

BAB 1 PENDAHULUAN

1.1 Latar belakang

Stroke merupakan penyakit yang timbul akibat terputusnya suplai darah menuju otak karena terdapat semburan pada pembuluh darah atau terjadi sumbatan berupa darah yang menggumpal. Suplai oksigen dan nutrisi pada otak menjadi terhenti dan dapat menyebabkan rusaknya jaringan pada otak. Gejala paling umum yang terjadi pada penyakit stroke adalah pada bagian wajah, lengan atau tungkai, sering terasa lemah atau mati rasa. Gejala lainnya yaitu kesulitan berbicara atau memahami perkataan orang lain, pusing, kebingungan, kesulitan dalam melihat dengan satu atau kedua mata, kesulitan dalam berjalan, hilangnya keseimbangan, sakit kepala tanpa sebab, pingsan atau tidak sadar. Efek stroke bergantung pada bagian otak mana yang terluka dan seberapa parah pengaruhnya. Kematian secara mendadak dapat terjadi ketika seorang pasien mengalami stroke yang sangat parah (WHO, 2014). Penyakit stroke menjadi penyebab penyebab kecacatan nomor satu dan kematian nomor tiga di dunia setelah penyakit jantung dan kanker. Penyakit stroke di Indonesia memiliki nilai prevalensi yang cukup tinggi. Berdasarkan data Riskedas, prevalensi stroke di Indonesia pada tahun 2013 mengalami kenaikan jika dibandingkan dengan data Riskesdas 2007 dengan nilai angka 8,3%, naik mencapai angka 12,1% per 1.000 penduduk (RI, 2013).

Penyakit stroke dapat terjadi karena beberapa faktor, diantaranya tekanan darah, riwayat fibrilasi atrium, kolesterol, diabetes dan lain sebagainya. Selama ini penanganan penyakit stroke dilakukan secara manual, dimana pasien melakukan pemeriksaan pada dokter spesialis penyakit syaraf. Kemudian dilakukan diagnosis pada pasien dengan cara mengajukan pertanyaan berupa keluhan yang dirasakan oleh pasien serta faktor-faktor yang dapat memicu terjadinya stroke. Sehingga akan didapatkan suatu kesimpulan tingkat risiko penyakit stroke pada pasien. Kegiatan semacam ini dapat menimbulkan permasalahan yaitu membutuhkan waktu dan biaya yang tidak sedikit. Berdasarkan penjelasan tersebut, maka diperlukan suatu aplikasi diagnosis tingkat risiko penyakit stroke agar stroke dapat segera diatasi sesuai dengan tingkat risikonya. Tingkat risiko penyakit stroke dibedakan menjadi 3, yaitu tingkat risiko stroke rendah, tingkat risiko stroke sedang dan tingkat risiko stroke tinggi.

Pemilihan metode yang tepat untuk deteksi tingkat risiko penyakit stroke sangat dibutuhkan karena berpengaruh pada hasil yang akan ditampilkan. Penelitian ini menggunakan data yang diambil dari data kondisi kesehatan masyarakat Kota Malang di berbagai Puskesmas di Kota Malang pada tahun 2015. Namun data yang didapatkan berupa data campuran, yaitu data numerik dan kategoris. Maka penelitian ini dapat memanfaatkan penggabungan metode *K-Nearest Neighbor* dan *Naïve Bayes*. Pada penelitian sebelumnya yang dilakukan oleh Elma Zannatul Ferdousy, dkk menunjukkan bahwa pendekatan sederhana

yang diusulkan (algoritme *K-Nearest Neighbor* dan *Naïve Bayes*) dapat memperbaiki kinerja klasifikasi *Naïve Bayes* biasa melalui penggabungan dua pengklasifikasian, dimana data yang digunakan mengandung atribut numerik dan kategoris. Akurasi pada penelitian menunjukkan angka 89.58% pada dataset *central bank*, melebihi nilai akurasi pada metode lain dengan data yang sama (Ferdousy, et al., 2013).

Algoritme *K-Nearest Neighbor* disingkat KNN digunakan untuk mengklasifikasikan objek berdasarkan data pembelajaran yang jarak tetangganya paling dekat atau memiliki nilai selisih yang kecil dengan objek tersebut. Tujuan dari algoritme ini adalah untuk mengklasifikasikan objek baru berdasarkan dari atribut, data latih dan data uji. Prinsip umum dari algoritme ini adalah menentukan nilai K pada data latih untuk menentukan KNN berdasarkan ukuran jarak. Selanjutnya nilai mayoritas dari KNN akan menjadi dasar untuk menentukan jenis kelas atau kategori dari sample berikutnya (Shouman, et al., 2012). Algoritme *Naïve Bayes* merupakan pengklasifikasian dengan metode probabilitas dan statistik, yaitu memprediksi peluang di masa depan berdasarkan pengalaman dimasa sebelumnya sehingga dikenal sebagai Teorema Bayes. Teorema ini kemudian dikombinasikan dengan *Naïve* dimana kondisi antar atribut saling bebas tidak terikat. Klasifikasi *Naïve Bayes* diasumsikan bahwa ada atau tidak ciri tertentu dari sebuah kelas tidak ada hubungannya dengan ciri dari kelas lainnya (Bustami, 2014).

Berdasarkan pada latar belakang diatas, maka perlu adanya suatu aplikasi diagnosis tingkat risiko penyakit stroke menggunakan metode *K-Nearest Neighbor* dan *Naïve Bayes* karena data yang didapat menggunakan atribut numerik dan kategoris. Pada aplikasi, terdapat karakteristik gejala penyakit stroke, seperti tekanan darah, riwayat fibrilasi atrium, merokok, kolesterol, diabetes, aktifitas fisik, diet dan riwayat keluarga yang akan digunakan dalam perhitungan untuk menentukan tingkat risiko penyakit stroke yang diderita oleh pasien. Metode *K-Nearest Neighbor* dan *Naïve Bayes* menghasilkan akurasi yang cukup baik dalam penelitian sebelumnya. Sehingga dengan menggunakan algoritme ini diharapkan dapat membantu dalam menentukan tingkat risiko penyakit stroke.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang diatas dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana penerapan metode *K-Nearest Neighbor* dan *Naïve Bayes* untuk diagnosis tingkat risiko penyakit stroke?
2. Bagaimana hasil akurasi pengujian pada sistem diagnosis tingkat risiko penyakit stroke menggunakan metode *K-Nearest Neighbor* dan *Naïve Bayes*?

1.3 Tujuan

Tujuan yang ingin dicapai terkait dengan penelitian ini adalah:

1. Menerapkan metode *K-Nearest Neighbor* dan *Naïve Bayes* untuk diagnosis tingkat risiko penyakit stroke.
2. Mengetahui hasil akurasi pengujian dari sistem diagnosis tingkat risiko penyakit stroke menggunakan metode *K-Nearest Neighbor* dan *Naïve Bayes*.

1.4 Manfaat

Manfaat penelitian yang didapatkan dari penelitian ini, yaitu:

1. Dapat membantu dalam melakukan diagnosis tingkat risiko penyakit stroke.
2. Hasil penelitian dapat digunakan sebagai referensi dalam penelitian mengenai penyakit stroke serta pengembangan metode *K-Nearest Neighbor* dan *Naïve Bayes*.
3. Dapat memberikan pelayanan, penanganan dan dapat memberikan informasi yang berguna untuk masyarakat yang mengalami gejala awal penyakit stroke agar dapat ditangani dengan baik dan tepat.

1.5 Batasan Masalah

Berdasarkan latar belakang dan rumusan masalah diatas, agar permasalahan yang dirumuskan dapat lebih terfokus, maka penelitian ini dibatasi dalam beberapa hal, yaitu:

1. Data yang digunakan diambil dari skripsi sebelumnya yang berjudul "Sistem Pakar Deteksi Dini Penyakit Stroke Menggunakan Metode Fuzzy Naïve Bayes". Merupakan data kondisi kesehatan masyarakat Kota Malang di Puskesmas Arjuno, Puskesmas Ciptomulyo, Puskesmas Bareng, Puskesmas Arjowinangun, Puskesmas Gribig dan Puskesmas Janti pada tahun 2015.
2. Jumlah data sebanyak 150 data.
3. Data atribut penyakit meliputi tekanan darah, riwayat fibrilasi atrium, merokok, kolesterol, diabetes, aktifitas fisik, diet, riwayat keluarga serta kelas untuk penentuan tingkat risiko stroke.
4. Keluaran berupa hasil diagnosis tingkat risiko penyakit stroke dengan hasil risiko tingkat rendah, sedang atau tinggi.

1.6 Sistematika Pembahasan

Sistematika penyusunan laporan ditunjukkan untuk mencapai tujuan yang ada meliputi beberapa bab, sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika pembahasan mengenai sistem diagnosis tingkat risiko penyakit stroke menggunakan metode *K-Nearest Neighbor* dan *Naïve Bayes*.

BAB II LANDASAN KEPUSTAKAAN

Pada bab ini berisi tentang dasar teori yang digunakan untuk mendukung dalam penelitian diagnosis tingkat risiko penyakit stroke menggunakan metode *K-Nearest Neighbor* dan *Naïve Bayes*. Dasar teori meliputi Penyakit Stroke, Algoritme *K-Nearest Neighbor*, Algoritme *Naïve Bayes* serta Algoritme *K-Nearest Neighbor* dan *Naïve Bayes*.

BAB III METODOLOGI

Metodologi terdiri dari kajian pustaka, pengumpulan data, perancangan sistem, implementasi sistem, pengujian dan analisa sistem serta kesimpulan dalam pembuatan sistem untuk diagnosis tingkat risiko penyakit stroke menggunakan metode *K-Nearest Neighbor* dan *Naïve Bayes*.

BAB IV PERANCANGAN SISTEM

Pada bab ini berisi tentang hasil perancangan sistem untuk implementasi sistem diagnosis tingkat risiko penyakit stroke menggunakan metode *K-Nearest Neighbor* dan *Naïve Bayes*.

BAB V IMPLEMENTASI

Pada bagian ini menguraikan tentang implementasi sistem diagnosis tingkat risiko penyakit stroke menggunakan metode *K-Nearest Neighbor* dan *Naïve Bayes*,

BAB VI PENGUJIAN DAN ANALISIS

Pada bab ini membahas tentang pengujian yang disertai beberapa parameter pengujian dan analisis sistem diagnosis tingkat risiko penyakit stroke menggunakan metode *K-Nearest Neighbor* dan *Naïve Bayes*.

BAB VII PENUTUP

Bab ini menguraikan kesimpulan yang diperoleh pada sistem diagnosis tingkat risiko penyakit stroke menggunakan metode *K-Nearest Neighbor* dan *Naïve Bayes* serta menyertakan saran yang dapat digunakan untuk pengembangan selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Penyakit stroke digunakan sebagai objek dalam penelitian ini karena penyakit stroke merupakan salah satu penyakit yang memiliki tingkat kematian tertinggi. Oleh karena itu, dibutuhkan sebuah sistem untuk mengetahui tingkat risiko penyakit stroke supaya dapat membantu masyarakat dalam mendeteksi dan mempermudah para medis dalam menentukan tingkat risiko penyakit stroke pada pasien. Metode yang digunakan dalam penelitian ini adalah adalah *K-Nearest Neighbor-Naïve Bayes*. Metode ini dapat digunakan dalam diagnosis tingkat risiko penyakit stroke dengan mengelompokkan ke dalam tingkat risiko rendah, sedang atau tinggi berdasarkan atribut penyakit yang diderita oleh masyarakat. Penelitian dibawah ini, merujuk pada Tabel 2.1.

Penelitian pertama merupakan penelitian yang dilakukan oleh Emma Zannatul Ferdousy, dkk menggunakan perpaduan *Naïve Bayes Classifier* dan *K-Nearest Neighbor* untuk memprediksi posisi profitabilitas lembaga keuangan. Pada penelitian ini menunjukkan bahwa kombinasi kedua algoritme yang digunakan lebih baik daripada hanya menggunakan algoritme *Naïve Bayes* saja, dengan tingkat akurasi sebesar 89,58% (Ferdousy, et al., 2013).

Penelitian berikutnya dilakukan oleh Safira Indana pada tahun 2017 membahas mengenai deteksi dini penyakit stroke menggunakan metode *fuzzy naïve bayes* dan sistem pakar. Penelitian penulis menggunakan data yang terdapat pada penelitian ini. Pada penelitian ini metode *Fuzzy Naïve Bayes* dihitung menggunakan data penderita stroke sehingga dapat disimpulkan bahwa metode *Fuzzy Naïve Bayes* merupakan metode yang baik dengan hasil tingkat akurasi yang cukup optimal, yaitu sebesar 85,6% (Insani, et al., 2017).

Penelitian selanjutnya dilakukan oleh Vina Adelina menggunakan metode GA-Fuzzy Tsukamoto untuk klasifikasi tingkat risiko penyakit stroke. Telah ditemukan penelitian untuk memperkirakan risiko penyakit stroke menggunakan inferensi Logika Fuzzy, dari 15 data uji mendapatkan akurasi 60%. Untuk mengoptimalkan hasil klasifikasi, penelitian tersebut menggunakan inferensi Fuzzy Tsukamoto, sedangkan untuk optimasi fungsi derajat keanggotaan menggunakan Algoritme Genetika. Akurasi yang dihasilkan dengan dilakukan pengoptimalan batasan fungsi keanggotaan sebesar 86.66% (Adelina, et al., 2018).

Tabel 2.1 Kajian Pustaka

No.	Judul	Obyek	Metode	Hasil
1.	(Ferdousy, et al., 2013)	<i>Dataset Central Bank</i>	Kombinasi metode <i>Naïve Bayes Classifier</i> dan <i>K-Nearest Neighbor</i>	Penelitian ini menunjukkan performa yang lebih baik dibandingkan dengan menggunakan metode <i>Naïve Bayes</i> saja. Tingkat akurasi yang diperoleh sebesar 89,58% pada $K=21$
2.	(Insani, et al., 2017)	Deteksi Dini Penyakit Stroke	Fuzzy <i>Naïve Bayes</i> dengan Sistem Pakar	Sistem pakar deteksi dini penyakit stroke tersebut memiliki kinerja sistem yang mampu berjalan dengan baik sesuai kebutuhan fungsional. Hasil pengujian akurasi terhadap 30 data kasus uji menghasilkan persentase sebesar 85,6%
3.	(Adelina, et al., 2018)	Klasifikasi Tingkat Risiko Penyakit Stroke	Algoritme Genetika-Fuzzy Tsukamoto	Akurasi yang dihasilkan dengan dilakukan pengoptimalan batasan fungsi keanggotaan dengan algoritme genetika dan fuzzy tsukamoto sebesar 86.66%.

2.2 Penyakit Stroke

Menurut *WHO Task Force in Stroke and other Cerebrovascular Disease* (1989) penyakit stroke merupakan gangguan disfungsi *neurologist* akut yang terjadi secara mendadak (hanya dalam waktu beberapa detik) atau setidaknya secara cepat (dalam hitungan jam) dengan gejala dan tanda-tanda yang sesuai dengan daerah fokal otak yang terganggu yang disebabkan oleh gangguan peredaran darah. Gangguan fungsi otak fokal pada stroke mengakibatkan tanda klinis berkembang cepat, dengan waktu gejala yang berlangsung selama 24 jam atau lebih dan dapat menyebabkan kematian (WHO, 2006)

Stroke pada umumnya dialami oleh penderita sekitar umur 65 tahun dengan presentase sebesar 33,5%. Angka kejadian penyakit stroke pada laki-laki biasanya lebih banyak dari pada perempuan. Stroke dapat terjadi tanpa munculnya gejala dini, dan muncul sangat mendadak. Stroke merupakan salah satu penyebab kecacatan dan kematian tertinggi di dunia. Tidak hanya berdampak bagi para penyandang, namun kecacatan akibat stroke juga berdampak buruk bagi keluarganya (Pinzon & Asanti, 2010). Tanpa penanggulangan dan pencegahan yang tepat, stroke dapat menjadi penyebab utama dari kecacatan pada tahun 2030 (Arofah, 2011). Di Amerika Serikat, stroke menduduki peringkat ke-3 sebagai penyebab kematian setelah penyakit jantung dan kanker. Setiap tahunnya 500.000 orang Amerika terserang stroke, dengan 80% diantaranya terkena stroke iskemik (Adams, et al., 2005).

Stroke merupakan masalah neurologik paling utama yang ada di dunia, negara Indonesia merupakan negara di Asia yang memiliki jumlah penderita stroke terbesar. Rendahnya kesadaran akan faktor risiko stroke, belum optimalnya pelayanan stroke serta kurangnya pengenalan gejala stroke dan ketaatan terhadap program terapi untuk pencegahan stroke ulang, berpengaruh pada kejadian stroke berulang (Safitri, 2012). Pasien penderita stroke memiliki risiko yang tinggi untuk mengalami serangan stroke ulang. Dalam waktu 5 tahun serangan stroke ulang dapat terjadi dengan nilai persentase 30%-43%.

2.2.1 Jenis Penyakit Stroke

a. Stroke Iskemik

Stroke Iskemik merupakan stroke yang muncul akibat adanya hambatan atau sumbatan pada pembuluh darah otak tertentu sehingga daerah otak yang diberikan aliran darah oleh pembuluh darah tersebut tidak lagi mendapatkan persediaan energi dan oksigen. Hambatan atau sumbatan tersebut mengakibatkan jaringan sel-sel otak pada daerah tersebut tidak berfungsi kembali/mati (Quamila, 2017).

b. Stroke Hemoragik

Stroke Hemoragik diawali dari pembuluh darah yang melemah, kemudian pecah dan menumpahkan darah pada daerah sekitarnya. Darah yang bocor

mengalami penumpukan dan menyebabkan penghambatan pada jaringan otak disekitarnya. Kematian atau koma panjang akan terjadi jika pendarahan tetap berlanjut. Hampir 70% kasus stroke hemoragik terjadi pada penderita hipertensi (Quamila, 2017).

2.2.2 Faktor Risiko Penyakit Stroke

Faktor Risiko stroke adalah kondisi kesehatan atau penyakit yang ada pada seseorang yang berisiko terkena penyakit stroke. Apabila kondisi ini tidak segera diatasi maka dapat memperburuk keadaan dan dapat mengakibatkan terjadinya penyempitan atau pecahnya pembuluh darah otak. Berdasarkan *National Stroke Association* terdapat delapan faktor risiko pada penyakit stroke yaitu:

1. Tekanan Darah

Tekanan darah tinggi adalah penyebab stroke nomor satu. Tekanan darah adalah kekuatan darah yang mendorong dinding arteri. Tekanan darah tinggi menyebabkan jantung memompa lebih keras untuk memindahkan darah ke seluruh tubuh. Hal ini bisa melemahkan pembuluh darah dan merusak organ utama, seperti otak. Seseorang yang mempunyai tekanan darah tinggi memiliki risiko satu setengah kali lebih besar terkena stroke dibandingkan mereka yang secara konsisten memiliki tekanan darah optimal sebesar 120/80. Hal terpenting yang dapat dilakukan adalah mengendalikannya. Hal ini bisa dilakukan melalui kebiasaan makan sehat, aktivitas fisik, atau pengobatan (Association, n.d.).

2. Riwayat Fibrilasi Atrium

Fibrilasi Atrium merupakan jenis detak jantung yang memiliki detak tidak teratur. Meskipun dapat terjadi pada usia berapapun, Fibrilasi Atrium lebih sering dialami oleh pasien dengan usia di atas 65 tahun. Fibrilasi Atrium lebih sering terjadi pada pasien dengan penyakit jantung, tekanan darah tinggi atau diabetes. Fibrilasi Atrium menimbulkan risiko stroke karena memungkinkan darah menyatu di hati. Sumsum darah cenderung membentuk gumpalan, kemudian gumpalan darah tersebut dibawa ke otak dan dapat menjadi awal mula terjadinya stroke (Association, n.d.).

3. Merokok

Perokok memiliki risiko dua kali lipat terkena penyakit stroke jenis iskemik (dimana terjadi pembekuan darah pada otak), serta memiliki risiko empat kali lipat risiko terkena stroke jenis hemoragik (dimana pembuluh darah pada otak pecah). Merokok memicu lebih banyak produksi fibrinogen (faktor penggumpalan darah) sehingga menimbulkan aterosklerosis (Vitahealth, 2003).

4. Kolesterol

Kolesterol adalah zat lemak dalam darah. Kolesterol dapat diproduksi oleh tubuh atau ditemukan dalam makanan. Kolesterol tinggi di arteri bisa menghalangi aliran normal ke otak dan menyebabkan stroke. Dengan nilai

kolesterol yang tinggi, maka risiko penyakit jantung dan aterosklerosis juga akan meningkat. Total kadar kolesterol di bawah 200 mg/dl dianggap aman dan jika kadar kolesterol meninggi, maka dapat dikontrol melalui kebiasaan makan, aktivitas fisik dan obat (Association, n.d.).

5. Diabetes

Pada penderita diabetes, tubuh tidak menghasilkan cukup insulin (Diabetes Tipe I) atau sel-sel menolak insulin (Diabetes Tipe II). Tanpa insulin, tubuh tidak bisa mengolah gula yang merupakan bahan bakar dasar bagi sel-sel dalam tubuh. Orang dengan diabetes memiliki kemungkinan empat kali lebih besar terkena stroke daripada orang yang tidak memiliki diabetes, terutama karena penderita diabetes sering memiliki faktor risiko stroke lainnya, seperti tekanan darah tinggi, fibrilasi atrium, dan kolesterol tinggi. Penurunan berat badan, olahraga, perubahan kebiasaan makan, obat oral, dan suntikan insulin adalah beberapa cara untuk mengendalikan diabetes (Association, n.d.).

6. Aktifitas Fisik

Aktifitas fisik mencakup aktifitas yang membutuhkan pergerakan pada tubuh. Penelitian menunjukkan bahwa seseorang yang berolahraga minimal lima kali atau lebih dalam seminggu dapat mengurangi risiko penyakit stroke (Vitahealth, 2003).

7. Diet

Diet sehat membantu untuk meningkatkan kesehatan secara keseluruhan, mengurangi risiko penyakit stroke, dan membantu untuk mempertahankan berat badan yang sehat (Vitahealth, 2003).

8. Riwayat Keluarga

Stroke dapat terjadi akibat adanya faktor keturunan. Faktor genetik menjadi salah satu faktor penting dalam risiko penyakit stroke antara lain penyakit jantung, tekanan darah tinggi, cacat pada pembuluh darah dan diabetes. Faktor gaya hidup dalam suatu keluarga juga dapat mempengaruhi risiko penyakit stroke (Vitahealth, 2003).

2.3 K-Nearest Neighbor

K-Nearest Neighbor (KNN) merupakan salah satu metode yang umumnya digunakan dalam melakukan proses pengklasifikasian data. Algoritme *K-Nearest Neighbor* digunakan dalam pengklasifikasian objek berdasarkan pada data pembelajaran yang jarak tetangganya paling dekat atau memiliki nilai selisih yang kecil dengan objek. Tujuan dari algoritme ini adalah untuk mengklasifikasikan objek baru berdasarkan dari atribut, data uji dan data latih. Prinsip umum dari algoritme ini adalah menentukan nilai *K* pada data latih untuk memproses *K-Nearest Neighbor* berdasarkan ukuran jarak. Selanjutnya nilai mayoritas dari *K-Nearest Neighbor* akan menjadi dasar untuk menentukan jenis kelas atau kategori dari sample selanjutnya. Selain itu algoritme ini sendiri sering digunakan untuk

klasifikasi pada teknik *data mining* serta dapat digunakan untuk estimasi dan prediksi data.

K-Nearest Neighbor merupakan salah satu teknik *data mining* yang paling sederhana. Hal ini biasa disebut dengan klasifikasi memori berbasis, sebagai contoh data latih perlu berada di memori pada saat *run-time*. Ketika berhadapan dengan atribut yang berkelanjutan, perbedaan antara atribut dihitung menggunakan jarak *Euclidean*. Jika contoh pertama adalah $(a_1, a_2, a_3, \dots, a_n)$ dan contoh kedua adalah $(b_1, b_2, b_3, \dots, b_n)$, maka jarak di antaranya dihitung dengan menggunakan persamaan 2.1 dibawah: (Shouman, et al., 2012)

$$d = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 \dots \dots \dots (a_n - b_n)^2} \quad \mathbf{2.1}$$

Penjelasan rumus :

- a : data uji, digunakan untuk melakukan tes pada permodelan yang dihasilkan dari data latih.
- b : data latih, digunakan untuk mencari permodelan yang tepat.
- n : data ke-n

2.4 Naïve Bayes

Salah satu cara yang digunakan untuk mengatasi ketidakpastian pada data adalah dengan menggunakan probabilitas bayes. Persamaan 2.2 dan 2.3 dibawah menunjukkan persamaan *naïve bayes*

$$P(H|E) = \frac{P(E|H).P(H)}{P(E)} \quad \mathbf{2.2}$$

$$P(H|E) = P(E_1|H) * P(E_2|H) * \dots * P(E_n|H) * P(H) \quad \mathbf{2.3}$$

Dimana:

- P(H|E) : probabilitas hipotesis H jika diberikan evidence E
- P(E|H) : probabilitas munculnya evidence E jika diketahui hipotesis H
- P(H) : probabilitas hipotesis H tanpa memandang evidence apapun
- P(E) : probabilitas evidence E

Persamaan 2.2 dan 2.3 digunakan pada data dengan sifat kategoris. Terdapat beberapa kelemahan dari algoritme Naïve Bayes, yaitu tidak dapat membuat model untuk data yang terikat. Sedangkan kelebihan dari algoritme ini adalah mudah diimplementasikan dan memberikan hasil yang baik untuk banyak kasus (Syarli & Muin, 2016).

2.5 K-Nearest Neighbor dan Naïve Bayes

Ada beberapa pendekatan yang berhubungan dengan klasifikasi, misalnya algoritme berbasis statistik *Naïve Bayes Classifier* dan algoritme berbasis jarak *K-Nearest Neighbor*. Dalam kasus algoritme *Naïve Bayes*, salah satu faktor yang

harus ditangani adalah atribut dengan tipe numerik. Dalam algoritme ini seseorang harus menentukan probabilitas bersyarat untuk setiap nilai dari semua atribut. Untuk mengatasi permasalahan ini, harus dilakukan diskretisasi atribut numerik ke dalam beberapa kelas dengan menerapkan teknik diskretisasi dari berbagai pilihan yang tersedia. Jadi teknik yang digunakan untuk diskretisasi memainkan peran penting atas keakuratan metode ini. Beberapa upaya telah dilakukan untuk meningkatkan keakuratan algoritme *Naïve Bayes* dengan mengadopsi skema diskretisasi baru (Yang & Webb, 2002).

Dalam kasus algoritme *K-Nearest Neighbor*, memiliki situasi yang berlawanan. Algoritme ini memiliki masalah mengenai atribut kategoris. Sebagai algoritme yang mengklasifikasikan dari data pelatihan berdasarkan jarak, skema pengukuran jarak untuk data kategoris harus ada. Umumnya dilakukan dengan teknik *preprocessing*, yaitu merubah data kategoris menjadi numerik dengan nilai-nilai tertentu (Ferdousy, et al., 2013).

Metode *K-Nearest Neighbor* dan *Naïve Bayes* diusulkan agar dapat memperbaiki kelemahan dari masing-masing algoritme. Pada algoritme ini tidak diperlukan diskretisasi pada data numerik dan juga tidak perlu mengukur jarak antar atribut kategoris. Diharapkan kombinasi ini dapat meningkatkan kinerja algoritme tersebut. *K-Nearest Neighbor* digunakan dalam menghitung jarak pada data dengan sifat numerik, dimana hasil perhitungan tersebut digunakan untuk membentuk model klasifikasi. Model klasifikasi tersebut kemudian digunakan untuk metode *Naïve Bayes* dalam perhitungan data dengan sifat kategoris (Ferdousy, et al., 2013).

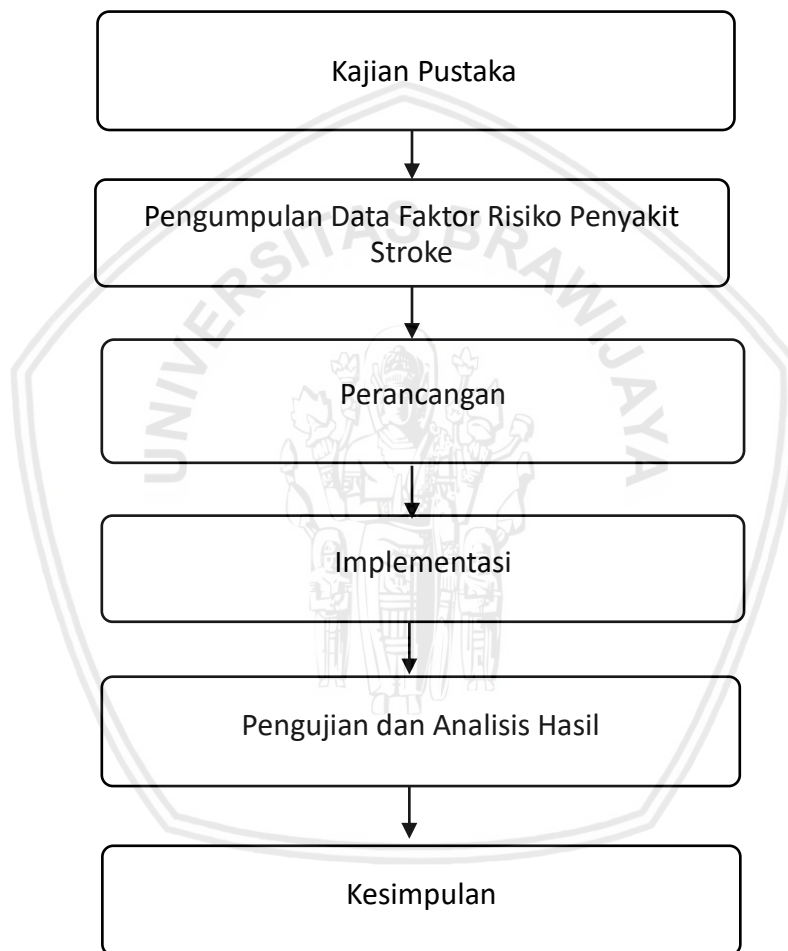
Langkah Pertama: Gunakan atribut dengan tipe numerik sebagai data latih dan data uji untuk proses klasifikasi menggunakan algoritme *K-Nearest Neighbor*

Langkah Kedua: Hasil dari pendekatan-pendekatan K kemudian dikumpulkan dan digunakan sebagai data latih untuk membangun model algoritma berikutnya, yaitu *Naïve Bayes*. Untuk proses algoritma *Naïve Bayes* hanya menggunakan data yang bersifat kategoris.

Langkah Ketiga: Gunakan model yang telah dibangun untuk proses klasifikasi.

BAB 3 METODOLOGI

Pada bab metodologi penelitian ini akan dibahas mengenai metode yang akan digunakan yaitu *K-Nearest Neighbor* dan *Naïve Bayes* beserta tahap-tahap implementasi diagnosis tingkat risiko penyakit stroke. Tahapan tersebut berisi kajian pustaka, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian dan analisis sistem dan penarikan kesimpulan. Gambar 3.1 menunjukkan tahapan metodologi dalam diagram blok metodologi penelitian.



Gambar 3. 1 Diagram Blok Metodologi Penelitian

3.1 Kajian Pustaka

Kajian pustaka pada penelitian ini mempelajari literature dan merujuk dari beberapa bidang ilmu dan informasi yang berhubungan dengan implementasi dari algoritme *K-Nearest Neighbor* dan *Naïve Bayes* yang akan digunakan untuk diagnosis tingkat risiko penyakit stroke. Literatur yang digunakan diperoleh dari

jurnal, bimbingan oleh dosen pembimbing, internet dan buku. Adapun literatur yang dipelajari adalah:

- a. Kajian Pustaka
- b. Penyakit Stroke
- c. Algoritme *K-Nearest Neighbor*
- d. Algoritme *Naïve Bayes*
- e. Algoritme *K-Nearest Neighbor* dan *Naïve Bayes*

3.2 Pengumpulan Data

Data pada penelitian ini, diambil dari penelitian sebelumnya yang dilakukan oleh Safira Adiputri Insani yang berjudul “Sistem Pakar Deteksi Dini Penyakit Stroke Menggunakan Metode *Fuzzy Naïve Bayes*”. Pengumpulan data pada penelitian ini menggunakan metode data sekunder, serta wawancara secara langsung. Data yang digunakan diambil dari data kondisi kesehatan masyarakat Kota Malang di Puskesmas Arjuno, Puskesmas Ciptomulyo, Puskesmas Bareng, Puskesmas Arjowinangun, Puskesmas Gribig dan Puskesmas Janti pada tahun 2015. Data atribut penyakit meliputi tekanan darah, riwayat fibrilasi atrium, merokok, kolesterol, diabetes, aktifitas fisik, diet, riwayat keluarga serta kelas untuk penentuan tingkat risiko stroke apakah termasuk dalam tingkat rendah, sedang atau tinggi. Data yang digunakan sebanyak 150 data.

3.3 Perancangan

Perancangan sistem berfungsi untuk merancang tahapan-tahapan pembangunan dari sistem yang akan dibangun. Setelah perancangan dikerjakan, perancangan yang dibangun akan dituangkan kedalam bentuk program komputer. Perancangan program yang dilakukan dibangun dalam beberapa tahapan, yaitu: deskripsi sistem, perancangan sistem, perhitungan manual, perancangan antarmuka dan perancangan pengujian.

3.4 Implementasi

Implementasi merupakan tahapan yang dikembangkan berdasarkan hasil analisis dan perancangan sistem. Tahap ini menandakan bahwa sistem siap untuk dioperasikan pada keadaan yang sebenarnya, sehingga akan diketahui apakah hasil yang diperoleh dari sistem sesuai dengan tujuan yang diinginkan. Dalam hal ini, akan dilakukan implementasi algoritme *K-Nearest Neighbor* dan *Naïve Bayes* untuk diagnosis tingkat risiko penyakit stroke. Tahapan-tahapan yang terdapat dalam implementasi antara lain:

1. Implementasi antarmuka (*interface*), menggunakan java netbeans.
2. Implementasi algoritme dengan metode *K-Nearest Neighbor* dan *Naïve Bayes* yang diimplementasikan menggunakan bahasa pemrograman java.

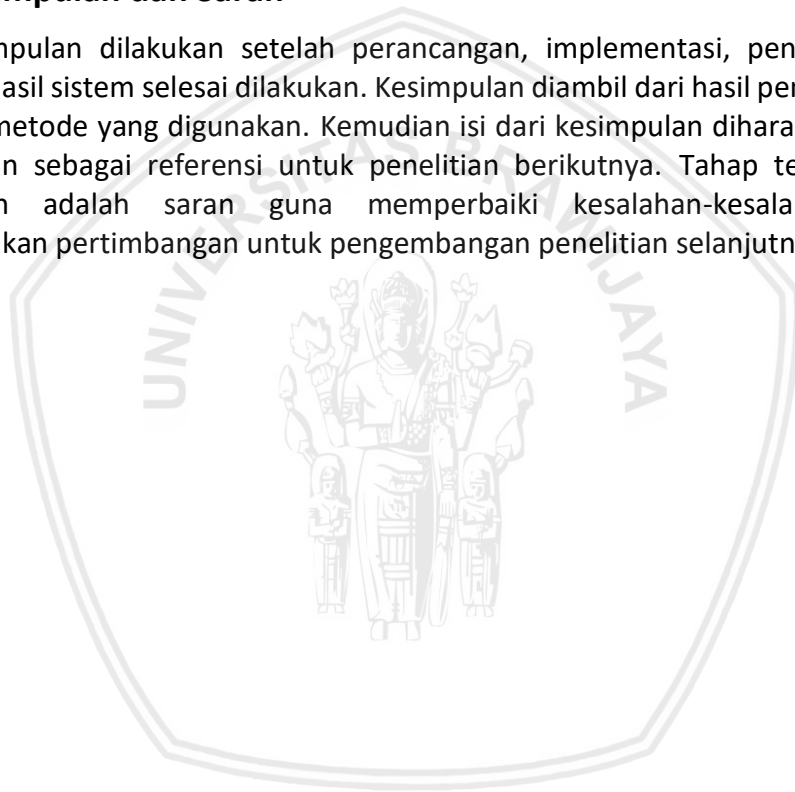
3. Implementasi yang dihasilkan merupakan data hasil diagnosis tingkat risiko penyakit stroke berdasarkan atribut stroke.

3.5 Pengujian dan Analisis Hasil

Pengujian pada tahap ini dilakukan untuk menguji apakah sistem dapat berjalan sesuai dengan yang telah dirancang sebelumnya. Tahap pengujian bertujuan agar pengguna tidak menemui error atau bug dalam sistem. Pengujian dilakukan dengan membagi data kedalam 2 variasi. Dimana, variasi pertama akan memiliki label kelas seimbang, sedangkan variasi kedua memiliki label kelas tidak seimbang. Kemudian dilakukan analisis hasil pada sistem.

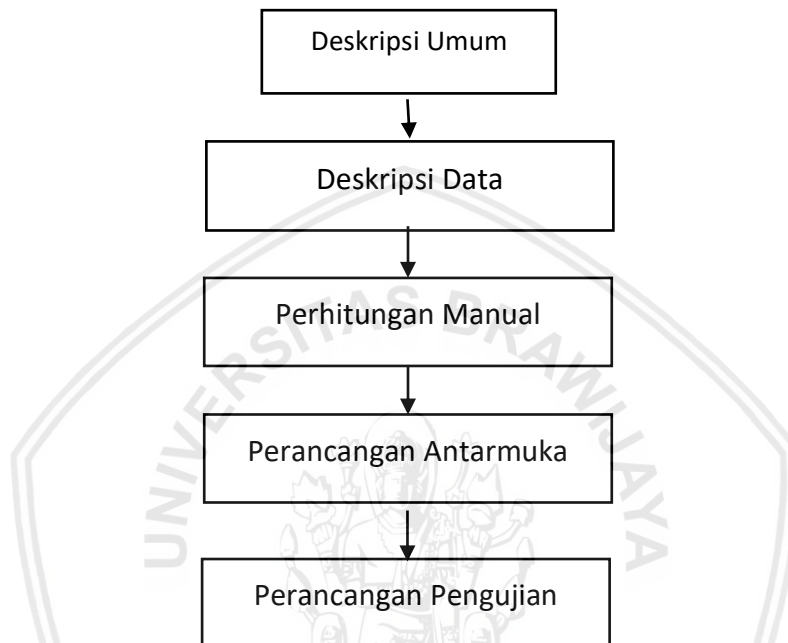
3.6 Kesimpulan dan Saran

Kesimpulan dilakukan setelah perancangan, implementasi, pengujian dan analisis hasil sistem selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis metode yang digunakan. Kemudian isi dari kesimpulan diharapkan dapat digunakan sebagai referensi untuk penelitian berikutnya. Tahap terakhir dari penulisan adalah saran guna memperbaiki kesalahan-kesalahan serta memberikan pertimbangan untuk pengembangan penelitian selanjutnya.



BAB 4 PERANCANGAN

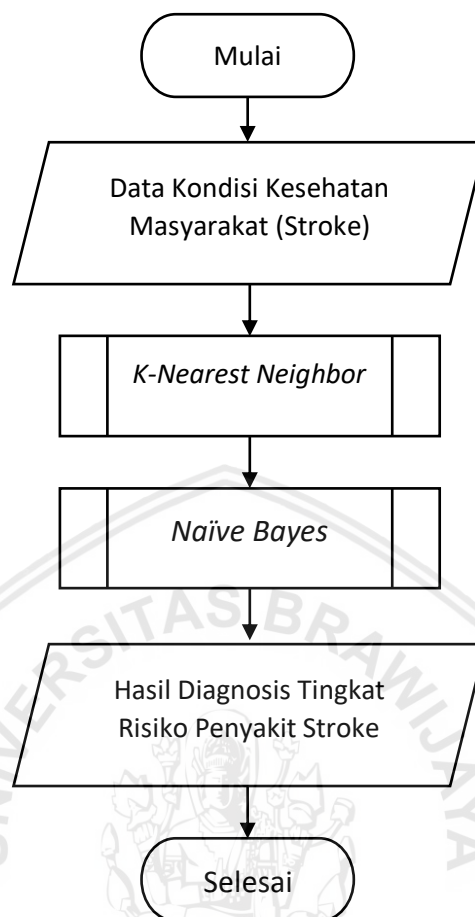
Bab ini membahas tentang analisis kebutuhan dan perancangan untuk membantu dalam melakukan diagnosis tingkat risiko penyakit stroke menggunakan metode *K-Nearest Neighbor* dan *Naïve Bayes*. Terdapat beberapa tahap dalam melakukan perancangan, seperti yang ditunjukkan dalam Gambar 4.1 dibawah ini:



Gambar 4.1 Pohon Perancangan

4.1 Perancangan Proses

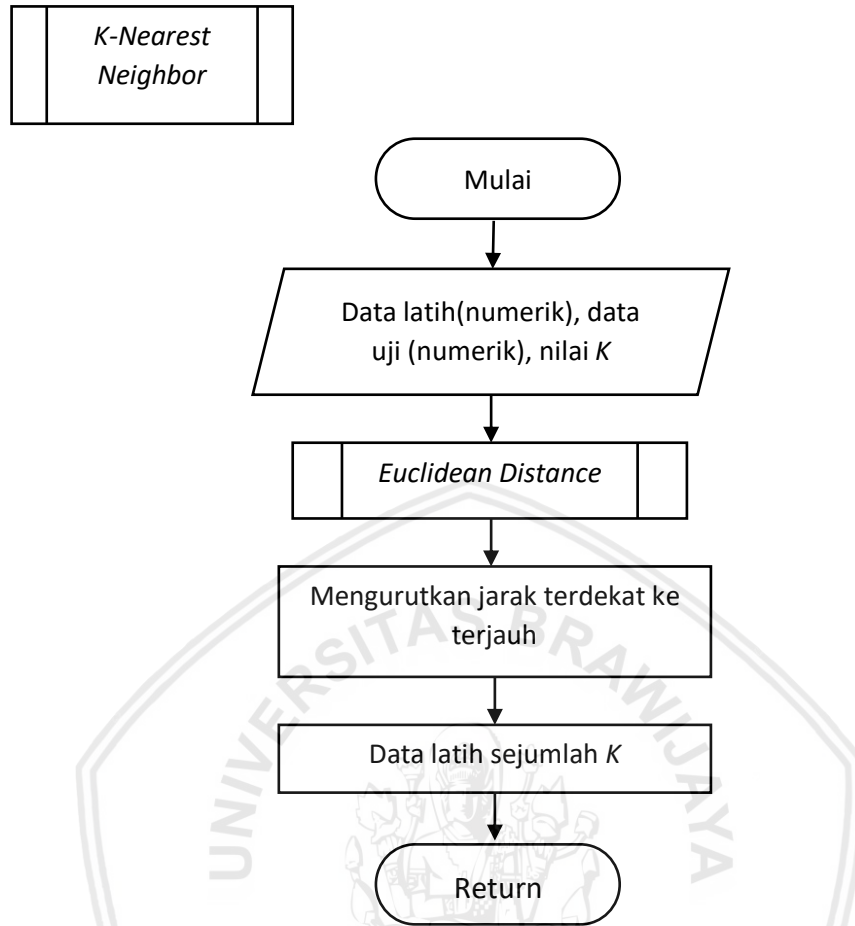
Perancangan proses menggunakan metode *K-Nearest Neighbor* dan *Naïve Bayes* yang digunakan untuk proses diagnosis tingkat risiko penyakit stroke, apakah termasuk dalam tingkat resiko rendah, sedang atau tinggi. Terdapat beberapa tahapan untuk melakukan proses diagnosis tingkat risiko penyakit stroke. Tahapan awal adalah menggunakan metode *K-Nearest Neighbor* dengan atribut bertipe numerik sebagai data latih. Kemudian hasil dari proses sebelumnya yaitu data latih sejumlah K digunakan untuk tahap selanjutnya, yaitu perhitungan dengan mengambil nilai dengan tipe kategoris menggunakan metode *Naïve Bayes*. Tahap akhir merupakan pengujian yang dilakukan pada hasil diagnosis tingkat risiko penyakit stroke. Pengujian dilakukan dengan menguji tingkat akurasi dari hasil diagnosis tersebut. Gambar 4.2 menunjukkan tahapan proses perhitungan.



Gambar 4.2 Tahapan Proses Diagnosis Tingkat Risiko Penyakit Stroke

4.1.1 Proses *K-Nearest Neighbor*

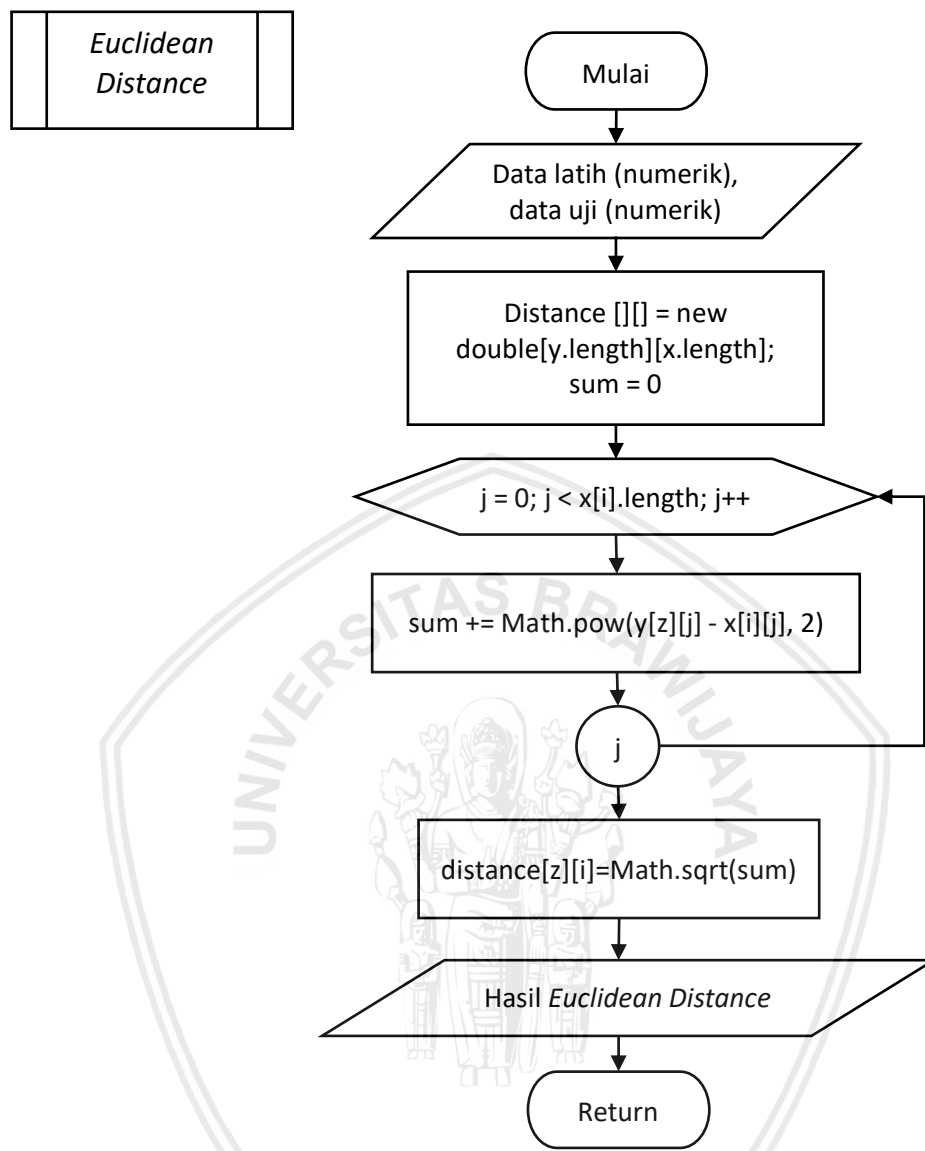
Proses dari algoritme *K-Nearest Neighbor* diawali dengan mengklasifikasikan data uji dan nilai K . Kemudian dari data latih dan data uji tersebut, hanya diambil data yang bersifat numerik. Proses berikutnya dilakukan perhitungan *euclidean distance* dengan menghitung jarak masing-masing antara data uji dan data latih. Hasil dari *euclidean distance* kemudian diurutkan dari jarak tetangga terdekat sampai terjauh. Kemudian dilakukan pengambilan data hasil pengurutan sesuai dengan nilai K yang telah ditentukan untuk di proses dengan algoritme selanjutnya. Gambar 4.3 menunjukkan proses algoritme *K-Nearest Neighbor* dalam diagram alir.



Gambar 4.3 Diagram Alir Proses *K-Nearest Neighbor*

4.1.1.1 Proses Hitung Jarak *Euclidean Distance*

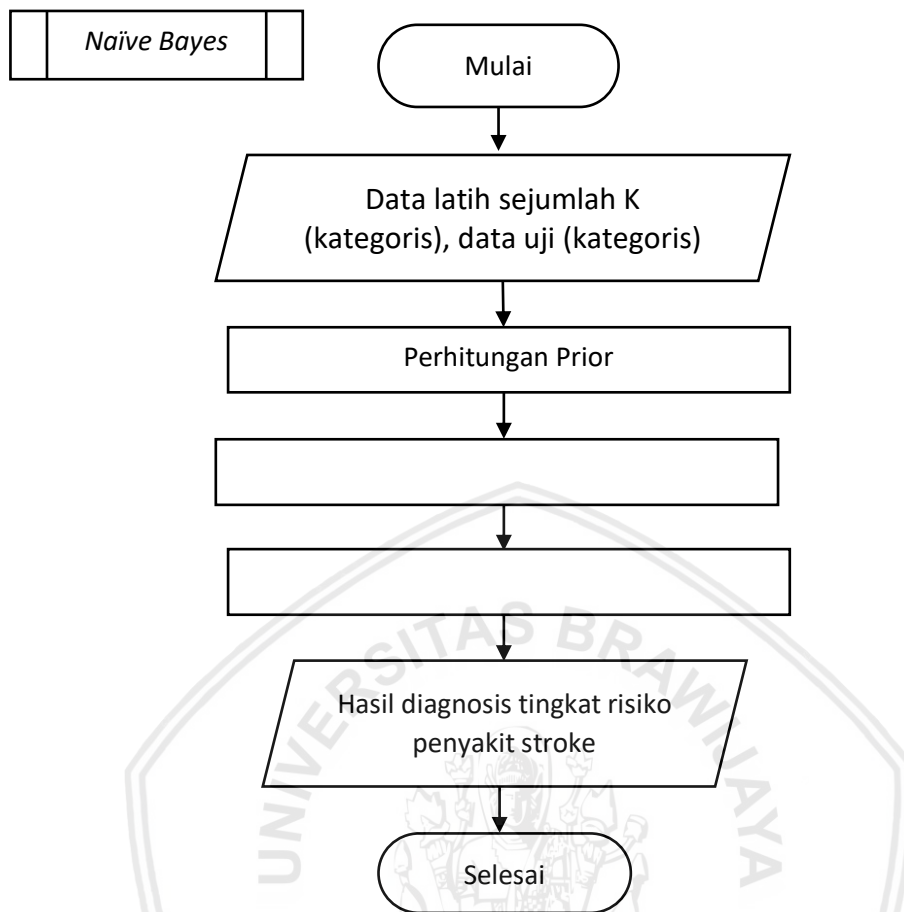
Proses hitung jarak *Euclidean Distance* dilakukan dengan memasukkan dataset penyakit stroke berupa data latih dan data uji yang bersifat numerik. Sistem pertama-tama melakukan proses inialisasi, lalu dilakukan tahap perhitungan jarak *Euclidean* seperti dalam persamaan (2.1). Keluaran proses merupakan hasil perhitungan jarak *Euclidean*, yaitu jarak data uji terhadap data latih. Pada Gambar 4.4 dapat dilihat proses hitung jarak *Euclidean*.



Gambar 4.4 Proses Hitung *Euclidean*

4.1.2 Proses *Naïve Bayes*

Proses *Naïve Bayes* diawali dengan memproses data uji yang diperoleh dari proses perhitungan *K-Nearest Neighbor* bersifat kategoris. Kemudian dilakukan perhitungan prior tiap kelas, yang dilanjutkan dengan menghitung probabilitas masing-masing atribut, yaitu likelihood. Tahap berikutnya yaitu mengalikan nilai prior dan likelihood pada tiap kelas, kemudian perkalian tersebut menghasilkan nilai posterior. Akhir dari proses ini adalah membandingkan label kelas dengan nilai posterior terbesar, sehingga akan didapatkan hasil berupa tingkat risiko penyakit stroke. Gambar 4.5 menunjukkan proses algoritme *Naïve Bayes* dalam diagram alir.



Gambar 4.5 Diagram Alir Proses *Naïve Bayes*

4.2 Deskripsi Data

Pengolahan data dilakukan pada 150 data kondisi kesehatan masyarakat Kota Malang, dengan 8 atribut, yaitu tekanan darah, riwayat fibrilasi atrium, merokok, kolesterol, diabetes, aktifitas fisik, diet, riwayat keluarga. Berikut merupakan penjelasan mengenai 8 atribut dari diagnosis tingkat risiko penyakit stroke:

1. Tekanan Darah

Terdapat 2 macam pengukuran tekanan darah, yaitu tekanan darah sistolik dan tekanan darah diastolik. Tabel 4.1, 4.2 dan 4.3 menunjukkan cara menentukan tekanan darah. Tabel 4.1 menunjukkan aturan tingkat tekanan darah. Tabel 4.2 menunjukkan penentuan tekanan darah. Tabel 4.3 merupakan table hasil dari penentuan tingkat tekanan darah.

Tabel 4.1 Aturan Tekanan Darah

Kode	Jenis Tekanan Darah	Rentang
S1	Sistolik Tinggi	>140
S2	Sistolik Sedang	120-140
S3	Sistolik Rendah	<120
D1	Diastolik Tinggi	>90

Kode	Jenis Tekanan Darah	Rentang
D2	Diastolik Sedang	80-90
D3	Diastolik Rendah	<80

Tabel 4.2 Penentuan Tekanan Darah

Kode	Tekanan Darah
TD1	Jika S1 dan D1 maka Tekanan Darah Tinggi
TD2	Jika S1 dan D2 maka Tekanan Darah Tinggi
TD3	Jika S1 dan D3 maka Tekanan Darah Tinggi
TD4	Jika S2 dan D1 maka Tekanan Darah Tinggi
TD5	Jika S2 dan D2 maka Tekanan Darah Sedang
TD6	Jika S2 dan D3 maka Tekanan Darah Sedang
TD7	Jika S3 dan D1 maka Tekanan Darah Tinggi
TD8	Jika S3 dan D2 maka Tekanan Darah Sedang
TD9	Jika S3 dan D3 maka Tekanan Darah Rendah

Tabel 4.3 Nilai Tekanan Darah

Tekanan Darah	Tingkat
Tekanan Darah > 140/90	Tinggi
Tekanan Darah 120-140/80-90	Sedang
Tekanan Darah < 120/80	Rendah

2. Riwayat Fibrilasi Atrium

Riwayat Fibrilasi Atrium merupakan suatu kondisi ketika serambi atau atrium jantung berdenyut tidak beraturan. Terdapat 3 kelompok dalam Riwayat Fibrilasi Atrium yaitu tidak merasakan detak jantung, detak jantung teratur dan detak jantung tidak teratur.

3. Merokok

Merokok dapat menyebabkan stroke, karena menghasilkan variasi detak jantung. Merokok dikelompokkan menjadi tidak merokok, kadang dan merokok.

4. Kolesterol

Faktor ke empat adalah kadar kolesterol pada tubuh. Banyaknya kadar kolesterol sangat berpengaruh pada tubuh. Kolesterol rendah merupakan tubuh dengan kadar kolesterol di bawah 200 mg/dL. Kemudian, kadar kolesterol antara 200-240 mg/dL termasuk dalam kadar kolesterol sedang. Kadar kolesterol diatas 240 mg/dL tergolong dalam kadar kolesterol tinggi.

5. Diabetes

Gula darah acak atau gula darah sewaktu menjadi penentu penyakit diabetes. Berikut merupakan rentang untuk diabetes:

- a. Jika gula darah acak mencapai lebih dari 200
- b. Jika gula darah acak mencapai nilai 160-200
- c. Jika gula darah acak memiliki nilai kurang dari 160

6. Aktifitas Fisik

Aktifitas fisik merupakan kegiatan yang dilakukan agar tubuh tetap sehat seperti olahraga. Ketika seseorang teratur dalam berolahraga, maka risiko terkena penyakit stroke akan semakin rendah. Aktifitas fisik dapat dikategorikan menjadi tiga, yaitu tidak pernah berolahraga, kadang-kadang dan rutin berolahraga.

7. Diet

Untuk mengukur indeks nilai diet dapat dilakukan dengan menggunakan perhitungan IMT atau Indeks Masa Tubuh. Persamaan 4.1 menunjukkan perhitungan indeks diet menggunakan IMT dan penentuan diet dapat dilihat pada Tabel 4.4.

$$IMT = \frac{BB}{TB^2} \tag{4.1}$$

Keterangan:

BB: Berat Badan (Kilogram)

TB: Tinggi Badan (meter)

Tabel 4.4 Nilai Diet

Diet	Rentang
Obesitas	IMT > 30
<i>Overweight</i>	IMT antara 25-30
Normal	IMT < 25

8. Riwayat Keluarga

Faktor risiko dari riwayat keluarga dibatasi kakek, nenek, ayah, dan ibu. Pasien yang memiliki riwayat keluarga dengan penyakit stroke, maka tingkat resikonya adalah stroke tinggi. Pasien yang tidak memiliki riwayat keluarga dengan penyakit stroke, penentuan tingkat resiko tergantung dari kriteria lainnya.

4.3 Perhitungan Manual

Perhitungan manual dilakukan untuk mengetahui proses perhitungan sebenarnya yang terjadi di dalam sistem. Perhitungan manual akan mempermudah dalam pembuatan program.

Tabel 4.5 Data Latih

No	Tekanan Darah	Diabetes	Riwayat Keluarga	Merokok	Kolesterol	Aktifitas Fisik	Diet	Riwayat Fabrilasi Atrium	Tingkat Risiko
1	Rendah	180	Iya	Tidak	100	Rutin Olahraga	20.3	Beraturan	Tinggi
2	Tinggi	230	Tidak Tahu	Merokok	251	Kadang	32.44	Tidak Beraturan	Tinggi
3	Tinggi	210	Tidak Tahu	Merokok	241	Tidak Pernah Olahraga	32.18	Tidak Tahu	Tinggi
4	Tinggi	250	Tidak Tahu	Merokok	249	Kadang	26.35	Tidak Tahu	Tinggi
5	Tinggi	280	Tidak Ada	Merokok	205	Kadang	30.99	Tidak Beraturan	Tinggi
6	Tinggi	265	Tidak Tahu	Merokok	239	Tidak Pernah Olahraga	31.02	Tidak Beraturan	Tinggi
7	Tinggi	350	Tidak Ada	Merokok	247	Tidak Pernah Olahraga	34.06	Tidak Beraturan	Tinggi
8	Sedang	220	Tidak Tahu	Merokok	260	Tidak Pernah Olahraga	31.1	Tidak Beraturan	Tinggi
9	Tinggi	256	Tidak Ada	Merokok	186	Tidak Pernah Olahraga	30.48	Tidak Beraturan	Tinggi
10	Tinggi	270	Tidak Tahu	Merokok	347	Tidak Pernah Olahraga	30.57	Tidak Tahu	Tinggi
11	Tinggi	298	Tidak Tahu	Merokok	248	Tidak Pernah Olahraga	30.44	Tidak Tahu	Tinggi
12	Tinggi	240	Tidak Tahu	Kadang	248	Kadang	33.4	Tidak Tahu	Sedang
13	Sedang	120	Tidak Ada	Kadang	120	Kadang	23.1	Tidak Tahu	Sedang
14	Rendah	210	Tidak Tahu	Kadang	260	Kadang	34.25	Tidak Tahu	Sedang
15	Tinggi	180	Tidak Tahu	Kadang	274	Kadang	37.05	Tidak Tahu	Sedang
16	Sedang	190	Tidak Tahu	Kadang	275	Kadang	31.25	Tidak Tahu	Sedang
17	Rendah	210	Tidak Ada	Tidak	145	Tidak Pernah Olahraga	32.7	Beraturan	Sedang
18	Sedang	206	Tidak Tahu	Merokok	210	Kadang	24.6	Tidak Tahu	Sedang
19	Sedang	234	Tidak Tahu	Kadang	205	Kadang	31.89	Tidak Tahu	Sedang
20	Rendah	225	Tidak Tahu	Kadang	210	Kadang	38.14	Tidak Tahu	Sedang
21	Tinggi	182	Tidak Tahu	Kadang	208	Kadang	34.25	Tidak Tahu	Sedang
22	Sedang	186	Tidak Tahu	Kadang	219	Kadang	30.67	Tidak Tahu	Sedang
23	Rendah	300	Tidak Ada	Tidak	100	Rutin Olahraga	33.9	Tidak Beraturan	Rendah
24	Sedang	129	Tidak Ada	Kadang	209	Rutin Olahraga	25.41	Beraturan	Rendah
25	Rendah	137	Tidak Ada	Kadang	217	Rutin Olahraga	27.64	Beraturan	Rendah

No	Tekanan Darah	Diabetes	Riwayat Keluarga	Merokok	Kolesterol	Aktifitas Fisik	Diet	Riwayat Fabrilasi Atrium	Tingkat Risiko
26	Sedang	168	Tidak Ada	Kadang	189	Rutin Olahraga	28.2	Beraturan	Rendah
27	Rendah	172	Tidak Ada	Kadang	192	Rutin Olahraga	28.84	Beraturan	Rendah
28	Sedang	149	Tidak Ada	Kadang	194	Rutin Olahraga	25.36	Beraturan	Rendah
29	Rendah	138	Tidak Ada	Kadang	186	Rutin Olahraga	27.99	Beraturan	Rendah
30	Sedang	172	Tidak Ada	Tidak	164	Rutin Olahraga	25.22	Beraturan	Rendah
31	Rendah	180	Tidak Ada	Tidak	180	Rutin Olahraga	27.27	Beraturan	Rendah
32	Sedang	139	Tidak Ada	Tidak	158	Rutin Olahraga	28.41	Beraturan	Rendah
33	Rendah	149	Tidak Ada	Tidak	167	Rutin Olahraga	29.21	Beraturan	Rendah
34	Sedang	169	Tidak Ada	Kadang	210	Rutin Olahraga	28.08	Beraturan	Rendah
35	Rendah	170	Tidak Ada	Kadang	205	Rutin Olahraga	21.08	Beraturan	Rendah
36	Sedang	151	Tidak Ada	Kadang	217	Rutin Olahraga	24.22	Beraturan	Rendah
37	Tinggi	296	Tidak Tahu	Merokok	271	Tidak Pernah Olahraga	31.46	Tidak Beraturan	Tinggi
38	Tinggi	305	Tidak Tahu	Merokok	326	Tidak Pernah Olahraga	21.3	Tidak Beraturan	Tinggi
39	Sedang	193	Tidak Tahu	Kadang	285	Kadang	31.05	Beraturan	Sedang
40	Rendah	195	Tidak Tahu	Kadang	260	Kadang	31.43	Beraturan	Sedang

Tabel 4.6 Data Uji

No	Tekanan Darah	Diabetes	Riwayat Keluarga	Merokok	Kolesterol	Aktifitas Fisik	Diet	Riwayat Fabrilasi Atrium	Tingkat Risiko
1	Rendah	159	Tidak Ada	Kadang	226	Rutin Olahraga	20.2	Beraturan	Rendah

Contoh perhitungan:

Tabel 4.5 dan 4.6 merupakan tabel data latih dan data uji. Data latih berisi 40 data pasien yang diambil secara acak dari keseluruhan 150 data. Sedangkan data uji berisi 1 buah data yang akan diujikan menggunakan metode yang diteliti.

4.3.1 Menentukan Nilai K

Langkah pertama yang dilakukan adalah menentukan nilai K untuk mengetahui nilai mayor pada akhir perhitungan *k-nearest neighbor*. Nilai K yang digunakan pada perhitungan ini adalah $K=5$.

4.3.2 Menghitung Jarak *Euclidean*

Berikutnya dilakukan perhitungan antara data latih dan data uji yang bersifat numerik menggunakan rumus jarak *Euclidean*. Data numerik yang digunakan adalah data diabetes, kolesterol dan diet. Tabel 4.7 dan 4.8 menunjukkan data latih dan data uji yang bersifat numerik.

Tabel 4.7 Data Latih Numerik

No	Diabetes	Kolesterol	Diet	Tingkat Risiko
1	180	100	20.3	Tinggi
2	230	251	32.44	Tinggi
3	210	241	32.18	Tinggi
4	250	249	26.35	Tinggi
5	280	205	30.99	Tinggi
6	265	239	31.02	Tinggi
7	350	247	34.06	Tinggi
8	220	260	31.1	Tinggi
9	256	186	30.48	Tinggi
10	270	347	30.57	Tinggi
11	298	248	30.44	Tinggi
12	240	248	33.4	Sedang
13	120	120	23.1	Sedang
14	210	260	34.25	Sedang
15	180	274	37.05	Sedang
16	190	275	31.25	Sedang
17	210	145	32.7	Sedang
18	206	210	24.6	Sedang
19	234	205	31.89	Sedang
20	225	210	38.14	Sedang
21	182	208	34.25	Sedang
22	186	219	30.67	Sedang
23	300	100	33.9	Rendah
24	129	209	25.41	Rendah
25	137	217	27.64	Rendah

No	Diabetes	Kolesterol	Diet	Tingkat Risiko
26	168	189	28.2	Rendah
27	172	192	28.84	Rendah
28	149	194	25.36	Rendah
29	138	186	27.99	Rendah
30	172	164	25.22	Rendah
31	180	180	27.27	Rendah
32	139	158	28.41	Rendah
33	149	167	29.21	Rendah
34	169	210	28.08	Rendah
35	170	205	21.08	Rendah
36	151	217	24.22	Rendah
37	296	271	31.46	Tinggi
38	305	326	21.3	Tinggi
39	193	285	31.05	Sedang
40	195	260	31.43	Sedang

Tabel 4.8 Data Uji Numerik

No	Diabetes	Kolesterol	Diet	Tingkat Risiko
1	159	226	20.2	?

a. Perhitungan Kolom Pertama Baris Pertama

$$d_{(1,1)} = \sqrt{(a_{1,1} - b_{1,1})^2 + (a_{1,2} - b_{1,2})^2 + (a_{1,3} - b_{1,3})^2}$$

$$d_{(1,1)} = \sqrt{(159 - 180)^2 + (226 - 100)^2 + (20.2 - 20.3)^2}$$

$$= 127.7380523$$

b. Perhitungan Kolom Pertama Baris Kedua

$$d_{(1,1)} = \sqrt{(a_{1,1} - b_{2,1})^2 + (a_{1,2} - b_{2,2})^2 + (a_{1,3} - b_{2,3})^2}$$

$$d_{(1,1)} = \sqrt{(159 - 230)^2 + (226 - 251)^2 + (20.2 - 32.44)^2}$$

$$= 76.26150798$$

Tabel 4.9 menunjukkan nilai jarak *Euclidean* antar data latih beserta tingkat risiko. *Euclidean*

Tabel 4.9 Jarak Antar Data Latih

No	Jarak	Tingkat Risiko
1	127.7380523	Tinggi
2	76.26150798	Tinggi
3	54.49330601	Tinggi
4	94.06286462	Tinggi
5	123.2818888	Tinggi
6	107.3409167	Tinggi
7	192.6502001	Tinggi
8	70.68104413	Tinggi
9	105.4261751	Tinggi
10	164.5282252	Tinggi
11	141.1022948	Tinggi
12	84.96611089	Sedang
13	112.9841139	Sedang
14	62.88404011	Sedang
15	55.03564754	Sedang
16	59.02628652	Sedang
17	96.53108308	Sedang
18	49.84335462	Sedang
19	78.75694319	Sedang
20	70.24132402	Sedang
21	32.40991361	Sedang
22	29.79296729	Sedang
23	189.5908489	Rendah
24	34.87325766	Rendah
25	24.90689864	Rendah
26	38.91015292	Rendah
27	37.41189116	Rendah
28	33.92087263	Rendah
29	45.8441283	Rendah
30	63.54683627	Rendah
31	51.05864178	Rendah
32	71.35407557	Rendah
33	60.51594914	Rendah
34	20.4473568	Rendah
35	23.72286661	Rendah
36	12.69489661	Rendah
37	144.6402005	Tinggi
38	176.9666918	Tinggi
39	68.95449587	Sedang
40	50.77512088	Sedang

4.3.3 Mengurutkan Jarak Terdekat

Setelah menghitung jarak menggunakan *Euclidean Distance*, kemudian dilakukan pengurutan jarak untuk mendapatkan jarak terdekat yaitu jarak yang mempunyai nilai terkecil hingga terbesar seperti yang ditunjukkan pada Tabel 4.10.

Tabel 4.10 Hasil Pengurutan

No	Jarak	Tingkat Risiko
36	12.69489661	Rendah
34	20.4473568	Rendah
35	23.72286661	Rendah
25	24.90689864	Rendah
22	29.79296729	Sedang
21	32.40991361	Sedang
28	33.92087263	Rendah
24	34.87325766	Rendah
27	37.41189116	Rendah
26	38.91015292	Rendah
29	45.8441283	Rendah
18	49.84335462	Sedang
40	50.77512088	Sedang
31	51.05864178	Rendah
3	54.49330601	Tinggi
15	55.03564754	Sedang
16	59.02628652	Sedang
33	60.51594914	Rendah
14	62.88404011	Sedang
30	63.54683627	Rendah
39	68.95449587	Sedang
20	70.24132402	Sedang
8	70.68104413	Tinggi
32	71.35407557	Rendah
2	76.26150798	Tinggi
19	78.75694319	Sedang
12	84.96611089	Sedang
4	94.06286462	Tinggi
17	96.53108308	Sedang
9	105.4261751	Tinggi
6	107.3409167	Tinggi
13	112.9841139	Sedang
5	123.2818888	Tinggi
1	127.7380523	Tinggi
11	141.1022948	Tinggi

No	Jarak	Tingkat Risiko
37	144.6402005	Tinggi
10	164.5282252	Tinggi
38	176.9666918	Tinggi
23	189.5908489	Rendah
7	192.6502001	Tinggi

4.3.4 Pengambilan Data Latih Sebanyak K

Setelah mengurutkan data sesuai dengan jarak terdekat, ambil data sesuai dengan jumlah nilai K yang sudah ditentukan sebelumnya, yaitu $K=5$. Data tersebut akan digunakan sebagai data latih untuk perhitungan selanjutnya. Pengambilan data latih sebanyak K ditunjukkan pada Tabel 4.11.

Tabel 4.11 Pengambilan Data Latih Sebanyak Nilai K

No	Jarak	Tingkat Risiko
36	12.69489661	Rendah
34	20.4473568	Rendah
35	23.72286661	Rendah
25	24.90689864	Rendah
22	29.79296729	Sedang

4.3.5 Menghitung Probabilitas Prior

Langkah awal untuk perhitungan Naïve Bayes adalah dengan menghitung nilai probabilitas/kemungkinan setiap tingkat risiko yang sama. Label kelas tingkat risiko yang sama dijumlahkan, kemudian dibagi dengan jumlah data latih. Tabel 4.12 menunjukkan data latih baru yang bersifat kategoris, hasil perhitungan K -Nearest Neighbor. Tabel 4.13 menunjukkan data uji dengan sifat kategoris.

Tabel 4.12 Data Latih Kategoris

No	Tekanan Darah	Riwayat Keluarga	Merokok	Aktifitas Fisik	Riwayat Fibrilasi Atrium	Risiko
36	Sedang	Tidak Ada	Kadang	Rutin Olahraga	Beraturan	Rendah
34	Sedang	Tidak Ada	Kadang	Rutin Olahraga	Beraturan	Rendah
35	Rendah	Tidak Ada	Kadang	Rutin Olahraga	Beraturan	Rendah
25	Rendah	Tidak Ada	Kadang	Rutin Olahraga	Beraturan	Rendah
22	Sedang	Tidak Tahu	Kadang	Kadang Olahraga	Tidak Tahu	Sedang

Tabel 4.13 Data Uji Kategoris

No	Tekanan Darah	Riwayat Keluarga	Merokok	Aktifitas Fisik	Riwayat Fabrilasi Atrium
1	Rendah	Tidak Ada	Kadang	Rutin Olahraga	Beraturan



Perhitungan Prior:

$$P(\text{Risiko} = \text{Rendah}) = \frac{4}{5} = 0,8$$

$$P(\text{Risiko} = \text{Sedang}) = \frac{1}{5} = 0,2$$

$$P(\text{Risiko} = \text{Tinggi}) = \frac{0}{5} = 0$$

4.3.6 Menghitung Probabilitas Likelihood

Pada tahap ini dilakukan perhitungan probabilitas pada tiap atribut faktor risiko penyakit stroke. Tabel 4.14 menunjukkan hasil probabilitas likelihood yang sudah dimasukkan dalam table. Berikut merupakan contoh perhitungan likelihood:

$$P(\text{Tekanan Darah} = \text{Rendah} | \text{Risiko} = \text{Rendah}) = \frac{2}{4} = 0.50$$

$$P(\text{Tekanan Darah} = \text{Rendah} | \text{Risiko} = \text{Sedang}) = 0$$

$$P(\text{Tekanan Darah} = \text{Rendah} | \text{Risiko} = \text{Tinggi}) = 0$$

$$P(\text{Riwayat Keluarga} = \text{Tidak Ada} | \text{Risiko} = \text{Rendah}) = \frac{4}{4} = 1.00$$

$$P(\text{Riwayat Keluarga} = \text{Tidak Ada} | \text{Risiko} = \text{Sedang}) = 0$$

$$P(\text{Riwayat Keluarga} = \text{Tidak Ada} | \text{Risiko} = \text{Tinggi}) = 0$$

Tabel 4.14 Probabilitas Likelihood

Risiko Stroke	Tekanan Darah=Rendah	Riwayat Keluarga=Tidak Ada	Merokok=Kadang	Aktifitas Fisik=Rutin Olahraga	Riwayat Fibrilasi Atrium=Beraturan
Rendah	0.5	1	1	1	1
Sedang	0	0	1	0	0
Tinggi	0	0	0	0	0

4.3.7 Menghitung Posterior

Tahap akhir dari perhitungan *Naïve Bayes* adalah perhitungan posterior dengan cara mengkalikan semua probabilitas likelihood dalam satu kelas risiko, kemudian dikali dengan probabilitas prior kelas risiko. Kemudian seluruh hasil akan dibandingkan, label data uji dipilih jika label kelas memiliki nilai tertinggi. Berikut merupakan perhitungan posterior:

a. Kelas Risiko Rendah:

$$P(\text{Tekanan Darah} = \text{Rendah} \mid \text{Risiko} = \text{Rendah}) * P(\text{Riwayat Keluarga} = \text{Tidak Ada} \mid \text{Risiko} = \text{Rendah}) * P(\text{Merokok} = \text{Kadang} \mid \text{Risiko} = \text{Rendah}) * P(\text{Aktifitas Fisik} = \text{Rutin Olahraga} \mid \text{Risiko} = \text{Rendah}) * P(\text{Riwayat Fibrilasi Atrium} = \text{Beraturan} \mid \text{Risiko} = \text{Rendah}) * P(\text{Risiko} = \text{Rendah})$$

$$= 0.5 * 1 * 1 * 1 * 1 * 0.8$$

$$= 0.4$$

b. Kelas Risiko Sedang:

$$P(\text{Tekanan Darah} = \text{Rendah} \mid \text{Risiko} = \text{Sedang}) * P(\text{Riwayat Keluarga} = \text{Tidak Ada} \mid \text{Risiko} = \text{Sedang}) * P(\text{Merokok} = \text{Kadang} \mid \text{Risiko} = \text{Sedang}) * P(\text{Aktifitas Fisik} = \text{Rutin Olahraga} \mid \text{Risiko} = \text{Sedang}) * P(\text{Riwayat Fibrilasi Atrium} = \text{Beraturan} \mid \text{Risiko} = \text{Sedang}) * P(\text{Risiko} = \text{Sedang})$$

$$= 0 * 0 * 1 * 0 * 0 * 0.2$$

$$= 0$$

c. Kelas Risiko Tinggi:

$$P(\text{Tekanan Darah} = \text{Rendah} \mid \text{Risiko} = \text{Tinggi}) * P(\text{Riwayat Keluarga} = \text{Tidak Ada} \mid \text{Risiko} = \text{Tinggi}) * P(\text{Merokok} = \text{Kadang} \mid \text{Risiko} = \text{Tinggi}) * P(\text{Aktifitas Fisik} = \text{Rutin Olahraga} \mid \text{Risiko} = \text{Tinggi}) * P(\text{Riwayat Fibrilasi Atrium} = \text{Beraturan} \mid \text{Risiko} = \text{Tinggi}) * P(\text{Risiko} = \text{Tinggi})$$

$$= 0$$

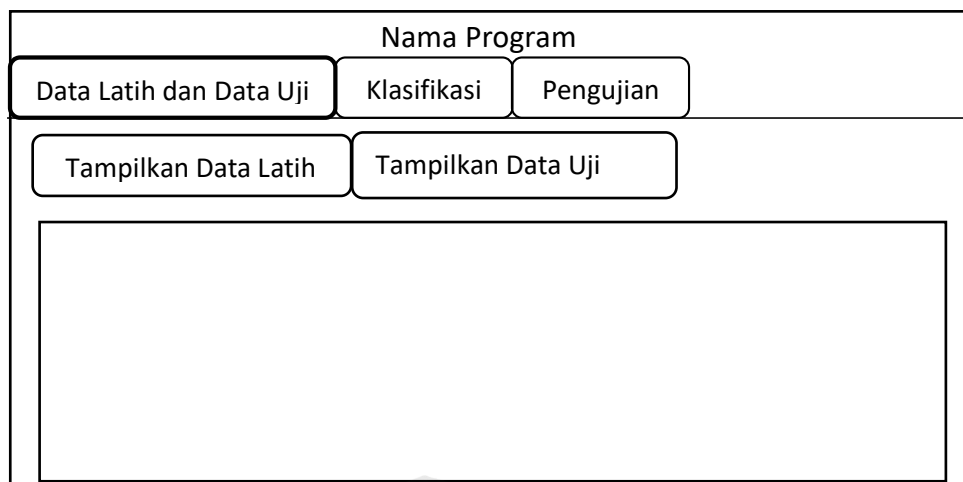
Maka dapat disimpulkan bahwa kelas risiko rendah memiliki nilai paling tinggi dibandingkan dengan kelas risiko sedang dan tinggi. Data Uji diklasifikasikan memiliki tingkat risiko rendah dengan nilai 0,4.

4.4 Perancangan Antarmuka

Perancangan antarmuka bertujuan sebagai sarana untuk menggambarkan antarmuka dalam sistem yang ingin dibuat. Antarmuka sistem sangat diperlukan agar pengguna dapat menjalankan sistem dengan baik dan benar. Penekanan pada antarmuka yaitu memiliki antarmuka yang baik, mudah dipahami dan memiliki tombol yang mudah dikenali.

4.4.1 Perancangan Antarmuka Tampilan Data

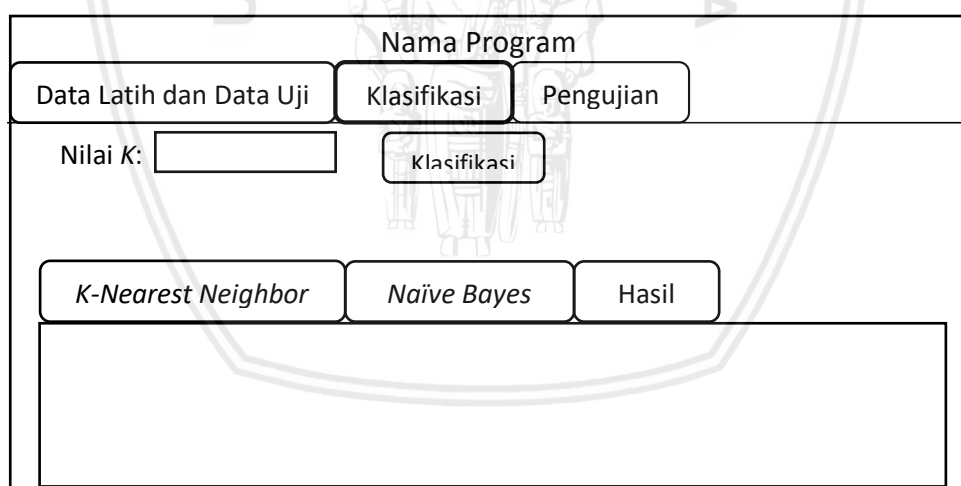
Halaman tampilan data merupakan halaman tampilan awal dari diagnosis tingkat risiko penyakit stroke yang memiliki dua menu didalamnya. Pada halaman ini terdapat menu untuk menampilkan data latih dan data uji pada aplikasi. Gambar 4.6 menunjukkan perancangan antarmuka tampilan data.



Gambar 4.6 Perancangan Antarmuka Tampilan Data

4.4.2 Perancangan Antarmuka Klasifikasi

Terdapat beberapa menu pada tampilan antarmuka klasifikasi. Pengguna dapat memasukkan nilai K untuk proses klasifikasi. Setelah memasukan nilai K , maka akan dilakukan klasifikasi pada program menggunakan metode *K-Nearest Neighbor* dan *Naïve Bayes* dengan menekan tombol klasifikasi. Gambar 4.7 menunjukkan perancangan antarmuka klasifikasi

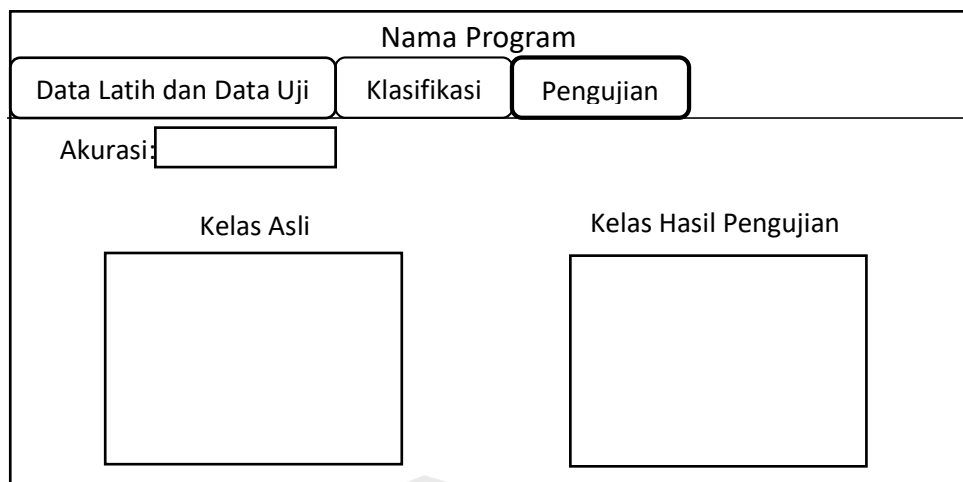


Gambar 4.7 Perancangan Antarmuka Klasifikasi

4.4.3 Perancangan Antarmuka Pengujian

Pada bagian perancangan antarmuka pengujian terdapat tampilan kelas asli dari data uji dan tampilan kelas data uji yang telah diklasifikasi serta akurasi dari hasil pengujian yang telah dilakukan. Gambar 4.8 menunjukkan perancangan antarmuka untuk pengujian.





Gambar 4.8 Perancangan Antarmuka Pengujian

4.5 Perancangan Pengujian

Perancangan ini berisi proses pengujian akurasi risiko penyakit stroke dari hasil yang dikeluarkan oleh sistem menggunakan metode *K-Nearest Neighbor* dan *Naïve Bayes*. Proses pengujian pada penelitian ini adalah pengujian pengaruh sebaran data pada kelas dengan data latih seimbang dan data latih tidak seimbang.

4.5.1 Pengujian Data Kelas Seimbang dan Tidak Seimbang

Pengujian dilakukan menggunakan data latih dengan label seimbang dan tidak seimbang. Perancangan pengujian data kelas seimbang dan tidak seimbang dapat dilihat pada Tabel 4.15.

Tabel 4.15 Pengujian Data Kelas Seimbang dan Tidak Seimbang

Data Latih	Akurasi (%)									
	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8	K=9	K=10

BAB 5 IMPLEMENTASI

Bab implementasi berisi tentang hasil penelitian berupa implementasi sistem yang dibangun bersumber pada perancangan yang telah dilakukan pada bab sebelumnya. Bab implementasi meliputi spesifikasi sistem, batasan implementasi, implementasi algoritme dan implementasi antarmuka (interface).

5.1 Spesifikasi Sistem

Pada tahap ini, dibutuhkan spesifikasi sistem yang sesuai dengan kebutuhan sistem. Spesifikasi sistem yang dibutuhkan untuk menerapkan algoritme ini antara lain berupa spesifikasi perangkat keras dan spesifikasi perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Dalam implementasi penerapan metode *K-Nearest Neighbor* dan *Naïve Bayes* untuk deteksi risiko penyakit stroke, memiliki spesifikasi perangkat keras seperti berikut:

Tabel 5.1 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
Prosesor	AMD A10-9600P RADEON R5, 2.40 GHz
RAM	4.00 GB

5.1.2 Spesifikasi Perangkat Lunak

Dalam implementasi penerapan metode *K-Nearest Neighbor* dan *Naïve Bayes* untuk deteksi risiko penyakit stroke, memiliki spesifikasi perangkat lunak seperti berikut:

Tabel 5.2 Spesifikasi Perangkat Lunak

Nama	Spesifikasi
Sistem Operasi	Windows 10
Bahasa Pemrograman	Java
<i>Tools</i> pemrograman	Netbeans IDE 8.2

5.2 Batasan Implementasi

Batasan implementasi mengacu pada penjelasan yang telah dijelaskan pada bab pendahuluan, bahwa pada penelitian ini menerapkan algoritme *K-Nearest Neighbor* dan *Naïve Bayes* untuk deteksi risiko penyakit stroke. Oleh sebab itu, diberikan beberapa batasan implementasi terhadap analisis algoritma ini, sebagai berikut:

1. Dibangun dengan bahasa pemrograman Java
2. Metode klasifikasi menggunakan *K-Nearest Neighbor* dan *Naïve Bayes*
3. *Input* pada sistem berupa nilai *K*

4. *Output* pada sistem berupa hasil klasifikasi, apakah penderita termasuk dalam kelas risiko rendah, sedang atau tinggi.

5.3 Implementasi Algoritme

Implementasi algoritme berisi tentang hasil penelitian berupa sistem yang dibangun berdasarkan pada perancangan sebelumnya, yaitu potongan kode program penerapan metode *K-Nearest Neighbor* dan *Naïve Bayes* untuk deteksi risiko penyakit stroke.

5.3.1 Implementasi Algoritme *K-Nearest Neighbor*

Proses klasifikasi yang pertama menggunakan algoritme *K-Nearest Neighbor*. Terdapat 2 proses perhitungan *K-Nearest Neighbor*, yaitu perhitungan dan pengurutan jarak.

5.3.1.1 Implementasi Algoritme Perhitungan Jarak

Perhitungan jarak pada algoritme *K-Nearest Neighbor* menggunakan perhitungan *euclidean distance*. Perhitungan *euclidean distance* ditunjukkan pada Kode Program 5.1.

```

1 public double[][] euclidean(double[][] y, double[][] x,
2 int i, int z) {
3     double[][] distance = new double[y.length][x.length];
4     double sum = 0;
5     for (int j = 0; j < x[i].length; j++) {
6         sum += Math.pow(y[z][j] - x[i][j], 2);
7     }
8     distance[z][i] = Math.sqrt(sum);
9     return distance;
10 }

```

Kode Program 5.1 Implementasi Algoritme Perhitungan Jarak

Penjelasan implementasi pada Kode Program 5.1 adalah sebagai berikut:

1. Baris 3-4 merupakan inisialisasi dari nilai variable *distance/euclidean* dan *sum*
2. Baris 5 digunakan sebagai perulangan sebanyak kolom *x*, yaitu data latih yang digunakan untuk melakukan perhitungan jarak antara data uji dengan data latih
3. Baris 6 merupakan proses perhitungan jarak menggunakan *euclidean distance*
4. Baris 8 digunakan dalam proses penyimpanan nilai jarak ke dalam variable *distance*
5. Baris 9 mengembalikan nilai *distance*

5.3.1.2 Implementasi Algoritme Pengurutan Jarak

Langkah selanjutnya adalah pengurutan jarak pada proses sebelumnya, dengan cara mengurutkan dari jarak terdekat hingga terjauh. Implementasi algoritme pengurutan jarak ditunjukkan pada Kode Program 5.2.


```

1 public void sorting(double[][] jarak, String[][] kls,
2 int[][] Ind) {
3
4     int n = DataLatih.length;
5     String[][] kelasBaru = new String [DataUji.length]
6     [DataLatih.length];
7     for (int i = 0; i < DataUji.length; i++) {
8         for (int z = 1; z < n; z++) {
9             for (int j = 0; j < n - z; j++) {
10                if (jarak[i][j] > jarak[i][j + 1]) {
11                    double temp = jarak[i][j];
12                    jarak[i][j] = jarak[i][j + 1];
13                    jarak[i][j + 1] = temp;
14                    // sorting kategori berdasarkan jarak
15                    String a = kls[i][j];
16                    kls[i][j] = kls[i][j + 1];
17                    kls[i][j + 1] = a;
18
19                    int b = Ind[i][j];
20                    Ind[i][j] = Ind[i][j + 1];
21                    Ind[i][j + 1] = b;
22                }
23            }
24        }
25    }
26 }

```

Kode Program 5.2 Implementasi Algoritme Pengurutan Jarak

Penjelasan dari Kode Program 5.2 adalah sebagai berikut:

6. Baris 4-6 merupakan inisialisasi dari nilai variable n dan kelasBaru
7. Baris 7-9 merupakan method perulangan untuk pengurutan jarak
8. Baris 10-13 merupakan proses pengurutan jarak dari yang terdekat hingga yang terjauh
9. Baris 15-17 merupakan proses pengurutan kelas berdasarkan pada jarak
10. Baris 19-21 merupakan proses pengurutan nomor index pada data berdasarkan jarak

5.3.2 Implementasi Algoritme Naïve Bayes

Metode klasifikasi yang dilakukan setelah proses *K-Nearest Neighbor* selesai adalah klasifikasi menggunakan Algoritme *Naïve Bayes*. Terdapat tiga proses utama dalam perhitungan *Naïve Bayes*, yaitu perhitungan prior pada tiap kelas, perhitungan likelihood pada setiap gejala penyakit dan perhitungan posterior.

5.3.2.1 Implementasi Algoritme Prior

Proses awal yang dilakukan pada algoritme *Naïve Bayes* adalah melakukan perhitungan prior. Perhitungan peluang pada kelas yang berada pada data latih, menghasilkan nilai prior. Berikut ini merupakan implementasi algoritme perhitungan prior yang ditunjukkan pada Kode Program 5.3.

```

1 public void Prior() {
2     int totalBaris = K;
3     int TotalRRendah, TotalRSedang, TotalRTinggi;
4     peluangKelasRendah = new double[DataUji.length];
5     peluangKelasSedang = new double[DataUji.length];
6     peluangKelasTinggi = new double[DataUji.length];
7
8     for (int i = 0; i < DataUji.length; i++) {
9         TotalRRendah = 0;
10        TotalRSedang = 0;
11        TotalRTinggi = 0;
12        for (int j = 0; j < K; j++) {
13            if (kelas[i][j].equalsIgnoreCase("Risiko
14                Rendah")) {
15                TotalRRendah++;
16            } else if (kelas[i][j].equalsIgnoreCase("Risiko
17                Sedang")) {
18                TotalRSedang++;
19            } else {
20                TotalRTinggi++;
21            }
22        }
23        peluangKelasRendah[i] = (double) TotalRRendah /
24            totalBaris;
25        peluangKelasSedang[i] = (double) TotalRSedang /
26            totalBaris;
27        peluangKelasTinggi[i] = (double) TotalRTinggi /
28            totalBaris;
29    }
30 }

```

Kode Program 5.3 Implementasi Algoritme Prior

Penjelasan dari Kode Program 5.3 adalah sebagai berikut:

1. Baris 2 merupakan inialisasi nilai dari totalBaris
2. Baris 4-6 merupakan inialisasi ukuran *array* untuk variabel yang digunakan pada perhitungan prior
3. Baris 8-11 merupakan perulangan untuk total masing-masing label kelas risiko penyakit stroke
4. Baris 12-21 merupakan proses perhitungan jumlah masing-masing label kelas risiko penyakit stroke
5. Baris 23-30 merupakan proses perhitungan prior masing-masing label kelas stroke

5.3.2.2 Implementasi Algoritme Likelihood

Setelah melakukan perhitungan prior, kemudian dilanjutkan dengan perhitungan likelihood, yaitu menghitung peluang kemunculan karakteristik-karakteristik sampel masing-masing atribut pada masing-masing kelas. Implementasi algoritme perhitungan likelihood ditunjukkan pada *Kode Program 5.4*.

```
1 public void Likelihood() {
2     int RisikoRendah, RisikoSedang, RisikoTinggi;
3     int[] totalKelasRRendah = new int[DataUji.length];
4     int[] totalKelasRSedang = new int[DataUji.length];
5     int[] totalKelasRTinggi = new int[DataUji.length];
6     peluangTD1 = new double[DataUji.length];
7     peluangRK1 = new double[DataUji.length];
8     peluangMerokok1 = new double[DataUji.length];
9     peluangAF1 = new double[DataUji.length];
10    peluangRiwayatFA1 = new double[DataUji.length];
11    peluangTD2 = new double[DataUji.length];
12    peluangRK2 = new double[DataUji.length];
13    peluangMerokok2 = new double[DataUji.length];
14    peluangAF2 = new double[DataUji.length];
15    peluangRiwayatFA2 = new double[DataUji.length];
16    peluangTD3 = new double[DataUji.length];
17    peluangRK3 = new double[DataUji.length];
18    peluangMerokok3 = new double[DataUji.length];
19    peluangAF3 = new double[DataUji.length];
20    peluangRiwayatFA3 = new double[DataUji.length];
21
22    for (int i = 0; i < DataUji.length; i++) {
23        RisikoRendah = 0;
24        RisikoSedang = 0;
25        RisikoTinggi = 0;
26        for (int j = 0; j < K; j++) {
27            if (kelas[i][j].equalsIgnoreCase("Risiko
28 Rendah")) {
29                RisikoRendah++;
30            } else if
31 (kelas[i][j].equalsIgnoreCase("Risiko Sedang")) {
32                RisikoSedang++;
33            } else {
34                RisikoTinggi++;
35            }
36        }
37        totalKelasRRendah[i] = RisikoRendah++;
38        totalKelasRSedang[i] = RisikoSedang++;
39        totalKelasRTinggi[i] = RisikoTinggi++;
40    }
41    for (int i = 0; i < DataUji.length; i++) {
42        if (jmlTD1[i] == 0 || totalKelasRRendah[i] ==
43 0) {
44            peluangTD1[i] = 0;
45        } else {
46            peluangTD1[i] = (double) jmlTD1[i] /
47 totalKelasRRendah[i];
48        }
49        if (jmlRK1[i] == 0 || totalKelasRRendah[i] ==
50 0) {
51            peluangRK1[i] = 0;
52        } else {
53            peluangRK1[i] = (double) jmlRK1[i] /
54 totalKelasRRendah[i];
55        }
56        if (jmlMerokok1[i] == 0 || totalKelasRRendah[i]
57 == 0) {
58            peluangMerokok1[i] = 0;
59        } else {
```

```
60         peluangMerokok1[i] = (double)
61     jmlMerokok1[i] / totalKelasRRendah[i];
62     }
63     if (jmlAF1[i] == 0 || totalKelasRRendah[i] ==
64     0) {
65         peluangAF1[i] = 0;
66     } else {
67         peluangAF1[i] = (double) jmlAF1[i] /
68     totalKelasRRendah[i];
69     }
70     if (jmlRiwayatFA1[i] == 0 ||
71     totalKelasRRendah[i] == 0) {
72         peluangRiwayatFA1[i] = 0;
73     } else {
74         peluangRiwayatFA1[i] = (double)
75     jmlRiwayatFA1[i] / totalKelasRRendah[i];
76     }
77     if (jmlTD2[i] == 0 || totalKelasRSedang[i] ==
78     0) {
79         peluangTD2[i] = 0;
80     } else {
81         peluangTD2[i] = (double) jmlTD2[i] /
82     totalKelasRSedang[i];
83     }
84     if (jmlRK2[i] == 0 || totalKelasRSedang[i] ==
85     0) {
86         peluangRK2[i] = 0;
87     } else {
88         peluangRK2[i] = (double) jmlRK2[i] /
89     totalKelasRSedang[i];
90     }
91     if (jmlMerokok2[i] == 0 || totalKelasRSedang[i]
92     == 0) {
93         peluangMerokok2[i] = 0;
94     } else {
95         peluangMerokok2[i] = (double)
96     jmlMerokok2[i] / totalKelasRSedang[i];
97     }
98     if (jmlAF2[i] == 0 || totalKelasRSedang[i] ==
99     0) {
100         peluangAF2[i] = 0;
101     } else {
102         peluangAF2[i] = (double) jmlAF2[i] /
103     totalKelasRSedang[i];
104     }
105     if (jmlRiwayatFA2[i] == 0 ||
106     totalKelasRSedang[i] == 0) {
107         peluangRiwayatFA2[i] = 0;
108     } else {
109         peluangRiwayatFA2[i] = (double)
110     jmlRiwayatFA2[i] / totalKelasRSedang[i];
111     }
112     if (jmlTD3[i] == 0 || totalKelasRTinggi[i] ==
113     0) {
114         peluangTD3[i] = 0;
115     } else {
116         peluangTD3[i] = (double) jmlTD3[i] /
117     totalKelasRTinggi[i];
118     }
```

```

119         if (jmlRK3[i] == 0 || totalKelasRTinggi[i] ==
120     0) {
121             peluangRK3[i] = 0;
122         } else {
123             peluangRK3[i] = (double) jmlRK3[i] /
124     totalKelasRTinggi[i];
125         }
126         if (jmlMerokok3[i] == 0 || totalKelasRTinggi[i]
127     == 0) {
128             peluangMerokok3[i] = 0;
129         } else {
130             peluangMerokok3[i] = (double)
131     jmlMerokok3[i] / totalKelasRTinggi[i];
132         }
133         if (jmlAF3[i] == 0 || totalKelasRTinggi[i] ==
134     0) {
135             peluangAF3[i] = 0;
136         } else {
137             peluangAF3[i] = (double) jmlAF3[i] /
138     totalKelasRTinggi[i];
139         }
140         if (jmlRiwayatFA3[i] == 0 ||
141     totalKelasRTinggi[i] == 0) {
142             peluangRiwayatFA3[i] = 0;
143         } else {
144             peluangRiwayatFA3[i] = (double)
145     jmlRiwayatFA3[i] / totalKelasRTinggi[i];
146         }
147     }
148 }
149

```

Kode Program 5.4 Implementasi Algoritme Likelihood

Penjelasan dari Kode Program 5.4 adalah sebagai berikut:

1. Baris 2 adalah variabel kelas yang dideklarasikan untuk digunakan dalam algoritme likelihood
2. Baris 3-5 merupakan proses pembuatan array satu dimensi untuk total tiap label kelas
3. Baris 6-20 merupakan inisialisasi ukuran *array* masing-masing atribut untuk masing-masing label kelas
4. Baris 22 merupakan perulangan sebanyak data uji untuk menghitung jumlah masing-masing label kelas
5. Baris 23-25 merupakan inisialisasi nilai variabel untuk proses perhitungan jumlah label kelas
6. Baris 26 merupakan perulangan sebanyak nilai *K* yang di input oleh user
7. Baris 27-34 merupakan perintah untuk memberikan syarat kondisi pada label kelas
8. Baris 37-39 untuk perhitungan jumlah pada setiap label kelas

9. Baris 41 merupakan perulangan sebanyak data uji untuk menghitung likelihood
10. Baris 42-48 untuk menghitung likelihood, jika peluang atribut tekanan darah dengan risiko rendah tidak ada maka bernilai 0, sedangkan jika ada maka peluang tekanan darah dengan kelas risiko rendah akan dibagi total kelas risiko rendah
11. Baris 49-55 untuk menghitung likelihood, jika peluang atribut riwayat keluarga dengan risiko rendah tidak ada maka bernilai 0, sedangkan jika ada maka peluang riwayat keluarga dengan kelas risiko rendah akan dibagi total kelas risiko rendah
12. Baris 56-62 untuk menghitung likelihood, jika peluang atribut merokok dengan risiko rendah tidak ada maka bernilai 0, sedangkan jika ada maka peluang merokok dengan kelas risiko rendah akan dibagi total kelas risiko rendah
13. Baris 63-69 untuk menghitung likelihood, jika peluang atribut aktifitas fisik dengan risiko rendah tidak ada maka bernilai 0, sedangkan jika ada maka peluang aktifitas fisik dengan kelas risiko rendah akan dibagi total kelas risiko rendah
14. Baris 70-76 untuk menghitung likelihood, jika peluang atribut fibrilasi atrium dengan risiko rendah tidak ada maka bernilai 0, sedangkan jika ada maka peluang fibrilasi atrium dengan kelas risiko rendah akan dibagi total kelas risiko rendah
15. Baris 77-83 untuk menghitung likelihood, jika peluang atribut tekanan darah dengan risiko sedang tidak ada maka bernilai 0, sedangkan jika ada maka peluang tekanan darah dengan kelas risiko sedang akan dibagi total kelas risiko sedang
16. Baris 84-90 untuk menghitung likelihood, jika peluang atribut riwayat keluarga dengan risiko sedang tidak ada maka bernilai 0, sedangkan jika ada maka peluang riwayat keluarga dengan kelas risiko sedang akan dibagi total kelas risiko sedang
17. Baris 91-97 untuk menghitung likelihood, jika peluang atribut merokok dengan risiko sedang tidak ada maka bernilai 0, sedangkan jika ada maka peluang merokok dengan kelas risiko sedang akan dibagi total kelas risiko sedang
18. Baris 98-104 untuk menghitung likelihood, jika peluang atribut aktifitas fisik dengan risiko sedang tidak ada maka bernilai 0, sedangkan jika ada maka peluang aktifitas fisik dengan kelas risiko sedang akan dibagi total kelas risiko sedang
19. Baris 105-111 untuk menghitung likelihood, jika peluang fibrilasi atrium dengan risiko sedang tidak ada maka bernilai 0, sedangkan jika ada maka

- peluang fibrilasi atrium dengan kelas risiko sedang akan dibagi total kelas risiko sedang
20. Baris 112-118 untuk menghitung likelihood, jika peluang atribut tekanan darah dengan risiko tinggi tidak ada maka bernilai 0, sedangkan jika ada maka peluang tekanan darah dengan kelas risiko tinggi akan dibagi total kelas risiko tinggi
 21. Baris 119-125 untuk menghitung likelihood, jika peluang atribut riwayat keluarga dengan risiko tinggi tidak ada maka bernilai 0, sedangkan jika ada maka peluang riwayat keluarga dengan kelas risiko tinggi akan dibagi total kelas risiko tinggi
 22. Baris 126-132 untuk menghitung likelihood, jika peluang atribut merokok dengan risiko tinggi tidak ada maka bernilai 0, sedangkan jika ada maka peluang tekanan darah dengan kelas merokok akan dibagi total kelas risiko tinggi
 23. Baris 133-139 untuk menghitung likelihood, jika peluang atribut aktifitas fisik dengan risiko tinggi tidak ada maka bernilai 0, sedangkan jika ada maka peluang aktifitas fisik dengan kelas risiko tinggi akan dibagi total kelas risiko tinggi
 24. Baris 140-146 untuk menghitung likelihood, jika peluang atribut fibrilasi atrium dengan risiko tinggi tidak ada maka bernilai 0, sedangkan jika ada maka peluang fibrilasi atrium dengan kelas risiko tinggi akan dibagi total kelas risiko tinggi

5.3.2.3 Implementasi Algoritme Posterior

Tahap paling akhir dalam perhitungan algoritme *Naïve Bayes* adalah perhitungan posterior. Posterior diawali dengan mengalikan nilai prior dan nilai likelihood pada tiap kelas, kemudian hasil akhir nilai posterior akan digunakan untuk menentukan klasifikasi kelas sample dengan cara membandingkan nilai posterior tiap kelas, kemudian akan diambil nilai terbesar pada perhitungan tersebut. Implementasi algoritme perhitungan posterior ditunjukkan pada Kode Program 5.5.

```

1      public void Posterior() {
2          totalPeluangRendah = new double[DataUji.length];
3          totalPeluangSedang = new double[DataUji.length];
4          totalPeluangTinggi = new double[DataUji.length];
5          HasilAkhir = new String[DataUji.length];
6          for (int i = 0; i < DataUji.length; i++) {
7              totalPeluangRendah[i] = peluangKelasRendah[i] *
8  peluangTD1[i] * peluangRK1[i] * peluangMerokok1[i] *
9  peluangAF1[i] * peluangRiwayatFA1[i];
10             totalPeluangSedang[i] = peluangKelasSedang[i] *
11  peluangTD2[i] * peluangRK2[i] * peluangMerokok2[i] *
12  peluangAF2[i] * peluangRiwayatFA2[i];
13
14

```

```

15         totalPeluangTinggi[i] = peluangKelasTinggi[i] *
16         peluangTD3[i] * peluangRK3[i] * peluangMerokok3[i] *
17         peluangAF3[i] * peluangRiwayatFA3[i];
18
19         if (totalPeluangRendah[i] >
20         totalPeluangSedang[i] && totalPeluangRendah[i] >
21         totalPeluangTinggi[i]) {
22             HasilAkhir[i] = "Risiko Rendah";
23         } else if (totalPeluangRendah[i] <
24         totalPeluangSedang[i] && totalPeluangSedang[i] >
25         totalPeluangTinggi[i]) {
26             HasilAkhir[i] = "Risiko Sedang";
27         } else if (totalPeluangTinggi[i] >
28         totalPeluangRendah[i] && totalPeluangSedang[i] <
29         totalPeluangTinggi[i]) {
30             HasilAkhir[i] = "Risiko Tinggi";
31         } else {
32             HasilAkhir[i] = kelasMayor[i];
33         }
34     }

```

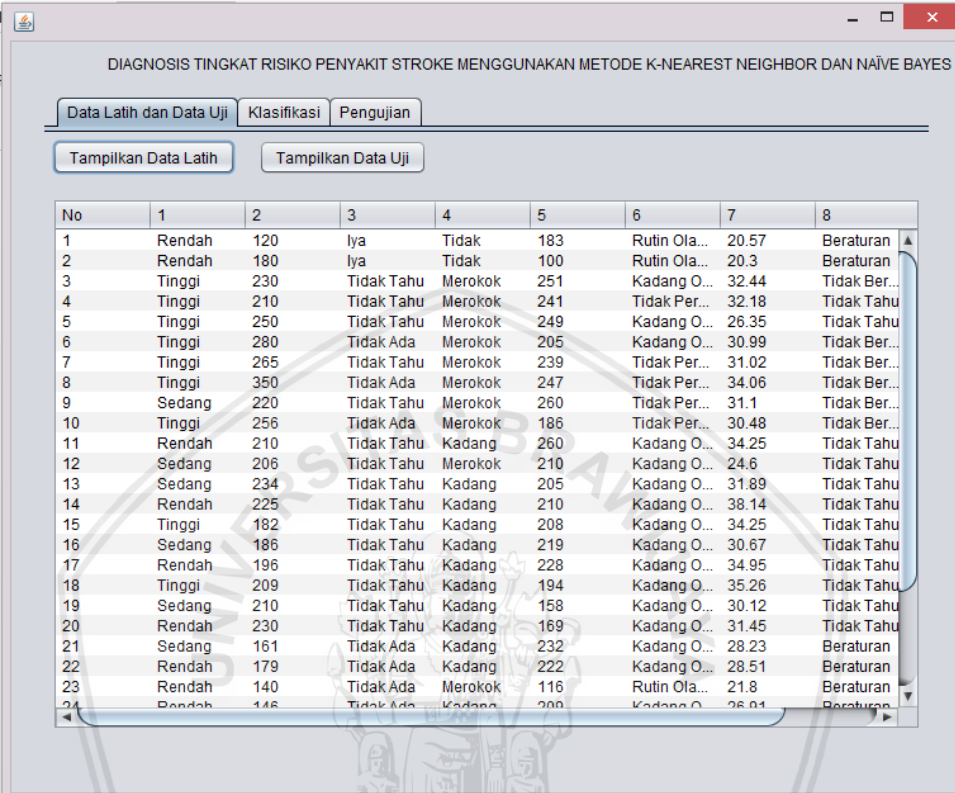
Kode Program 5.5 Implementasi Algoritme Posterior

Penjelasan dari Kode Program 5.5 adalah sebagai berikut:

1. Baris 2-4 inialisasi ukuran *array* dari data uji untuk variabel yang digunakan dalam perhitungan posterior
2. Baris 5 merupakan inialisasi HasilAkhir
3. Baris 6 merupakan perulangan sebanyak data uji untuk perhitungan posterior
4. Baris 7-15 proses perhitungan posterior masing-masing kelas dengan mengkalikan perhitungan prior dan likelihood
5. Baris 17-20 menunjukkan proses perbandingan total peluang risiko penyakit. Jika lebih banyak risiko rendah, maka hasil akhir risiko rendah
6. Baris 21-24 menunjukkan proses perbandingan total peluang risiko penyakit. Jika lebih banyak risiko sedang, maka hasil akhir risiko sedang
7. Baris 25-28 menunjukkan proses perbandingan total peluang risiko penyakit. Jika lebih banyak risiko tinggi, maka hasil akhir risiko tinggi
8. Baris 30 merupakan proses penentuan label kelas data uji saat nilai posterior dari 2 label kelas mempunyai nilai yang sama. Maka label kelas terpilih merupakan kelas mayor pada perhitungan *K-Nearest Neighbor* sebelumnya.

5.4 Implementasi Antarmuka Tampilan Data

Gambar 5.1 menunjukkan implementasi antarmuka tampilan data yang dapat menampilkan data latih beserta data uji untuk aplikasi *K-Nearest Neighbor* dan *Naïve Bayes*.

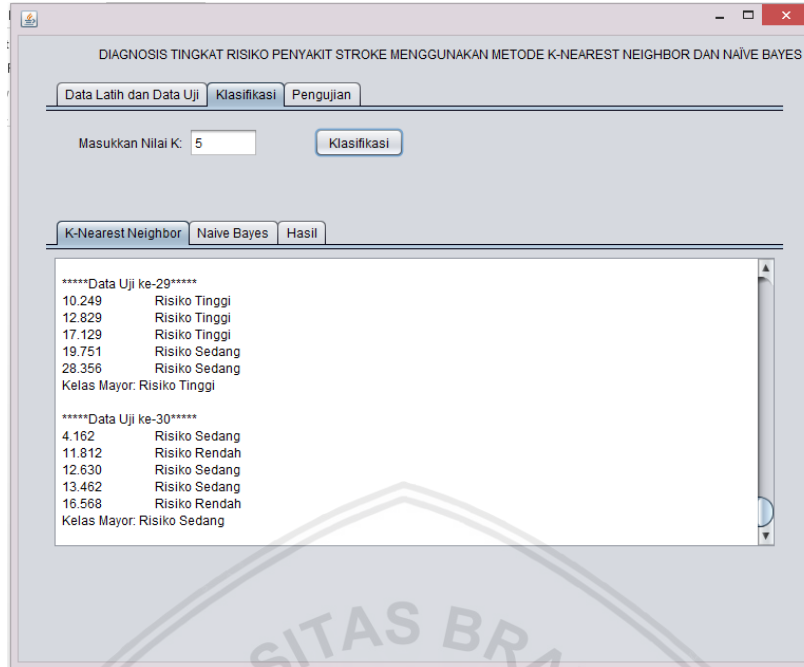


No	1	2	3	4	5	6	7	8
1	Rendah	120	Iya	Tidak	183	Rutin Ola...	20.57	Beraturan
2	Rendah	180	Iya	Tidak	100	Rutin Ola...	20.3	Beraturan
3	Tinggi	230	Tidak Tahu	Merokok	251	Kadang O...	32.44	Tidak Ber...
4	Tinggi	210	Tidak Tahu	Merokok	241	Tidak Per...	32.18	Tidak Tahu
5	Tinggi	250	Tidak Tahu	Merokok	249	Kadang O...	26.35	Tidak Tahu
6	Tinggi	280	Tidak Ada	Merokok	205	Kadang O...	30.99	Tidak Ber...
7	Tinggi	265	Tidak Tahu	Merokok	239	Tidak Per...	31.02	Tidak Ber...
8	Tinggi	350	Tidak Ada	Merokok	247	Tidak Per...	34.06	Tidak Ber...
9	Sedang	220	Tidak Tahu	Merokok	260	Tidak Per...	31.1	Tidak Ber...
10	Tinggi	256	Tidak Ada	Merokok	186	Tidak Per...	30.48	Tidak Ber...
11	Rendah	210	Tidak Tahu	Kadang	260	Kadang O...	34.25	Tidak Tahu
12	Sedang	206	Tidak Tahu	Merokok	210	Kadang O...	24.6	Tidak Tahu
13	Sedang	234	Tidak Tahu	Kadang	205	Kadang O...	31.89	Tidak Tahu
14	Rendah	225	Tidak Tahu	Kadang	210	Kadang O...	38.14	Tidak Tahu
15	Tinggi	182	Tidak Tahu	Kadang	208	Kadang O...	34.25	Tidak Tahu
16	Sedang	186	Tidak Tahu	Kadang	219	Kadang O...	30.67	Tidak Tahu
17	Rendah	196	Tidak Tahu	Kadang	228	Kadang O...	34.95	Tidak Tahu
18	Tinggi	209	Tidak Tahu	Kadang	194	Kadang O...	35.26	Tidak Tahu
19	Sedang	210	Tidak Tahu	Kadang	158	Kadang O...	30.12	Tidak Tahu
20	Rendah	230	Tidak Tahu	Kadang	169	Kadang O...	31.45	Tidak Tahu
21	Sedang	161	Tidak Ada	Kadang	232	Kadang O...	28.23	Beraturan
22	Rendah	179	Tidak Ada	Kadang	222	Kadang O...	28.51	Beraturan
23	Rendah	140	Tidak Ada	Merokok	116	Rutin Ola...	21.8	Beraturan
24	Rendah	146	Tidak Ada	Kadang	200	Kadang O...	26.01	Beraturan

Gambar 5.1 Implementasi Antarmuka Tampilan Data

5.5 Implementasi Antarmuka Klasifikasi

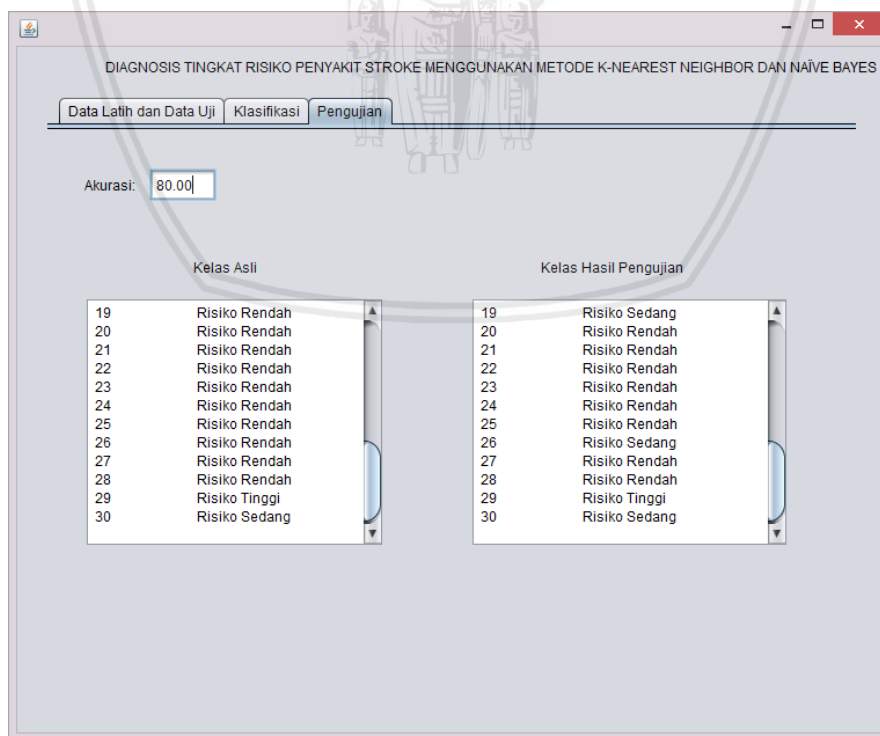
Pada implementasi antarmuka klasifikasi memiliki beberapa menu dengan masing-masing fungsi. Terdapat perintah untuk memasukkan nilai *K* yang diinginkan pengguna, agar dapat digunakan dalam proses klasifikasi menggunakan metode *K-Nearest Neighbor* dan *Naïve Bayes*. Gambar 5.2 menunjukkan implementasi antarmuka klasifikasi.



Gambar 5.2 Implementasi Antarmuka Klasifikasi

5.6 Implementasi Antarmuka Pengujian

Gambar 5.3 menunjukkan implementasi antarmuka pengujian. Pada tab pengujian menunjukkan hasil akurasi diagnosis beserta dengan kelas asli pada data uji dan kelas hasil dari klasifikasi diagnosis dengan algoritme yang diusulkan.



Gambar 5.3 Implementasi Antarmuka Pengujian



BAB 6 PENGUJIAN DAN ANALISIS

Bab pengujian dan analisis berisi tentang pengujian dan analisis sistem diagnosis tingkat risiko penyakit stroke menggunakan metode *K-Nearest Neighbor* dan *Naïve Bayes*. Pengujian pada bab ini berisi tentang pengujian pengaruh sebaran data pada kelas data latih seimbang dan tidak seimbang.

Pengujian pengaruh sebaran data dilakukan untuk mengetahui pengaruh sebaran data dengan kelas yang seimbang dan tidak seimbang pada hasil akurasi. Dalam pengujian ini, data latih ditentukan sebelumnya sejumlah 30, 45 dan 60 data pada masing-masing kelas. Sedangkan data uji yang digunakan berjumlah sama, yaitu 30 data agar mempermudah dalam menganalisis hasil akurasi sebaran data seimbang dan tidak seimbang dengan jumlah data uji tetap pada data latih yang berbeda.

6.1 Pengujian Sebaran Data Kelas Seimbang

Pada pengujian sebaran data kelas seimbang ini, menggunakan 30, 45 dan 60 data latih dengan komposisi data jumlah risiko penyakit stroke yang sama. Sebagai contoh dalam data latih 30, terdapat 10 data dengan kelas risiko rendah, 10 data dengan kelas risiko sedang dan 10 data dengan kelas risiko tinggi. Data uji yang digunakan pada pengujian ini berjumlah 30 data dan menggunakan nilai K mulai dari $K=1$ sampai $K=30$. Hasil pengujian sebaran data pada kelas seimbang dapat dilihat pada Tabel 6.1, 6.2 dan 6.3.

Tabel 6.1 Hasil Pengujian Sebaran Data Kelas Seimbang $K=1-10$

Data Latih	Akurasi (%)									
	$K=1$	$K=2$	$K=3$	$K=4$	$K=5$	$K=6$	$K=7$	$K=8$	$K=9$	$K=10$
30	60	50	80	70	80	80	83.33	80	86.67	90
45	60	53.34	80	66.67	80	83.34	86.67	83.34	86.67	83.34
60	63.34	53.33	73.34	70	86.67	86.67	86.67	83.34	86.67	86.67

Tabel 6.2 Hasil Pengujian Sebaran Data Kelas Seimbang $K=11-20$

Data Latih	Akurasi (%)									
	$K=11$	$K=12$	$K=13$	$K=14$	$K=15$	$K=16$	$K=17$	$K=18$	$K=19$	$K=20$
30	86	93.34	86.67	93.34	90	90	90	86.67	83.34	76.67
45	83.34	90	90	90	96.67	96.67	96.67	96.67	96.67	96.67
60	86.67	86.67	86.67	86.67	86.67	90	93.34	93.34	96.67	93.34

Tabel 6.3 Hasil Pengujian Sebaran Data Kelas Seimbang K=21-30

Data Latih	Akurasi (%)									
	K=21	K=22	K=23	K=24	K=25	K=26	K=27	K=28	K=29	K=30
30	73.33	73.33	73.33	73.33	73.33	73.33	73.33	70	76.67	80
45	96.67	96.67	93.33	93.33	93.33	93.33	93.33	90	90	90
60	93.34	93.34	93.34	93.34	93.34	93.34	93.34	93.34	90	90

Pada tabel 6.1, 6.2 dan 6.3 menunjukkan hasil akurasi tertinggi berada pada jumlah data latih 45 dengan nilai K=15,16,17,18,19,20,21 dan 22 yang memiliki akurasi sebesar 96.67%. Sedangkan nilai terendah terdapat pada jumlah data latih 30 dengan nilai K=2 yang memiliki nilai akurasi sebesar 50%. Pada data latih 30 menunjukkan peningkatan akurasi yang kurang stabil dari nilai K=1-14 dan mengalami penurunan akurasi dimulai dari nilai K=15. Pada data latih 45 mengalami hal yang sama dengan data latih 30, yaitu terdapat peningkatan yang kurang stabil, dimulai dari K=1-22 dan mengalami penurunan akurasi dimulai dari nilai K=23. Sedangkan pada data latih 60 menunjukkan peningkatan akurasi pada nilai K=1-19 dan penurunan akurasi pada nilai K=20.

6.2 Pengujian Sebaran Data Kelas Tidak Seimbang

Pengujian sebaran data kelas tidak seimbang menggunakan 30, 45 dan 60 data latih dengan komposisi data jumlah risiko penyakit stroke tidak sama/berbeda. Sebagai contoh dalam data latih 30, terdapat 8 data dengan kelas risiko rendah, 8 data dengan kelas risiko sedang dan 14 data dengan kelas risiko tinggi. Data uji yang digunakan pada pengujian ini berjumlah 30 data. Hasil pengujian sebaran data pada kelas tidak seimbang dapat dilihat pada Tabel 6.4, 6.5 dan 6.6.

Tabel 6.4 Hasil Pengujian Sebaran Data Kelas Tidak Seimbang K=1-10

Data Latih	Akurasi (%)									
	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8	K=9	K=10
30	53.34	53.34	60	63.34	70	70	76.67	86.67	90	96.67
45	56.67	56.67	70	66.67	70	70	73.34	73.34	80	86.67
60	56.64	56.64	70	73.34	70	73.34	76.67	73.34	80	86.67

Tabel 6.5 Hasil Pengujian Sebaran Data Kelas Tidak Seimbang K=11-20

Data Latih	Akurasi (%)									
	K=11	K=12	K=13	K=14	K=15	K=16	K=17	K=18	K=19	K=20
30	96.67	96.67	96.67	96.67	86.67	83.34	83.34	76.67	73.34	73.34
45	90	90	93.34	93.34	93.34	93.34	93.34	96.67	96.67	96.67
60	86.67	86.67	93.34	93.34	93.34	93.34	93.34	96.67	80	100

Tabel 6.6 Hasil Pengujian Sebaran Data Kelas Tidak Seimbang K=21-30

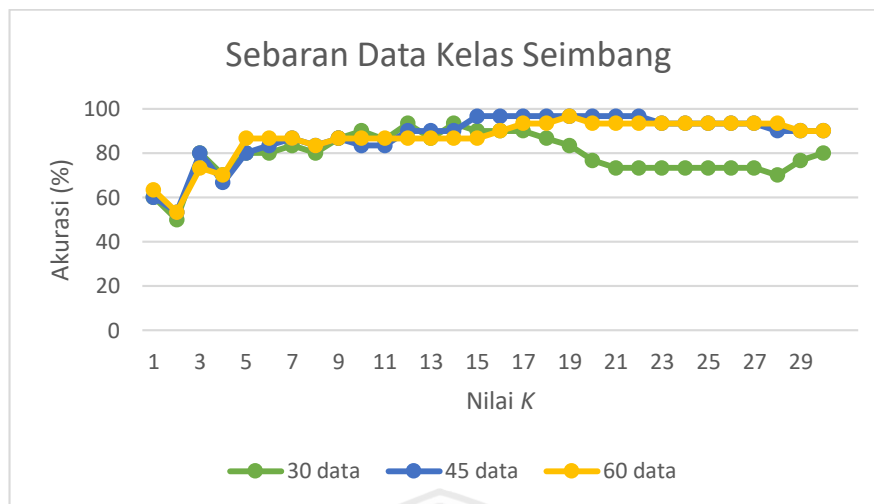
Data Latih	Akurasi (%)									
	K=21	K=22	K=23	K=24	K=25	K=26	K=27	K=28	K=29	K=30
30	73.34	70	70	70	70	76.67	76.67	80	86.67	90
45	96.67	96.67	96.67	96.67	96.67	96.67	96.67	96.67	96.67	96.67
60	100	100	100	100	100	100	100	100	100	100

Tabel 6.3 dan 6.4 menunjukkan hasil akurasi tertinggi berada pada jumlah data latih 60 dengan nilai $K=20-30$ yang memiliki akurasi stabil sebesar 100%. Sedangkan nilai terendah terdapat pada jumlah data latih 30 dengan nilai $K=1$ dan 2 yang memiliki akurasi sebesar 53.34%. Pada data latih 30 menunjukkan peningkatan akurasi yang stabil dari nilai $K=1-14$ dan mengalami penurunan akurasi dimulai dari nilai $K=15$. Sedangkan pada data latih 45 dan data latih 60 menunjukkan peningkatan akurasi yang cukup stabil.

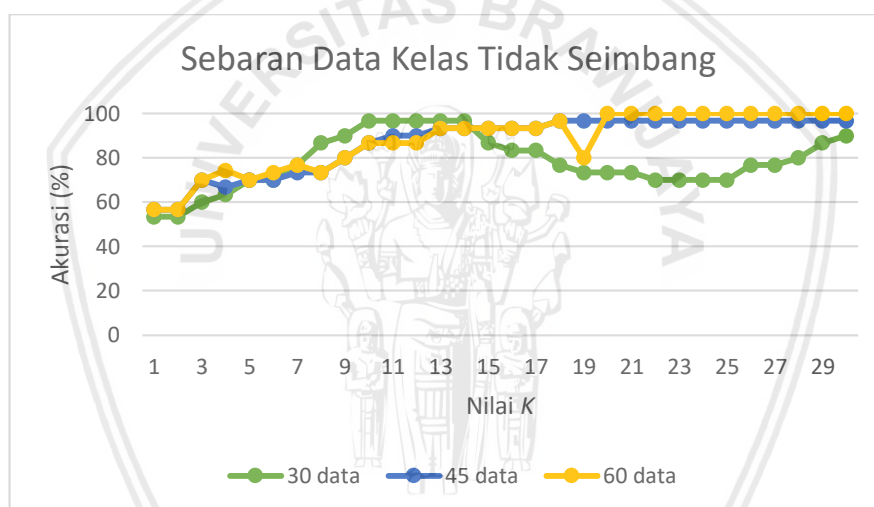
6.3 Analisis Hasil Pengujian

Dari hasil akurasi masing-masing sebaran data kelas seimbang dan tidak seimbang dengan 30 nilai K dan 30, 45 dan 60 data latih pada Tabel 6.1, 6.2, 6.3, 6.4, 6.5 dan 6.6 menghasilkan nilai akurasi yang beragam. Nilai K pada pengujian ini memiliki pengaruh yang besar dalam menentukan hasil klasifikasi. Karena metode *Naïve Bayes* pada aplikasi ini menggunakan data latih yang berasal dari nilai K pada klasifikasi *K-Nearest Neighbor* sebelumnya untuk kemudian dihitung peluangnya.

Grafik hasil pengujian pengaruh sebaran data kelas seimbang ditunjukkan pada Gambar 6.1 dan grafik hasil pengujian data kelas tidak seimbang ditunjukkan pada Gambar 6.2.



Gambar 6.1 Grafik Hasil Pengujian Pengaruh Sebaran Data Seimbang



Gambar 6.2 Grafik Hasil Pengujian Pengaruh Sebaran Data Tidak Seimbang

Berdasarkan grafik pada Gambar 6.1 dan 6.2, pada pengujian sebaran kelas seimbang dan tidak seimbang memiliki nilai akurasi terendah dengan nilai $K=2$. Hal ini terjadi karena metode terakhir yang digunakan adalah *naïve bayes* dengan peluang $K=2$, maka data latih yang digunakan untuk perhitungan probabilitas *naïve bayes* hanya berjumlah 2 data. Data latih tersebut diambil dari sorting *K-Nearest Neighbor* sebelumnya, dimana $K=2$ bernilai sangat sedikit untuk dijadikan data latih perhitungan *Naïve Bayes*. Pada saat dilakukan perhitungan peluang kemunculan data, terlalu banyak data yang bernilai 0, yang mengakibatkan terjadinya kesalahan dalam klasifikasi. Jika nilai akhir pada *Naïve Bayes* bernilai 0, maka akan diambil hasil klasifikasi pada kelas mayor proses *K-Nearest Neighbor* sebelumnya.

Akurasi data seimbang dan tidak seimbang pada pengujian yang sudah dilakukan menunjukkan bahwa dengan nilai K yang semakin besar, maka peluang pada proses *Naïve Bayes* akan semakin besar, sehingga dapat mempengaruhi hasil



klasifikasi. Sebaran data kelas tidak seimbang memiliki akurasi yang lebih baik dan stabil daripada sebaran data kelas seimbang. Berdasarkan analisis pada atribut data uji yang digunakan, data latih dengan sebaran kelas tidak seimbang memiliki kedekatan nilai atribut yang lebih baik dibandingkan data latih dengan sebaran kelas seimbang. Sehingga, hal tersebut mempengaruhi hasil akurasi dari masing-masing sebaran data. Pada sebaran data kelas seimbang maupun tidak seimbang nilai $K > 5$ sudah dapat menghasilkan nilai akurasi yang cukup baik.



BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan pada penelitian diagnosis tingkat risiko penyakit stroke menggunakan metode *K-Nearest Neighbor* dan *Naïve Bayes*, maka dapat diambil kesimpulan sebagai berikut:

1. Metode kombinasi *K-Nearest Neighbor* dan *Naïve Bayes* dapat diimplementasikan untuk melakukan diagnosis tingkat risiko penyakit stroke dengan 8 atribut dan 3 jenis tingkatan risiko penyakit stroke, yaitu tingkat risiko rendah, sedang dan tinggi. Langkah pertama yang dilakukan adalah menggunakan atribut dengan tipe numerik sebagai data latih dan data uji untuk proses klasifikasi menggunakan algoritme *K-Nearest Neighbor*. Kemudian menentukan K tetangga terdekat untuk proses algoritme *K-Nearest Neighbor* berdasarkan atribut-atribut numerik. Gunakan nilai kumpulan dari tetangga K yang didapatkan pada langkah sebelumnya sebagai data latih dan gunakan untuk membangun algoritme *Naïve Bayes* berdasarkan atribut-atribut bernilai kategoris. Proses *Naïve Bayes* dimulai dengan menghitung nilai prior, likelihood dan posterior. Hasil klasifikasi berupa nilai tertinggi pada posterior, kemudian menjadi label kelas untuk data yang di uji. Jika terjadi kesamaan nilai tertinggi pada posterior, maka nilai mayor pada proses *K-Nearest Neighbor* akan menjadi label kelas data uji.
2. Pengujian yang dilakukan pada penelitian ini adalah pengujian pengaruh sebaran data pada kelas data latih seimbang dan tidak seimbang. Nilai akurasi tertinggi yang diperoleh pada data kelas seimbang adalah 96.67% dengan data latih sejumlah 45 dan nilai $K=15-22$. Sedangkan pada data latih tidak seimbang, menunjukkan akurasi tertinggi sebesar 100% dengan jumlah data latih sebanyak 60 dan nilai $K=20-30$.

7.2 Saran

Untuk penelitian berikutnya, saran yang dapat diberikan dari penelitian Diagnosis Tingkat Risiko Penyakit Stroke Menggunakan Metode *K-Nearest Neighbor* dan *Naïve Bayes* adalah sebagai berikut:

1. Penelitian ini dilakukan dengan atribut serta dataset dalam jumlah yang kecil, karenanya tidak dapat dinyatakan sebagai pendekatan terbaik. Diharapkan pada penelitian berikutnya terdapat penambahan beberapa atribut serta data agar variasi data lebih beragam, sehingga dapat menghasilkan akurasi yang lebih baik.
2. Melakukan *Laplace Smoothing* untuk menghindari nilai likelihood=0 pada klasifikasi, sehingga probabilitas posterior tidak turun menjadi 0.