimiliki sistem ini. Adapun saran yang diberikan sebagai berikut.

- a. Berdasarkan penelitian Siringoringo & Perangin-angin (2017), hasil akurasi yang didapatkan terjadi peningkatan sehingga perlu dilakukan hibridisasi menggunakan metode Modified Particle Swarm Optimization untuk menentukan nilai K dan M terbaik.

DAFTAR PUSTAKA

- Bungin, B. (2007). *Penelitian Kualitatif: Komunikasi, Ekonomi, Kebijakan Publik, Dan Ilmu Sosial Lainnya*. Jakarta: Kencana.
- El-Radhi, A. S., Carroll, J., & Klein, N. (2009). *Clinical manual of fever in children. Clinical Manual of Fever in Children*. <https://doi.org/10.1007/978-3-540-78598-9>
- Gorunescu, F. (2011). *Data Mining: Concepts, Models and Techniques*. Berlin: Springer.
- Indonesia. (2008). *Peraturan Menteri Kesehatan Nomor 269 Tahun 2008 Tentang Rekam Medis*. Jakarta: Menteri Kesehatan Republik Indonesia.
- Informatikalogi. (2017). Text Preprocessing. Retrieved from <https://informatikalogi.com/text-preprocessing/>
- Keller, J. M., Gray, M. R., & Givens, James A, J. (1985). A Fuzzy K-Nearest Neighbor Algorithm. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, SMC-16*(4), 580–585.
- Markhan, K. (2014). Simple Guide to Confusion Matrix Terminology. Retrieved from <https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/>
- Massaron, L., & Mueller, J. P. (2015). *Python for Data Science For Dummies*. New Jersey: John Wiley & Sons, inc.
- MathWorks. (2013a). Supervised Learning. Retrieved from <https://www.mathworks.com/discovery/supervised-learning.html>
- MathWorks. (2013b). Unsupervised Learning. Retrieved from <https://www.mathworks.com/discovery/unsupervised-learning.html>
- MathWorks. (2016). What Is Machine Learning? 3 things you need to know. Retrieved from <https://www.mathworks.com/discovery/machine-learning.html>
- Mozilla. (2018). Django introduction. Retrieved from <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction>
- Mulyati, S., Kusumadewi, S., & Rosita, L. (2012). Model Sistem Pendukung Keputusan Untuk Diagnosis Penyakit Anak Dengan Gejala Demam Menggunakan Naive Bayes. In *Seminar Nasional Informatika Medis III (SNIMed III)* (pp. 50–54). Yogyakarta.
- Nugraha, S. D., Regasari, R., Putri, M., & Wihandika, R. C. (2017). Penerapan Fuzzy K-Nearest Neighbor (FK-NN) Dalam Menentukan Status Gizi Balita. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 1(9), 925–932. Retrieved from <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/278>

- Prato, S. (2013). What is Text Mining? Retrieved from <https://ischool.syr.edu/infospace/2013/04/23/what-is-text-mining/>
- Siringoringo, R., & Perangin-angin, R. (2017). Hibridisasi Metode Fuzzy K-Nearest Neighbor Dengan Metode Modified Particle Swarm Optimization Pada Pengklasifikasian Penyakit Tanaman Kedelai. *Sinkron Jurnal & Penelitian Teknik Informatika*, 2(2), 6–12.
- Soedarmo, S. S. P., Garna, H., Hadinegoro, S. R. S., & Satari, H. I. (2010). *Buku Ajar Infeksi & Pediatri Tropis* (2nd ed.). Jakarta: Badan Penerbit IDAI.
- Wafiyah, F., Hidayat, N., & Perdana, R. S. (2017). Implementasi Algoritma Modified K-Nearest Neighbor (MKNN) untuk Klasifikasi Penyakit Demam. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komunikasi*, 1(10), 1210–1219. Retrieved from <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/364>

LAMPIRAN

LAMPIRAN A

Tabel Dataset

100_28.7_24_tidak_7.5_68.0_demam 4 hari_batuk_pilek_makan sulit_minum sulit	bronkitis
114_37.2_30_tidak_10.0_70.0_demam 4 hari_demam naik turun_makan sulit_minum sulit_BAB cair_BAB 4 kali	bronkitis
110_36.0_28_tidak_8.7_75.0_sesak nafas_batuk 2 hari_pilek 2 hari_demam naik turun_asma	bronkitis
122_39.3_32_tidak_9.0_78.0_demam 1 hari_batuk_pilek_kejang	bronkitis
126_39.5_24_tidak_6.0_65.0_batuk 7 hari_sesak nafas_nafas cepat_demam naik turun	bronkitis
120_39.5_24_tidak_10.0_86.0_demam 2 hari_muntah 5 kali_batuk_pilek_makan sulit_kejang	bronkitis
156_37.7_40_tidak_5.2_57.0_batuk 7 hari_pilek 7 hari_demam	bronkitis
160_37.5_34_tidak_2.7_47.0_batuk 2 hari_pilek_sesak nafas_makan sulit_lemas	bronkitis
108_37.0_28_tidak_10.0_88.0_batuk 12 hari_demam 3 hari_muntah_BAK sedikit_BAB normal_mimisan	bronkitis
115_39.4_28_ya_35.0_138.0_badan kemerahan_demam 6 hari_mata merah_demam naik turun_muntah_batuk	campak
128_38.9_24_ya_65.0_161.0_demam 4 hari_demam naik turun_pusing_batuk_pilek_mual_muntah_BAK sedikit_BAB normal	campak
122_39.4_30_tidak_47.0_160.0_demam 3 hari_batuk_pilek_pusing_mata merah	campak
86_38.8_26_ya_48.0_160.0_demam 3 hari_demam naik turun_mual_BAB cair_batuk_pilek_nyeri tenggorokan	campak
130_38.9_28_tidak_28.0_125.0_demam 4 hari_batuk_pilek_mimisan_BAK normal_muntah_pusing_nyeri perut	campak
120_38.1_32_tidak_21.5_35.0_demam 6 hari_demam naik turun_badan kemerahan_muntah_BAB cair	campak
100_36.6_32_tidak_63.0_159.0_bintik merah_demam_nyeri perut	campak
110_38.7_24_ya_14.0_100.0_demam 4 hari_demam naik turun_muntah_badan kemerahan	campak

126_37.5_30_tidak_13.0_88.0_demam 1 hari_badan kemerahan_batuk_pilek_muntah_BAB cair_mata merah	campak
112_38.5_30_tidak_11.0_88.0_demam 5 hari_demam naik turun_badan kemerahan_makan sedikit_minum sedikit_batuk_pilek	campak
96_38.5_26_ya_73.0_165.0_demam naik turun_demam 4 hari_muntah_nyeri ulu hati_pusing_mual_lemas	dengue
122_36.0_24_tidak_19.0_107.0_demam 6 hari_mimisan	dengue
116_38.1_28_tidak_25.0_126.0_demam 3 hari_batuk_pilek_mimisan_menggigil	dengue
96_34.4_28_tidak_43.0_151.0_demam 4 hari_mual_batuk	dengue
82_38.5_28_tidak_14.0_94.0_demam 1 hari_kejang	dengue
100_39.7_34_ya_43.0_155.0_demam 3 hari_demam naik turun_mual_pusing_BAK normal_BAB normal	dengue
127_37.8_30_ya_25.0_127.0_demam naik turun_demam 5 hari_nyeri perut_muntah_makan sedikit_BAB cair	dengue
120_38.8_26_ya_14.0_101.0_demam naik turun_demam 4 hari_makan sedikit_lemas_mual_muntah_BAK normal_mimisan	dengue
100_40.1_24_ya_83.0_178.0_demam 5 hari_mual_muntah_batuk_nyeri tenggorokan_BAB cair	dengue
84_38.0_24_tidak_16.0_99.0_demam 4 hari_bintik merah	dengue
124_38.7_26_tidak_45.0_148.0_demam 2 hari_mual_muntah 4 hari_batuk_pilek_makan sedikit_nyeri perut	dengue
132_35.6_26_tidak_9.0_70.0_demam 6 hari_demam naik turun_lemas_makan sedikit_muntah_BAB cair_bintik merah	dengue
112_36.6_28_tidak_38.0_127.0_demam 4 hari_mimisan	dengue
110_38.7_24_tidak_42.0_160.0_demam 5 hari_pusing_nyeri kepala_mimisan_nyeri perut_lemas	dengue
100_36.7_24_tidak_19.0_123.0_demam 4 hari_mimisan_batuk_makan sedikit_bintik merah	dengue
108_37.3_26_tidak_34.0_135.0_demam 5 hari_demam naik turun_nyeri perut	dengue
112_38.1_28_tidak_15.0_99.0_demam 6 hari_demam naik turun_bintik merah	dengue
105_37.0_22_tidak_53.0_150.0_muntah_demam_nyeri ulu hati_pusing	dengue

88_34.0_24_ya_50.0_165.0_nyeri kepala_mual_muntah_nyeri ulu hati_pusing_demam	diare
128_38.1_28_ya_3.7_32.0_BAB cair sejak 4 hari_BAB cair sebanyak 5 kali_demam 5 hari	diare
120_37.7_30_ya_9.0_83.0_demam 3 hari_BAB cair sejak 3 hari_BAB cair warna kuning_BAB cair lendir_tidak berdarah_BAB tidak ada ampas_muntah_perut kembung_makan sedikit_minum sedikit	diare
155_38.8_32_tidak_11.0_75.0_demam 4 hari_muntah 3 hari_BAB cair sebanyak 5 kali	diare
135_36.3_28_ya_13.7_112.0_BAB cair sejak 1 hari_BAB cair sebanyak 3 kali_muntah_makan sedikit_minum sedikit_lemas	diare
126_39.8_30_tidak_17.0_70.0_kejang_batuk_demam 1 hari_pilek_mual_muntah_BAB cair_ada jatuh_ada terbentur	diare
86_37.2_24_tidak_55.0_152.0_nyeri perut_mual_muntah_BAB cair sebanyak 3 kali_lemas_pusing	diare
100_36.7_24_ya_50.0_150.0_muntah_BAB cair sebanyak 2 kali_mules_nyeri ulu hati_muntah_demam	diare
86_37.0_26_tidak_50.0_165.0_muntah_BAB cair sebanyak 5 kali_BAB ada ampas_demam_lemas	diare
98_37.4_22_ya_48.0_153.0_BAB cair sejak 3 hari_BAB cair berwarna merah_nyeri perut_mual_muntah_lemas_pusing_demam	diare
112_36.8_30_tidak_9.0_75.0_demam_muntah_BAB cair	diare
102_36.3_24_tidak_26.0_123.0_muntah_nyeri perut_pusing_demam	diare
122_36.8_30_tidak_7.0_61.0_muntah_makan sedikit_minum sedikit_muntah isi susu_muntah isi lendir_BAB cair sebanyak 3 kali_demam_batuk_pilek	diare
112_36.2_28_tidak_52.0_135.0_BAB cair sejak 1 hari_BAB cair sebanyak 5 kali_demam_lemas_makan normal_minum normal	gea
128_38.0_28_tidak_5.3_63.0_BAB cair sejak 1 hari_BAB cair sebanyak 5 kali_muntah 3 kali_demam	gea
132_38.0_26_tidak_9.0_82.0_BAB cair sejak 1 hari_BAB cair sebanyak 5 kali_muntah 5 kali_demam 1 hari_kembung	gea
120_37.1_28_tidak_12.0_72.0_BAB cair sejak 4 hari_muntah 1 kali_BAB cair sebanyak 2 kali_muntah tiap makan_muntah tiap minum	gea



128_38.1_32_tidak_9.0_82.0_BAB cair sejak 1 hari_BAB cair sebanyak 10 kali_muntah 3 kali_demam	gea
124_37.8_30_ya_6.2_60.0_muntah 1 hari_muntah 10 kali_BAK sedikit_dehidrasi_BAB cair sebanyak 2 kali_demam_batuk_pilek	gea
111_37.7_34_tidak_10.0_86.0_muntah 3 kali_muntah 1 hari_muntah isi makanan_BAB cair sebanyak 2 kali_BAB cair sejak 1 hari_makan sedikit_minum sedikit_mual_demam 1 hari	gea
116_38.2_32_tidak_8.0_75.0_muntah 6 kali_muntah 1 hari_BAB cair sebanyak 7 kali_demam_makan sedikit_minum sedikit	gea
100_36.4_30_tidak_10.0_79.0_BAB cair sejak 1 hari_BAB cair sebanyak 6 kali_muntah 4 kali_demam_makan sedikit_minum sedikit	gea
130_36.1_24_tidak_10.0_81.0_muntah 1 hari_muntah 10 kali_BAB cair_demam	gea
110_37.9_34_tidak_16.0_94.0_demam 2 hari_muntah 4 kali_pilek_makan sedikit	gea
120_37.1_30_tidak_5.3_6.1_BAB cair sejak 14 hari_demam 1 hari	gea
111_38.5_28_tidak_10.0_80.0_BAB cair sejak 2 hari_ada ampas_mual_muntah_demam	gea
110_39.0_40_tidak_6.5_62.0_BAB cair sebanyak 5 kali_muntah 10 kali_demam 5 hari_minum sedikit	gea
130_39.8_48_tidak_7.0_62.0_BAB cair sejak 1 hari_ada ampas_muntah_demam 1 hari_minum sedikit_BAK normal	gea
114_36.9_30_tidak_12.0_81.0_BAB cair sejak 1 hari_BAB cair sebanyak 10 kali_demam_muntah	gea
116_36.7_30_ya_31.0_123.0_muntah 5 kali_muntah 1 hari_nyeri perut	gea
121_38.5_24_ya_58.0_158.0_demam 1 hari_batuk_nyeri tenggorokan	farangitis
100_37.4_22_tidak_44.0_162.0_demam 2 hari_mual_makan sedikit_lemas_pusing_nyeri perut_nyeri tenggorokan	farangitis
102_36.0_28_ya_65.0_160.0_makan sedikit_mual_muntah_batuk_pilek_sesak nafas_bengkak gusi_lemas_demam	farangitis
101_37.5_32_tidak_12.0_100.0_batuk_demam_muntah_batuk darah	farangitis
98_36.0_24_tidak_48.0_153.0_pusing_lemas_nyeri ulu hati_mual_menggigil_demam naik turun	farangitis

88_37.6_24_ya_60.0_165.0_keram kedua kaki_demam 1 hari_batuk_pilek_mual_muntah	farangitis
128_37.2_89_tidak_10.0_87.0_demam 4 hari_demam naik turun_muntah_lemas	farangitis
123_39.1_28_tidak_40.0_151.0_demam 1 hari_pusing_mual_nyeri tenggorokan	farangitis
112_39.9_26_ya_36.0_138.0_demam 1 hari_nyeri kepala_mual_muntah	farangitis
137_39.1_24_tidak_50.0_155.0_demam 2 hari_menggigil_mimisan_gusi berdarah_nyeri tenggorokan_batuk	farangitis
120_37.0_36_tidak_7.3_69.0_demam 2 hari_batuk_pilek	pnemonia
146_37.4_30_tidak_4.1_50.0_batuk_pilek_demam_sesak nafas	pnemonia
160_38.3_34_ya_7.5_69.0_demam 2 hari_batuk_pilek_muntah	pnemonia
132_38.7_30_tidak_15.0_98.0_demam 5 hari_demam naik turun_batuk_pilek_makan sedikit_bibir kering	pnemonia
111_37.0_30_ya_55.0_155.0_batuk_batuk darah_demam naik turun_demam 14 hari_muntah_pusing_sesak nafas	pnemonia
120_38.4_36_tidak_61.0_165.0_sesak nafas_batuk_demam_riwayat tb	pnemonia
110_37.5_26_ya_15.0_98.0_demam 14 hari_batuk_pilek_mimisan_nyeri tenggorokan_muntah	pnemonia
128_37.0_60_tidak_4.5_54.0_batuk_sesak nafas_BAB cair	pnemonia
120_37.0_24_tidak_6.0_69.0_batuk_pilek_sesak nafas	pnemonia
132_39.6_40_tidak_8.0_74.0_sesak nafas_batuk_demam 3 hari_demam naik turun	pnemonia
120_36.5_32_tidak_11.0_98.0_sesak nafas_batuk_pilek_demam	pnemonia
124_30.0_36_tidak_5.2_50.0_batuk_pilek_sakit mata_BAB cair sejak 15 hari_sesak nafas	pnemonia
110_37.5_24_ya_20.0_124.0_demam 1 hari_nyeri perut_mual_muntah_batuk	typhoid
128_39.8_27_ya_12.0_97.0_demam naik turun_demam 2 hari_muntah_batuk_pilek_BAB normal_BAK normal_minum sedikit	typhoid
110_38.2_24_tidak_45.0_155.0_demam 14 hari_demam naik turun_menggigil	typhoid
129_38.7_25_ya_25.0_120.0_demam 2 hari_demam naik turun_tidak BAB 2 hari_batuk_pilek_nyeri ulu hati_mual_makan sedikit	typhoid
132_37.2_30_ya_13.0_86.0_demam 3 hari_muntah_lemas_BAB cair_BAK normal_batuk_pilek_sesak nafas	typhoid

100_38.8_26_tidak_65.0_150.0_demam 3 hari_mual_muntah_makan sedikit_minum sedikit_lemas_BAB cair	typhoid
98_36.2_26_ya_67.0_165.0_demam 2 hari_tidak BAB 2 hari_makan sedikit_minum sedikit_BAK sedikit_mual_batuk_nyeri ulu hati	typhoid
82_36.0_24_ya_57.0_160.0_muntah_makan sedikit_mual_nyeri ulu hati_demam naik turun_demam 3 hari	typhoid
120_38.4_24_tidak_15.0_97.0_demam 2 hari_demam naik turun_ada getah_pilek_makan sedikit_minum sedikit_bibir kering_lemas_muntah	typhoid
100_37.5_24_ya_16.0_123.0_demam 7 hari_batuk_pilek_makan sedikit_lemas_nyeri ulu hati_BAB sedikit_BAK sedikit	typhoid
110_36.0_30_ya_18.0_98.0_kembung_nyeri perut_muntah 1 hari_muntah 2 kali_BAB cair sejak 1 hari_BAB cair 1 kali_demam	typhoid
124_38.1_28_tidak_37.0_155.0_demam 14 hari_demam naik turun_muntah_makan sedikit_pucat	typhoid
120_37.6_120_tidak_13.0_96.0_demam naik turun_demam 2 hari_mual_muntah_makan sedikit	typhoid
124_36.6_24_ya_13.0_96.0_BAB keras_makan sedikit_demam_batuk	typhoid
108_37.4_24_tidak_30.0_134.0_demam 5 hari_demam naik turun_sakit perut_nyeri kepala_muntah	typhoid
140_38.8_30_tidak_6.1_60.0_demam 1 hari_muntah 5 kali_BAB cair sebanyak 2 kali	typhoid
128_37.5_28_tidak_9.5_87.0_muntah 5 kali_BAB cair sebanyak 2 kali_makan sedikit_demam 2 hari_lemas	typhoid

LAMPIRAN 2

Surat Keterangan Plagiasi

	UNIVERSITAS ISLAM INDONESIA DIREKTORAT PERPUSTAKAAN
Kampus Terpadu UII Jl. Kaliurang Km. 14,5, Yogyakarta 55584, INDONESIA Telp: (0274) 898 444 Psw. 2301 - 2324; Fax: (0274) 898 444 Psw. 2091 http://library.uui.ac.id ; e-mail: perpustakaan@uui.ac.id	
<hr/> <u>SURAT KETERANGAN CEK PLAGIASI</u> <hr/>	
No.1002995310 /Perpus/10/Div.PP/III/2018	
<i>Assalamu'alaikum wr. wb.</i>	
Yang bertanda tangan di bawah ini:	
Nama	: Ismanto
NIK	: 861002112
Jabatan	: Kepala Divisi Pelayanan Pemakai Direktorat Perpustakaan
Dengan ini menerangkan bahwa:	
Nama	: Rizky Karunia Putra
Nomor Mahasiswa	: 14523064
Fakultas / Prodi	: FTI / Teknik Informatika
Judul Karya Ilmiah	: Sistem Klasifikasi Penyakit Dengan Gejala Demam Pada Anak Menggunakan Metode Fuzzy K-Nearest Neighbor (FK-NN)
Berdasarkan cek plagiasi dengan menggunakan aplikasi Turnitin karya ilmiah yang bersangkutan di atas terdapat kesamaan kata sebanyak 4 (Empat) % .	
Demikian surat keterangan dibuat agar dapat dipergunakan sebagaimana mestinya.	
<i>Wassalamu'alaikum wr. wb.</i>	
Yogyakarta, 17 September 2018	
Kepala Divisi Pelayanan Pemakai Direktorat Perpustakaan	
 Asmanto	
NIK: 861002112	