

**KLASIFIKASI PENURUNAN FUNGSI KOGNITIF PASIEN
STROKE MENGGUNAKAN METODE KLASIFIKASI *RANDOM
FOREST***

SKRIPSI

Disusun oleh:

Muhammad Shidqi Fadlilah

NIM: 155150207111021



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

KLASIFIKASI PENURUNAN FUNGSI KOGNITIF PASIEN STROKE TROMBOSIS
MENGUNAKAN METODE KLASIFIKASI *RANDOM FOREST*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Muhammad Shidqi Fadlilah
NIM: 155150207111021

Skripsi ini telah diuji dan dinyatakan lulus pada
2 Januari 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II



Randy Cahya Wihandika, S.ST., M.Kom
NIK: 201405 880206 1 001



Bayu Rahayudi, S.T., M.T.
NIP: 19740712 200604 1 001

Mengetahui
Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T., Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 17 Desember 2018



Muhammad Shidqi Fadlilah

NIM: 155150207111021

PRAKATA

Puji dan syukur penulis panjatkan kepada Allah SWT, karena dengan rahmat, karunia, dan hidayah-Nya penulis dapat menyelesaikan penelitian dan penulisan skripsi dengan judul “Klasifikasi Penurunan Fungsi Kognitif Pada Pasien Stroke Dengan Menggunakan Metode *Random forest*”.

Penulis juga ingin menyampaikan rasa hormat dan terima kasih kepada semua pihak yang telah membantu dan mendukung penulis hingga penulis dapat menyelesaikan penelitian dan penulisan skripsi ini. Ada pula pihak yang dimaksud adalah:

1. Bapak Randy Cahya Wihandika, S.ST., M.Kom. selaku dosen pembimbing pertama yang dengan kesabaran penuh telah membina, membimbing, meluruskan niat, dan memberi arahan penulis untuk menyelesaikan skripsi ini dengan baik dan benar.
2. Bapak Bayu Rahayudi, S.T, M.T. selaku dosen pembimbing kedua yang telah membimbing, mengevaluasi, dan mengajarkan penulis banyak hal hingga selesainya penelitian dan penulisan skripsi ini.
3. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D., Bapak Ir Heru Nurwasito, M.Kom., Bapak Suprpto, S.T, M.T., Bapak Edy Santoso, S.Si, M.Kom. selaku Dekan, Wakil Dekan I, Wakil Dekan II dan Wakil Dekan III Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Tri Astoto Kurniawan selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak Agus Wahyu Widodo, S.T, M.Cs selaku ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
6. Bapak Adam Hendra Brata, S.Kom., M.T., selaku dosen penasihat akademik yang telah memberikan saran dan nasihat selama menempuh masa studi.
7. Seluruh Dosen Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan ilmu dan pengetahuannya kepada penulis selama masa studi hingga skripsi ini selesai.
8. Seluruh Sivitas Akademika Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan bantuan kepada penulis selama masa studi hingga skripsi ini selesai.
9. Bapak Ir. Rachmad Widiyanto, M.Si., dan Ibu Ir. Nila Astiti, M.M., selaku orang tua penulis, Nasya Amalina, selaku adik penulis dan seluruh keluarga besar penulis yang telah memberikan motivasi dan bimbingan hingga penulis dapat menyelesaikan penelitian dan penulisan skripsi ini.
10. Suryani Agustin, yang tidak hanya hadir dikala penulis sedang bahagia, namun juga hadir menyemangati dan meluruskan kembali niat penulis

dikala penulis sempat putus asa untuk melakukan penelitian dan penulisan skripsi ini.

11. Muhammad Rafi Farhan, selaku tutor penulis yang dengan penuh kesabaran mencari solusi dikala penulis kesulitan dalam pengerjaan program klasifikasi ini.
12. Kevin Azwega, Rahmat Faizal, Dwi Suci Ariska Yanti, Khalisma Frinta, dan Indriya Dewi Onantya, selaku penasihat dan pengingat penulis dalam hal kebaikan hingga penulis dapat menyelesaikan penelitian dan penulisan skripsi ini.
13. Serta semua pihak yang telah mendukung dan membantu penulis dalam penelitian dan penulisan skripsi ini yang dengan berat hati tidak dapat disebutkan satu persatu.

Penulis sadar betul dengan hasil penelitian dan penulisan skripsi ini masih jauh dari kata sempurna. Maka dari itu penulis sangat mengharapkan kritik dan saran pembaca kepada penulis untuk menjadi bahan pembelajaran untuk penelitian selanjutnya di waktu yang akan datang. Semoga skripsi ini bermanfaat bagi yang membaca dan menggunakannya.

Malang, 02 Januari 2019

Penulis

mshidqifadlilah@gmail.com

ABSTRAK

Muhammad Shidqi Fadlilah, KLASIFIKASI PENURUNAN FUNGSI KOGNITIF PASIEN STROKE MENGGUNAKAN METODE KLASIFIKASI RANDOM FOREST

Pembimbing: Randy Cahya Wihandika, S.ST., M.Kom. dan Bayu Rahayudi, S.T., M.T.

Stroke merupakan salah satu penyakit yang menyerang segala golongan manusia tanpa memandang ras, jenis kelamin, dan usia. Salah satu dampak dari penyakit stroke adalah penurunan fungsi kognitif. Otak manusia memiliki banyak saraf yang beberapa diantaranya adalah mengatur kerja fungsi kognitif manusia. Berdasarkan penelitian yang dilakukan oleh Wardani (2015), faktor yang memengaruhi penurunan fungsi kognitif terdiri dari tiga belas faktor. Dengan jumlah yang tidak sedikit tersebut, maka dibutuhkan sistem yang dapat mendeteksi penurunan fungsi kognitif pada pasien stroke. Maka dalam penelitian ini, dilakukan klasifikasi penurunan fungsi kognitif pada pasien stroke dengan menggunakan metode klasifikasi random forest berdasarkan data yang didapatkan dari Rumah Sakit Saiful Anwar pada bulan Juli sampai dengan Oktober tahun 2014. Metode random forest dipilih karena metode ini baik untuk data kategorikal seperti pada kasus faktor-faktor pengaruh penurunan fungsi kognitif pada pasien stroke ini. Berdasarkan hasil pengujian dengan lima kombinasi data yang terdiri dari 19 data uji dan 76 data latih, jumlah tree terbaik yang didapatkan adalah 100 tree. Hasil rata-rata akurasi yang didapatkan dari semua percobaan adalah 53,094%. Hal ini disebabkan oleh beberapa faktor yang menjadi error pada sistem tersebut. Salah satu faktor yang menyebabkan ketidaksempurnaan sistem ini adalah penyebaran data latih yang tidak merata kelasnya.

Kata kunci: *random forest*, *Decision Tree*, Fungsi Kognitif, Stroke

ABSTRACT

Muhammad Shidqi Fadlilah, Classification System for Decreasing Cognitive Function of Stroke Patients Using the Random Forest Classification Method

Supervisor: Randy Cahya Wihandika, S.ST., M.Kom. dan Bayu Rahayudi, S.T., M.T.

Stroke is a disease that attacks all human, regardless of race, gender, and age. One of the effects of stroke is a decreased cognitive function. A human brain has many nerves, one of them is regulating the work of the human's cognitive function. Based on research by Wardhani (2015), factors that decreasing cognitive function consist of thirteen factors. So, a system that can detect a decreased cognitive function on a stroke patient is needed. So in this research, we make a system that could be used to classify the decreasing cognitive function using a random forest method. the random forest was chosen because this method is good for categorical data. Based on the testing result, the best tree that builds in this system was 100 trees. The average result of the accuracy obtained from all experiments were 53.094%. That number means that the system is still far from perfect. One of the factors that caused this system's imperfection was the distribution of training classes were not evenly distributed.

Keyword: random forest, Decision Tree, Cognitive Function, Stroke

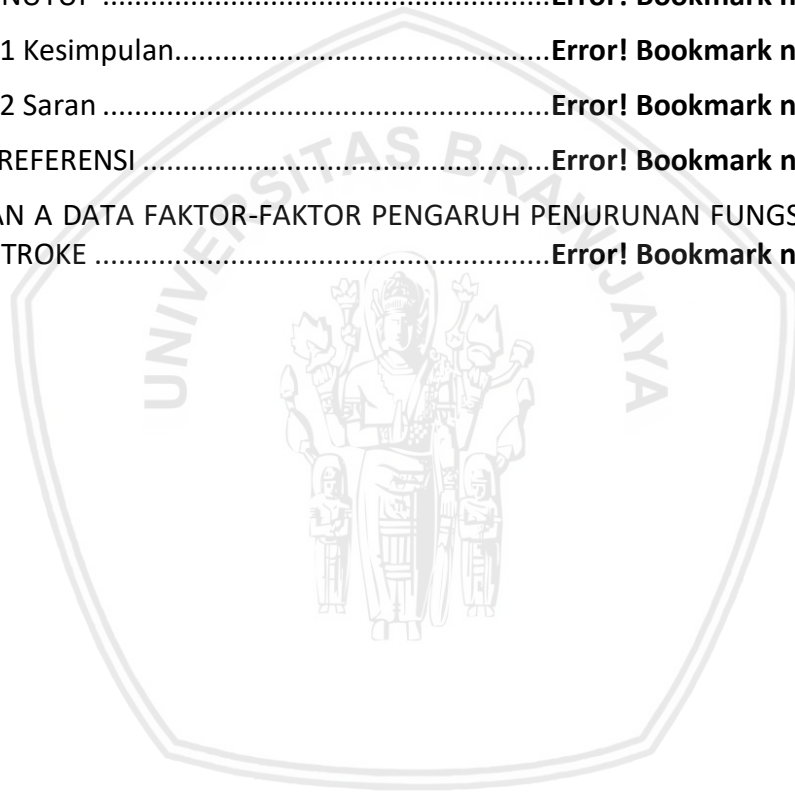
DAFTAR ISI

KLASIFIKASI PENURUNAN FUNGSI KOGNITIF PASIEN STROKE MENGGUNAKAN METODE KLASIFIKASI RANDOM FOREST.....	i
PROGRAM STUDI TEKNIK INFORMATIKA.....	i
JURUSAN TEKNIK INFORMATIKA.....	i
FAKULTAS ILMU KOMPUTER.....	i
UNIVERSITAS BRAWIJAYA	i
MALANG.....	i
2018.....	i
PENGESAHAN	Error! Bookmark not defined.
PERNYATAAN ORISINALITAS	ii
PRAKATA.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiv
BAB 1 PENDAHULUAN.....	Error! Bookmark not defined.
1.1 Latar belakang.....	Error! Bookmark not defined.
1.2 Rumusan Masalah.....	Error! Bookmark not defined.
1.3 Tujuan	Error! Bookmark not defined.
1.4 Manfaat.....	Error! Bookmark not defined.
1.5 Batasan Masalah.....	Error! Bookmark not defined.
1.6 Sistematika Pembahasan	Error! Bookmark not defined.
BAB 2 LANDASAN KEPUSTAKAAN	Error! Bookmark not defined.
2.1 Kajian Pustaka	Error! Bookmark not defined.
2.2 Klasifikasi.....	Error! Bookmark not defined.
2.2.1 Komponen Dasar Klasifikasi	Error! Bookmark not defined.
2.2.2 Metode Klasifikasi	Error! Bookmark not defined.
2.3 Classification and Regression Trees (CART).....	Error! Bookmark not defined.
2.4 Random Forest.....	Error! Bookmark not defined.

2.4.1 Tahap Pengacakan Data	Error! Bookmark not defined.
2.4.2 Tahapan Pembangunan CART	Error! Bookmark not defined.
2.4.3 Tahap <i>Voting</i>	Error! Bookmark not defined.
2.4.4 <i>Error Random Forest</i>	Error! Bookmark not defined.
2.4.5 Kondisi Berhenti	Error! Bookmark not defined.
2.5 Stroke	Error! Bookmark not defined.
2.6 Fungsi Kognitif.....	Error! Bookmark not defined.
2.6.2 Penurunan Fungsi Kognitif	Error! Bookmark not defined.
2.6.3 Faktor-Faktor yang Penurunan Fungsi Kognitif	Error! Bookmark not defined.
2.7 Evaluasi	Error! Bookmark not defined.
BAB 3 METODOLOGI	Error! Bookmark not defined.
3.1 Tipe Penelitian	Error! Bookmark not defined.
3.2 Strategi Penelitian.....	Error! Bookmark not defined.
3.3 Pengambilan Data	Error! Bookmark not defined.
3.4 Implementasi	Error! Bookmark not defined.
3.5 Pengujian	Error! Bookmark not defined.
3.6 Kesimpulan.....	Error! Bookmark not defined.
BAB 4 ANALISIS DAN PERANCANGAN	Error! Bookmark not defined.
4.1 Analisis Kebutuhan	Error! Bookmark not defined.
4.2 Identifikasi Permasalahan.....	Error! Bookmark not defined.
4.3 Perancangan Algoritme	Error! Bookmark not defined.
4.3.1 Proses <i>random forest</i>	Error! Bookmark not defined.
4.3.2 Proses <i>Training</i>	Error! Bookmark not defined.
4.3.3 Proses Pembangunan <i>Tree</i>	Error! Bookmark not defined.
4.3.4 Proses Perhitungan Entropi	Error! Bookmark not defined.
4.3.5 Proses <i>Testing</i>	Error! Bookmark not defined.
4.4 Perhitungan Manual	Error! Bookmark not defined.
4.4.1 Random Data.....	Error! Bookmark not defined.
4.4.2 Membangun <i>Decision Tree</i>	Error! Bookmark not defined.
4.4.3 <i>Voting</i>	Error! Bookmark not defined.
4.4.4 Pengujian Akurasi.....	Error! Bookmark not defined.

4.5 Perancangan Pengujian	Error! Bookmark not defined.
4.5.1 Pengujian <i>K-fold cross validation</i> Seleksi Fitur	Error! Bookmark not defined.
4.5.2 Pengujian <i>K-fold cross validation</i> Jumlah tree	Error! Bookmark not defined.
BAB 5 IMPLEMENTASI	Error! Bookmark not defined.
5.1 Kode Program	Error! Bookmark not defined.
5.1.1 Method <i>main</i>	Error! Bookmark not defined.
5.1.2 Method <i>multi_tree</i>	Error! Bookmark not defined.
5.1.3 Method <i>train</i>	Error! Bookmark not defined.
5.1.4 Method <i>random_data</i>	Error! Bookmark not defined.
5.1.5 Method <i>splitting</i>	Error! Bookmark not defined.
5.1.6 Method <i>baca_kelas_unik</i>	Error! Bookmark not defined.
5.1.7 Method <i>entropy</i>	Error! Bookmark not defined.
5.1.8 Method <i>baca_kategori_fitur</i>	Error! Bookmark not defined.
5.1.9 Method <i>test</i>	Error! Bookmark not defined.
5.1.10 Method <i>voting</i>	Error! Bookmark not defined.
5.1.11 Method <i>akurasi</i>	Error! Bookmark not defined.
BAB 6 PENGUJIAN DAN ANALISIS	Error! Bookmark not defined.
6.1 Pengujian <i>K-Fold Cross Validation</i> Jumlah Fitur	Error! Bookmark not defined.
6.1.1 Hasil Pengujian 13 Fitur	Error! Bookmark not defined.
6.1.2 Hasil Pengujian 12 Fitur	Error! Bookmark not defined.
6.1.3 Hasil Pengujian 11 Fitur	Error! Bookmark not defined.
6.1.4 Hasil Pengujian 10 Fitur	Error! Bookmark not defined.
6.1.5 Hasil Pengujian 9 Fitur	Error! Bookmark not defined.
6.1.6 Hasil Pengujian 8 Fitur	Error! Bookmark not defined.
6.1.7 Hasil Penujian Seleksi Fitur	Error! Bookmark not defined.
6.2 Pengujian <i>K-Fold Cross Validation</i> Jumlah Tree	Error! Bookmark not defined.
6.2.1 Hasil Pengujian Kombinasi <i>Folds</i> 1	Error! Bookmark not defined.
6.2.2 Hasil Pengujian Kombinasi <i>Folds</i> 2	Error! Bookmark not defined.
6.2.3 Hasil Pengujian Kombinasi <i>Folds</i> 3	Error! Bookmark not defined.

6.2.4 Hasil Pengujian Kombinasi <i>Folds</i> 4..	Error! Bookmark not defined.
6.2.5 Hasil Pengujian Kombinasi <i>Folds</i> 5..	Error! Bookmark not defined.
6.2.6 Hasil Pengujian Jumlah <i>Tree</i>	Error! Bookmark not defined.
6.3 Analisis Hasil Pengujian.....	Error! Bookmark not defined.
6.3.1 Analisis Hasil Pengujian Jumlah Fitur	Error! Bookmark not defined.
6.3.2 Analisis Hasil Pengujian Jumlah <i>Tree</i>	Error! Bookmark not defined.
BAB 7 PENUTUP	Error! Bookmark not defined.
7.1 Kesimpulan.....	Error! Bookmark not defined.
7.2 Saran	Error! Bookmark not defined.
DAFTAR REFERENSI	Error! Bookmark not defined.
LAMPIRAN A DATA FAKTOR-FAKTOR PENGARUH PENURUNAN FUNGSI KOGNITIF PASIEN STROKE	Error! Bookmark not defined.



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	Error! Bookmark not defined.
Tabel 3.1 Jadwal Penelitian	Error! Bookmark not defined.
Tabel 4.1 Data Faktor Penurunan Fungsi Kognitif Pasien Stroke	Error! Bookmark not defined.
Tabel 4.2 Tabel Data Latih Manualisasi.....	Error! Bookmark not defined.
Tabel 4.3 Tabel Data Uji Manualisasi	Error! Bookmark not defined.
Tabel 4.4 Data Latih Random Manualisasi 1	Error! Bookmark not defined.
Tabel 4.5 Data Latih Random Manualisasi 2	Error! Bookmark not defined.
Tabel 4.6 Entropi Kelas Iterasi 1.....	Error! Bookmark not defined.
Tabel 4.7 Entropi Fitur Usia Iterasi 1.....	Error! Bookmark not defined.
Tabel 4.8 Entropi dan <i>Gain</i> Fitur Sex Iterasi 1.....	Error! Bookmark not defined.
Tabel 4.9 Entropi dan <i>Gain</i> Fitur Stroke Iterasi 1.....	Error! Bookmark not defined.
Tabel 4.10 Entropi dan <i>Gain</i> Fitur Hipertensi Iterasi 1	Error! Bookmark not defined.
Tabel 4.11 Entropi dan <i>Gain</i> Fitur Hiperkolestrol Iterasi 1	Error! Bookmark not defined.
Tabel 4.12 Entropi dan <i>Gain</i> Fitur Diabetes Melitus Iterasi 1	Error! Bookmark not defined.
Tabel 4.13 Entropi dan <i>Gain</i> Fitur Atrial Fibrilasi Iterasi 1	Error! Bookmark not defined.
Tabel 4.14 Entropi dan <i>Gain</i> Fitur Rokok Iterasi 1	Error! Bookmark not defined.
Tabel 4.15 Entropi dan <i>Gain</i> Fitur Alkohol Iterasi 1 ..	Error! Bookmark not defined.
Tabel 4.16 Entropi dan <i>Gain</i> Fitur <i>Hemisfer</i> Iterasi 1	Error! Bookmark not defined.
Tabel 4.17 Entropi dan <i>Gain</i> Fitur CT Scan Iterasi 1..	Error! Bookmark not defined.
Tabel 4.18 Entropi dan <i>Gain</i> Fitur Pendidikan Iterasi 1	Error! Bookmark not defined.
Tabel 4.19 Entropi dan <i>Gain</i> Fitur BMI Iterasi 1.....	Error! Bookmark not defined.
Tabel 4.20 Tabel Pengurutan <i>Gain</i> Iterasi 1	Error! Bookmark not defined.
Tabel 4.21 Contoh Data Uji <i>Voting</i> 1.....	Error! Bookmark not defined.
Tabel 4.22 Contoh Data Uji <i>Voting</i> 2.....	Error! Bookmark not defined.
Tabel 4.23 Tabel Akurasi Manualisasi	Error! Bookmark not defined.
Tabel 5.1 Kode Program <i>Method</i> Main.....	Error! Bookmark not defined.

Tabel 5.2 Tabel Penjelasan Kode Program Mehtod Main**Error! Bookmark not defined.**

Tabel 5.3 Kode Program *Method* multi_tree**Error! Bookmark not defined.**

Tabel 5.4 Penjelasan Kode Program *Method* multi_tree**Error! Bookmark not defined.**

Tabel 5.5 Kode Program *Method* train**Error! Bookmark not defined.**

Tabel 5.6 Penjelasan Kode Program *Method* Train ..**Error! Bookmark not defined.**

Tabel 5.7 Kode Program *Method* random_data**Error! Bookmark not defined.**

Tabel 5.8 Penjelasan Kode Proram *Method* random_data**Error! Bookmark not defined.**

Tabel 5.9 Kode Program *Method* Splitting**Error! Bookmark not defined.**

Tabel 5.10 Penjelasan Kode Program *Method* Splitting**Error! Bookmark not defined.**

Tabel 5.11 Kode Program *Method* baca_kelas_unik **Error! Bookmark not defined.**

Tabel 5.12 Penjelasan Kode Program *Method* baca_kelas_unik**Error! Bookmark not defined.**

Tabel 5.13 Kode Program *Method* entropy.....**Error! Bookmark not defined.**

Tabel 5.14 Penjelasan Kode Program *Method* Entropy**Error! Bookmark not defined.**

Tabel 5.15 Kode Program *Method* baca_kategori_fitur**Error! Bookmark not defined.**

Tabel 5.16 Penjelasan Kode Program *Method* baca_kategori_fitur.....**Error! Bookmark not defined.**

Tabel 5.17 Kode Program *Method* test**Error! Bookmark not defined.**

Tabel 5.18 Penjelasan Kode Program *Method* test...**Error! Bookmark not defined.**

Tabel 5.19 Kode Program *Method* voting**Error! Bookmark not defined.**

Tabel 5.20 Penjelasan Kode Program *Method* voting**Error! Bookmark not defined.**

Tabel 5.21 Kode Program *Method* akurasi.....**Error! Bookmark not defined.**

Tabel 5.22 Penjelasan Kode Program *Method* akurasi**Error! Bookmark not defined.**

DAFTAR GAMBAR

Gambar 4.1 Diagram Alir <i>random forest</i>	Error! Bookmark not defined.
Gambar 4.2 Diagram Alir Proses <i>random forest</i>	Error! Bookmark not defined.
Gambar 4.3 Diagram Alir Proses <i>train</i>	Error! Bookmark not defined.
Gambar 4.4 Diagram Alir Proses Pembangunan <i>Tree</i>	Error! Bookmark not defined.
Gambar 4.5 Diagram Alir Proses Perhitungan Entropi.....	Error! Bookmark not defined.
Gambar 4.6 Diagram Alir Proses <i>Testing</i>	Error! Bookmark not defined.
Gambar 4.7 <i>Split Tree</i> Iterasi 1.....	Error! Bookmark not defined.
Gambar 4.8 <i>Split Tree</i> Iterasi 4.....	Error! Bookmark not defined.
Gambar 4.9 <i>Tree Voting</i> Data Uji 1	Error! Bookmark not defined.
Gambar 4.10 <i>Tree Voting</i> Data Uji 1	Error! Bookmark not defined.
Gambar 6.1 Hasil Pengujian 13 Fitur.....	Error! Bookmark not defined.
Gambar 6.2 Hasil Pengujian 12 Fitur.....	Error! Bookmark not defined.
Gambar 6.3 Hasil Pengujian 11 Fitur.....	Error! Bookmark not defined.
Gambar 6.4 Hasil Pengujian 10 Fitur.....	Error! Bookmark not defined.
Gambar 6.5 Hasil Pengujian 9 Fitur.....	Error! Bookmark not defined.
Gambar 6.6 Hasil Pengujian 8 Fitur.....	Error! Bookmark not defined.
Gambar 6.7 Grafik Nilai Rata-rata Akurasi Pada Enam Jumlah Fitur	Error! Bookmark not defined.
Gambar 6.8 Hasil Akurasi Pengujian <i>Folds</i> 1	Error! Bookmark not defined.
Gambar 6.9 Hasil Akurasi Pengujian <i>Folds</i> 2	Error! Bookmark not defined.
Gambar 6.10 Hasil Akurasi Pengujian <i>Folds</i> 3	Error! Bookmark not defined.
Gambar 6.11 Hasil Akurasi Pengujian <i>Folds</i> 4	Error! Bookmark not defined.
Gambar 6.12 Hasil Akurasi Pengujian <i>Folds</i> 5	Error! Bookmark not defined.
Gambar 6.13 Grafik Nilai Rata-rata Akurasi Pada Lima Jumlah <i>Tree</i>	Error! Bookmark not defined.

BAB 1 PENDAHULUAN

Bab satu menjabarkan latar belakang penulisan proposal skripsi, rumusan masalah berdasarkan objek penelitian, tujuan penulisan proposal skripsi, batasan masalah penulisan, manfaat penulisan, dan sistematika penulisan proposal skripsi.

1.1 Latar belakang

Penyakit stroke adalah penyakit yang penderitanya mengalami gangguan vaskular. Hal tersebut disebabkan oleh pecahnya pembuluh darah di otak yang menimbulkan gangguan kognitif berdasarkan lokasi pecahnya pembuluh darah di otak penderita tersebut. Gangguan kognitif terdiri dari penurunan kesadaran, gangguan *visuospatial*, gangguan pembelajaran non-verbal, gangguan berkomunikasi, dan penurunan tingkat atensi penderita (Hanas, et al., 2015).

Dari beberapa penyakit di dunia yang dapat menyerang manusia dan menyebabkan kematian, stroke merupakan salah satu penyakit yang menyerang segala golongan manusia tanpa memandang ras, jenis kelamin, dan usia. Menurut organisasi kesehatan dunia, *World Health Organization (WHO)*, stroke atau *Cerebrovascular disease* ini adalah tanda klinis yang berkembang akibat gangguan fungsi otak karena adanya penyumbatan atau terjadi pecah pembuluh darah pada otak penderita dengan gejala yang berlangsung lebih dari 24 jam (Nugraha, et al., 2017).

WHO menyebutkan ada sekitar 795.000 orang di Amerika Serikat terserang penyakit stroke setiap tahunnya. Dari 795.000 orang tersebut, 610.000 di antaranya merupakan stroke perdananya atau bukan status kekambuhan stroke. Sedangkan 185.000 orang lainnya merupakan kasus kekambuhan stroke yang berulang. Dari empat juta orang di Amerika Serikat yang mengalami stroke, 15-30% pasien pasca stroke menderita cacat yang permanen. Di Indonesia, data Riset Kesehatan Dasar tahun 2013 menyebutkan bahwa tujuh dari seribu orang terkena stroke. Hal tersebut dapat dikatakan dari setiap tujuh orang meninggal, satu di antaranya meninggal dikarenakan stroke (Malik & Maulina, 2015).

Salah satu dampak dari penyakit stroke adalah penurunan fungsi kognitif. Otak manusia memiliki banyak saraf yang beberapa diantaranya adalah mengatur kerja fungsi kognitif manusia. Penurunan fungsi kognisi manusia dimulai pada usia 30 tahun dan berlangsung secara bertahap atau progresif. Penyebab penurunan fungsi kognitif ini bisa karena vaskular, hal ini dapat diartikan sebagai gangguan yang berlanjut dari *mild cognitive impairment* sampai *diemsia*. Hal ini dikarenakan adanya keterlibatan kerusakan vaskular yang berguna untuk memori, kognisi, dan kebiasaan. Kelainan kognitif akibat vaskular dapat dideskripsikan sebagai kelainan heterogen dengan berbagai tipe *lesi cerebrovascular* yang berbeda dan akan berkembang sampai dengan tahap akhirnya yaitu *demensia* (Jellinger, 2013).

Pendeteksian penurunan fungsi kognitif pada pasien stroke ini perlu dilakukan dengan menggunakan komputer. Hal ini ditujukan untuk mempersingkat waktu dalam proses pendeteksian. Seperti penelitian yang dilakukan oleh Nugraha (2017), menggunakan 203 data rekam medik pasien yang menderita penyakit stroke yang ada di Rumah Sakit Umum Daerah Udata Palu. Penelitian tersebut mendapatkan tingkat keberhasilan dari klasifikasi yang dilakukan tersebut sebesar 89,65% tingkat akurasi. Berdasarkan penelitian yang dilakukan oleh Wardhani (2015), faktor yang memengaruhi penurunan fungsi kognitif terdiri dari 13 faktor. Dengan jumlah yang tidak sedikit tersebut, maka dibutuhkan sistem yang dapat mendeteksi penurunan fungsi kognitif pada pasien stroke. Meskipun sebuah sistem tidak akan mendapatkan hasil yang sempurna dalam tingkat akurasi, pengguna sistem tersebut dalam hal ini adalah penderita stroke tetap dapat menggunakannya untuk mengetahui deteksi awal sebelum memeriksakannya ke dokter lebih lanjut untuk hasil yang lebih maksimal.

Dari penjabaran tentang stroke dan faktor penurunan fungsi kognitif di atas, maka dalam penelitian kali ini dilakukan klasifikasi penurunan fungsi kognitif pada pasien stroke dengan menggunakan metode klasifikasi *random forest* berdasarkan data yang didapatkan. Metode *random forest* dipilih karena metode ini baik untuk data kategorikal seperti pada kasus faktor-faktor pengaruh penurunan fungsi kognitif pada pasien stroke ini. Metode ini menggunakan *voting* dari sejumlah decision tree yang dibangun dari data latih yang ada. Penelitian mengenai klasifikasi menggunakan *random forest* terlebih dahulu dilakukan oleh Fitriani (2007) untuk mengklasifikasi penerimaan tenaga kerja baru dan oleh Rukmigayatri (2015) untuk klasifikasi kemunculan titik panas pada lahan gambut.

1.2 Rumusan Masalah

Berdasarkan penjabaran latar belakang di atas, maka dapat dirumuskan permasalahan dalam penelitian ini yaitu:

1. Berapa jumlah fitur yang optimal untuk mengklasifikasi penurunan fungsi kognitif pada pasien stroke?
2. Berapa jumlah *tree* yang optimal pada proses pelatihan *random forest* untuk mengklasifikasi penurunan fungsi kognitif pada pasien stroke?
3. Berapa tingkat akurasi klasifikasi penurunan fungsi kognitif pada pasien stroke dengan menggunakan metode *random forest*?

1.3 Tujuan

Tujuan dari penelitian ini berdasarkan rumusan masalah dan latar belakang yang telah dijabarkan di atas adalah:

1. Mendapatkan jumlah fitur yang optimal pada klasifikasi penurunan fungsi kognitif pada pasien stroke menggunakan metode klasifikasi *random forest*.
2. Mendapatkan jumlah *tree* yang optimal pada klasifikasi penurunan fungsi kognitif pada pasien stroke menggunakan metode klasifikasi *random forest*.

3. Mendapatkan tingkat akurasi klasifikasi penurunan fungsi kognitif pada pasien stroke dengan menggunakan metode *random forest*

1.4 Manfaat

Manfaat yang diharapkan dari penelitian ini adalah klasifikasi penurunan fungsi kognitif pada pasien stroke menggunakan metode klasifikasi *random forest* yang dibuat dapat membantu mengklasifikasi pasien stroke yang mengalami penurunan fungsi kognitif dengan lebih efisien.

1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Data latih dan data uji yang digunakan dalam membangun klasifikasi ini adalah dari data penelitian yang dilakukan di Rumah Sakit Saiful Anwar Malang pada bulan Juli s.d. Oktober tahun 2014.
2. Kelas hasil klasifikasi pada sistem ini adalah Normal, Kemungkinan, dan Gangguan.
3. Fitur-fitur yang digunakan dalam klasifikasi penurunan fungsi kognitif pada pasien stroke adalah usia, jenis kelamin, riwayat pendidikan, riwayat stroke, nilai BMI (*Body Mass Index*), riwayat hipertensi, diabetes melitus, riwayat hiperkolesterol, riwayat merokok, letak *lesi*, atrial fibrilasi, dan riwayat gagal ginjal.

1.6 Sistematika Pembahasan

Bagian ini berisi struktur proposal skripsi ini mulai bab pendahuluan sampai bab penutup dan deskripsi singkat dari masing-masing bab.

BAB 1 PENDAHULUAN

Bab 1 menjelaskan permasalahan yang ada tentang penyakit stroke dan faktor penurunan fungsi kognitif dan alasan penggunaan metode *random forest* untuk menyelesaikan masalah tersebut yakni mengklasifikasi dari data yang didapatkan. Selain itu dalam bab ini juga membahas tentang rumusan masalah, tujuan, batasan masalah, serta sistematika penulisan dari proposal skripsi ini.

BAB 2 LANDASAN KEPUSTAKAAN

Bab 2 menjelaskan lebih rinci tentang stroke dan faktor penurunan fungsi kognitif serta penjelasan tentang metode *random forest* secara lebih dalam. Selain itu dalam bab ini juga dijabarkan referensi kajian pustaka yang digunakan untuk memperkuat penelitian.

BAB 3 METODOLOGI

Bab 3 menguraikan langkah pengerjaan dalam proses penelitian mengenai klasifikasi stroke dan faktor penurunan fungsi kognitif yang meliputi tipe

penelitian, strategi penelitian, proses pengambilan data, implementasi, dan pengujian dari klasifikasi yang dilakukan.

BAB 4 ANALISIS DAN PERANCANGAN

Bab 4 menjabarkan analisis dan perancangan dari klasifikasi stroke dan faktor penurunan fungsi kognitif yang didalamnya akan dibahas juga mengenai penggunaan metode *random forest* dan cara kerjanya. Selain itu pada bab ini juga menjelaskan perancangan pengujian yang dilakukan terhadap sistem.

BAB 5 IMPLEMENTASI

Bab 5 menjelaskan pembahasan tentang implementasi dasar teori yang telah dibahas dalam bab sebelumnya yaitu KLASIFIKASI PENURUNAN FUNGSI KOGNITIF PASIEN STROKE MENGGUNAKAN METODE KLASIFIKASI RANDOM FOREST.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab pengujian dan analisis membahas hasil dari percobaan setiap skenario pengujian yang telah dijabarkan pada perancangan pengujian. Pengujian pada klasifikasi ini dilakukan dengan dua pengujian, pengujian seleksi fitur dan pengujian jumlah *tree*. Setiap pengujian tersebut dilakukan dengan menggunakan metode pengujian *K-fold cross validation*. Hal ini ditujukan untuk mengetahui tingkat keberhasilan dari klasifikasi yang telah dilakukan.

BAB 7 PENUTUP

Di dalam bab terakhir ini dituliskan kesimpulan dari penelitian dan implementasi KLASIFIKASI PENURUNAN FUNGSI KOGNITIF PASIEN STROKE MENGGUNAKAN METODE KLASIFIKASI RANDOM FOREST. Selain itu juga akan dituliskan saran untuk pengembang penelitian yang serupa di kemudian hari.

BAB 2 LANDASAN KEPUSTAKAAN

Bab 2 menjelaskan lebih rinci tentang stroke dan faktor penurunan fungsi kognitif serta penjelasan tentang metode *random forest* secara lebih dalam. Selain itu dalam bab ini juga dijabarkan referensi kajian pustaka yang digunakan untuk memperkuat penelitian.

2.1 Kajian Pustaka

Dalam subbab berikut ini menjelaskan tentang kajian pustaka yang penulis peroleh dari penelitian-penelitian yang sudah dilakukan sebelumnya yang selaras dengan topik penelitian yang penulis lakukan kali ini. Penelitian tersebut dijelaskan pada tabel kajian pustaka yang ditunjukkan pada Tabel 2.1.

Penelitian yang dilakukan oleh Rahmi Fitriani (2015) menyimpulkan bahwa penggunaan *random forest* dalam klasifikasi cukup efektif. Hal tersebut terlihat dari hasil grafik persentase *error* yang membandingkan dengan metode klasifikasi lain. Data yang mereka gunakan dalam penelitiannya adalah data numerik yang diambil dari delapan *dataset* yang terdapat pada *UCI Machine Learning Repository*.

Pada penelitian lain yang dilakukan oleh Resa Rukmigayatri (2015) juga mengatakan hal yang tidak jauh berbeda. Akurasi yang didapatkan dalam mengklasifikasi kemunculan titik panas pada lahan gambut di Sumatera dan Kalimantan adalah 87,40% untuk Sumatera dan 72,50% untuk Kalimantan. Dari tingkat akurasi tersebut, metode *random forest* dapat digunakan untuk melakukan klasifikasi yang penulis kerjakan saat ini.

Penelitian yang dilakukan Laras Binarwati (2017) menyatakan bahwa hasil dari klasifikasi menggunakan metode *random forest* yang dioptimalkan menggunakan algoritme genetika menghasilkan akurasi yang lebih tinggi dengan nilai 91% – 95% disbanding dengan hanya menggunakan *random forest* pada umumnya yang hanya mendapat nilai akurasi sebesar 40% – 95%.

Penelitian yang dilakukan oleh Deny Wiria Nugraha (2017) beserta rekan-rekan menyatakan mendapatkan tingkat keberhasilan dari klasifikasi yang dilakukan tersebut sebesar 89,65% tingkat akurasi. Penelitian tersebut menggunakan 203 data rekam medik pasien yang menderita penyakit stroke yang ada di Rumah Sakit Umum Daerah Udata Palu.

Penelitian terakhir yang dilakukan oleh Yustia Ika Wardhani (2015) di Rumah Sakit Saiful Anwar, Malang, menyatakan bahwa ada beberapa faktor yang dapat memengaruhi penurunan fungsi kognitif pada pasien penderita stroke. Hasil penelitian mereka adalah klasifikasi pasien apakah mengalami m penurunan fungsi kognitif atau tidak.

Tabel 2.1 Kajian Pustaka

No.	Judul	Objek	Metode	Hasil
1	Analisis dan Implementasi Algoritme <i>random forest</i> sebagai Sebuah <i>Classifier</i> dalam <i>Data Mining</i>	Data Numerik	<i>random forest</i>	Klasifikasi Data Numerik dengan Akurasi Tinggi
2	Klasifikasi Kemunculan Titik Panas pada Lahan Gambut di Sumatera dan Kalimantan Menggunakan Algoritme <i>random forest</i>	Data lahan gambut pulau Sumatera dan Kalimantan dari tahun 1990-2002	<i>random forest</i>	Klasifikasikan 3 titik dari data baru Kalimantan dan Sumatera
3	Implementasi Algoritme Genetika untuk Optimalisasi <i>random forest</i> Dalam Proses Klasifikasi Penerimaan Tenaga Kerja Baru: Studi Kasus PT.XYZ	Sumber Data Manusia untuk Penerimaan Tenaga Kerja	<i>random forest</i> dengan optimasi algoritme genetika	Pola Prediksi Penerimaan Tenaga Kerja
4	Klasifikasi Penyakit Stroke Menggunakan Metode <i>Naïve Bayes Classifier</i> (Studi Kasus pada Rumah Sakit Umum Daerah Udata Palu)	Data rekam medik pasien penderita stroke	<i>Naïve Bayes Classifier</i>	Klasifikasi Penyakit Stroke
5.	Faktor-Faktor yang Memengaruhi Penurunan Fungsi Kognitif Pasien Stroke di Rumah Sakit Saiful Anwar Malang	Pasien Stroke di Rumah Sakit Saiful Anwar	Chi Square Test	Faktor-faktor yang memengaruhi penurunan fungsi kognitif pada pasien penderita stroke trombosit

2.2 Klasifikasi

Klasifikasi adalah proses pengelompokan suatu objek yang belum memiliki kelas atau kelompok ke dalam suatu kelas atau kelompok yang telah ditentukan sebelumnya kelompok atau kelasnya. Pada klasifikasi terdapat dua proses utama yang dilakukan. Proses pertama adalah proses *training* atau pelatihan dari data latih yang dimiliki. Data latih adalah kumpulan data yang terdiri dari data-data yang telah memiliki kelas yang dijadikan acuan sistem untuk menentukan kelas dari objek baru. Proses yang kedua adalah proses *training*. Dalam proses ini, data uji yang telah memiliki kelas akan diuji kembali dengan menggunakan klasifikasi yang dilakukan dari data latih yang ada untuk menentukan ketepatan dari sistem yang telah dibuat (Binarwati, et al., 2017).

2.2.1 Komponen Dasar Klasifikasi

Untuk memperkuat sebuah proses klasifikasi, maka diperlukan beberapa komponen di dalamnya. Komponen tersebut antara lain adalah:

1. Kelas. Variabel bertipe kategorikal yang merepresntasikan kelompok sebuah objek. Kelas juga merupakan hasil *output* dari klasifikasi.
2. Fitur. Variabel yang memengaruhi klasifikasi kelas pada objek. Fitur ini akan menjadi sebuah *predictor* dalam proses klasifikasi.
3. Data Latih. Sekumpulan data berisikan kelas beserta fitur-fitur yang memengaruhi kelas tersebut.
4. Data Uji. Sekumpulan data yang telah memiliki kelas atau kelompok yang digunakan sebagai data penguji ketepatan klasifikasi yang telah dilakukan.

2.2.2 Metode Klasifikasi

Klasifikasi memiliki banyak metode untuk penyelesaiannya. Semua algoritme memiliki tingkat akurasi yang berbeda berdasarkan dari jumlah data yang digunakan maupun jumlah fitur yang terdapat pada data latih. Algoritme yang ada untuk melakukan klasifikasi antara lain adalah:

1. *Decision/Classification Tree*.
2. *Bayesian Classifier/Naïve Bayes Classifier*
3. *Neural Networks*
4. *Statical Analysis*
5. *random forest*
6. *K-Nearest Neighbour*
7. *Rule-based Methods*
8. *Memory Based Reasoning*

2.3 Classification and Regression Trees (CART)

Classification and Regression Trees (CART) merupakan salah satu metode klasifikasi dan regresi data dalam bentuk pohon keputusan. Pohon klasifikasi dibentuk saat *output* yang dihasilkan adalah berupa data kategori. Sedangkan pohon regresi dibentuk saat *output* yang dihasilkan adalah berupa data numerik (Breiman, 1984).

Proses klasifikasi dengan menggunakan CART bekerja dengan membangun sebuah pohon keputusan dari sebuah data latih yang ada. Setiap *leaf* atau daun dari pohon keputusan yang telah dibangun adalah kelas-kelas yang ada pada data latih. Setelah terbentuk sebuah pohon keputusan, maka terbentuk sebuah aturan dari setiap *leaf* yang ada untuk dilakukan klasifikasi data baru yang belum memiliki kelas atau kelompok. Aturan tersebut berjumlah *leaf* yang terbentuk pada pohon keputusan. Aturan yang dimaksud berisikan *node* dan *edge* yang menuju *leaf* tersebut.

2.4 Random Forest

Metode *random forest* merupakan sebuah perkembangan lebih lanjut dari metode CART. Metode ini dapat melakukan klasifikasi dan regresi. Yang membedakan dari CART adalah jumlah *tree* yang digunakan dalam metode ini lebih dari satu. Hasil klasifikasi atau regresi dari setiap *tree* akan dilakukan *voting* mencari kelas terbanyak yang dihasilkan. Kata *random* pada nama metode ini diartikan sebagai data latih yang digunakan dalam pembangunan *tree* dalam metode ini adalah hasil pengacakan secara tidak beraturan (Binarwati, et al., 2017).

2.4.1 Tahap Pengacakan Data

Pengacakan data dilakukan untuk mendapatkan data pada data latih secara acak untuk membedakan antara satu *tree* dengan *tree* yang lainnya. Tidak jarang terdapat redundansi data pada hasil pengacakan data latih. Data tersebut digunakan untuk menentukan kelas menggunakan pohon CART. Sehingga hasil klasifikasi dari setiap *tree* yang terbangun memiliki kelas yang berbeda.

2.4.2 Tahapan Pembangunan CART

Langkah selanjutnya setelah proses pengacakan data berdasarkan data latih adalah menumbuhkan pohon CART dari data *random* yang dihasilkan. Pemilihan fitur pemisah dalam CART adalah menggunakan nilai *gain* yang dihasilkan. Untuk mendapatkan nilai *gain* pada setiap fitur pada data latih, terlebih dahulu dilakukan perhitungan entropi kelas dan entropi setiap fitur tersebut seperti pada persamaan 2.1 dan 2.2.

$$Entropy(S) = \sum_{j=1}^k -p_j \log_2 p_j \quad (2.1)$$

Keterangan:

S : Himpunan Kelas

K : Jumlah Kelas

P_j: Probabilitas Kelas dalam Himpunan

$$Gain(S, A) = Entropy(S) - \sum_{j=1}^k Entropy(A) \quad (2.2)$$

Keterangan:

S : Himpunan Kelas

A : Himpunan Fitur

2.4.3 Tahap Voting

Tahap *voting* pada metode *random forest* dilakukan untuk menentukan kelas akhir dari kelas-kelas yang telah dihasilkan dari sejumlah *tree* yang telah dibangun pada proses sebelumnya. *Voting* sendiri dilakukan dengan mencari jumlah kelas yang dihasilkan dari sejumlah *tree* tersebut pada setiap data uji yang ada.

2.4.4 Error Random Forest

Dikarenakan ada proses pengacakan data sebelum melakukan pembangunan *tree*, metode *random forest* ini terdapat *error*. Hal ini terjadi ketika ada data yang tidak ikut ke dalam data yang diacak. Selain itu, redundansi data juga menjadi salah satu faktor *error* pada *random forest*.

2.4.5 Kondisi Berhenti

Dalam *random forest* terbentuk lebih dari satu *decision tree* yang berasal dari hasil random data latih yang dimiliki. Pembangunan *tree* tersebut akan berhenti pada kondisi tertentu. Seperti yang ada pada metode CART, pembangunan *decision tree* pada *random forest* akan berhenti apabila data latih yang digunakan sudah homogen atau dengan jumlah *tree* tertentu.

2.5 Stroke

Stroke adalah istilah untuk menggambarkan perubahan neurologis yang terjadi karena gangguan aliran darah pada otak penderita. Penanganan yang cepat dan tepat dibutuhkan pada penderita stroke dikarenakan perubahan neurologi ini terjadi secara mendadak. Stroke merupakan kondisi darurat dikarenakan penderita akan mengalami penurunan oksigen dan aliran darah yang dapat menyebabkan kerusakan otak permanen (Malik & Maulina, 2015).

Penyakit stroke dapat menyebabkan kecacatan dalam jangka yang Panjang bagi penderitanya. Data menunjukkan bahwa angka kecacatan karena stroke lebih tinggi dari angka kematian. Perbandingan kecacatan stroke dengan kematian

tersebut adalah empat banding satu. Penderita stroke yang mengalami cacat adalah penderita yang berusia 45 tahun atau lebih.

Klasifikasi stroke terbagi menjadi dua, yaitu stroke pendarahan dan stroke *iskemik*. 85% penyakit stroke diakibatkan oleh adanya penyumbatan atau yang disebut stroke *iskemik*, sisanya diakibatkan oleh adanya pendarahan pada otak penderita (Stroke Accosiation, 2013).

Stroke *iskemik* dibagi lagi menjadi beberapa subtype yaitu stroke *emboli* dan stroke thrombosis. Stroke thrombosis diakibatkan oleh proses arteriosklerosis yang menyumbat pembuluh darah besar penderita yaitu arteri serebral dan servikal. Sedangkan stroke *emboli* diakibatkan karena adanya gumpalan darah yang berasal dari bagian lain sistem darah di arteri, misalnya yang berasal dari *lesi* jantung (Wardhani, 2015).

2.6 Fungsi Kognitif

Kognitif atau *cognition* yang artinya mengetahui dapat diartikan sebagai berkenan, bersifat, atau ditandai dengan aktivitas pikiran yang membuat seseorang menyadari objek pikiran dalam hal ini kognisi meliputi semua aspek menerima, berpikir, dan mengingat. Fungsi kognitif dimaksudkan untuk menunjukkan kemampuan belajar, menerima, dan mengolah dari manusia terhadap informasi dari lingkungan sekitarnya (Wardhani, 2015).

Fungsi kognitif dibagi menjadi empat bagian yaitu masukan, penyimpanan, pemrosesan, dan keluaran di mana dijelaskan lebih lanjut:

1. Fungsi masukan yang termasuk kemampuan untuk memilih, mendapatkan, mengklasifikasikan, dan mengintegrasikan informasi.
2. Fungsi penyimpanan adalah kemampuan mendapatkan kembali apa yang telah dimasukkan dari fungsi sebelumnya. Fungsi ini dapat juga disebut memori dan pembelajaran.
3. Proses berpikir yang termasuk mengorganisir mental dan juga informasi.
4. Fungsi keluaran atau fungsi ekspresif yang berarti bagaimana informasi akan disampaikan atau dikomunikasikan.

Fungsi kognitif juga meliputi tentang bahasa, memori, atensi, orientasi, dan kemampuan konstruksional. Beberapa domain fungsi kognisi yang biasanya dites dalam pemeriksaan *neuropsychologist* adalah atensi, memori, bahasa, persepsi visual, dan kemampuan visuospatial, serta fungsi eksekusi (Hofgren, 2009).

2.6.2 Penurunan Fungsi Kognitif

Penurunan fungsi kognitif pada manusia rata-rata dimulai pada usia 30 tahun dan penurunan itu berlangsung secara progresif. Beberapa faktor yang memengaruhi penurunan fungsi ini adalah faktor vaskular, hal ini dapat diartikan sebagai gangguan yang berlanjut dari *mild cognitive impairment* sampai *demensia*. Penurunan fungsi tersebut dikarenakan adanya keterlibatan kerusakan vaskular yang berguna untuk memori, kognisi, dan kebiasaan. Kelainan kognitif akibat

vaskular dapat dideskripsikan sebagai kelainan heterogen dengan berbagai tipe *lesi cerebrovascular* yang berbeda dan akan berkembang sampai dengan tahap akhirnya yaitu *demesia* (Jellinger, 2013).

2.6.3 Faktor-Faktor yang Penurunan Fungsi Kognitif

Dua dari tiga pasien stroke mengalami gangguan atau penurunan fungsi kognitif setelah terserang penyakit stroke. Sedangkan satu pertiga sisanya mengalami *demensia*. Penurunan fungsi kognitif ini sepuluh kali lebih beresiko pada pasien yang memiliki riwayat stroke dibandingkan orang biasa yang tidak punya riwayat stroke.

Penderita gangguan kognitif ini menimbulkan tanda-tanda yang tidak terlalu signifikan pada tiga bulan setelah terkena penyakit stroke. Tanda-tanda selanjutnya mengalami variasi penurunan fungsi selama 15 tahun. Berdasarkan penelitian yang dilakukan *Douiri*, 40,8% penderita stroke mengalami penurunan fungsi kognitif, 18,4% pasien mengalami stroke thrombosis, 11,2% stroke *emboli*, dan 11,2% lainnya tidak diketahui *patogenesisnya*.

2.6.3.1 Usia

Usia merupakan salah satu faktor yang memengaruhi penurunan fungsi kognitif pada pasien stroke. Hal ini dikarenakan penderita stroke berusia lanjut mengalami gangguan anti radikal bebas seperti *glutathione hydrogen peroxidase* dan *superoksida dismutase* (SOD). Penurunan anti radikal bebas ini menyebabkan terjadinya beberapa penyakit seperti kanker, penyakit jantung dan pembuluh darah, pikun, katarak, dan salah satunya adalah penurunan fungsi kognitif. Itulah alasan mengapa pasien penderita stroke, prevalensi gangguan kognitif meningkat seiring dengan meningkatnya usia.

2.6.3.2 Jenis Kelamin

Dari berbagai penelitian mengenai stroke, dikatakan wanita lebih sedikit mengalami penyakit stroke dikarenakan wanita terlindungi oleh hormon *esterogen* dan *endogen* (Alkayed & al., 1998). Hal ini menandakan bahwa jenis kelamin menjadi salah satu faktor penurunan fungsi kognitif pada pasien stroke.

2.6.3.3 Tekanan Darah

Alasan mengapa tekanan darah menjadi salah satu faktor yang memengaruhi penurunan fungsi kognitif pada pasien stroke adalah, tekanan darah tinggi atau hipertensi dikaitkan sebagai faktor kumulatif dalam penurunan fungsi kognitif bersama dengan faktor obesitas. Tingginya tekanan darah meningkatkan jumlah kebocoran darah pada otak (Wreksoatmodjo, 2014).

2.6.3.4 Kolesterol

Tingginya kadar kolesterol akan menyebabkan terjadinya *aterosklerosis*. *Aterosklerosis* adalah penyempitan yang terjadi pada dinding pembuluh darah dikarenakan penumpukan plak pada arteri. Plak tersebut terbentuk dari zat lemak, kalsium, dan *fibrin* (zat dalam darah). Hal ini dapat menyebabkan penyumbatan

pada pembuluh darah pada otak yang menyebabkan stroke (Wreksoatmodjo, 2014). Proses *aterosklerosis* adalah etologi dari stroke (Aaronson, 2012).

2.6.3.5 Diabetes Melitus

Orang yang memiliki riwayat diabetes melitus enam kali berisiko untuk terjadinya stroke. Hal ini dikarenakan gula darah yang tidak seimbang menyebabkan penyumbatan pembuluh darah pada otak. Secara tidak langsung, penderita diabetes melitus ini juga mengalami penurunan fungsi kognitif (Wreksoatmodjo, 2014).

2.6.3.6 Volume Kerusakan Otak

Pasien stroke sudah dipastikan mengalami kerusakan pada otak. Namun volume dan luas kerusakan otak setiap pasien berbeda. Pasien yang mengalami kerusakan otak lebih dari 100ml akan mengalami penurunan fungsi dibandingkan dengan pasien yang mengalami kerusakan kurang dari 100ml. Salah satu skala yang digunakan untuk menilai angka keparahan stroke adalah NHSS (*National Institute of Health Stroke Scale*).

2.6.3.7 Letak Kerusakan Otak

Letak kerusakan otak juga memengaruhi menurunnya fungsi kognitif pada pasien stroke. Letak kerusakan otak di *hemisfer* kanan maupun kiri memunculkan wujud gejala yang tidak sama. Hal ini terjadi dikarenakan adanya proses lateralisasi dari fungsi tertentu ke salah satu *hemisfer*. Pada *hemisfer* kiri akan menimbulkan gangguan kemampuan menulis, kemampuan membaca, kemampuan memori, dan kemampuan mengingat. Sedangkan pada *hemisfer* kanan akan menimbulkan gangguan fungsi *visuomotor*, fungsi pengabaian, memori visual, dan koordinasi motorik (Harsono, 2007).

2.6.3.8 Pendidikan

Pada penelitian yang dilakukan Bannet (2003) mengatakan bahwa adanya penurunan fungsi kognitif yang kuat berdasarkan tingkat edukasi formal. Penderita stroke dengan tingkat pendidikan rendah akan lebih menurun fungsi kognitifnya dibandingkan dengan penderita stroke yang riwayat pendidikannya tinggi (Lee, 2003). Hal tersebut dikarenakan pendidikan berhubungan dengan fungsi kognitif berdasarkan pada potensi kognitif yang didapatkan sejak lahir. Teori lain menjelaskan tentang *synaptic reverse hypothesis* di mana orang yang berpendidikan tinggi punya lebih banyak *synaps* di otak dibandingkan dengan orang yang berpendidikan di bawah standar atau rendah. Pada saat *synaps* tersebut rusak, maka *synaps* lain akan menggantikan dan mampu mengkompensasi dengan baik terhadap hilangnya suatu kemampuan otak.

2.6.3.9 Merokok

Nikotin pada rokok dengan skala kecil dan jangka pendek menyebabkan efek positif dan menguntungkan untuk fungsi kognitif manusia terutama fungsi atensi, belajar, dan daya ingat. Namun hal tersebut tidak berlaku untuk rokok jangka

panjang. Merokok dengan skala yang besar dapat menurunkan fungsi kognitif manusia dikarenakan akan menurunkan kadar anti oksidan dan meningkatkan jumlah plak pada otak pasien stroke (Wreksoatmodjo, 2014).

2.6.3.10 Obesitas

Obesitas sangat berhubungan dengan kolesterol tinggi, diabetes, dan tekanan darah tinggi. Ketiga faktor tersebut sangat berpengaruh terhadap penurunan fungsi kognitif pada pasien stroke. Selain itu, penderita stroke yang obesitas mengalami penyempitan pembuluh darah pada otak dikarenakan timbunan lemak yang ada pada otak sudah lebih dari angka normal. Hal tersebut tentu memengaruhi penurunan fungsi kognitif pada penderitanya (Wreksoatmodjo, 2014).

2.6.3.11 Alkohol

Pada peminum alkohol berat didapatkan penurunan fungsi pembelajaran dan memori sebagaimana adanya perubahan personalitas dan individual. Konsumsi alkohol kronis dapat menyebabkan kerusakan, hal ini akan berkembang menjadi penurunan fungsi kognitif yang *irreversible*. Dengan demikian alkohol sangat berpengaruh terhadap penurunan fungsi kognitif pada manusia.

2.6.3.12 Riwayat Stroke

Stroke merupakan penyakit yang dikarenakan penyumbatan darah pada otak. Selain itu stroke juga dapat disebabkan oleh kebocoran pembuluh darah pada otak yang membuat saraf pada otak membusuk. Hal ini dapat menyebabkan penurunan fungsi kognitif pada manusia karena bagian otak yang mengalami kerusakan itu tidak dapat bekerja dalam melakukan fungsi kognitif (Shin, 2013).

2.7 Evaluasi

Tahap evaluasi dilakukan setelah semua proses pada pembangunan sistem ini telah selesai. Evaluasi bertujuan untuk mengetahui seberapa baik sistem yang telah dibangun tersebut. Pada proses evaluasi pada KLASIFIKASI PENURUNAN FUNGSI KOGNITIF PASIEN STROKE MENGGUNAKAN METODE KLASIFIKASI RANDOM FOREST dilakukan dengan beberapa tahap yaitu evaluasi parameter yang optimal dan evaluasi akhir dengan parameter optimal yang telah didapatkan sebelumnya. Proses tersebut akan dijelaskan lebih lanjut pada perancangan pengujian pada Bab Perancangan.

Kedua proses tersebut dievaluasi dengan mencari nilai akurasi dari setiap pengujianinya. Perhitungan akurasi dilakukan dengan rumus seperti pada persamaan 2.3.

$$\text{Akurasi} = \frac{\text{Jumlah Klasifikasi Sesuai}}{\text{Jumlah Data Uji}} \times 100\% \quad (2.3)$$

BAB 3 METODOLOGI

Dalam bab 3 ini menguraikan langkah pengerjaan dalam proses penelitian mengenai klasifikasi stroke dan faktor penurunan fungsi kognitif yang meliputi tipe penelitian, strategi penelitian, proses pengambilan data, implementasi, dan pengujian dari klasifikasi yang dilakukan

3.1 Tipe Penelitian

Penelitian yang dilakukan oleh penulis adalah jenis penelitian non-implementatif deskriptif yang memfokuskan pada kasus klasifikasi penurunan fungsi kognitif pada pasien stroke berdasarkan hasil analisis terhadap data pasien stroke yang telah diperoleh. Hasil produk dari penelitian ini adalah pengujian akurasi dari klasifikasi yang dilakukan dari data yang telah didapatkan.

3.2 Strategi Penelitian

Strategi penelitian yang dilakukan penulis adalah melakukan implementasi metode *random forest* terhadap data pasien stroke guna mendapatkan jawaban dari rumusan masalah yang telah dijabarkan pada Bab 1.

Dalam penelitian ini dilakukan beberapa langkah untuk menyelesaikannya. Jadwal pengerjaan dari setiap langkah-langkah yang dilakukan guna mencapai tujuan penelitian ini digambarkan pada Tabel 3.1.

Tabel 3.1 Jadwal Penelitian

No	Uraian	Agustus				September				Oktober				November				Desember			
		Minggu ke-																			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Studi Kepustakaan																				
2	Pengambilan Data																				
3	Implementasi																				
4	Pengujian																				
5	Kesimpulan																				

3.3 Pengambilan Data

Tahapan pengambilan data ini dilakukan menggunakan data yang diambil dari data penelitian Fakultas Kedokteran Universitas Brawijaya pada Rumah Sakit Saiful Anwar Malang pada bulan Juli sampai dengan Oktober pada tahun 2014 yang di dalamnya terdapat data pasien stroke thrombosis.

Data yang didapatkan adalah data klasifikasi yang kelas akhirnya adalah Normal, Kemungkinan, dan Gangguan. Serta fitur yang digunakan untuk mendapatkan kelas tersebut berjumlah 13 fitur yang diantaranya adalah usia, jenis kelamin, pendidikan, riwayat stroke, nilai BMI (*Body Mass Index*), riwayat hipertensi, diabetes melitus, riwayat hiperkolesterol, riwayat merokok, letak *lesi*, atrial fibrilasi, dan gagal ginjal.

Data data yang didapatkan dalam penelitian sebelumnya tersebut berjumlah 95 data. Dari 95 data tersebut, 76 akan digunakan sebagai data latih, 19 sisanya akan digunakan sebagai data uji. Penyebaran dari 95 data latih yang dimiliki adalah 32 normal, 31 kemungkinan, dan 7 gangguan.

3.4 Implementasi

Implementasi adalah penerapan klasifikasi sesuai dengan perancangan yang telah dibuat terlebih dahulu yaitu manualisasi data latih yang didapat. Implementasi pada penelitian kali ini dilakukan dengan menggunakan bahasa pemrograman Python. Pemilihan bahasa pemrograman Python sendiri dikarenakan teknik penulisan bahasa pemrograman Python tidak sulit. Penulisan kode dilakukan tanpa harus menuliskan tipe data sebelumnya dan variabelnya yang dinamis.

Pada tahap implementasi dilakukan klasifikasi dari data latih yang didapatkan menggunakan metode *random forest*. Metode *random forest* dipilih berdasarkan hasil dari penelitian yang dilakukan Rukmigayatri (2015), metode ini menghasilkan tingkat akurasi sebesar 87,40%.

Hasil yang disajikan dalam klasifikasi yang dilakukan penulis dalam penelitian kali ini adalah tanpa *user interface*. Proses *input* untuk melakukan pengujian di langkah berikutnya. Proses *input* yang dilakukan adalah angka untuk menentukan jumlah pembangunan *decision tree* yang dilakukan dalam metode *random forest*.

Implementasi klasifikasi ini dilakukan dengan beberapa perangkat keras dan juga perangkat lunak untuk mendukung terbangunnya sistem. Adapun perangkat keras dan perangkat lunak yang digunakan untuk membangun sistem ini adalah:

1. Perangkat Keras:
 - a. MSI GL62M 7RDX Laptop
 - b. 4096MB RAM
 - c. Intel® Core™ i7-770HQ CPU
 - d. NVIDIA GeForce GTX 1050
2. Perangkat Lunak:
 - a. Anaconda Promt
 - b. Spyder
 - c. Microsoft Office

3.5 Pengujian

Pengujian sistem dilakukan guna mengetahui nilai akurasi dari klasifikasi penyakit penyakit stroke dan faktor penurunan fungsi kognitif yang dibangun ini. Pengujian yang dilakukan adalah menggunakan 19 data uji dari 95 data yang 76 sisanya telah menjadi data latih. Data uji akan diolah menggunakan metode *random forest* untuk mendapatkan klasifikasinya. Sistem dikatakan akurat dengan nilai akurasi tertentu apabila hasil dari data uji tersebut memiliki kelas yang sama dengan kelas aslinya.

3.6 Kesimpulan

Setelah melalui beberapa tahapan penelitian mulai dari studi kepustakaan, pengambilan data, implementasi, serta pengujian, maka dapat ditarik sebuah kesimpulan penelitian serta saran yang dapat dijadikan motivasi untuk mengembangkan sistem yang lebih baik dan sempurna ke depannya.



BAB 4 ANALISIS DAN PERANCANGAN

Pada bab analisis dan perancangan menjabarkan tentang analisis kebutuhan sistem, identifikasi permasalahan objek, perancangan algoritme *random forest*, dan perancangan skenario untuk pengujian di langkah selanjutnya.

4.1 Analisis Kebutuhan

Pada penelitian kali ini, penulis melakukan implementasi program menggunakan bahasa pemrograman Python. Sedangkan perangkat lunak yang digunakan untuk implementasi tersebut adalah Spyder yang merupakan editor untuk menjalankan bahasa pemrograman Python.

4.2 Identifikasi Permasalahan

Permasalahan yang akan diselesaikan pada penelitian kali ini adalah klasifikasi penurunan fungsi kognitif pada pasien stroke. Klasifikasi ini diselesaikan dengan metode klasifikasi *random forest*. Hasil akhir yang diperoleh adalah nilai akurasi penggunaan metode *random forest* untuk klasifikasi pada kasus penurunan fungsi kognitif pada pasien stroke.

Klasifikasi penurunan fungsi kognitif pada pasien stroke ini menggunakan data yang diambil dari skripsi mahasiswa Fakultas Kedokteran Universitas Brawijaya yang sebelumnya telah melakukan penelitian mengenai faktor-faktor yang memengaruhi penurunan fungsi kognitif pada pasien stroke yang dilakukan di Rumah Sakit Saiful Anwar, Malang. Faktor-faktor tersebut yang pada akhirnya penulis jadikan fitur pada data latih yang penulis gunakan.

Parameter yang digunakan dalam perhitungan dengan menggunakan metode *random forest* adalah jumlah *tree* yang dibangun dari hasil acak data latih yang digunakan. Proses klasifikasi *random forest* dibagi menjadi 2 bagian yaitu *training* dan *testing*. Pada proses *training* data latih akan diacak sejumlah n . Dari setiap data yang telah diacak tersebut akan dibuat masing-masing *decision tree*. Setelah semua *tree* mendapatkan hasil klasifikasinya, maka akan dilakukan *voting* yang akan menjadi hasil akhir klasifikasi dari metode *random forest*. Selanjutnya pada proses *testing* akan dilakukan pengujian pada 19 data dari 95 *dataset* yang di mana 76 data set tersebut menjadi data latih. 19 data uji tersebut kan dilakukan klasifikasi dengan metode yang telah dibangun sebelumnya dan akan dilihat hasil klasifikasinya. Besar persentase kebenaran antar data yang telah diuji dengan kelas asli data uji tersebut akan menjadi nilai akurasi dari klasifikasi penurunan fungsi kognitif pada pasien stroke ini.

Berikut adalah beberapa sampel data yang digunakan pada penelitian kali ini yang ditunjukkan pada Tabel 4.1. sebanyak 10 data (data lengkap terdapat pada lampiran).

Tabel 4.1 Data Faktor Penurunan Fungsi Kognitif Pasien Stroke

FITUR						Mini Mental State Examination (Kelas)
Nama	Usia	Sex	Pendidikan	Stroke	...	
SUK	55	1	2	2	...	Normal
MLY	50	1	4	2	...	Normal
OBH	68	2	2	1	...	Gangguan
APY	47	1	2	2	...	Kemungkinan
HWN	73	1	2	2	...	Kemungkinan
ATM	49	2	2	1	...	Gangguan
TRI	45	2	1	2	...	Kemungkinan
NUR	53	1	4	2	...	Kemungkinan
ARS	39	1	4	2	...	Normal
GFR	53	1	4	2	...	Normal

Keterangan untuk setiap fitur yang ada pada Tabel 4.1 adalah sebagai berikut:

1. Usia: Usia dari pasien stroke
2. Sex: Jenis kelamin dari pasien stroke. 1 menandakan wanita, 2 menandakan pria.
3. Pendidikan: Status pendidikan terakhir dari pasien stroke. 1 menggambarkan SD, 2 menggambarkan SMP, 3 menggambarkan SMA, 4 menggambarkan Pendidikan Tinggi.
4. Stroke: Riwayat stroke dari pasien stroke. 1 tidak ada riwayat stroke, 2 ada riwayat stroke.
5. Hipertensi: Kondisi tekanan darah dari pasien stroke. 1 menggambarkan tidak hipertensi, 2 menandakan hipertensi.
6. Hiperkolestrol: Riwayat kolesterol dari pasien stroke. 1 menandakan tidak ada riwayat kolesterol, 2 menandakan ada.
7. Diabetes Melitus: riwayat Diabetes Melitus dari pasien stroke. 1 menandakan tidak ada diabetes, 2 menandakan ada riwayat diabetes.
8. Atrial Fibrilasi: status kelainan jantung dari pasien stroke. 1 menandakan tidak ada kelainan, 2 menandakan ada kelainan.
9. Rokok: 1 menandakan bahwa pasien bukan perokok, 2 menandakan pasien adalah perokok
10. Alkohol: 1 menandakan bahwa pasien tidak pernah mengonsumsi alkohol, 2 menandakan pasien adalah pengonsumsi alkohol
11. BMI: Pembagian antara berat badan dengan tinggi badan dari pasien stroke

12. *Hemisfer*: letak *lesi* dari otak yang terkena stroke. 1 kanan, 2 kiri.
13. CT Scan: riwayat stroke dari pasien stroke. 1 menandakan tidak ada penyumbatan pembuluh darah, 2 adanya penyumbatan
14. GFR: riwayat gagal ginjal dari pasien stroke. Angka 90 adalah *stage* 1, 60 – 89 adalah *stage* 2, 30 – 59 adalah *stage* 3, 15 – 29 adalah *stage* 4, dan kurang dari 15 adalah *stage* 5

4.3 Perancangan Algoritme

Pada penelitian kali ini penulis menggunakan metode *random forest* untuk menyelesaikan masalah pada objek yang telah ditentukan di awal. Berikut adalah diagram alir dari proses klasifikasi menggunakan metode *random forest* yang akan ditunjukkan pada Gambar 4.1.

Langkah-langkah pada diagram alir *random forest* pada Gambar 4.1 adalah:

1. Sistem dimulai
2. Inisialisasi data latih dan data uji
3. Jumlah *tree* sebagai masukan
4. Menjalankan *random forest* dengan multi *tree*
5. Perulangan *i* sepanjang data uji
6. Memasukkan data uji kolom terakhir menjadi kelas data uji
7. Melakukan penilaian akurasi dengan *method* akurasi
8. Menghasilkan keluaran berupa hasil nilai akurasi
9. Sistem selesai

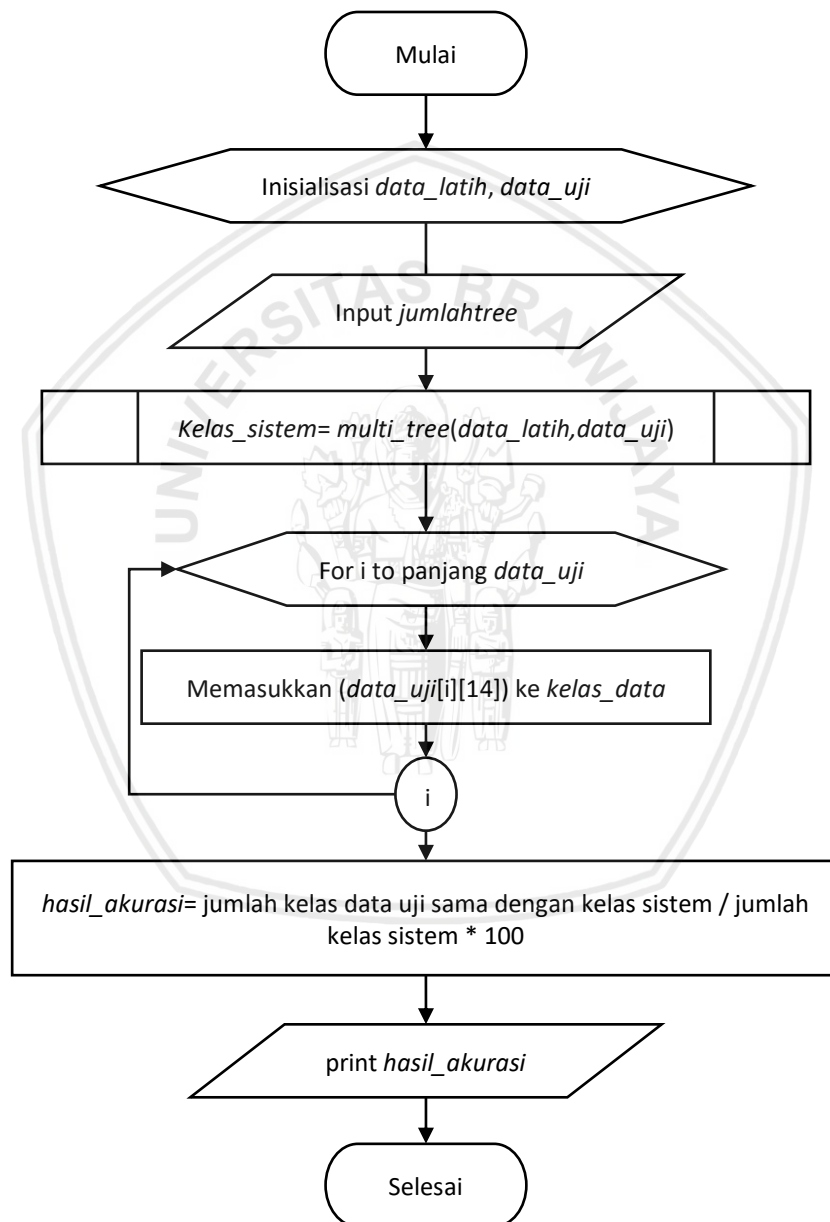
4.3.1 Proses *random forest*

Dalam proses *random forest* ini dilakukan *training* dari data latih dan dilanjutkan dengan melakukan *testing* dengan data uji yang ada. Setelah pengujian selesai, akan dilakukan *voting* terhadap hasil pengujian di proses sebelumnya. Lebih lengkap tentang proses *random forest* ini akan digambarkan pada Gambar 4.2.

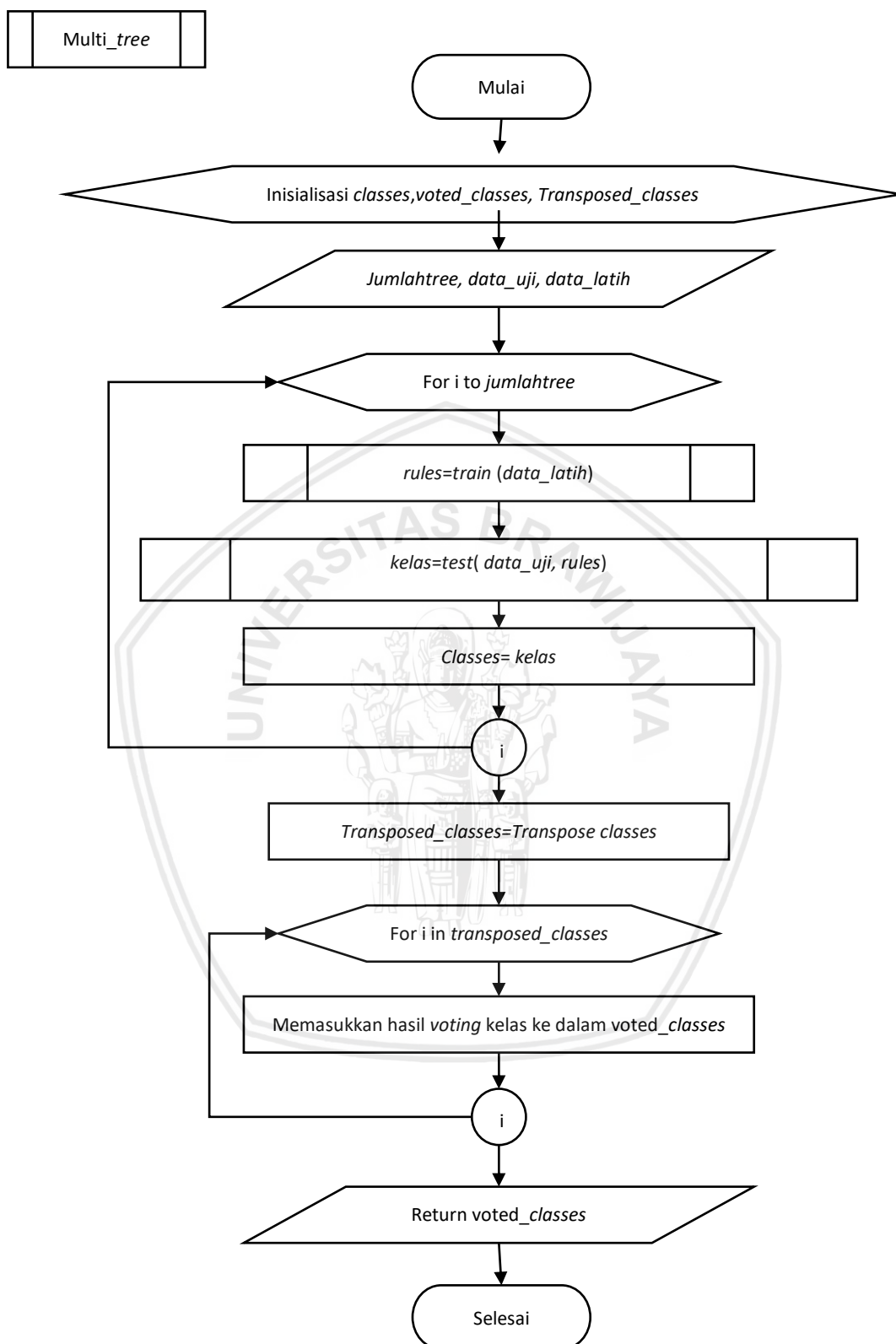
Langkah-langkah pada diagram alir proses *random forest* pada Gambar 4.2 adalah:

1. Sistem dimulai
2. Inisialisasi *classes* dan *voted_classes*
3. Jumlah *tree*, data uji, dan data latih sebagai input dari *method* ini
4. Perulangan *i* sepanjang jumlah *tree*
5. Melakukan *training* data latih
6. Melakukan *testing* terhadap data uji dan dimasukkan ke dalam *classes*

7. Melakukan transpose *classes* untuk devoting
8. Perulangan i sebanyak hasil transpose
9. Melakukan *voting*
10. Menghasilkan keluaran hasil *voting*
11. Sistem selesai



Gambar 4.1 Diagram Alir *random forest*



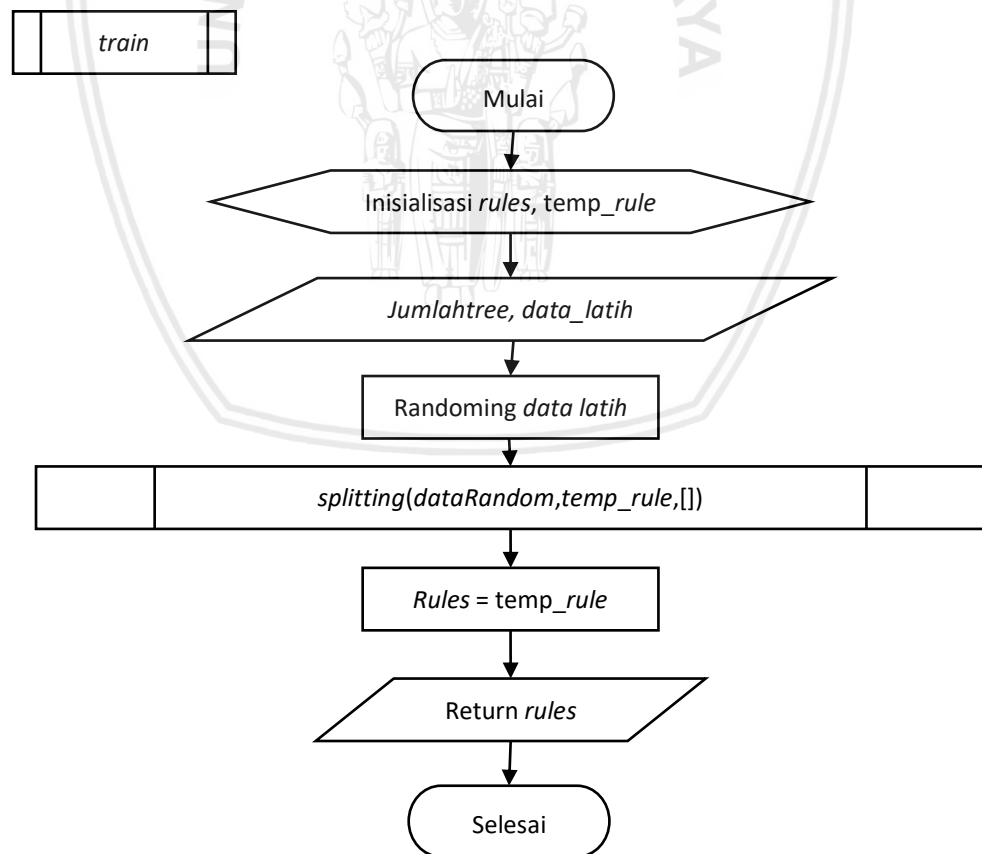
Gambar 4.2 Diagram Alir Proses *random forest*

4.3.2 Proses Training

Pada proses *training* ini dilakukan pembangunan *tree* sejumlah jumlah *tree* yang telah ditentukan sebelumnya. Data yang dilakukan pembangunan *tree* adalah hasil random dari data latih yang dimiliki. Proses *training* digambarkan pada diagram alir Gambar 4.3.

Langkah-langkah pada diagram alir proses *training* pada Gambar 4.3 adalah:

1. Sistem dimulai
2. Inisialisasi *rules* dan temp *rule*
3. Jumlah *tree* dan data latih sebagai inputan
4. Melakukan random pada data latih
5. Melakukan pembangunan *tree* pada data random
6. Memasukkan temp *rule* ke dalam *rules*
7. Mengembalikan nilai *rule* sebagai keluaran *method training*
8. Sistem selesai



Gambar 4.3 Diagram Alir Proses *train*

4.3.3 Proses Pembangunan *Tree*

Proses pembangunan *tree* ini dilakukan secara rekursif untuk membentuk node-node hingga leaf yang menunjukkan kelas dari *tree* tersebut. Data latih yang dibantung adalah hasil random dari data latih yang sebelumnya telah dilakukan. Proses pembangunan *tree* ini digambarkan pada Gambar 4.4.

Langkah-langkah pada diagram alir proses pembangunan *tree* pada Gambar 4.4 adalah:

1. Sistem dimulai
2. Inisialisasi split data
3. Data, temp *rule*, dan *rule* sebagai inputan
4. Membaca kelas yang ada pada data
5. Mencari index dengan *method* entropi
6. Mengurutkan isi fitur pada kolom indx disimpan dalam isi_fitur
7. Perulangan i sebanyak isi_fitur
8. Memasukkan array kosong ke split data sebanyak isi dari isi_fitur
9. Perulangan i sepanjang split data
10. Memisahkan data berdasarkan isis_fitur dimasukkan ke dalam split_data
11. Seleksi jika kelas masih lebih dari 1
12. Menambahkan *rule* berupa indx dan isi fitur tersebut
13. Melakukan rekursif *method* ini
14. Jika kelas tinggal 1
15. Menambahkan *rule* berupa indx dan isi fitur beserta kelas tersebut
16. Menghasilkan *rule* sebagai keluaran
17. Sistem selesai

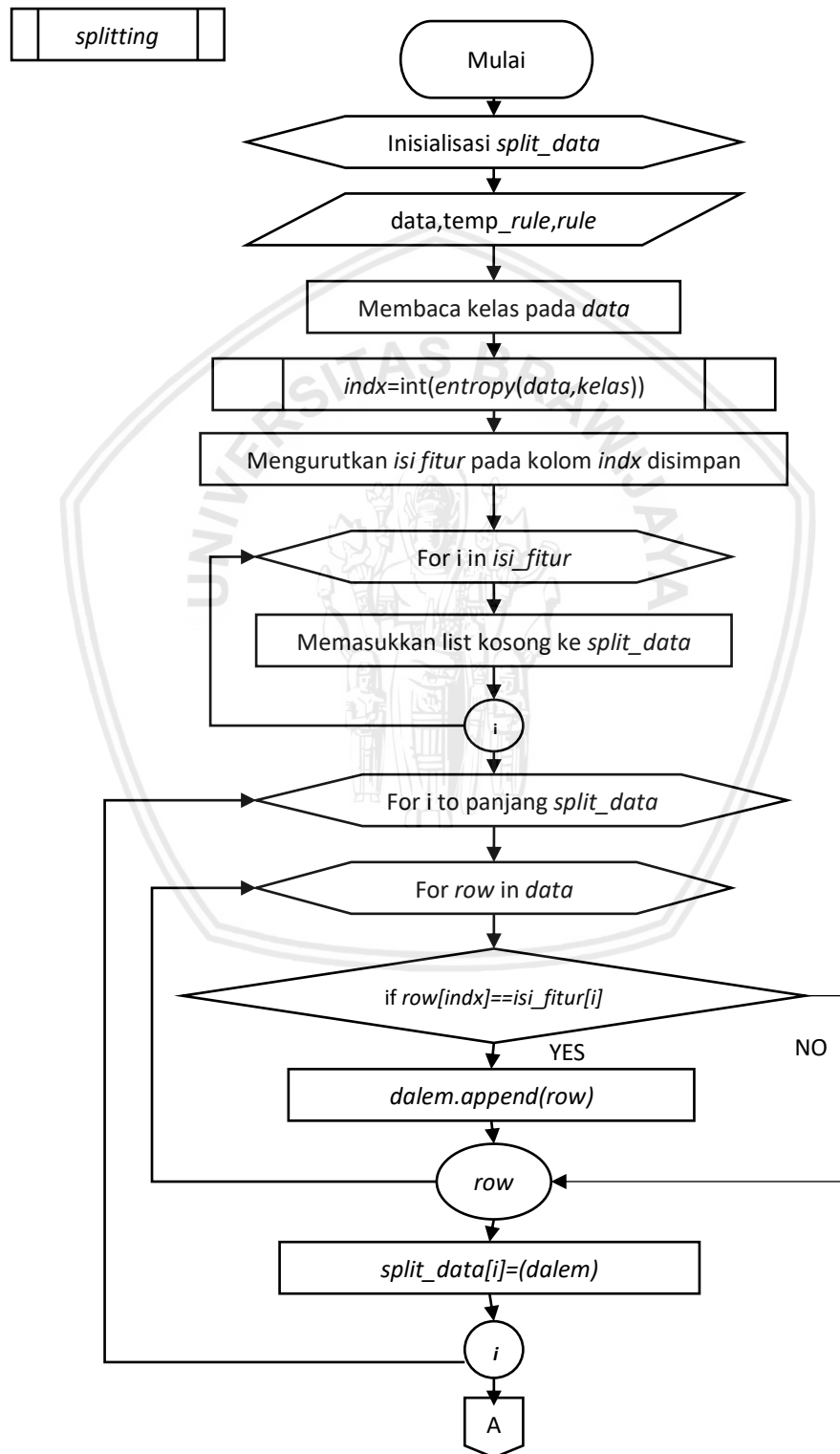
4.3.4 Proses Perhitungan Entropi

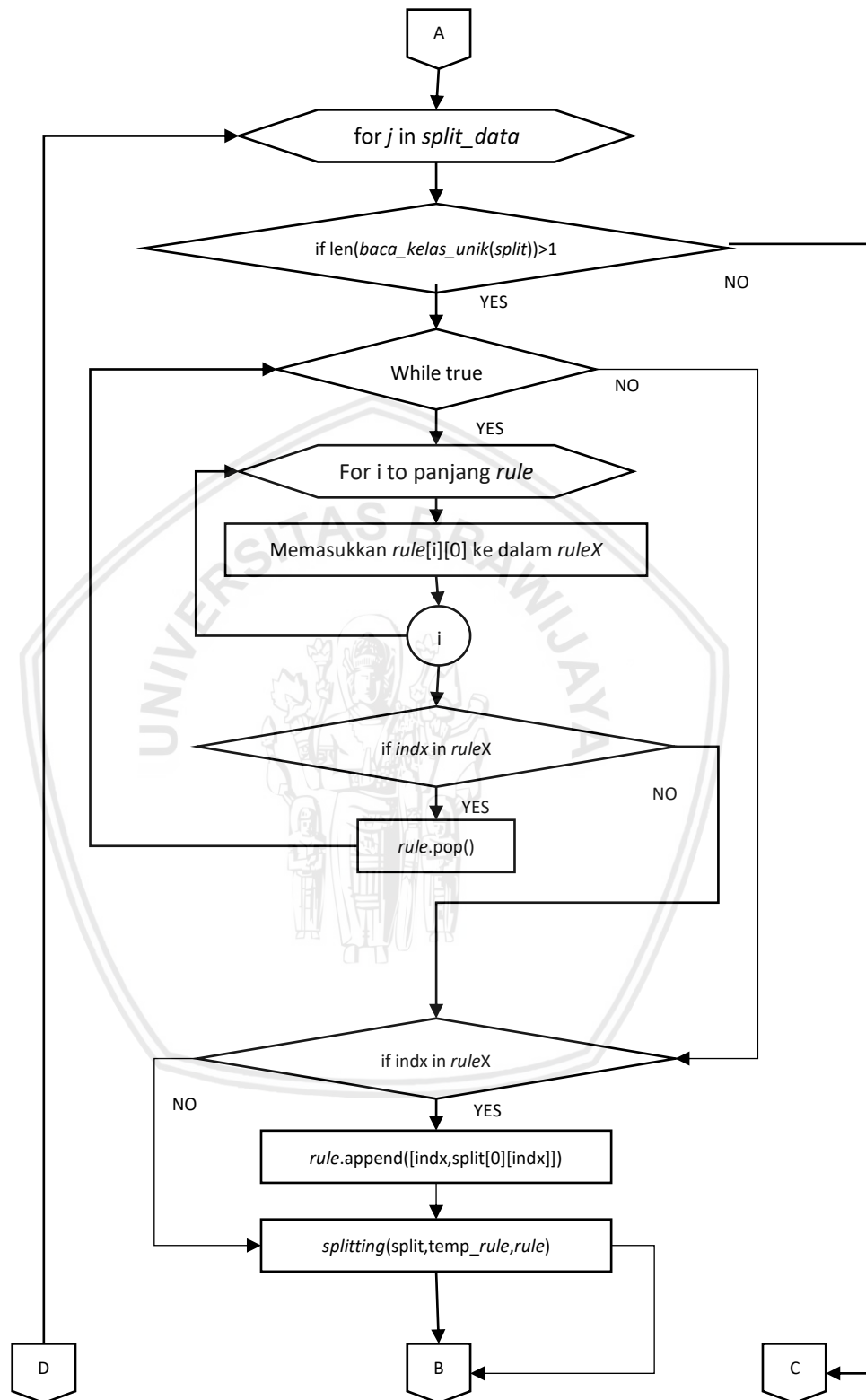
Proses perhitungan entropi dilakukan untuk menentukan fitur yang terbaik pada data latih untuk dilakukan *splitting*. Proses perhitungan ini dilakukan pada setiap kolom data latih yang sebagai fitur. Sistem ini mengembalikan nilai berupa indx. Proses perhitungan entropi ini digambarkan pada Gambar 4.5.

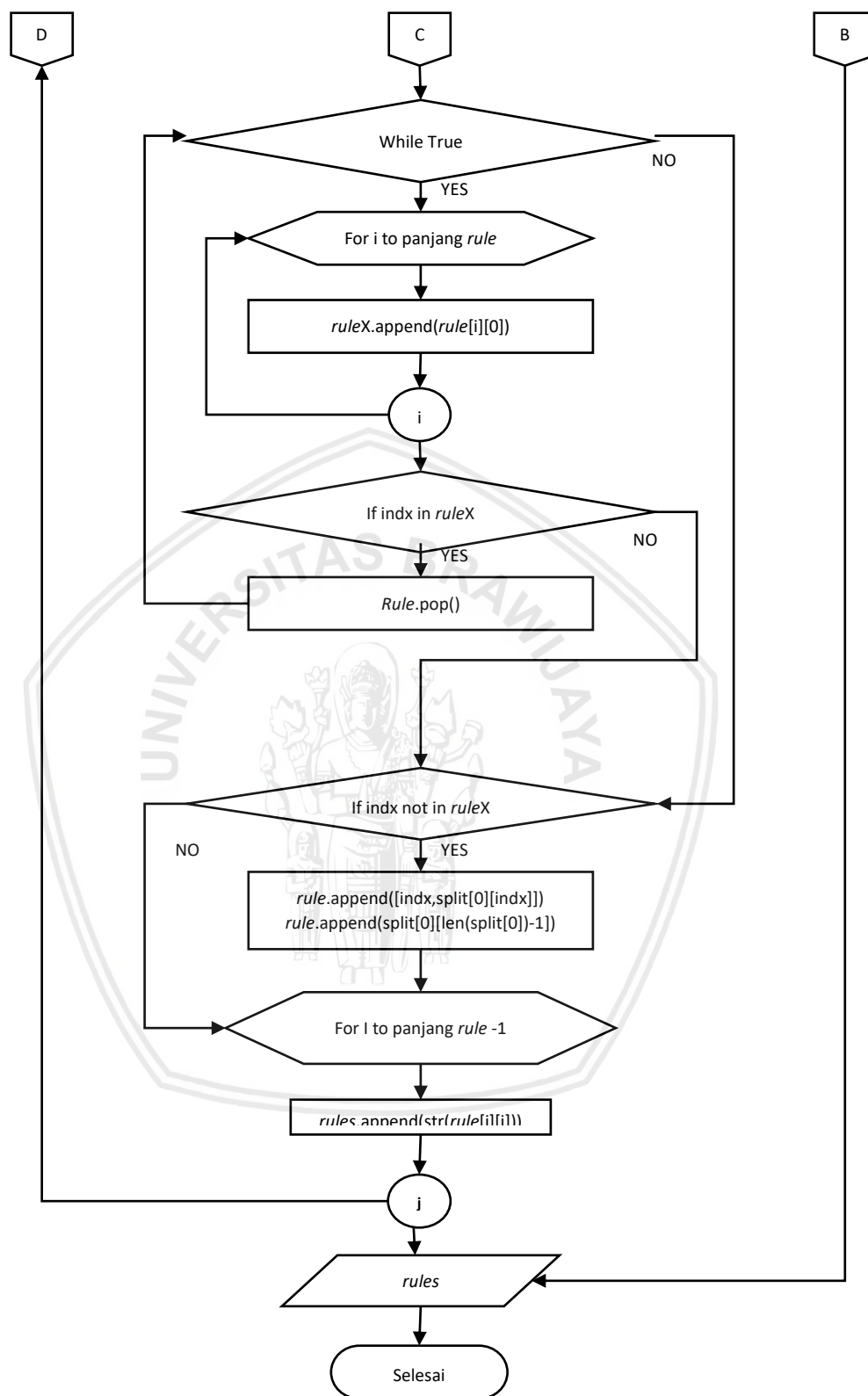
Langkah-langkah pada diagram alir proses perhitungan entropi pada Gambar 4.5 adalah:

1. Sistem dimulai
2. Inisialisasi *gain*
3. Data dan kelas sebagai masukan
4. Perulangan sepanjang kolom data

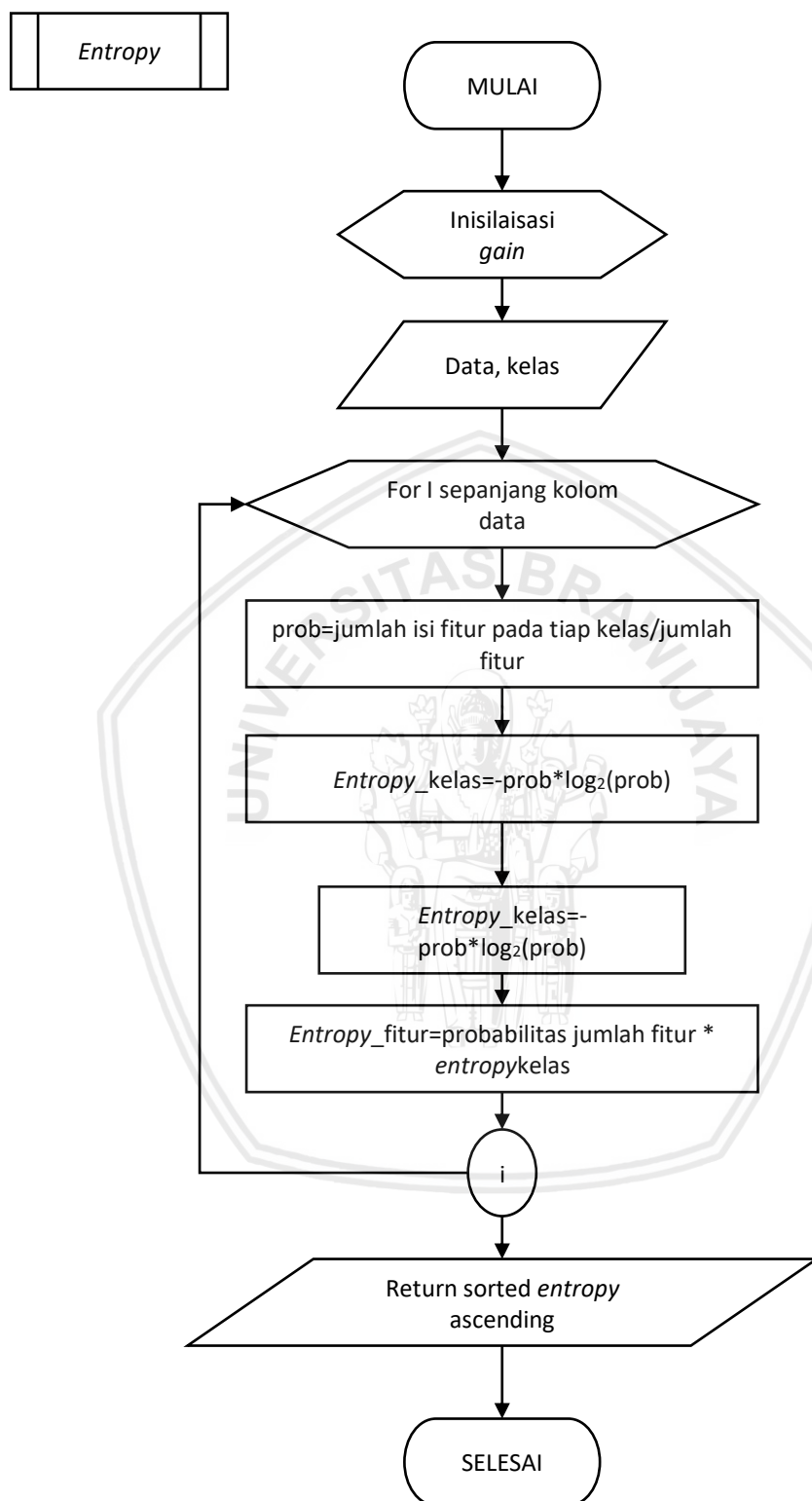
5. Melakukan perhitungan probabilitas fitur
6. Melakukan perhitungan entropi kelas dan entropi fitur
7. Sistem mengembalikan nilai *entropy* tertinggi berupa *indx*
8. Sistem selesai



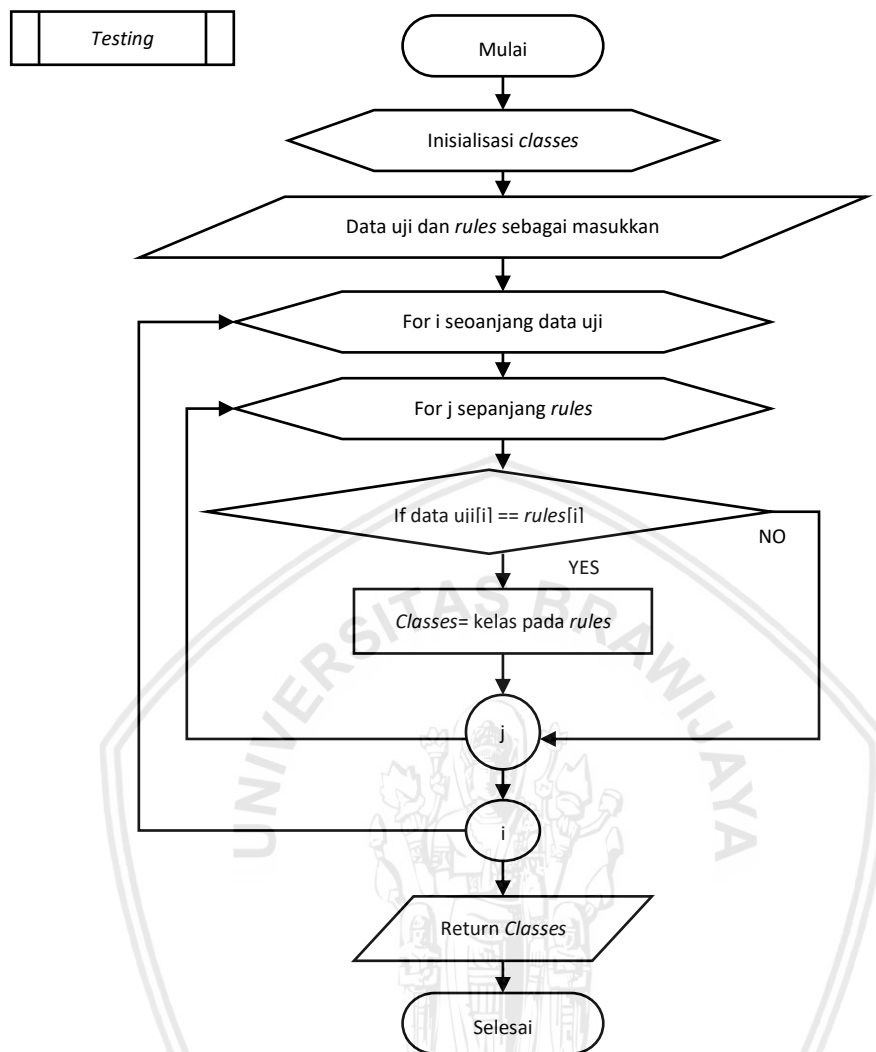




Gambar 4.4 Diagram Alir Proses Pembangunan Tree



Gambar 4.5 Diagram Alir Proses Perhitungan Entropi



Gambar 4.6 Diagram Alir Proses Testing

4.3.5 Proses Testing

Proses *testind* dilakukan pada data uji setelah dilakukan *training*. Proses ini bertujuan untuk menentukan kelas dari setiap data uji yang ada. Sistem ini mengeluarkan kelas kelas dari semua data uji yang ada. Proses *testing* ini digambarkan pada Gambar 4.6.

Langkah-langkah pada diagram alir proses *testing* pada Gambar 4.6 adalah:

1. Sistem dimulai
2. Inisialisasi *classes*
3. Data uji dan *rules* sebagai masukan
4. Perulangan sepanjang data uji
5. Perulangan sepanjang *rules*

6. Jika data uji sama dengan *rules*
7. Memasukkan kelas *rules* ke dalam *classes*
8. Mengembalikan *classes*
9. Sistem selesai

4.4 Perhitungan Manual

Perhitungan manual ditujukan untuk menggambarkan bagaimana klasifikasi yang dilakukan ini bekerja untuk mengolah sebuah data. Proses perhitungan manual melakukan perhitungan dengan sebagian data latih sebagai contoh. Data yang digunakan dalam perhitungan manual ini adalah 10 data latih (Tabel 4.2) dan 2 data uji (Tabel 4.3).

Tabel 4.2 Tabel Data Latih Manualisasi

Fitur						Mini Mental State Examination
Nama	Usia	Sex	Pendidikan	Stroke	...	
SUK	55	1	2	2	...	Normal
MLY	50	1	4	2	...	Normal
OBH	68	2	2	1	...	Gangguan
APY	47	1	2	2	...	Kemungkinan
HWN	73	1	2	2	...	Kemungkinan
ATM	49	2	2	1	...	Gangguan
TRI	45	2	1	2	...	Kemungkinan
NUR	53	1	4	2	...	Kemungkinan
ARS	39	1	4	2	...	Normal
GFR	53	1	4	2	...	Normal

Tabel 4.3 Tabel Data Uji Manualisasi

Fitur						Mini Mental State Examination
Nama	Usia	Sex	Pendidikan	Stroke	...	
JUM	86	2	2	1	...	Gangguan
ERF	43	1	1	2	...	Normal

4.4.1 Random Data

Langkah pertama yang dilakukan adalah mengacak data latih yang dimiliki untuk dibangun *tree* di langkah selanjutnya. Pada manualisasi kali ini data akan dirandom dua kali sehingga mendapatkan 3 data termasuk data asli.

Tabel 4.4 Data Latih Random Manualisasi 1

Fitur						Mini Mental State Examination
Nama	Usia	Sex	Pendidikan	Stroke	...	
ATM	49	2	2	1	...	Gangguan
NUR	53	1	4	2	...	Kemungkinan
ARS	39	1	4	2	...	Normal
HWN	73	1	2	2	...	Kemungkinan
APY	47	1	2	2	...	Kemungkinan
TRI	45	2	1	2	...	Kemungkinan
SUK	55	1	2	2	...	Normal
OBH	68	2	2	1	...	Gangguan
GFR	53	1	4	2	...	Normal
MLY	50	1	4	2	...	Normal

Tabel 4.5 Data Latih Random Manualisasi 2

Fitur						Mini Mental State Examination
Nama	Usia	Sex	Pendidikan	Stroke	...	
TRI	45	2	1	2	...	Kemungkinan
SUK	55	1	2	2	...	Normal
OBH	68	2	2	1	...	Gangguan
NUR	53	1	4	2	...	Kemungkinan
MLY	50	1	4	2	...	Normal
HWN	73	1	2	2	...	Kemungkinan
GFR	53	1	4	2	...	Normal
ATM	49	2	2	1	...	Gangguan
GFR	53	1	4	2	...	Normal
MLY	50	1	4	2	...	Normal

4.4.2 Membangun *Decision Tree*

Setelah data latih diacak random menjadi 3 tabel data set yang berbeda-beda, selanjutnya data tersebut dibangun *decision tree* masing-masing. Pada manualisasi ini akan digambarkan pembangunan 1 *tree* dari data latih asli. Langkah pertama pada pembangunan *tree* adalah mencari nilai entropi dari kelas yang ada pada data. Nilai entropi ini digunakan pada perhitungan *gain* di proses

selanjutnya. Adapun rumus entropi tersebut dijabarkan pada persamaan 2.1 dan 2.2.

4.4.2.1 Iterasi

Perhitungan entropi kelas ini dilakukan pada tiap iterasi. Hasil dari perhitungan entropi pada iterasi 1 tergambar pada Tabel 4.6.

Tabel 4.6 Entropi Kelas Iterasi 1

	Diagnosis		
	Normal	Kemungkinan	Gangguan
	4	4	2
	0,04816	0,048164799	0,012041
Entropy	-0,108370798		

Selanjutnya dilakukan perhitungan entropi dan probabilitas pada fitur usia. Perhitungan tersebut digambarkan pada Tabel 4.7.

Tabel 4.7 Entropi Fitur Usia Iterasi 1

	Usia				
	Normal	Kemungkinan	Gangguan		
	<60	4	3	1	8
	>=60	0	1	1	2
	<60	0,07526	0,042332343	0,004704	
	>=60	0	0,075257499	0,075257	
Entropy	<60	-0,122293436			
	>=60	-0,150514998			
Prob	<60	0,8			
	>=60	0,2			
E(Diagnosa,Usia)	-0,127937748				
Gain(Diagnosa, Usia)	0,01956695				

Selanjutnya dilakukan perhitungan entropi dan *gain* pada fitur sex. Perhitungan tersebut digambarkan pada Tabel 4.8.

Tabel 4.8 Entropi dan Gain Fitur Sex Iterasi 1

		Sex			
		Normal	Kemungkinan	Gangguan	
	1	4	3	0	7
	2	0	1	2	3
	1	0,0983	0,055291224	0	
	2	0	0,033447777	0,133791	
Entropy	1	-0,153586732			
	2	-0,167238886			
Prob	1	0,7			

	2	0,3
E(Diagnosa, Sex)	-0,157682379	
Gain(Diagnosa, Sex)	0,04931158	

Selanjutnya dilakukan perhitungan entropi dan *gain* pada fitur stroke. Perhitungan tersebut digambarkan pada Tabel 4.9.

Tabel 4.9 Entropi dan *Gain* Fitur Stroke Iterasi 1

		Stroke			
		Normal	Kemungkinan	Gangguan	
	1	0	0	2	2
	2	4	4	0	8
	1	0	0	0,30103	
	2	0,07526	0,075257499	0	
Entropy	1	-0,301029996			
	2	-0,150514998			
Prob	1	0,2			
	2	0,8			
E(Diagnosa, Stroke)		-0,180617997			
Gain(Diagnosa, Stroke)		0,072247199			

Selanjutnya dilakukan perhitungan entropi dan *gain* pada fitur hipertensi. Perhitungan tersebut digambarkan pada Tabel 4.10.

Tabel 4.10 Entropi dan *Gain* Fitur Hipertensi Iterasi 1

		Hipertensi			
		Normal	Kemungkinan	Gangguan	
	1	3	4	2	9
	2	1	0	0	1
	1	0,03345	0,059462715	0,014866	
	2	0,30103	0	0	
Entropy	1	-0,107776171			
	2	-0,301029996			
Prob	1	0,9			
	2	0,1			
E(Diagnosa, Hipertensi)	-0,127101554				
Gain(Diagnosa, Hipertensi)	0,018730755				

Selanjutnya dilakukan perhitungan entropi dan *gain* pada fitur hiperkolesterol. Perhitungan tersebut digambarkan pada Tabel 4.11.

Tabel 4.11 Entropi dan *Gain* Fitur Hiperkolestrol Iterasi 1

		Hiperkolestrol			
		Normal	Kemungkinan	Gangguan	
	1	0	1	0	1
	2	4	3	2	9
	1	0	0,301029996	0	
	2	0,05946	0,033447777	0,014866	
Entropy	1	-0,301029996			
	2	-0,107776171			
Prob	1	0,1			
	2	0,9			
E(Diagnosa,Hiperkolestrol)	-0,127101554				
Gain(Diagnosa, Hiperkolestrol)	0,018730755				

Selanjutnya dilakukan perhitungan entropi dan *gain* pada fitur diabetes melitus. Perhitungan tersebut digambarkan pada Tabel 4.12.

Tabel 4.12 Entropi dan *Gain* Fitur Diabetes Melitus Iterasi 1

		DM			
		Normal	Kemungkinan	Gangguan	
	1	1	0	0	1
	2	3	4	2	9
	1	0,30103	0	0	
	2	0,03345	0,059462715	0,014866	
Entropy	1	-0,301029996			
	2	-0,107776171			
Prob	1	0,1			
	2	0,9			
E(Diagnosa, DM)	-0,127101554				
Gain(Diagnosa, DM)	0,018730755				

Selanjutnya dilakukan perhitungan entropi dan *gain* pada fitur atrial fibrilasi. Perhitungan tersebut digambarkan pada Tabel 4.13.

Tabel 4.13 Entropi dan *Gain* Fitur Atrial Fibrilasi Iterasi 1

		AF			
		Normal	Kemungkinan	Gangguan	
	1	0	0	0	0
	2	4	4	2	10
	1	0	0	0	
	2	0,04816	0,048164799	0,012041	
	1	0			
Entropy	1	0			

	2	-0,108370798
Prob	1	0
	2	1
E(Diagnosa, AF)	-0,108370798	
Gain(Diagnosa, AF)	0	

Selanjutnya dilakukan perhitungan entropi dan *gain* pada fitur rokok. Perhitungan tersebut digambarkan pada Tabel 4.14.

Tabel 4.14 Entropi dan *Gain* Fitur Rokok Iterasi 1

		Rokok			
		Normal	Kemungkinan	Gangguan	
	1	0	1	0	1
	2	4	3	2	9
	1	0	0,301029996	0	
	2	0,05946	0,033447777	0,014866	
Entropy	1	-0,301029996			
	2	-0,107776171			
Prob	1	0,1			
	2	0,9			
E(Diagnosa,Rokok)		-0,127101554			
Gain(Diagnosa, Rokok)		0,018730755			

Selanjutnya dilakukan perhitungan entropi dan *gain* pada fitur alkohol. Perhitungan tersebut digambarkan pada Tabel 4.15.

Tabel 4.15 Entropi dan *Gain* Fitur Alkohol Iterasi 1

		Alkohol			
		Normal	Kemungkinan	Gangguan	
	1	0	0	1	1
	2	4	4	1	9
	1	0	0	0,30103	
	2	0,05946	0,059462715	0,003716	
Entropy	1	-0,301029996			
	2	-0,12264185			
Prob	1	0,1			
	2	0,9			
E(Diagnosa, Alkohol)	-0,140480665				
Gain(Diagnosa, Alkohol)	0,032109866				

Selanjutnya dilakukan perhitungan entropi dan *gain* pada fitur *hemisfer*. Perhitungan tersebut digambarkan pada Tabel 4.16.

Tabel 4.16 Entropi dan *Gain* Fitur *Hemisfer* Iterasi 1

		Hemisfer			
		Normal	Kemungkinan	Gangguan	
	1	2	2	2	6
	2	2	2	0	4
	1	0,03345	0,033447777	0,033448	
	2	0,07526	0,075257499	0	
Entropy	1	-0,100343332			
	2	-0,150514998			
Prob	1	0,6			
	2	0,4			
E(Diagnosa,Hemisfer)		-0,120411998			
Gain(Diagnosa, Hemisfer)		0,0120412			

Selanjutnya dilakukan perhitungan entropi dan *gain* pada fitur ct scan. Perhitungan tersebut digambarkan pada Tabel 4.17.

Tabel 4.17 Entropi dan *Gain* Fitur CT Scan Iterasi 1

		CT Scan			
		Normal	Kemungkinan	Gangguan	
	1	1	1	1	3
	2	3	3	1	7
	1	0,03345	0,033447777	0,033448	
	2	0,05529	0,055291224	0,006143	
Entropy	1	-0,100343332			
	2	-0,116725917			
Prob	1	0,3			
	2	0,7			
E(Diagnosa,CT Scan)	-0,111811141				
Gain(Diagnosa, CT Scan)	0,003440343				

Selanjutnya dilakukan perhitungan entropi dan *gain* pada fitur pendidikan. Perhitungan tersebut digambarkan pada Tabel 4.18.

Tabel 4.18 Entropi dan *Gain* Fitur Pendidikan Iterasi 1

		Pendidikan			
		Normal	Kemungkinan	Gangguan	
	1	0	1	0	1
	2	1	2	2	5
	3	0	0	0	0

	4	3	1	0	4
	1	0	0,301029996	0	
	2	0,01204	0,048164799	0,048165	
	3	0	0	0	
	4	0,16933	0,018814375	0	
Entropy	1	-0,301029996			
	2	-0,108370798			
	3	0			
	4	-0,188143747			
Prob	1	0,1			
	2	0,5			
	3	0			
	4	0,4			
E(Diagnosa,Pendidikan)	-0,159545898				
Gain(Diagnosa, Pendidikan)	0,051175099				

Selanjutnya dilakukan perhitungan entropi dan *gain* pada fitur BMI. Perhitungan tersebut digambarkan pada Tabel 4.19.

Tabel 4.19 Entropi dan *Gain* Fitur BMI Iterasi 1

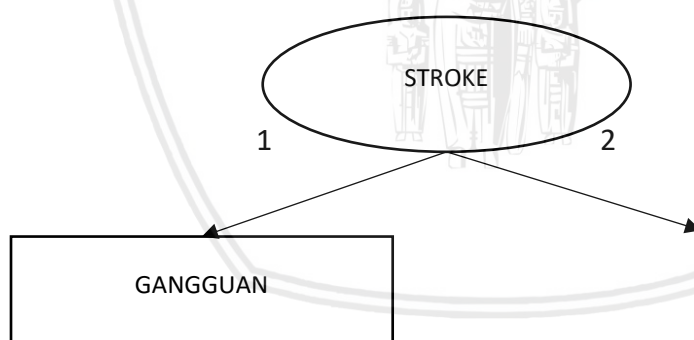
		BMI			
		Normal	Kemungkinan	Gangguan	
	Under	0	1	0	1
	Normal	4	2	1	7
	Over	0	1	1	2
	Obese	0	0	0	0
	Under	0	0,301029996	0	
	Normal	0,0983	0,024573877	0,006143	
	Over	0	0	0	
	Obese	0	0	0	
Entropy	Under	-0,301029996			
	Normal	-0,129012855			
	Over	0			
	Obese	0			
Prob	Under	0,1			
	Normal	0,7			
	Over	0,2			
	Obese	0			
E(Diagnosa,BMI)	-0,120411998				
Gain(Diagnosa, BMI)	0,0120412				

Setelah mendapatkan nilai *gain* dari setiap fitur, maka langkah selanjutnya adalah pengurutan nilai *gain* dari *gain* yang terbesar ke nilai *gain* yang terkecil. Pengurutan tersebut tergambar pada Tabel 4.20.

Tabel 4.20 Tabel Pengurutan *Gain* Iterasi 1

<i>Gain</i> (Diagnosa, Stroke)	0,0722
<i>Gain</i> (Diagnosa, Pendidikan)	0,0512
<i>Gain</i> (Diagnosa, Sex)	0,0493
<i>Gain</i> (Diagnosa, Alkohol)	0,0321
<i>Gain</i> (Diagnosa, Usia)	0,0196
<i>Gain</i> (Diagnosa, Hipertensi)	0,0187
<i>Gain</i> (Diagnosa, Hiperkolesterol)	0,0187
<i>Gain</i> (Diagnosa, DM)	0,0187
<i>Gain</i> (Diagnosa, Rokok)	0,0187
<i>Gain</i> (Diagnosa, Hemisfer)	0,012
<i>Gain</i> (Diagnosa, BMI)	0,012
<i>Gain</i> (Diagnosa, CT Scan)	0,0034
<i>Gain</i> (Diagnosa, AF)	0

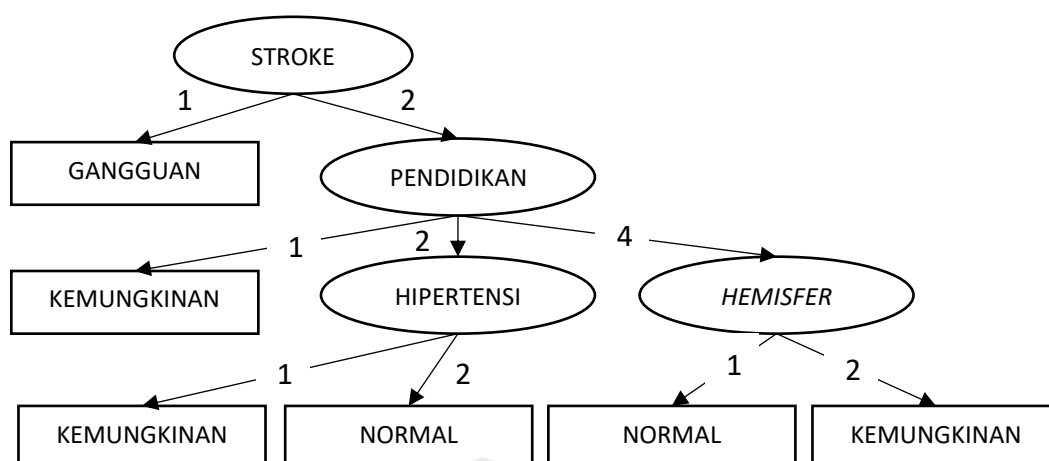
Hasil dari pengurutan *gain* mendapatkan nilai *gain* terbesar. *Gain* terbesar dari fitur tersebut yang menjadikan fitur tersebut sebagai fitur yang akan displit. Hasil *tree* yang didapatkan adalah seperti Gambar 4.7.



Gambar 4.7 Split Tree Iterasi 1

Proses yang telah dijalankan pada iterasi pertama di atas akan dilakukan secara berulang sehingga tidak ada lagi data yang bisa dicabang atau dengan kata lain, kelas pada tabel terakhir sudah sama, tidak ada perbedaan.

Setelah semua dilakukan akan terbentuk *tree* seperti pada Gambar 4.8. *Tree* tersebut yang akan menjadi patokan pada pengujian data uji di proses selanjutnya.



Gambar 4.8 Split Tree Iterasi 4

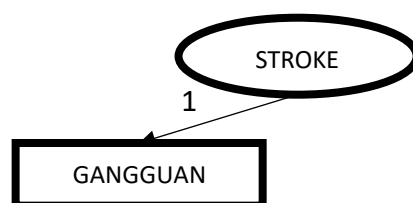
4.4.3 Voting

Setelah *tree* dari masing-masing data yang dirandom sejumlah (t) yang dimasukkan terbangun, langkah selanjutnya adalah *voting*. Data uji yang *training* dicari klasifikasinya dari *tree* yang telah terbangun di tahap sebelumnya. Dari semua hasil klasifikasi tersebut akan dilakukan *voting* seperti contoh di bawah ini.

Tabel 4.21 Contoh Data Uji Voting 1

Fitur						Mini Mental State Examination
Nama	Usia	Sex	Pendidikan	Stroke	...	
JUM	86	2	2	1	...	Gangguan

Data uji diatas dimasukkan ke dalam proses *training tree* yang telah dibangun pada contoh di langkah sebelumnya mendapatkan klasifikasi dengan kelas akhir adalah “Gangguan” yang di mana adalah seperti kelas target pada data uji seperti yang tergambar pada Gambar 4.9.



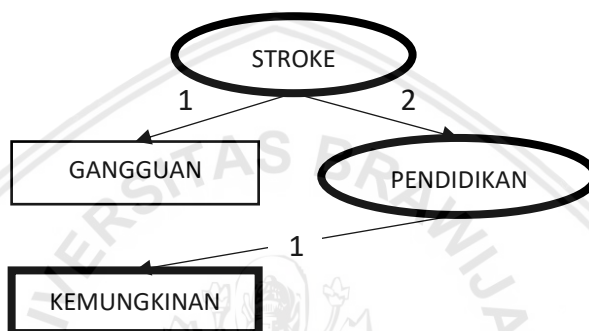
Gambar 4.9 Tree Voting Data Uji 1

Setelah mendapatkan hasil klasifikasi dari semua *tree*, yang dalam manualisasi ini dibuatkan 1 *tree*, maka dilakukan *voting*. Hasil *voting* yang didapatkan pada data uji tersebut adalah kelas “Gangguan” dari jumlah *tree* sebanyak 1.

Tabel 4.22 Contoh Data Uji Voting 2

Fitur						Mini Mental State Examination
Nama	Usia	Sex	Pendidikan	Stroke	...	
ERF	43	1	1	2	...	Normal

Data uji diatas dimasukkan ke dalam proses *training tree* yang telah dibangun pada contoh di langkah sebelumnya mendapatkan klasifikasi dengan kelas akhir adalah “Kemungkinan” yang di mana berbeda dengan kelas target pada data uji seperti yang tergambar pada Gambar 4.10.



Gambar 4.10 Tree Voting Data Uji 1

Setelah mendapatkan hasil klasifikasi dari semua *tree*, yang dalam manualisasi ini dibuatkan 1 *tree*, maka dilakukan *voting*. Hasil *voting* yang didapatkan pada data uji tersebut adalah kelas “Kemungkinan” dari jumlah *tree* sebanyak 1.

4.4.4 Pengujian Akurasi

Pengujian akurasi ini dilakukan untuk menentukan nilai ketepatan dari sistem yang telah dibangun menggunakan data latih dan data uji yang telah ditentukan. Pada manualisasi ini terdapat 2 data uji yang sudah lebih dahulu di *training*. Hasil *training* pada manualisasi ini tergambar pada Tabel 4.23.

Tabel 4.23 Tabel Akurasi Manualisasi

Data Uji	Hasil Sistem	Target
1	Gangguan	Gangguan
2	Kemungkinan	Normal

Perhitungan akurasi ini didapatkan dari jumlah klasifikasi yang sesuai dengan target dibagi dengan jumlah keseluruhan data uji. Rumus akurasi digambarkan pada persamaan 2.3.

$$\begin{aligned} \text{Akurasi} &= \frac{1}{2} \times 100\% \\ &= 50\% \end{aligned}$$

4.5 Perancangan Pengujian

Perancangan pengujian membahas tentang rencana pengujian dari klasifikasi yang dilakukan. Pengujian sendiri ditujukan untuk menentukan tingkat validasi dari sistem yang telah dibuat. Nilai validasi tersebut yang dapat menyimpulkan berapa tingkat keberhasilan dari sistem yang telah dibuat.

4.5.1 Pengujian *K-fold cross validation* Seleksi Fitur

Pengujian seleksi fitur adalah pengujian untuk menentukan jumlah fitur yang digunakan dalam proses klasifikasi. Seleksi fitur dilakukan dengan menggunakan metode *information gain* dengan menggunakan rumus yang sama dengan proses pemisahan pada pembangunan *tree*. Rumus yang dimaksud telah dijabarkan pada Persamaan 2.1 dan Persamaan 2.2.

Pengujian seleksi fitur ini dilakukan dengan metode *K-fold cross validation*. Metode pengujian ini adalah salah satu metode yang dilakukan untuk mengevaluasi kinerja dari kombinasi data latih dan data uji. Seluruh *dataset* yang dimiliki dibagi menjadi lima *fold*, yaitu F1, F2, F3, F4, dan F5. Setiap *fold* tersebut berisikan 19 data. Dari 5 subset data tersebut dibuat menjadi lima kombinasi yang terdiri dari empat *folds* data latih dan satu *fold* data uji.

Masing-masing dari kombinasi yang terbentuk, dilakukan pengujian jumlah fitur dari mulai 8 fitur sampai dengan 13. Pengujian yang telah dirancang tersebut dilakukan sebanyak lima kali dikarenakan data yang dilatih terlebih dahulu diacak secara *random* pada proses *training*. Dari pengujian tersebut didapatkan jumlah fitur berdasarkan hasil nilai akurasi yang didapatkan. Nilai akurasi tersebut adalah rata-rata dari setiap eksekusi pengujian. Jumlah *tree* yang digunakan pada pengujian seleksi fitur menggunakan *k-fold cross validation* ini adalah 50 *tree*.

4.5.2 Pengujian *K-fold cross validation* Jumlah *tree*

Pengujian yang dilakukan pada penelitian kali ini adalah pengujian jumlah *tree* yang dibangun pada metode *random forest*. Pengujian dilakukan dengan metode *K-fold cross validation*. Metode pengujian ini adalah salah satu metode yang dilakukan untuk mengevaluasi kinerja dari kombinasi data latih dan data uji. Seluruh *dataset* yang dimiliki dibagi menjadi lima *fold*, yaitu F1, F2, F3, F4, dan F5. Setiap *fold* tersebut berisikan 19 data. Dari 5 subset data tersebut dibuat menjadi lima kombinasi yang terdiri dari empat *fold* data latih dan satu *fold* data uji.

Masing-masing dari kombinasi yang terbentuk, dilakukan pengujian jumlah *tree* dari mulai 50 *trees*, 100 *trees*, 150 *trees*, 200 *trees*, dan 250 *trees*. Pengujian yang telah dirancang tersebut dilakukan sebanyak lima kali dikarenakan data yang dilatih terlebih dahulu diacak secara *random* pada proses *training*.

BAB 5 IMPLEMENTASI

Bab 5 menjelaskan pembahasan tentang implementasi dasar teori yang telah dibahas dalam bab sebelumnya menjadi sebuah sistem jadi yaitu KLASIFIKASI PENURUNAN FUNGSI KOGNITIF PASIEN STROKE MENGGUNAKAN METODE KLASIFIKASI RANDOM FOREST.

5.1 Kode Program

Dari perancangan algoritme dan perhitungan manual pada tahap sebelumnya, maka penulis mengimplementasikan metode *random forest* pada data latih yang telah dimiliki dengan bahasa pemrograman Python. Berikut adalah kode program yang telah diimplementasikan.

5.1.1 Method main

Berikut adalah kode program dari *method* main yang dijabarkan pada Tabel 5.1.

Tabel 5.1 Kode Program Method Main

1	def main():
2	data_latih = baca_data('DataStroke.csv')
3	data_uji = convert_data(baca_data('DataUji.csv'))
4	jumlahtree=int(input("Jumlah Tree: "))
5	kelas_sistem=(multi_tree(data_latih,jumlahtree,data_uji)
6	)
7	kelas_data_uji=[]
8	for i in range(len(data_uji)):
9	kelas_data_uji.append(data_uji[i][14])
10	hasil_akurasi=akurasi(kelas_data_uji,kelas_sistem)
11	print("\n")
12	print(kelas_data_uji,"\n")
13	print(kelas_sistem,"\n")
14	print(hasil_akurasi)

Berikut adalah penjelasan dari kode program di atas yang dijabarkan pada Tabel 5.2.

Tabel 5.2 Tabel Penjelasan Kode Program Mehtod Main

1	Inisialisasi <i>method</i> main
2-3	Inisialisasi data latih dan data uji
4	Inputan bertipe integer disimpan pada variable jumlah <i>tree</i>
5	Melakukan <i>method</i> <i>multi_tree</i> dan disimpan di variable kelas_sistem
7-9	Mengambil kelas asli dari setiap data uji disimpan di kelas_data_uji
10	Mencari nilai akurasi dengan <i>method</i> akurasi disimpan di hasil_akurasi
11-14	Mencetak keluaran kelas_data_uji, kelas_sistem, dan hasil_akurasi

5.1.2 Method *multi_tree*

Berikut adalah kode program dari *method multi_tree* yang dijabarkan pada Tabel 5.3.

Tabel 5.3 Kode Program *Method multi_tree*

1	<code>def multi_tree(data_latih, jumlahtree, data_uji):</code>
2	<code> classes=[]</code>
3	<code> for i in range(jumlahtree):</code>
4	<code> rules=train(copy.deepcopy(data_latih))</code>
5	<code> kelas=test(copy.deepcopy(data_uji), copy.deepcopy(rules))</code>
6	<code> classes.append(kelas)</code>
7	<code> transposed_classes=[[classes[j][i] for j in</code>
8	<code>range(len(classes))] for i in range(len(classes[0]))]</code>
9	<code> voted_classes=[]</code>
10	<code> for isi in transposed_classes:</code>
11	<code> voted_classes.append(voting(isi))</code>
12	<code> return(voted_classes)</code>
13	

Berikut adalah penjelasan dari kode program di atas yang dijabarkan pada Tabel 5.4.

Tabel 5.4 Penjelasan Kode Program *Method multi_tree*

1	Inisialisasi <i>method multi_tree</i> dengan parameter <i>data_latih</i> , <i>jumlahtree</i> , dan <i>data_uji</i>
2	Inisialisasi <i>classes</i>
3	Perulangan sebanyak jumlah <i>tree</i>
4	Melakukan <i>training</i> disimpan di <i>rules</i>
5-6	Melakukan <i>testing</i> disimpan di kelas
7	Menambahkan setiap kelas ke dalam <i>classes</i>
8-10	Melakukan transpose pada <i>classes</i>
11-12	Setiap isi pada <i>transposed_classes</i> dilakukan <i>voting</i>
13	Mengembalikan nilai hasil <i>voting</i>

5.1.3 Method *train*

Berikut adalah kode program dari *method train* yang dijabarkan pada Tabel 5.5.

Tabel 5.5 Kode Program *Method train*

1	<code>def train(dataLatih):</code>
2	<code> try:</code>
3	<code> os.remove("rules.csv")</code>
4	<code> except:</code>
5	<code> print("file tidak ada")</code>
6	<code> rules=[]</code>
7	<code> temp_rule=[]</code>
8	<code> dataRandom=random_data(convert_data(dataLatih))</code>
9	<code> splitting(copy.deepcopy(dataRandom), temp_rule, [])</code>

10	<code>rules=baca_data("rules.csv")</code>
11	<code>return rules</code>

Berikut adalah penjelasan dari kode program di atas yang dijabarkan pada Tabel 5.6.

Tabel 5.6 Penjelasan Kode Program *Method Train*

1	Inisialisasi <i>method train</i> dengan parameter data latih
2-5	Try catch untuk erorr apabila tidak teradpat <i>rules.csv</i> pada <i>folder</i>
6-7	Inisialisasi <i>rules</i> dan tempr ule
8	Melakukan random data dengan <i>method</i> random data
9	Melakukan pembangunan <i>tree</i> dengan <i>method splitting</i>
10	Memasukkan <i>rules</i> ke dalam <i>rules.csv</i>
11	Mengembalikan variable <i>rules</i>

5.1.4 *Method random_data*

Berikut adalah kode program dari *method random_data* yang dijabarkan pada Tabel 5.7.

Tabel 5.7 Kode Program *Method random_data*

1	<code>def random_data(dataLatih):</code>
2	<code> indx = [random.randrange(0,len(dataLatih)) for i in</code>
3	<code> range(0,len(dataLatih))]</code>
4	<code> x=[]</code>
5	<code> for i in indx:</code>
6	<code> x.append(dataLatih[i])</code>
7	<code> return x</code>

Berikut adalah penjelasan dari kode program di atas yang dijabarkan pada Tabel 5.8.

Tabel 5.8 Penjelasan Kode Proram *Method random_data*

1	Inisialisasi <i>method random_data</i> dengan parameter data latih
2-3	Memasukkan data secara random sepanjang datalatih
4-6	Setiap isi dalam indx dimasukkan ke dalam x
7	Mengembalikan nilai x

5.1.5 *Method splitting*

Berikut adalah kode program dari *method splitting* yang dijabarkan pada Tabel 5.9.

Tabel 5.9 Kode Program *Method Splitting*

1	<code>def splitting(data,temp_rule,rule):</code>
2	<code> kelas=baca_kelas_unik(data)</code>
3	<code> indx=int(entropy(data,kelas))</code>
4	<code> isi_fitur=sorted(baca_kategori_fitur(data,indx))</code>

```

5 split_data=[]
6 if True:
7     for isi in isi_fitur:
8         split_data.append([])
9     for i in range(len(split_data)):
10        dalem=[]
11        for row in data:
12            if row[indx]==isi_fitur[i]:
13                dalem.append(row)
14        split_data[i]=(dalem)
15 for split in split_data:
16     if len(baca_kelas_unik(split))>1:
17         while True:
18             ruleX=[]
19             for i in range(len(rule)):
20                 ruleX.append(rule[i][0])
21             if indx in ruleX:
22                 rule.pop()
23             else:
24                 break
25             if indx not in rule:
26                 rule.append([indx,split[0][indx]])
27             splitting(split,temp_rule,rule)
28         else:
29             while True:
30                 ruleX=[]
31                 for i in range(len(rule)):
32                     ruleX.append(rule[i][0])
33                 if indx in ruleX:
34                     rule.pop()
35                 else:
36                     break
37                 if indx not in rule:
38                     rule.append([indx,split[0][indx]])
39                     rule.append(split[0][len(split[0])-1])
40             rules=[]
41             for i in range(len(rule)-1):
42                 for j in range(len(rule[i])):
43                     rules.append(str(rule[i][j]))
44             rules.append(rule[len(rule)-1])
45             tulis_data("rules.csv",rules)

```

Berikut adalah penjelasan dari kode program di atas yang dijabarkan pada Tabel 5.10.

Tabel 5.10 Penjelasan Kode Program *Method Splitting*

1	Inisialisasi <i>method splitting</i> dengan parameter data, temp_rule, dan rule
2	Membaca kelas pada data disimpan di kelas
3	Mencari index dengan <i>method</i> entropi
4	Membaca isi fitur disimpan di isi_fitur
5	Inisialisasi list split_data
6-8	Menambahkan list kosong pada split_data sejumlah isi dalam isi_fitur
9-13	Memasukkan data berdasarkan isi dalam isi_fitur ke dalam split_data

15	Perulangan setiap isi pada Split_data
16	Jika kelas pada isi Split_data lebih dari 1
17-26	Mencatat indx dan isi dalam isi_fitur ke dalam <i>rule</i>
27	Melakukan rekursif <i>method splitting</i>
28	Jika kelas pada isi Split_data sudah tinggal 1
29-45	Mencatat indx, isi dalam isi_fitur, dan kelas tersebut pada <i>rule</i>
46	Memasukkan setiap <i>rule</i> ke dalam <i>rules</i>
47	Membuat file <i>rules.csv</i>

5.1.6 Method baca_kelas_unik

Berikut adalah kode program dari *method* *baca_kelas_unik* yang dijabarkan pada Tabel 5.11.

Tabel 5.11 Kode Program Method *baca_kelas_unik*

1	def <i>baca_kelas_unik</i> (data):
2	x=[]
3	for i in range(len(data)):
4	kelas=data[i][len(data[i])-1]
5	if kelas not in x:
6	x.append(kelas)
7	return x

Berikut adalah penjelasan dari kode program di atas yang dijabarkan pada Tabel 5.12.

Tabel 5.12 Penjelasan Kode Program Method *baca_kelas_unik*

1	Inisialisasi <i>baca_kelas_unik</i> dengan parameter data
2	Inisialisasi x list kosong
3	Perulangan i sepanjang data
4	Memasukan kelas data ke dalam kelas
5	Jika kelas tidak ada di x
6	Kelas ditambahkan ke dalam x
7	Mengembalikan nilai x

5.1.7 Method entropy

Berikut adalah kode program dari *method entropy* yang dijabarkan pada Tabel 5.13.

Tabel 5.13 Kode Program Method *entropy*

1	def <i>entropy</i> (data, kelas):
2	gain=[]
3	for fitur in range(len(data[0])-1):
4	isi_fitur= <i>baca_kategori_fitur</i> (data, fitur)
5	<i>entropy_fitur</i> =0
6	for isi in isi_fitur:
7	n_fitur=0
8	n_isi_fitur=0
9	for baris in range(len(data)):

```

10         if data[baris][fitur]==isi:
11             n_isi_fitur+=1
12         entropy_kelas=0
13         n_fitur+=n_isi_fitur
14         for kls in kelas:
15             n_isi_kelas=0
16             for baris in range(len(data)):
17                 if data[baris][fitur]==isi and
18 data[baris][len(data[baris])-1]==kls:
19                     n_isi_kelas+=1
20                     prob=n_isi_kelas/n_isi_fitur
21                     entropy_kelas-= prob*math.log(prob,2) if
22 prob>0 else 0
23             entropy_fitur+=(n_fitur/len(data))*entropy_kelas
24         gain.append([entropy_fitur,fitur])
25     return sorted(gain)[0][1]

```

Berikut adalah penjelasan dari kode program di atas yang dijabarkan pada Tabel 5.14

Tabel 5.14 Penjelasan Kode Program *Method Entropy*

1	Inisialisasi <i>method</i> entropi dengan parameter data dan kelas
2	Inisialisasi <i>gain</i>
3	Perulangan setiap fitur
4-13	Menambah 1 untuk setiap data yang memiliki fitur di setiap isi fitur
14	Perulangan setiap kelas
17-19	Menambah 1 untuk setiap data yang memiliki kelas di setiap kelas
20	Menghitung probabilitas fiutr
21	Menghitung <i>entropy</i> kelas
22	Menghitung entropi fitur
23-24	Mengurutkan <i>entropy</i> fitur terkecil untuk dikembalikan

5.1.8 *Method baca_kategori_fitur*

Berikut adalah kode program dari *method* *baca_kategori_unik* yang dijabarkan pada Tabel 5.15.

Tabel 5.15 Kode Program *Method* *baca_kategori_fitur*

```

1 def baca_kategori_fitur(data,n):
2     x=[]
3     for i in range(len(data)):
4         if data[i][n] not in x:
5             x.append(data[i][n])
6     return x

```

Berikut adalah penjelasan dari kode program di atas yang dijabarkan pada Tabel 5.16.

Tabel 5.16 Penjelasan Kode Program *Method* baca_kategori_fitur

1	Inisialisasi <i>method</i> main
2-3	Inisialisasi data latih dan data uji
4	Inputan bertipe integer disimpan pada variable jumlah <i>tree</i>
5	Melakukan <i>method</i> <i>multi_tree</i> dan disimpan di variable kelas_sistem
7-9	Mengambil kelas asli dari setiap data uji disimpan di kelas_data_uji
10	Mencari nilai akurasi dengan <i>method</i> akurasi disimpan di hasil_akurasi
11-14	Mencetak keluaran kelas_data_uji, kelas_sistem, dan hasil_akurasi

5.1.9 Method test

Berikut adalah kode program dari *method test* yang dijabarkan pada Tabel 5.17

Tabel 5.17 Kode Program *Method test*

1	def test(dataUji, rules):
2	converted_data_uji=[]
3	converted_rules=[]
4	classes=[]
5	for i in range(len(dataUji)):
6	dalem=[]
7	for j in range(len(dataUji[i])-1):
8	dalem.append(int(dataUji[i][j]))
9	converted_data_uji.append(dalem)
10	for i in range(len(rules)):
11	dalem=[]
12	for j in range(len(rules[i])-1):
13	dalem.append(int(rules[i][j]))
14	dalem.append(rules[i][len(rules[i])-1])
15	converted_rules.append(dalem)
16	for data in converted_data_uji:
17	kelas=""
18	test_rule=copy.deepcopy(converted_rules)
19	try:
20	while type(test_rule[0][0]) != str:
21	test_rule_first_value=[]
22	for rule in test_rule:
23	if rule[1] not in test_rule_first_value:
24	test_rule_first_value.append(rule[1])
25	if data[test_rule[0][0]] not in
26	test_rule_first_value:
27	kelas="MissingValue"
28	else:
29	new_rules=[]
30	for rule_test_rule in test_rule:
31	if
32	rule_test_rule[1]==data[rule_test_rule[0]]:
33	new_rules.append(rule_test_rule)
34	new_rules_2=[]
35	for isi_new_rule in new_rules:
36	isi_new_rule.pop(0)
37	isi_new_rule.pop(0)
38	new_rules_2.append(isi_new_rule)
39	test_rule=new_rules_2

40	kelas=test_rule[0][0]
41	classes.append(kelas)
42	test_rule=copy.deepcopy(converted_rules)
45	except:
46	classes.append('X')
47	return classes

Berikut adalah penjelasan dari kode program di atas yang dijabarkan pada Tabel 5.18.

Tabel 5.18 Penjelasan Kode Program *Method test*

1	Inisialisasi <i>method test</i> dengan parameter
2-4	Inisialisasi <i>classes</i> , <i>converted_data_uji</i> dan <i>converted_rules</i>
5-9	Mengkonversi data uji disimpan dalam <i>converted_data_uji</i>
10-15	Mengkonversi <i>rules</i> ke disimpan ke dalam <i>converted_rules</i>
16	Perulangan sepanjang <i>converted_data_uji</i>
17-42	Jika setiap <i>converted_data_uji</i> telah menemukan kelas dari proses seleksi aturan pada <i>rules</i> , maka kelas tersebut dimasukkan ke dalam variabel kelas
45	Memasukkan setiap variabel kelas ke dalam <i>classes</i>
47	Mengembalikan nilai <i>classes</i>

5.1.10 Method voting

Berikut adalah kode program dari *method voting* yang dijabarkan pada Tabel 5.19.

Tabel 5.19 Kode Program *Method voting*

1	def <i>voting</i> (inilist):
2	word={}
3	for i in inilist:
4	if i not in word:
5	word[i]=1
6	else:
7	word[i]+=1
8	return max(word.items(),key=operator.itemgetter(1))[0]

Berikut adalah penjelasan dari kode program di atas yang dijabarkan pada Tabel 5.20.

Tabel 5.20 Penjelasan Kode Program *Method voting*

1	Inisialisasi <i>method voting</i>
2	Inisialisasi dictionary word
3	Perulangan i sebanyak inilist
4	Jika i tidak ada pada word

5	Word[i] = 1
6	Selain itu
7	Word[i] ditambah 1
8	Mengembalikan hasil <i>voting</i>

5.1.11 Method akurasi

Berikut adalah kode program dari *method* akurasi yang dijabarkan pada Tabel 5.21.

Tabel 5.21 Kode Program Method akurasi

1	def akurasi(asli,sistem):
2	benar=0
3	for i in range(len(asli)):
4	if asli[i]==sistem[0]:
5	benar+=1
6	return (benar/len(sistem))*100

Berikut adalah penjelasan dari kode program di atas yang dijabarkan pada Tabel 5.22.

Tabel 5.22 Penjelasan Kode Program Method akurasi

1	Inisialisasi <i>method</i> akurasi
2	Inisialisasi benar =0
3	Perulangan sepanjang variable asli
4	Jika asli sama dengan sistem
5	Benar ditambah 1
6	Mengembalikan nilai benar/panjang sistem *100

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab pengujian dan analisis membahas hasil dari percobaan setiap skenario pengujian yang telah dijabarkan pada perancangan pengujian. Pengujian pada klasifikasi ini dilakukan dengan dua pengujian, pengujian seleksi fitur dan pengujian jumlah *tree*. Setiap pengujian tersebut dilakukan dengan menggunakan metode pengujian *k-fold cross validation*. Hal ini ditujukan untuk mengetahui tingkat keberhasilan dari klasifikasi yang dilakukan.

6.1 Pengujian *K-Fold Cross Validation* Jumlah Fitur

Pengujian seleksi fitur adalah pengujian untuk menentukan jumlah fitur yang digunakan dalam proses klasifikasi. Seleksi fitur dilakukan dengan menggunakan metode *information gain* dengan menggunakan rumus yang sama dengan proses pemisahan pada pembangunan *tree*. Rumus yang dimaksud telah dijabarkan pada Persamaan 2.1 dan Persamaan 2.2.

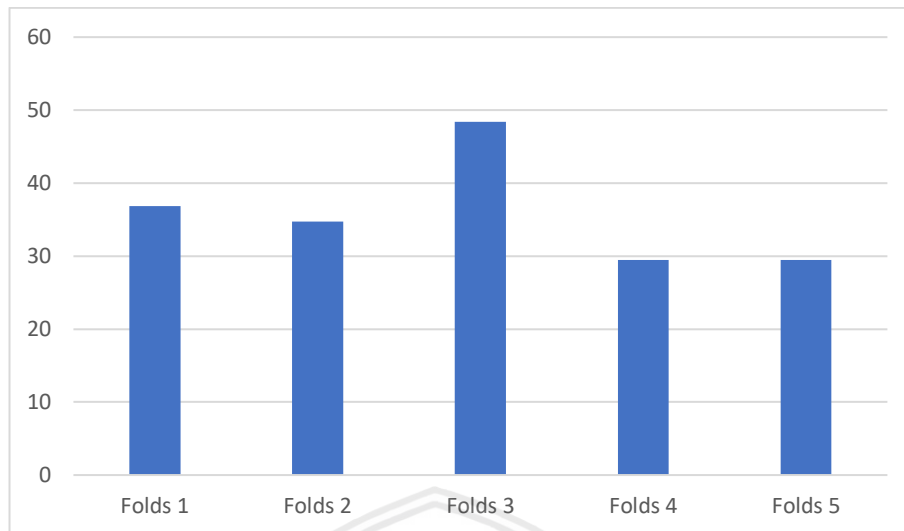
Pengujian seleksi fitur ini dilakukan dengan metode *K-fold cross validation*. Metode pengujian ini adalah salah satu metode yang dilakukan untuk mengevaluasi kinerja dari kombinasi data latih dan data uji. Seluruh *dataset* yang dimiliki dibagi menjadi lima *fold*, yaitu F1, F2, F3, F4, dan F5. Setiap *fold* tersebut berisikan 19 data. Dari 5 subset data tersebut dibuat menjadi lima kombinasi yang terdiri dari empat *folds* data latih dan satu *fold* data uji.

Masing-masing dari kombinasi yang terbentuk, dilakukan pengujian jumlah fitur dari mulai 8 fitur sampai dengan 13. Pengujian yang telah dirancang tersebut dilakukan sebanyak lima kali dikarenakan data yang dilatih terlebih dahulu diacak secara *random* pada proses *training*. Dari pengujian tersebut didapatkan jumlah fitur berdasarkan hasil nilai akurasi yang didapatkan. Nilai akurasi tersebut adalah rata-rata dari setiap eksekusi pengujian.

Jika pada setiap percobaan tersebut menghasilkan nilai akurasi nol atau *error*, maka dilakukan kembali hingga maksimal 10 kali. Jika setelah 10 kali dijalankan tetap menghasilkan nilai nol, maka nilai akurasi pada percobaan tersebut adalah nol. Jumlah *tree* yang digunakan pada pengujian seleksi fitur menggunakan *k-fold cross validation* ini adalah 50 *tree*.

6.1.1 Hasil Pengujian 13 Fitur

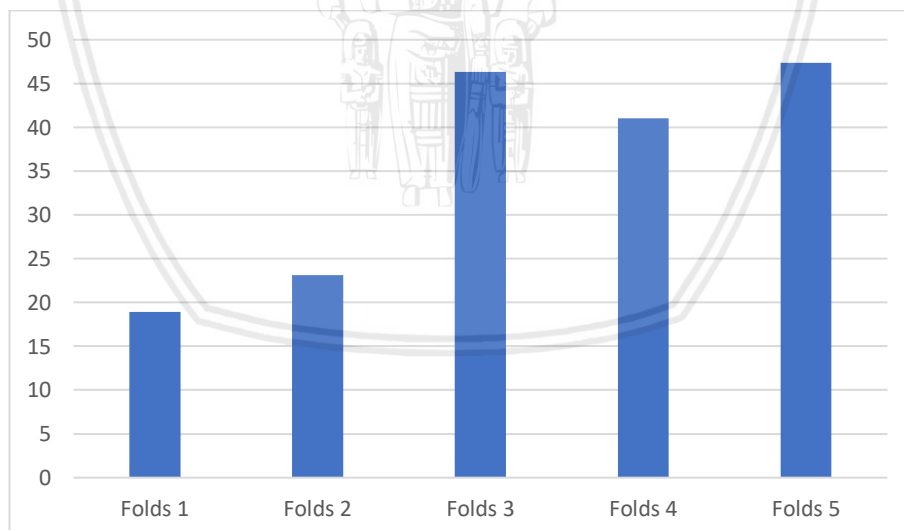
Berikut adalah hasil pengujian seleksi fitur dengan menggunakan jumlah fitur adalah 13 fitur yang digambarkan pada Gambar 6.1. Berdasarkan hasil pengujian dengan menggunakan 13 fitur pada lima kombinasi *folds*, didapatkan akurasi rata-rata adalah 35,78%. Kombinasi terbaik terdapat pada kombinasi 3 dengan F3 sebagai data uji dan *fold* lainnya sebagai data latih. Pada gambar, sumbu x adalah *folds* yang digunakan dan sumbu y adalah tingkat akurasi yang didapatkan.



Gambar 6.1 Hasil Pengujian 13 Fitur

6.1.2 Hasil Pengujian 12 Fitur

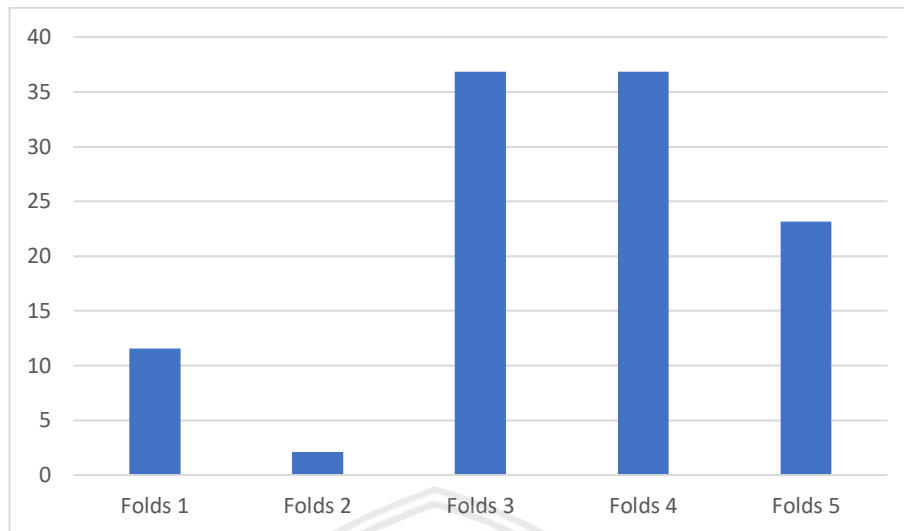
Berikut adalah hasil pengujian seleksi fitur dengan menggunakan jumlah fitur adalah 12 fitur yang digambarkan pada Gambar 6.2. Berdasarkan hasil pengujian dengan menggunakan 12 fitur pada lima kombinasi *folds*, didapatkan akurasi rata-rata adalah 35,36%. Kombinasi terbaik terdapat pada kombinasi 5 dengan F5 sebagai data uji dan *fold* lainnya sebagai data latih. Pada gambar, sumbu x adalah folds yang digunakan dan sumbu y adalah tingkat akurasi yang didapatkan.



Gambar 6.2 Hasil Pengujian 12 Fitur

6.1.3 Hasil Pengujian 11 Fitur

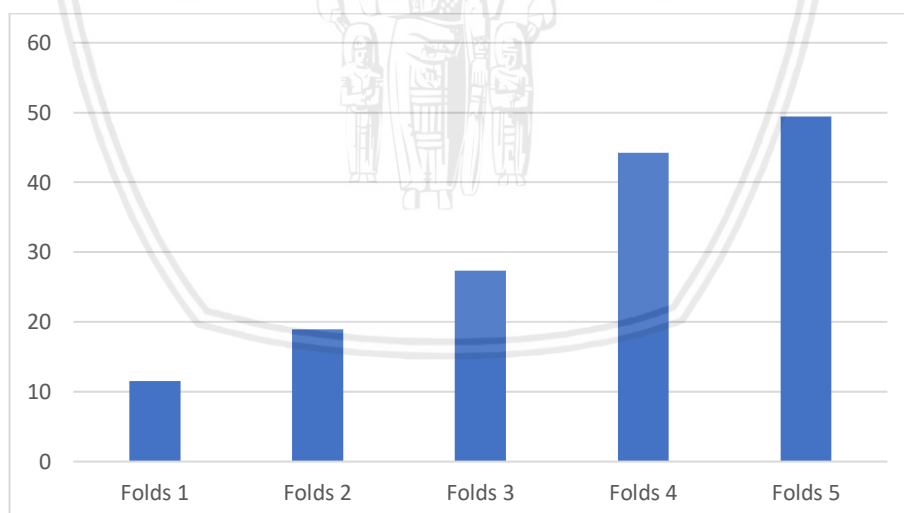
Berikut adalah hasil pengujian seleksi fitur dengan menggunakan jumlah fitur adalah 11 fitur yang digambarkan pada Gambar 6.3. Berdasarkan hasil pengujian dengan menggunakan 11 fitur pada lima kombinasi *folds*, didapatkan akurasi rata-rata adalah 22,1%. Kombinasi terbaik terdapat pada kombinasi 3 dengan F3 sebagai data uji dan *fold* lainnya sebagai data latih.



Gambar 6.3 Hasil Pengujian 11 Fitur

6.1.4 Hasil Pengujian 10 Fitur

Berikut adalah hasil pengujian seleksi fitur dengan menggunakan jumlah fitur adalah 10 fitur yang digambarkan pada Gambar 6.4. Berdasarkan hasil pengujian dengan menggunakan 10 fitur pada lima kombinasi *folds*, didapatkan akurasi rata-rata adalah 30,31%. Kombinasi terbaik terdapat pada kombinasi 5 dengan F5 sebagai data uji dan *fold* lainnya sebagai data latih. Pada gambar, sumbu x adalah folds yang digunakan dan sumbu y adalah tingkat akurasi yang didapatkan.

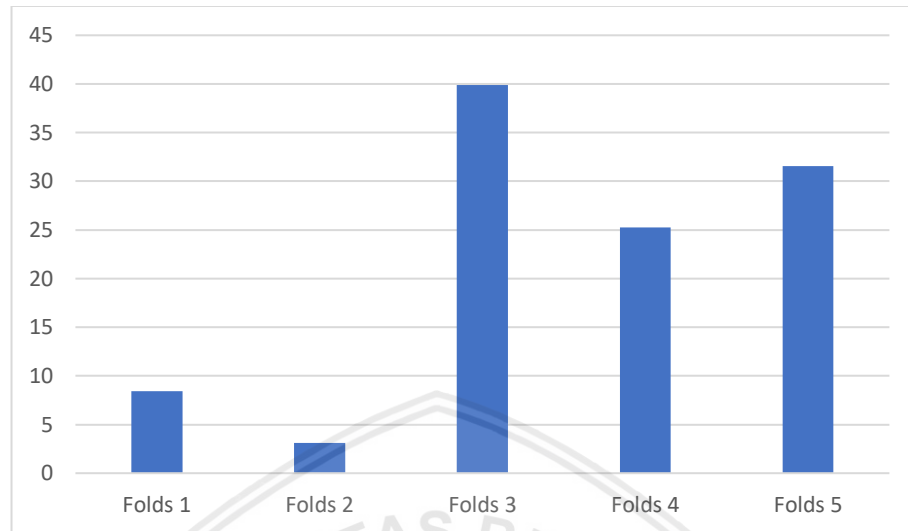


Gambar 6.4 Hasil Pengujian 10 Fitur

6.1.5 Hasil Pengujian 9 Fitur

Berikut adalah hasil pengujian seleksi fitur dengan menggunakan jumlah fitur adalah 9 fitur yang digambarkan pada Gambar 6.5. Berdasarkan hasil pengujian dengan menggunakan 9 fitur pada lima kombinasi *folds*, didapatkan akurasi rata-rata adalah 21,68%. Kombinasi terbaik terdapat pada kombinasi 3 dengan F3

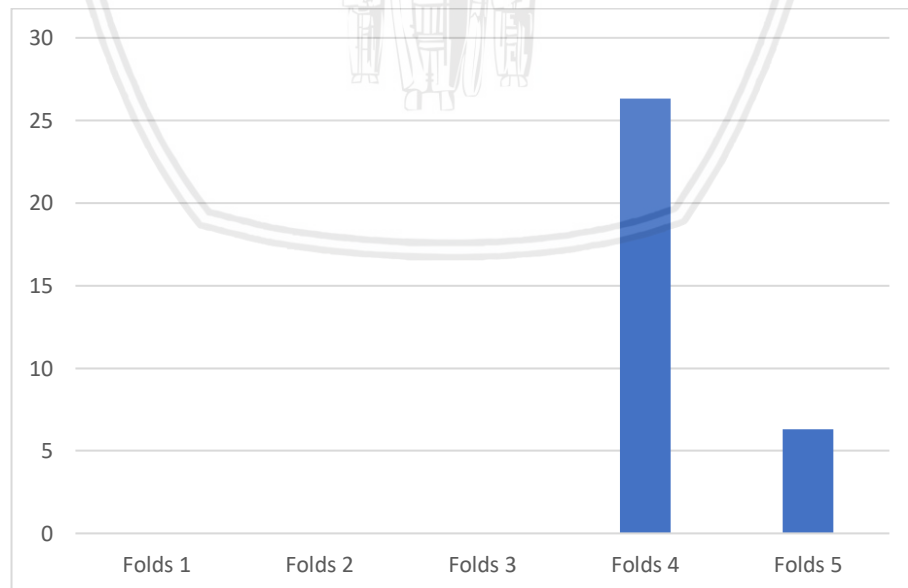
sebagai data uji dan *fold* lainnya sebagai data latih. Pada gambar, sumbu x adalah folds yang digunakan dan sumbu y adalah tingkat akurasi yang didapatkan.



Gambar 6.5 Hasil Pengujian 9 Fitur

6.1.6 Hasil Pengujian 8 Fitur

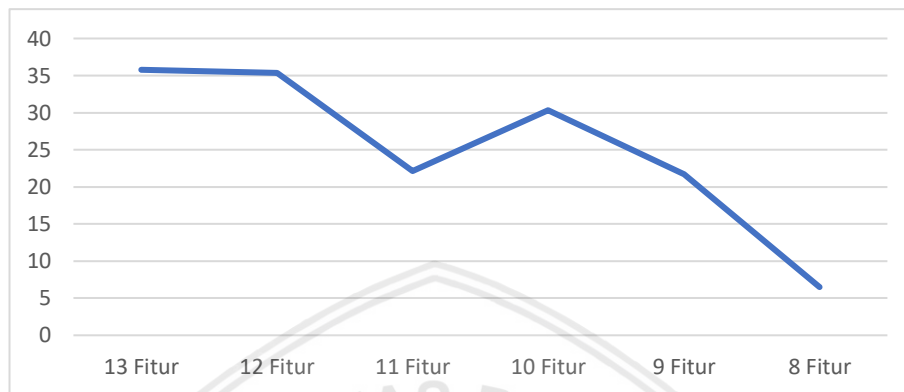
Berikut adalah hasil pengujian seleksi fitur dengan menggunakan jumlah fitur adalah 8 fitur yang digambarkan pada Gambar 6.6. Berdasarkan hasil pengujian dengan menggunakan 8 fitur pada lima kombinasi *folds*, didapatkan akurasi rata-rata adalah 6,5%. Kombinasi terbaik terdapat pada kombinasi 4 dengan F4 sebagai data uji dan *fold* lainnya sebagai data latih. Pada gambar, sumbu x adalah folds yang digunakan dan sumbu y adalah tingkat akurasi yang didapatkan.



Gambar 6.6 Hasil Pengujian 8 Fitur

6.1.7 Hasil Penujian Seleksi Fitur

Pada hasil pengujian seleksi fitur di atas, didapatkan jumlah fitur terbaik untuk klasifikasi ini adalah 13 fitur. Seperti yang tergambar pada grafik, semakin sedikit fitur yang digunakan, semakin kecil pula nilai akurasi yang dihasilkan. Grafik akurasi pengujian seleksi fitur tersebut tergambar pada Gambar 6.7.



Gambar 6.7 Grafik Nilai Rata-rata Akurasi Pada Enam Jumlah Fitur

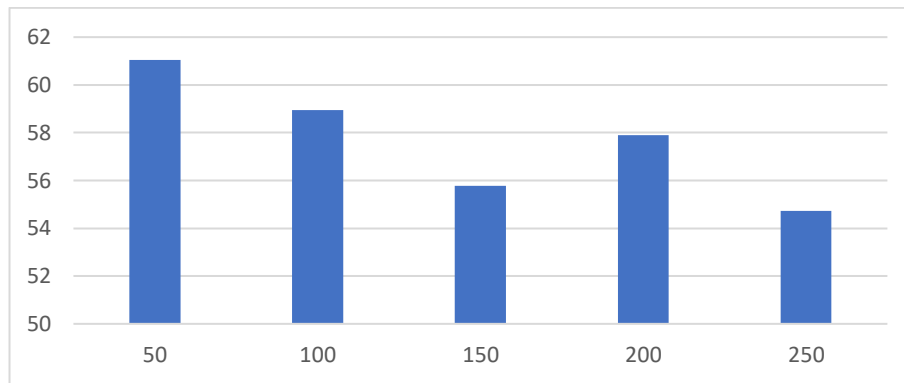
6.2 Pengujian *K-Fold Cross Validation* Jumlah *Tree*

Pada pengujian sebelumnya didapatkan jumlah fitur yang terbaik adalah 13 fitur. Maka dari itu pada pengujian ini digunakan 13 fitur untuk proses klasifikasi. Pengujian ini dilakukan untuk mengetahui kombinasi data latih dan data uji terbaik dari *dataset* yang dimiliki.

Jumlah keseluruhan *dataset* adalah 95. Dari data tersebut dibagi menjadi lima *folds* yang masing-masing *fold* terdiri dari 19 data. Kombinasi data latih dan data uji dari *folds* yang ada telah tergambar pada perancangan pengujian. Berikut adalah hasil pengujian dengan menggunakan metode *K-fold cross validation*.

6.2.1 Hasil Pengujian Kombinasi *Folds* 1

Pada pengujian ini dilakukan klasifikasi menggunakan *random forest* dengan kombinasi F1 sebagai data uji, F2, F3, F4, dan F5 sebagai data latih. Pengujian dilakukan dengan lima jumlah *tree* yang berbeda, 50, 100, 150, 200, dan 250 *tree*. Grafik akurasi berbanding jumlah *tree* tersebut tergambar pada Gambar 6.8. Pada gambar, sumbu x adalah jumlah *tree* yang digunakan dan sumbu y adalah tingkat akurasi yang didapatkan.

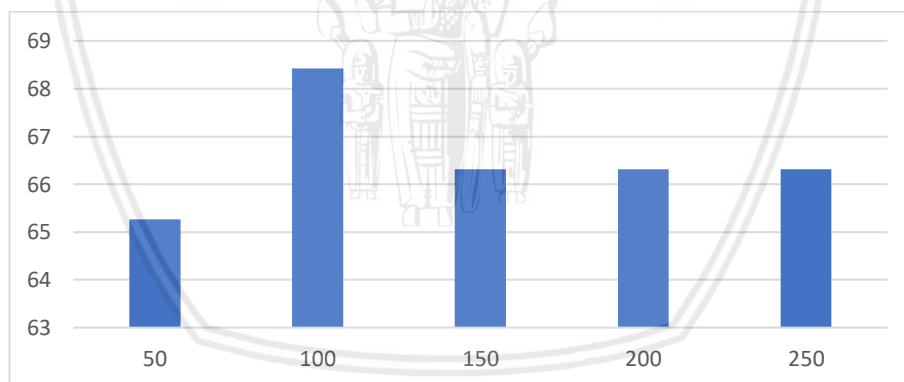


Gambar 6.8 Hasil Akurasi Pengujian Folds 1

Dari hasil pengujian pada kombinasi *folds* 1 dengan lima jumlah *tree* yang berbeda, 50, 100, 150, 200, dan 250 *tree*, didapatkan rata-rata akurasi sebesar 58,68%.

6.2.2 Hasil Pengujian Kombinasi Folds 2

Pada pengujian ini dilakukan klasifikasi menggunakan *random forest* dengan kombinasi F2 sebagai data uji, F1, F3, F4, dan F5 sebagai data latih. Pengujian dilakukan dengan lima jumlah *tree* yang berbeda, 50, 100, 150, 200, dan 250 *tree*. Grafik nilai akurasi berbanding jumlah *tree* tersebut tergambarkan pada Gambar 6.9. Pada gambar, sumbu x adalah jumlah *tree* yang digunakan dan sumbu y adalah tingkat akurasi yang didapatkan.

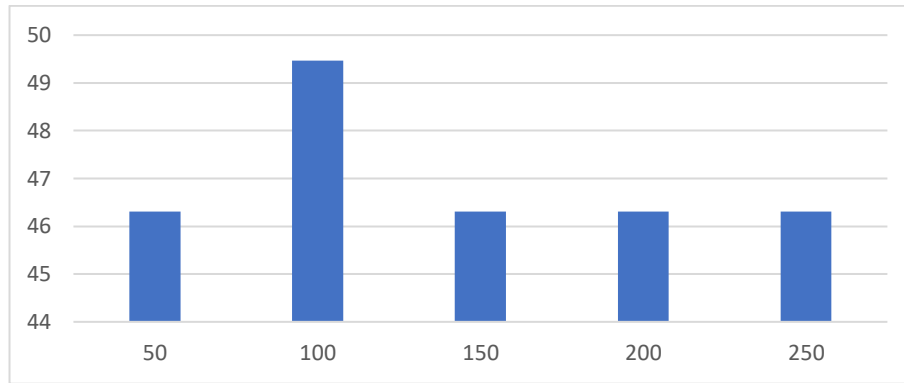


Gambar 6.9 Hasil Akurasi Pengujian Folds 2

Dari hasil pengujian pada kombinasi *folds* 2 dengan lima jumlah *tree* yang berbeda, 50, 100, 150, 200, dan 250 *tree*, didapatkan rata-rata akurasi sebesar 66,52%.

6.2.3 Hasil Pengujian Kombinasi Folds 3

Pada pengujian ini dilakukan klasifikasi menggunakan *random forest* dengan kombinasi F3 sebagai data uji, F2, F1, F4, dan F5 sebagai data latih. Pengujian dilakukan dengan lima jumlah *tree* yang berbeda, 50, 100, 150, 200, dan 250 *tree*. Grafik nilai akurasi berbanding jumlah *tree* tersebut tergambarkan pada Gambar 6.10. Pada gambar, sumbu x adalah jumlah *tree* yang digunakan dan sumbu y adalah tingkat akurasi yang didapatkan.

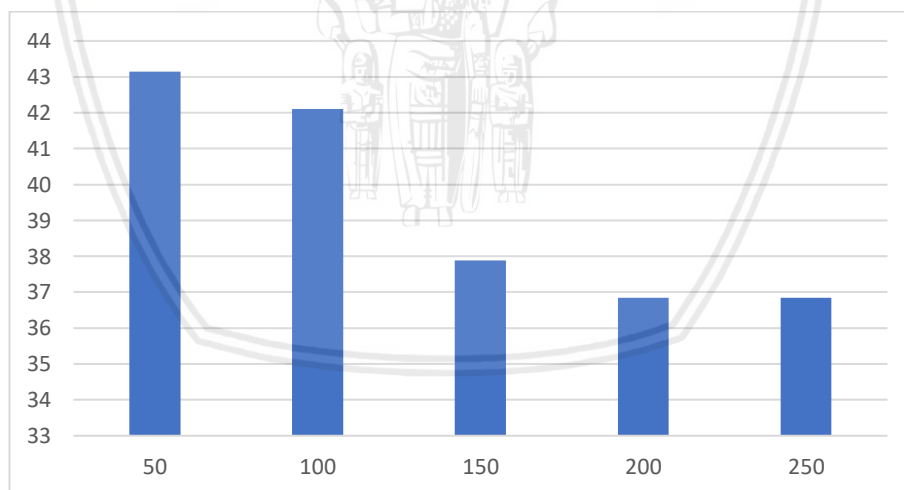


Gambar 6.10 Hasil Akurasi Pengujian *Folds* 3

Dari hasil pengujian pada kombinasi *folds* 3 dengan lima jumlah *tree* yang berbeda, 50, 100, 150, 200, dan 250 *tree*, didapatkan rata-rata akurasi sebesar 46,94%.

6.2.4 Hasil Pengujian Kombinasi *Folds* 4

Pada pengujian ini dilakukan klasifikasi menggunakan *random forest* dengan kombinasi F4 sebagai data uji, F2, F3, F1, dan F5 sebagai data latih. Pengujian dilakukan dengan lima jumlah *tree* yang berbeda, 50, 100, 150, 200, dan 250 *tree*. Grafik nilai akurasi berbanding jumlah *tree* tersebut tergambarkan pada Gambar 6.11. Pada gambar, sumbu x adalah jumlah *tree* yang digunakan dan sumbu y adalah tingkat akurasi yang didapatkan.



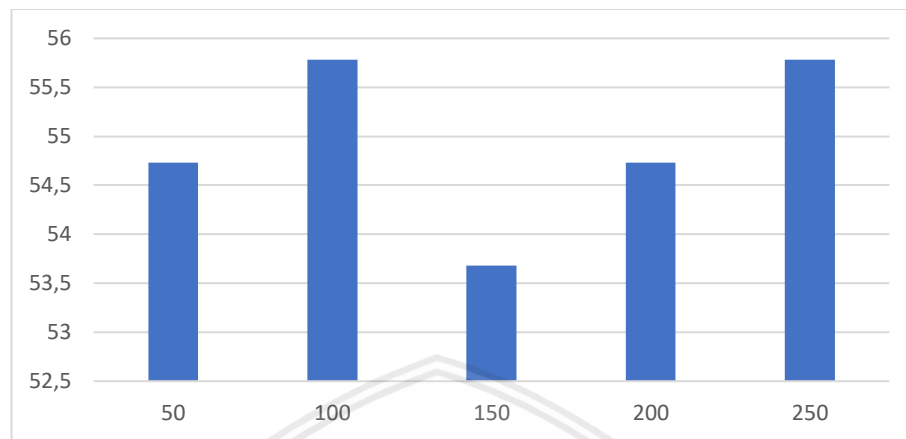
Gambar 6.11 Hasil Akurasi Pengujian *Folds* 4

Dari hasil pengujian pada kombinasi *folds* 4 dengan lima jumlah *tree* yang berbeda, 50, 100, 150, 200, dan 250 *tree*, didapatkan rata-rata akurasi sebesar 39,36%.

6.2.5 Hasil Pengujian Kombinasi *Folds* 5

Pada pengujian ini dilakukan klasifikasi menggunakan *random forest* dengan kombinasi F5 sebagai data uji, F2, F3, F4, dan F1 sebagai data latih. Pengujian dilakukan dengan lima jumlah *tree* yang berbeda, 50, 100, 150, 200, dan 250 *tree*.

Grafik nilai akurasi berbanding jumlah *tree* tersebut tergambarkan pada Gambar 6.12. Pada gambar, sumbu x adalah jumlah *tree* yang digunakan dan sumbu y adalah tingkat akurasi yang didapatkan.



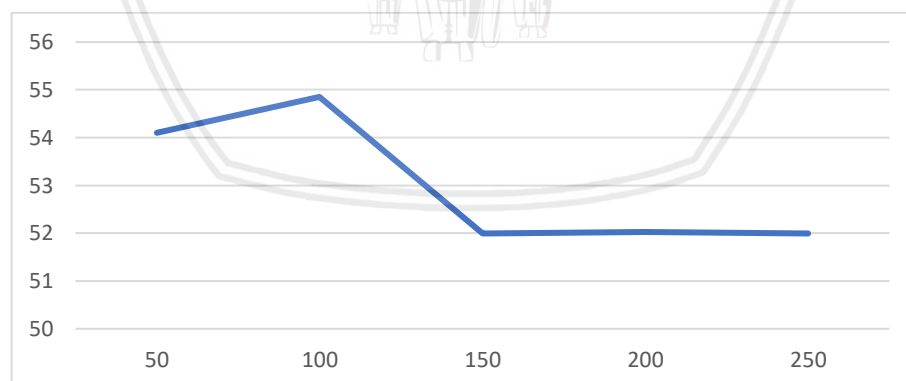
Gambar 6.12 Hasil Akurasi Pengujian *Folds* 5

Dari hasil pengujian pada kombinasi *folds* 5 dengan lima jumlah *tree* yang berbeda, 50, 100, 150, 200, dan 250 *tree*, didapatkan rata-rata akurasi sebesar 54,94%.

6.2.6 Hasil Pengujian Jumlah *Tree*

Pada hasil pengujian dengan lima jumlah *tree* yang berbeda, didapatkan jumlah *tree* yang paling baik untuk klasifikasi ini adalah 100 *tree*.

Seperti yang tergambarkan pada grafik, jumlah *tree* 50 ke 100 mengalami kenaikan, namun saat jumlah *tree* ditambah menjadi 150, nilai akurasi klasifikasi berkurang. Grafik akurasi klasifikasi berdasarkan jumlah *tree* tergambarkan pada Gambar 6.6.



Gambar 6.13 Grafik Nilai Rata-rata Akurasi Pada Lima Jumlah *Tree*

6.3 Analisis Hasil Pengujian

6.3.1 Analisis Hasil Pengujian Jumlah Fitur

Dari pengujian jumlah fitur dengan menyeleksi fitur pada dataset yang dimiliki dengan metode *information gain*, didapatkan bahwa fitur yang terbaik untuk klasifikasi ini adalah berjumlah 13 fitur. Dilihat dari grafik nilai rata-rata seleksi

fitur, semakin sedikit fitur yang digunakan, nilai akurasi yang dihasilkan semakin kecil. Hal ini disebabkan beberapa dari percobaan yang dilakukan dengan *k-fold cross validation* mengalami *error* atau memiliki nilai akurasi sama dengan nol.

Pada kasus percobaan yang mendapatkan nilai akurasi sama dengan nol, hal ini terjadi ketika pada data latih yang dilakukan pemecahan pada proses pembangunan *tree* masih memiliki dua atau lebih kelas namun semua fiturnya telah memiliki nilai yang sama. Hal tersebut menyebabkan data yang masih memiliki kelas lebih dari satu tersebut tidak dapat dipisah sehingga pembangunan *tree* tersebut dianggap gagal dan mendapatkan nilai akurasi sebesar nol.

Dari Gambar 6.1 sampai dengan Gambar 6.6, terlihat bahwa setiap kombinasi *folds* yang dijalankan, menghasilkan nilai akurasi yang berbeda. Ada dua faktor utama yang menyebabkan perbedaan tingkat akurasi dari setiap kombinasi *folds*. Faktor yang pertama adalah penyebaran data latih pada setiap kombinasi *folds*. Penyebaran kelas pada data latih yang semakin merata, semakin tinggi nilai akurasi yang dihasilkan. Faktor kedua adalah pengaruh dari proses pengacakan data pada proses *training*. Apabila pada proses pengacakan data latih tersebut menghasilkan data baru yang memiliki kelas yang merata, maka data *random* tersebut menghasilkan nilai akurasi yang tinggi.

6.3.2 Analisis Hasil Pengujian Jumlah Tree

Setelah dilakukan pengujian dengan menggunakan metode *K-fold cross validation*, didapatkan kombinasi terbaik adalah kombinasi 2 dengan F2 sebagai data uji dan sisanya menjadi data latih. Jumlah *tree* terbaik yang didapatkan adalah 100 *tree*. Hasil rata-rata akurasi yang didapatkan dari semua percobaan adalah 53,094%.

Pada penelusuran penulis terhadap kekurangan dari klasifikasi ini adalah dikarenakan penyebaran kelas pada dataset tidak merata. Data dengan kelas “Gangguan” memiliki jumlah yang tidak seimbang dengan kedua kelas lainnya. Hal ini berpengaruh saat pengacakan secara random pada proses *training*. Hal tersebut menyebabkan salah satu dari 150 *tree* yang terbangun, tidak terdapat kelas “Gangguan”, sehingga data uji dengan kelas asli “Gangguan” tidak dapat diklasifikasikan sebagai kelas “Gangguan”.

Masalah lain ditemukan ketika data uji memiliki *value* dari salah satu fitur yang tidak terdapat pada *value* fitur yang ada pada data latih yang telah diacak secara random. Seperti contoh data uji memiliki *value* “4” untuk fitur usia, namun pada *tree* yang terbangun tidak memiliki *value* “4” pada fitur usianya. Hal ini menyebabkan data uji tersebut tidak dapat diklasifikasikan atau pada implementasi kelasnya menjadi “X”.

BAB 7 PENUTUP

Bab penutup ini membahas kesimpulan dan saran dari seluruh proses penelitian klasifikasi penurunan fungsi kognitif pada pasien stroke menggunakan metode *random forest*.

7.1 Kesimpulan

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan pada tahap sebelumnya, maka dapat ditarik kesimpulan sebagai berikut:

1. Penerapan metode *random forest* untuk klasifikasi pada kasus penurunan fungsi kognitif pasien stroke ini dapat berjalan dengan rata-rata nilai ketepatan sebesar 53.094% berdasarkan pengujian *k-fold cross validation*.
2. Jumlah fitur yang digunakan dalam proses klasifikasi sangat berpengaruh. Fitur yang optimal untuk penelitian ini adalah 13 fitur.
3. Jumlah *tree* memengaruhi hasil dari klasifikasi pada kasus penurunan fungsi kognitif pasien stroke menggunakan metode *random forest*. Jumlah *tree* yang optimal untuk penelitian ini adalah 100 *trees*.
4. Nilai akurasi setiap pengujian dapat berubah-ubah dikarenakan data yang digunakan dalam proses *training* terlebih dahulu diacak secara *random*.

7.2 Saran

Saran yang diberikan oleh penulis untuk pembaca dan pengembang penelitian serupa selanjutnya di kemudian hari adalah sebagai berikut:

1. Untuk penelitian selanjutnya, diharapkan untuk menggunakan data latih yang lebih banyak dengan penyebaran kelas yang merata demi meningkatkan nilai akurasi klasifikasi.
2. Selain itu, diharapkan juga untuk penelitian selanjutnya menggunakan pembangunan *tree* selain dengan metode CART.
3. Demi mendapatkan hasil yang baik, diharapkan menggunakan kombinasi *fold* yang lebih banyak pada pengujian *k-fold cross validation*.

DAFTAR REFERENSI

- Aaronson, P. i., 2012. *At a Glance Sistem Kardiovaskular*. Jakarta: Erlangga.
- Alkayed, N. J. & al., e., 1998. Gender-linked Brain Injury in Experimental Stroke. *Stroke*, Volume 29.1, pp. 159-166.
- Anggorowati, L., 2013. Faktor Risiko Kanker Payudara Wanita. *Jurnal Kesehatan Masyarakat*.
- Binarwati, L., Mukhlash, I. & Soetrisno, 2017. Implementasi Algoritma Genetika untuk Optimalisasi *random forest* Dalam Proses Klasifikasi Penerimaan Tenaga Kerja Baru: Studi Kasus PT.XYZ. *Jurnal Sains dan Seni ITS*.
- Breiman, L., 1984. *Classification and Regression Trees*. New York: Chapman & Hall.
- Departemen Kesehatan RI, 2008. Buku Saku Pencegahan Kanker Leher Rahim & Kanker Payudara. *Direktorat Pengendalian Penyakit Tidak Menular*.
- Douri, A. & et. al, 2012. Prevalence of SPostroke Cognitive Impairment. *London Stroke Register 1995-2010*, pp. 44:138-145.
- Eveline, K., Purwanto, H. & Lestari, P., 2017. Faktor Klinis dan Histopatologi serta Hubungannya dengan Kekambuhan Pasca Operasi pada Pasien Kanker Payudara di RSUD DR. Soetomo, Januari-Juni 2015. *Fakultas Kedokteran Universitas Airlangga*.
- Fitriani Ab, R., Bijaksana, M. A. & Adiwijaya, 2007. Analisis dan Implementasi Algoritma *random forest* Sebagai Sebuah Classifier dalam Data Mining. *Tugas Akhir Telkom University*.
- Gylmour & et. al., 2007. Lesion Characteristic, NIH Stroke Scale, and Functional Recovery After Stroke. *American Journal of Physical Medicine & Rehabilitation*, 86(9), pp. 725-733.
- Hanas, M., Lestari, E. & Asni, A. K., 2015. Gambaran Fungsi Kognitif pada Pasien Pasca Stroke di Poliklinik Saraf RSUD Arifin Achmad Provinsi Riau.
- Harsono, 2007. *Kapita Selekta Neurologi Edisi ke-2*. Yogyakarta: Gajah Mada University Press.
- Hofgren, C., 2009. Screening of Cognitive Function. *Evaluation of Methods and Their Applicability in Neurological Rehabilitation*, pp. 9-12.
- Jellinger, K. A., 2013. Pathology and Pathogenesis of Vascular Cognitive Impairment - a Critical Update. *Frontiers in Aging Neoruscience*, Volume 5(17), pp. 1-19.
- Lee, 2003. Education, Other Socioeconomic Indicators, and Cognitive Function. *American Journal of Epidemiology*, Volume 157, pp. 712-720.
- Malik, A. & Maulina, M., 2015. Fungsi Kognitif Pasien Stroke Berdasarkan Mini Mental State Examination (MMSE) di Rumah Sakit Umum Cut Meutia Kabupaten Aceh Utara. *Fakultas Kedokteran Universitas Malikussaleh*.

Maria, I. L., Sainal, A. A. & Nyorong, M., 2017. Risiko Gaya Hidup Terhadap Kejadian Kanker Payudara Pada Wwanita. *Jurnal MKMI*.

Nugraha, D. W., Dodu, A. E. & Chandra, N., 2017. Klasifikasi Penyakit Stroke Menggunakan Metode Naïve Bayes Classifier (Studi Kasus pada Rumah Sakit Umum Daerah Udata Palu). *semanTIK*.

Rukmigayatri, R., 2015. Klasifikasi Kemunculan Titik Panas pada Lahan Gambut di Sumatera dan Kalimantan Menggunakan Algoritme *random forest*. *Fakultas Matematika dan Ilmu Pengetahuan Alam Institut Pertanian Bogor*.

Shin, 2013. Mechanisme of Recurrence in Subtypes of Ischemic Stroke. *AMA*, Volume 62, pp. 1232-1237.

Stroke Accosiation, 2013. [Online]

Available at: <http://www.stroke.org.uk/resource-sheet/stroke-statistiks>

[Accessed 22 Desember 2013].

Wardhani, Y. I., 2015. Faktor-Faktor yang Memengaruhi Penurunan Fungsi Kognitif Pasien Stroke di Rumah Sakit Saiful Anwar Malang.

Wreksoatmodjo, B. R., 2014. Beberapa Kondisi Fisik dan Penyakit yang Meruapakan Faktor Resiko Gangguan FUNgsi Kognitif. 41(1).

