Mukhamad Hasim Iswanto, S.Kom., M.Eng.

PEMILIHAN FITUR DATA MINING UNTUK DIAGNOSA TBC

Mukhamad Hasim Iswanto, S.Kom., M.Eng.

PEMILIHAN FITUR DATA MINING UNTUK DIAGNOSA TBC

Pustaka Ilmu
CV. Pustaka Ilmu Group

PEMILIHAN FITUR
DATA MINING
UNTUK DIAGNOSA
TBC

Penulis:
Mukhamad Hasim Iswanto, S.Kom., M.Eng.

Copyright © Pustaka Ilmu, 2024

viii+120 halaman; 14,5x21 cm

Hak cipta ada Pada Penulis

ISBN: 978-623-509-296-6

Penulis : Mukhamad Hasim Iswanto, S.Kom., M.Eng.
Editor : Ahmad Zayyadi, M.A., M.H.I.
Desain Cover : Nur Afandi
Layout : Nur Afandi

Penerbit Pustaka Ilmu
Griya Larasati No. 079 Tamantirto, Kasihan,
Bantul Yogyakarta Telp/Faks: (0274)4435538
E-mail: radaksipustakailmu@gmail.com
Website: https:// www.pustakailmu.co.id
Layanan WhatsApp: 081578797497

Anggota IKAPI

Cetakan I, November 2024

Marketing:
Griya Larasati No. 079 Tamantirto, Kasihan,
Bantul Yogyakarta Telp/Faks: (0274)4435538
E-mail: radaksipustakailmu@gmail.com
Website: https:// www.pustakailmu.co.id
Layanan WhatsApp: 0815728053639

© Hak Cipta dilindungi Undang-undang
All Rights Reserved

Dilarang memperbanyak sebagian atau seluruh isi buku ini
dalam bentuk apa pun tanpa izin tertulis dari Penerbit
Pustaka Ilmu Yogyakarta

PENGANTAR PENULIS

Puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat dan barokahnya sehingga penulis dapat menyelesaikan buku dengan judul *Pemilihan Fitur Data Mining Untuk Diagnosa TBC*.

Dalam melakukan penulisan buku ini penulis telah mendapatkan banyak dukungan dan bantuan dari berbagai pihak. Penulis mengucapkan terima kasih yang tak terhingga kepada semua pihak yang telah membantu selesainya penulisan buku ini.

Penulis menyadari sepenuhnya bahwa buku ini masih jauh dari sempurna, untuk itu semua jenis saran, kritik dan masukan yang bersifat membangun sangat penulis harapkan. Akhir kata, semoga buku ini dapat memberikan manfaat dan memberikan wawasan tambahan bagi para pembaca dan khususnya bagi penulis sendiri.

Yogyakarta, 15 Oktober 2024

Mukhamad Hasim Iswanto

ARTI LAMBANG DAN SINGKATAN

P (C $_j$|X) = Probabililas data record x yang mempunyai label kelas C_j

RST = *Rough Set Theory*

SVM = *Support Vector Machines*

IBk = *Instance Based Learning* dengan k-*nearesl neighbor*

ANFIS = *Adaptive Network Fuzzy Inference System*

PART = *Partial Decisian Tree*

REPTree = *Reduced Error Pruning Tree*

TP = *True Positive*

TN = True Negative

FP = *False Positive*

FN = *False Negative*

CFS = *Correlation Based Feature Selection*

MFS = *Motivated Feature Selection*

DAFTAR ISI

KATA PENGANTAR v

BAB I
PENDAHULUAN 1

BAB II
KOMPUTASI CERDAS MENDIAGNOSIS PENYAKIT 6

A. Tinjauan 6
B. Teori 9
1. Penyakit tuberkulosis (TBC) 9
2. Metode konvensional untuk mendiagnosis TBC 10
3. Metode komputasi cerdas mendiagnosis TBC 18
C. Rangkuman 55

BAB III
***PREPROCESSING* DIAGNOSIS 57**

A. Pendahuluan 57
B. Perangkat Lunak yang Digunakan 59
1. *Rosetta GUI* version 1.4.41 60
2. WEKA 3.7.10 60
C. Metode 61
1. Data preprocessing 62
2. Pemilihan Fitur 64
3. Pengacakan data 70
4. Klasifikasi data 72
5. Evaluasi performa 81

BAB IV
***PROSESSING* DIAGNOSIS** **84**
A. Pendahuluan 84
B. Data preprosessing 84
C. Pemilihan Fitur 88
D. Performa Klasifikasi 89
1. Performa Klasifikasi dengan 10-*fold cross validation* 89
2. Peringkat Skema Pemilihan Fitur 92
3. Penggabungan pemilihan fitur berbasis pakar medis dan komputer 95
E. Pembahasan 100
1. Performa Klasifikasi dan Peringkat Skema Pemilihan Fitur 100
2. Penggabungan Pemilihan Fitur Berbasis Pakar Medis dan Komputer 107
F. Rangkuman 109

BAB V
KESIMPULAN **111**

DAFTAR PUSTAKA 113

BAB I

PENDAHULUAN

Paru-paru adalah organ pada sistem pernapasan (respirasi) dan berhubungan dengan sistem peredaran darah (sirkulasi) *vertebrata* yang bernapas dengan udara. Fungsinya adalah menukar oksigen dari udara dengan karbon dioksida dari darah. Prosesnya disebut "pernapasan eksternal" atau bernapas. Paru-paru juga mempunyai fungsi nonrespirasi. Istilah kedokteran yang berhubungan dengan paru-paru sering mulai di *pulmo-*, dari kata Latin *pulmones* untuk paru-paru [1]. Paru-paru sangat erat kaitannya pada sistem pernapasan (respirasi) dan berhubungan dengan sistem peredaran darah. Tingkat pencemaran udara yang sangat tinggi organ yang terletak di bawah tulang rusuk ini memang mempunyai tugas yang berat yaitu menghirup berbagai bibit penyakit yang berkeliaran di udara. Ini semua dapat menimbulkan berbagai penyakit paru-paru. Pada organ paru terdapat beberapa penyakit yang dapat dikelompokkan menjadi dua yaitu penyakit paru spesifik (Tuberkulosis Paru) dan penyakit paru non spesifik seperti bronkitis, *pneumonia bronkiectasis*, asma *bronkiole*, tumor paru, dan lain-lain. Diantara penyakit paru, penyakit yang paling menular dan berbahaya adalah Tuberkulosis Paru (Tb paru).

Tuberkulosis (TBC) merupakan penyakit menular yang umum, dan dalam banyak kasus bersifat mematikan. Penyakit ini disebabkan oleh berbagai *strainmikobakteria*, umumnya

Mycobacterium tuberculosis. Tuberkulosis biasanya menyerang paru-paru, namun juga bisa berdampak pada bagian tubuh lainnya. Tuberkulosis menyebar melalui udara ketika seseorang dengan infeksi TB aktif batuk, bersin, atau menyebarkan butiran ludah mereka melalui udara [2]. Infeksi TB umumnya bersifat *asimtomatik* dan laten. Namun hanya satu dari sepuluh kasus infeksi laten yang berkembang menjadi penyakit aktif. Apabila Tuberkulosis tidak diobati maka lebih dari 50% orang yang terinfeksi bisa meninggal. TBC dapat menyerang siapa saja (tua, muda, laki-laki, perempuan). Sekitar 75% pasien TBC adalah kelompok usia yang paling produktif secara ekonomis (15-50 tahun). Diperkirakan seorang pasien TBC dewasa, akan kehilangan rata-rata waktu kerjanya 3 sampai 4 bulan. Survei prevalensi TB yang dilakukan di enam provinsi pada tahun 1983-1993 menunjukkan bahwa prevalensi TBC di Indonesia berkisar antara 0,2 – 0,65%. Menurut laporan penanggulangan TBC Global yang dikeluarkan oleh WHO pada tahun 2004, angka insidensi TBC pada tahun 2002 mencapai 555.000 kasus (256 kasus/100.000 penduduk), dan 46% diantaranya diperkirakan merupakan kasus baru [3].

Gejala klasik infeksi TB aktif yaitu batuk kronis dengan bercak darah sputum atau dahak, demam, berkeringat di malam hari, dan berat badan turun. (dahulu TB disebut penyakit "konsumsi" karena orang-orang yang terinfeksi biasanya mengalami kemerosotan berat badan). Infeksi pada organ lain menimbulkan gejala yang bermacam-macam. Diagnosis TB aktif bergantung pada hasil radiologi (biasanya melalui sinar-X dada) serta pemeriksaan *mikroskopis* dan pembuatan kultur mikrobiologis cairan tubuh. Sementara itu, diagnosis TB laten bergantung pada tes tuberkulin kulit/*tuberculin skin test* (TST)

dan tes darah. Pengobatan sulit dilakukan dan memerlukan pemberian banyak macam antibiotik dalam jangka waktu lama. Orang-orang yang melakukan kontak juga harus menjalani tes penapisan dan diobati bila perlu. Resistensi antibiotik merupakan masalah yang bertambah besar pada infeksi tuberkulosis resisten multi-obat (TBC MDR). Untuk mencegah TB, semua orang harus menjalani tes penapisan penyakit tersebut dan mendapatkan vaksinasi basil Calmette–Guérin.

Diagnosis awal TBC dapat dilakukan melalui pengkajian dari gejala klinis, pemeriksaan fisik kemudian uji lanjutan dapat dilakukan. Gejala klinis yang sering ditemui pada tuberkulosis paru adalah batuk yang tidak spesifik tetapi progresif. Pada pemeriksaan fisik kadang kita dapat menemukan suara yang khas tergantung seberapa luas dan dan seberapa jauh kerusakan jaringan paru yang terjadi. Uji lanjutan yang dilakukan yaitu gambaran radiologi atau *rontgen* paru dan pemeriksaan laboratorium klinis maupun bakteriologis. Pemeriksaan *rontgen* dapat menunjukkan gambaran yang bermacam macam dan tidak dapat dijadikan gambaran diagnostik yang absolut dari tuberculosis paru. Pada pemeriksaan laboratorium, peningkatan laju endap darah dapat menunjukan proses yang sedang aktif, tapi laju endap darah yang normal bukan berarti menyingkirkan adanya proses *tuberculosis*. Penemuan adanya Basil Tahan Asam (BTA) pada dahak, bilasan bronkus, bilasan lambung, cairan *pleura* atau jaringan paru adalah sangat penting untuk mendiagnosa TBC Paru. Sering dianjurkan untuk pemeriksaan dahak sebanyak 3 kali untuk dahak yang diambil pada *specimen* dahak dalam waktu 2 hari berturut-turut, yaitu Sewaktu-Pagi-Sewaktu. Pemeriksaan dahak dilakukan pada orang dewasa. Selain di lakukan pemeriksaan dahak juga

dilanjutkan dengan uji tuberkulin. Uji tuberkulin dilakukan dengan cara menyuntikkan secara intrakutan (yakni di dalam kulit), dengan tuberkulin PPD RT 23 kekuatan 2 TU (*Tuberculin Unit*). Pembacaan dilakukan 48-72 jam setelah penyuntikan, dan diukur diameter dari peradangan atau indurasi yang dinyatakan dalam milimeter. Dinyatakan positif apabila indurasi sebesar > 10 mm [4]. Uji tuberkulin dilakukan pada anak Balita ini lebih dipilih oleh ahli paru untuk mendiagnosis keberadaan TB pada pasien karena balita sulit untuk mengeluarkan dahak dengan akurasi yang tinggi meskipun *invasive*, mempunyai risiko, dan mahal.

Jika dilihat dari kekurangan uji ini, perlu dikembangkan sebuah metode yang mampu mendiagnosis TBC sebelum dilakukan uji tuberkulin dan pemeriksaan dahak. Tujuannya adalah untuk menghindari prosedur diagnosis TB yang *invasive*, memiliki risiko, waktu yang lama dan mahal terhadap pasien. Oleh karena itu, hal ini memotivasi pengembangan suatu metode komputer untuk dapat mendiagnosis keberadaan TBC. Metode komputer dapat menyediakan prosedur diagnosis TBC terhadap pasien dengan cara yang *non-invasive*, aman, waktu yang singkat dan lebih murah.

Banyak buku yang mengembangkan berbagai macam teknik komputasi cerdas untuk mendiagnosis penyakit paru. Metode *forward chaining* [5], metode *clustering* dan klasifikasi [6], dan data mining [7], [8], [9], [10], [11], diusulkan untuk mendiagnosis paru-paru. Pada diagnosis medis, reduksi data merupakan masalah penting. Data medis sering mengandung sejumlah besar fitur yang tidak relevan dan sejumlah kasus yang relatif sedikit sehingga dapat mempengaruhi kualitas diagnosis penyakit. Oleh karena itu, proses pemilihan fitur dapat

digunakan untuk menyeleksi fitur-fitur yang relevan pada data medis. Proses pemilihan fitur diusulkan dalam banyak buku untuk meningkatkan akurasi pada proses diagnosis paru-paru

Bakar dan Febriyani [7] melakukan satu proses pemilihan fitur berbasis komputer. Oleh karena itu tidak memberikan informasi penggunaan metode pemilihan fitur lainnya. Selain itu, buku ini juga menemukan perbedaan karakteristik yang digunakan dengan buku sebelumnya. Perbedaan ini jelas dapat mempengaruhi hasil pemilihan fitur dan performa diagnosis.

BAB II

KOMPUTASI CERDAS MENDIAGNOSIS PENYAKIT

A. Tinjauan

Banyak teknik komputasi cerdas yang telah dikembangkan pada buku sebelumnya. Metode *forward chaining* [5], metode *clustering* dan klasifikasi [6], dan data mining [7], [8], [9], [10], [11] diusulkan untuk mendiagnosis paru-paru.

Costa, dkk. [14] meneliti tentang identifikasi otomatis *mycrobacterium tuberculosis* melalui gambar dari mikroskop konvensional berdasarkan warna merah dan hijau menggunakan *global adaptive threshold segmentation*. Buku ini menggunakan *sample* dahak dari 10 pasien yang terinfeksi tuberkulosis. Hasil dari buku ini menunjukkan sensitivitas terbaik dan *false positive rate* masing-masing 76,65% dan 12%.

Semogan, dkk. [15] membangun sebuah sistem untuk mendiagnosis tuberkulosis dengan *decission support system* menggunakan *rule-based fuzzy*. Sitem ini ditujukan untuk dokter paru. Sistem akan menganaliss kelas tuberkulosis pasien. Sistem ini menyediakan *workplace* dimana dokter akan memasukkan nilai sesuai dengan gejala yang dialami pasien. Selanjutunya sistem akan meringkas dan mengalokasikan kelas dari TB yang diderita oleh psaien. Skor yang diberikan oleh dokter menyiratkan intensitas setiap gejala yang dialami pasien. Dalam perumusan/*fuzzy set*, rentang skor diklasifikasikan dalam setiap

gejala . Nilai-nilai atau skor telah menjalani proses *fuzzifikasi*. Sistem yang dibuat membutuhkan dokter untuk menentukan nilai yang diperlukan sesuai dengan gejala tuberkulosis. Inputan sistem dan skornya adalah sebagai berikut:

Kondisi A: Batuk (0-10), dan durasi batuk (0-30 hari).

Kondisi B: Suhu tubuh (33-45), dan durasi deman (0-30 hari).

Kondisi C: Perubahan warna dahak (0-10), dan dahak hidung (0-10).

Kondisi D: Menggigil sore (0-10), dan berkeringat malam (0-10).

Kondisi E: penurunan berat badan (0-20 kg), dan hilangnya nafsu makan (0-10).

Sistem ini menghasilkan 16 *rules* untuk kondisi A sampai E, dan 323 *rules set* (kombinasi dari 5 kondisi). Kelas tuberkulosis yang menjadi *output* sistem yaitu,

PTB1, PTB2, PTB3, PTB4, dan PTB5.

Ucar dan Karahoca [10] mengusulkan teknik data mining untuk diagnosis tuberkulosis. Ada beberapa metode yang digunakan pada buku ini, yaitu *adaptive neuro fuzzy inference system (ANFIS), multilayer perceptron, dan partial decision trees (PART).* Buku ini menggunakan 667 data pasien di sebuah klinik yang memiliki 30 atribut. Hasil dari buku ini menunjukan metode ANFIS lebih baik dibandingkan dengan metode *multilayer perceptron, dan partial decision trees* (PART).

Asha, dkk. [12] menggunakan metode klasifikasi *ensemble* seperti *bagging, adaBoostM1,* dan *random forest tree* untuk mendiagnosis penyakit tuberkulosis. Buku ini menggunakan dataset medis sebanyak 250 pasien penderita tuberkuosis terdiri dari 11 atribut gejala tuberkulosis dan satu atribut kelas. Hasil dari perbandingan ketiga metode

menunjukkan bahwa metode *bagging* dan *adaBoostM1* lebih baik dari pada *random forest tree,* dengan masing-masing akurasi 96%, 97%, dan 93%.

Ucar dkk. [11] menggunakan metode ANFIS dan *roughs Set* untuk mengklasifikasikan penyakit tuberkulosis. Dataset yang digunakan sebanyak 503 data pasien dengan 30 atribut yag berbeda. Dari buku tersebut diperoleh 8 *rule* untuk metode ANFIS dengan performa klasifikasi masing-masing adalah *sensitivity* 0,95, *specificity* 0,97, dan *correctness* 0,97, sementara untuk metode *Rough Set* diperoleh 30 *rule* dengan performa klasifikasi masing-masing adalah *sensitivity* 0,93, *specificity* 0,92, dan *correctness* 0,92.

Ali dkk. [15] menggunakan algoritme *naive bayes* dan *bayesian network* untuk melakukan klasifikasi tuberkulosis. Dataset yang digunakan diperoleh dari Epi-Lab., Sudan yang terdiri dari 54 atribut dan 714 data pasien. Hasil yang diperoleh menunjukkan algoritme *bayesian network* lebih unggul dibandingkan algoritme *naive bayes* dengan akurasi sebesar *93,2773%*.

Sebuah *review* dilakukan oleh Winarko [16] yang merangkum aplikasi metode *data mining* untuk diagnosis tuberkulosis telah dilakukan dan hasilnya dimana dapat dilihat bahwa dataset yang digunakan pada setiap buku yang dilakukan memiliki perbedaan jumlah atribut atau variabel. Dari hasil *review* tersebut disimpulkan bahwa variabel yang sering digunakan atau yang paling penting dalam mendiagnosis penyakit tuberkulosis adalah *sweating at night, coughing, fever, loss in weight, age,* dan *chest pain.* Sementara performa klasifikasi terbaik diperoleh pada algoritme SVM, bagging, dan

random forest dengan akurasi masing-masing sebesar 98,7%, 98,4%, dan 98,3%.

Bakar dan Febriyani [7] melakukan buku dengan *dataset*nya dari Unit Layanan Kesehatan Mandau, Riau terdiri dari 14 atribut/fitur dari *233* data pasien Dari 14 atribut/fitur diperoleh 8 atribut dari proses pemilihan fitur (*preprocessing*) yaitu jenis kelamin, umur, berat badan, demam, keringat malam, batuk > 3 minggu, dahak berdarah, basil tahan asam (BTA) dengan *preposesing discretization rough set application.* Metode yang digunakan klasifikasi dengan menguji beberapa algoritme klasifikasi. Dari hasil menunjukkan bahwa dengan algoritme *neural network* didapatkan akurasi 90,44 dan algortima *rough set* 92,14% dengan menggabungkan kedua algoritme tersebut menghasilkan algortima *rough set neural network* sebanyak 92,29 %.

B. Teori

1. Penyakit tuberkulosis (TBC)

Penyakit tuberkulosis paru adalah penyakit menular yang menyerang paru-paru,penyakit ini disebabkan oleh bakteri *mycobacterium tuberkulosis*. Cara penularan melalui ludah atau dahak penderita yang mengandung basil tuberculosis paru. Penyakit ini ditemukan oleh Rober Koch pada tahun 1882 [17]. Tuberkolusis disebabkan oleh bakteri berbentuk batang (basil) yang dikenal dengan nama *mycrobacterium tuberculosis* [18]. Pada waktu batuk butir-butir air ludah beterbangan di udara dan terhisap oleh orang yang sehat dan masuk kedalam parunya yang kemudian menyebabkan penyakit tuberkulosis paru (TBParu) [18]. Pada penderita TBC paru dapat memperlihatkan tanda-tanda seperti : batuk berdahak lebih dari tiga minggu, batuk

mengeluarkan darah , dada terasa sakit atau nyeri, terasa sesak pada waktu bernafas [18].

2. Metode konvensional untuk mendiagnosis TBC

Berdasarkan pedoman dan penatalaksanan di Indonesia[19], ada beberapa proses yang dilakukan untuk mendiagnosis adanya *mycrobacterium tuberculosis*, diantaranya:

a. Gejala klinis

Gejala klinis tuberkulosis dapat dibagi menjadi 2 golongan, yaitu gejala lokal dan gejala sistemik. Bila organ yang terkena adalah paru maka gejala lokal ialah gejala respiratori (gejala lokal sesuai organ yang terlibat)

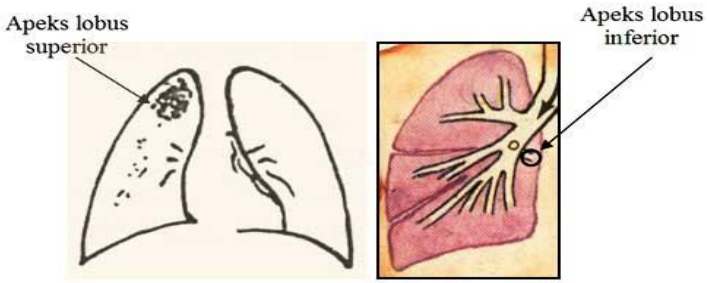
1. Gejala respiratorik
   a. batuk > 3 minggu
   b. batuk darah
   c. sesak napas
   d. nyeridada

Gejala respiratori ini sangat bervariasi, dari mulai tidak ada gejala sampai gejala yang cukup berat tergantung dari luas lesi. Terkadang pasien terdiagnosis pada saat *medical check up.* Bila *bronkus* belum terlibat dalam proses penyakit, inaka pasien mungkin tidak ada gejala batuk. Batuk yang pertama terjadi karena iritasi *bronkus,* dan selanjutnya batuk terjadi untuk membuang dahak ke luar.

2. Gejala sistemik
   a. demam
   b. gejala sistemik lain adalah malaise, keringat malam, anoreksia, dan berat badan menurun

b. Pemeriksaan jasmani

Pada pemeriksaan jasmani, kelainan yang akan dijumpai tergantung dari organ yang terlibat. Pada tuberkulosis paru, kelainan yang didapat tergantung luas kelainan struktur paru. Pada permulaan (awal) perkembangan penyakit umumnya tidak (atau sulit sekali) menemukan kelainan. Pada Gambar 2.1 menunjukan kelainan paru pada umumnya terletak di daerah *lobus superior* terutama daerah apeks dan segmen *posterior*, serta daerah apeks *lobus inferior* [19]. Pada pemeriksaan jasmani dapat ditemukan antara lain suara napas *bronkial, amforik,* suara napas melemah, ronki basah, tanda-tanda penarikan paru, *diafragma* dan *mediastinum.* Pada *pleuritis* tuberkulosis, kelainan pemeriksaan fisis tergantung dari banyaknya cairan di rongga *pleura.* Pada perkusi ditemukan pekak, pada *auskultasi* suara napas yang melemah sampai tidak terdengar pada sisi yang terdapat cairan. Pada limfa denitis tuberkulosis, terlihat pembesaran kelenjar getah bening, tersering di daerah leher kadang-kadang di daerah ketiak. Pembesaran kelenjar tersebut dapat menjadi *"cold abscess".*



Gambar 2.1 Paru: apeks lobus superior dan apeks lobus inferior [19]

c. Diagnosis dengan menggunakan mikroskop

Diagnosis *mycrobacterium tuberculosis* dengan menggunakan mikroskop merupakan teknik sederhana dan tidak membutuhkan biaya yang mahal. Dalam teknik ini digunakan metode pewarnaan Ziehl-Neelsen dengan warna merah untuk melihat bakteri. Dengan teknik ini bakteri dapat dengan jelas dilihat dengan mikroskop, sehingga dapat dengan mudah menghitung koloni dari *mycrobacterium.* Tetapi, beberapa mikroorganisme lain juga bisa dianggap sebagai *mycrobacteriun tuberculosis* [3] dan metode ini juga membutuhkan waktu yang cukup lama dan membutuhkan ahli mikrobiologi dalam proses deteksi [21].

d. **Pemeriksaan radiologis**

Pemeriksaan standar ialah foto thoraks PA. Pemeriksaan lain atas indikasi: foto lateral, top-lordotik, oblik, *CT-Scan.* Pada pemeriksaan foto thoraks, tuberkulosis dapat memberi gambaran bermacam-macam bentuk (*multiform*).

Gambaran radiologi yang dicurigai sebagai lesi TB aktif ialah:

a. Bayangan berawan / nodular di segmen apikal dan *posterior lobus* atas paru dan segmen *superior lobus* bawah
b. Kaviti, terutama lebih dari satu, dikelilingi oleh bayangan opak berawan atau nodular
c. Bayangan bercak milier
d. *Efusi pleura unilateral* (umumnya) atau bilateral (jarang)

Gambaran radiologik yang dicurigai lesi TB aktif ialah:

a. Fibrotik
b. Kalsifikasi

c. *Schwarte atiu* penebalan *pleura.*

Luluh paru *(destroyed Lung*):

a. Gambaran radiologi yang menunjukkan kerusakan jaringan paru yang berat, biasanya secara klinis disebut luluh paru. Gambaran radiologi luluh paru terdiri dari *atelektasis, ektasis/multikaviti* dan *fibrosis parenkim* paru. Sulit untuk menilai aktivitas lesi atau penyakit hanya berdasarkan gambaran radiologi tersebut.

b. Perlu dilakukan pemeriksaan bakteriologi untuk memastikan aktivitas proses penyakit

Luas lesi yang tampak pada foto thorak untuk kepentingan pengobatan dapat dinyatakan sebagai berikut (terutama pada kasus BTA negatif):

a. Lesi minimal

   Bila proses mengenai sebagian dari satu atau dua paru dengan luas tidak lebih dari sela iga 2 depan (volume paru yang terletak di atas *chondrostemal junction* dari iga kedua depan dan prosesus spinosus dari vertebra torakalis 4 atau korpus vertebra torakalis 5), serta tidak dijumpai kaviti.

b. Lesi luas

   Bila proses lebih luas dari lesi minimal.

e. Uji tuberkulin

Uji *tuberculin* [20] adalah salah satu metode yang digunakan untuk mendiagnosis infeksi TB yang dilakukan pada anak Balita. Ini sering digunakan untuk skrening individu dari infeksi laten dan menilai rata-rata infeksi TB pada populasi tertentu. Uji tuberkulin dilakukan untuk melihat seseorang

mempunyai kekebalan terhadap basil TB, sehingga sangat baik untuk mendeteksi infeksi TB.Tetapi uji tuberkulin ini tidak dapat untuk menentukan M.Tb tersebut aktif atau tidak aktif (latent). Oleh sebab itu harus dikonfirmasi dengan ada tidaknya gejala dan lesi pada foto thorak untuk mengetahui seseorang tersebut terdapat infeksi TB atau sakit TB.

Uji tuberkulin merupakan salah satu dasar kenyataan bahwa infeksi oleh M.Tb akan menyebabkan reaksi *delayed-type hypersensitivity* terhadap komponen antigen yang berasal dari ekstrak M.tb atau tuberkulin. Ada 2 perusahaan yang memproduksi tuberkulin (PPD) yaitu PPD dari USA : Parke-Davis (Aplisol) dan Tubersol. PPD yang dipakai ada 2 jenis yaitu PPD-S dibuat oleh Siebert dan Glenn tahun 1939 yang sampai sekarang digunakan sebagai standart Internasional. Sebagai dosis standart adalah Tuberkulin Unit (TU) PPD-S yang diartikan aktivitas uji tuberkulin ini dapat mengekskresikan 0.1 mg/0.1 ml PPD-S. Dosis lain yang pernah dilaporkan adalah dosis 1 dan 250 TU, tetapi dosis ini tidak digunakan karena akan menghasilkan reaksi yang kecil dan membutuhkan dosis yang besar. PPD jika diencerkan dapat diabsorsi oleh gelas dan plastik dalam jumlah yang bervariasi, sehingga untuk menghindarinya didalam sediaan PPD ditambah dengan Tween 80 untuk menghindari sediaan tersebut terabsorbsi. Standart tuberkulin ada 2 yaitu PPD-S dan PPD.RT 23, dibuat oleh Biological Standards Staten, Serum Institute, Copenhagen, Denmark. Dosis standart 5 TU PPD-S sama dengan dosis 1 / 2TU PPD RT 23.4 WHO merekomendasikan penggunaan 1 TU PPD RT 23 Tween 80 untuk penegakan diagnosis TB guna memisahkan terinfeksi TB dengan sakit TB.

Reaksi uji tuberkulin yang dilakukan secara intradermal akan menghasilkan hipersensitiviti tipe IV atau *delayed-type hypersensitivity* (DTH). Masuknya protein TB saat injeksi akan menyebabkan sel T tersensitisasi dan menggerakkan limfosit ke tempat suntikan. Limfosit akan merangsang terbentuknya indurasi dan vasodilatasi lokal, edema, deposit fibrin dan penarikan sel inflamasi ke tempat suntikan.

Reaksi tuberkulin merupakan reaksi DTH. Protein tuberkulin yang disuntikkan di kulit, kemudian diproses dan dipresentasikan ke sel dendritik/Langerhans ke sel T melalui molekul MHC-II.Sitokin yang diproduksi oleh sel T, akan membentuk molekul adhesi endotel. Monositkeluar dari pembuluh darah dan masuk ke tempat suntikan yang berkembang menjadi makrofag.Produk sel T dan makrofag menimbulkan edema dan bengkak. Test kulit positif maka akan tampak edema lokal atau infiltrat maksimal 48-72 jam setelah suntikan.

Uji tuberkulin dilakukan dengan injeksi 0,1 ml PPD secara intradermal dengan metode Mantoux) *divolar* / permukaan belakang lengan bawah. Injeksi tuberkulin menggunakan jarum gauge 27 dan *spuit* tuberkulin, saat melakukan injeksi harus membentuk sudut 30° antara kulit dan jarum. Penyuntikan dianggap berhasil jika pada saat menyuntikkan didapatkan indurasi diameter 6-10 mm. Uji ini dibaca dalam waktu 48-72 jam setelah suntikan. Hasil uji tuberkulin dicatat sebagai diameter indurasi bukan kemerahan dengan cara palpasi. Standarisasi digunakan diameter indurasi diukur secara *transversal* dari panjang axis lengan bawah dicatat dalam milimeter.

Hasil uji tuberkulin negatif dapat diartikan sebagai seseorang tersebut tidak terinfeksi dengan basil TB. Selain itu dapat juga oleh karena terjadi pada saat kurang dari 10 minggu sebelum imunologi seseorang terhadap basil TB terbentuk. Jika terjadi hasil yang negatif maka uji tuberkulin dapat diulang 3 bulan setelah suntikan pertama. Hasil uji tuberkulin yang positif dapat diartikan sebagai seseorang tersebut sedang terinfeksibasil TB. Terpenting disini adalah jika seseorang sedang terinfeksi M.tb apakah sedang terinfeksi atau sakit TB. Sehingga *guideline* ACHA menyebutkan jika hasil uji tuberkulin positif maka harus dikonfirmasikan dengan pemeriksaan fototoraks dan pemeriksaan dahak. Jika hasil fototoraks tersebut normal maka dapat dilakukan pemberian terapi TB laten, tetapi jika hasil fototoraks terjadi kelainan dan menunjukkan ke arahTB maka dapat dimasukkan dalam M.Tb aktif.

Spesiviciti uji tuberkulin dapat berubah menjadi 95-99% tergantung dari prevalensi infeksi bukan TB pada suatu populasi. Jika *spesiviciti* turun akan meningkatkan resiko *cross-reaction*. Curley mendapatkan *spesiviciti* uji tuberkulin meningkat dengan meningkatnya cut off point dengan 15mm. Manuhutu mendapatkan cut off point antara *reactor* dan *non-reactor* 12 mm. Pembacaan uji tuberkulin dilakukan dalam waktu 48-72 jam, tetapi dianjutkan untuk 72 jam. Hasil yang dilaporkan adalah indurasi lokal (bukan kemerahan) dengan palpasi, diameter transversal dan dicatat dalam millimeter. Interpretasi ukuran diameter uji tuberkulin Dengan dasar sensitiviti dan spesivisiti, prevalens iTB masing-masing kelompok dapat dibedakan.Terdapat 3 *cut-off point* yang direkomendasikan untuk mengartikan reaksi uji tuberculin.

Selain dengan uji tuberkulin yang dilakukan pada anak Balita juga dilakukan pada orang dewasa dengan pemeriksaan dahak dengan melakukan Pemeriksaan bakteriologi untuk menemukan kuman tuberkulosis mempunyai arti yang sangat penting dalam menegakkan diagnosis. Bahan untuk pemeriksaan bakteriologi ini dapat berasal dari dahak, cairan *pleura, liquor cerebrospinal*, bilasan *bronkus*, bilasan lambung, kurasan *bronkoalveolar (bronchoalveolar lavage*/BAL), urin, *faeces* dan jaringan biopsi (termasuk biopsi jarum halus/ BJH). Dahak diambil sebanyak 3 kali (SPS), yaitu:

a. Sewaktu / spot (dahak sewaktu saat kunjungan)
b. Pagi (keesokan harinya)
c. Sewaktu / spot (pada saat mengantarkan dahak pagi), atau

f. Pemeriksaan bakteriologi

Pemeriksaan bakteriologi untuk menemukan kuman tuberkulosis mempunyai arti yang sangat penting'dalam menegakkan diagnosis. Pemeriksaan ini dianggap sebagai metode *gold standard* dalam mendiagnosis secara pasti penyakit tuberkulosis paru. Bahan untuk pemeriksaan bakteriologi ini dapat berasal dari dahak, cairan *pleura*, *liquor cerebrospinal*, bilasan bronkus, bilasan lambung, kurasan bronkoal veolar (*bronchoal veolar lavage*/BAL), urin, *faeces* dan jaringan biopsi (termasuk biopsi jarum halus/BJH). Dahak diambil sebanyak 3 kali (SPS), yaitu:

a. Sewaktu / spot (dahak sewaktu saat kunjungan)
b. Pagi (keesokan harinya)
c. Sewaktu / spot (pada saat mengantarkan dahak pagi), atau
d. Setiap pagi 3 hari berturut-turut.

Pemeriksaan Uji Tuberkulin dan pemeriksaan bakteriologi pada dahak dianggap sebagai metode *gold standard* dalam mendiagnosis secara pasti penyakit tuberkulosis paru.

Metode *gold standard* yang menjadi acuan oleh para dokter untuk mendiagnosis penyakit TBC adalah *uji tuberkulin* dan pemeriksaan dahak Meskipun metode ini banyak digunakan oleh dokter, metode ini memiliki beberapa kekurangan yaitu merupakan teknik diagnosis yang invasive, mempunyai resiko bagi pasien, mahal dan waktu yang lama. Oleh karena itu, diusulkan penggunaan metode komputer untuk mendiagnosis penyakit TBC. Keuntungan dari metode komputer untuk mendiagnosis penyakit TBC adalah *non-invasive*, tidak memiliki resiko terhadap pasien, biaya lebih murah dan waktu yang tidak lama jika dibandingkan dengan tuberkulin *dan* pemeriksaan dahak.

3. Metode komputasi cerdas mendiagnosis TBC

Data mining adalah proses penggalian pengetahuan tersembunyi dari data. Metode ini dapat mengungkapkan pola dan hubungan antar sejumlah besar data dalam dataset yang tunggal atau tidak [16]. Metode pemilihan fitur diusulkan dalam banyak buku untuk meningkatkan akurasi pada proses diagnosis Paru-paru. Metode ini dapat digunakan untuk menghapus fitur-fitur yang tidak penting dari suatu dataset.

a. Penambangan data

Pada bagian ini diberikan penjelasan mengenai beberapa teori pada penambangan data yang digunakan pada buku ini. *Data mining* merupakan salah satu alat analisis yang banyak digunakan dalam bidang kesehatan, data mining terbukti dapat

mengungkap informasi berbeda yang tidak teridentifikasi dengan menggunakan metode statistik tradisional [21] Teori ini digunakan sebagai arahan dalam menyelesaikan masalah pada buku.

b. Diskretisasi

Diskretisasi merupakan proses konversi data yang bertipe numerik menjadi tipe data diskret atau kategori. *Boolean reasoning* merupakan metode diskretisasi yang digunakan pada buku ini dan termasuk kategori *supervised* [22]. Proses diskretisasi dengan metode *boolean reasoning* adalah mencari himpunan *basic cuts* dari sebuah fitur. Misalkan *A* adalah himpunan fitur dan *a* adalah fitur bertipe bilangan *real*, $a \in A$. Misalkan nilai-nilai fitur a adalah $Va = \{ v_1^a, V_2^a, ..., v_n^a \}$. Maka, himpunan *basic cuts A* dapat didefinisikan seperti persamaan (2-1) [23] [24]

$$Ba = \left\{ \left(a, \left(\frac{v_1^a + v_2^a}{2} \right) \right), \left(a, \left(\frac{v_2^a + v_3^a}{2} \right) \right), \left(a, \left(\frac{v_{n-1}{}^a + v_n^a}{2} \right) \right) \right\} \qquad (2\text{-}1)$$

Himpunan *cuts* dapat direduksi dengan menggunakan algoritme Johnson [25] . Misalkan P_i^a merupakan partisi dimana P_i^a adalah interval dari nilai $\left[v_n^a, v_{n+1}{}^a \right)$ Berdasarkan Komorowski, dkk. [26], tabel sistem informasi dari objek yang dapat dibedakan *(discern)* dan partisi dapat dibentuk dengan menggunakan format seperti Tabel 2.1

Tabel 2.1 Tabel sistem informasi dan partisi

		p_1^a	p_2^a	...	p_n^a	D
1	(x_i,x_j)	v_{11}	v_{12}	...	v_{1a}	1
...	...	...	...	...	...	...
M	(x_p,x_q)	v_{ml}	v_{m2}	...	v_{mn}	1

Pasangan objek (x_i, x_j),..., (x_p, x_q) memiliki nilai fitur keputusan yang berbeda. Kolom p_i^a, p_2^a,..., p_n^a merupakan seluruh kemungkinan partisi yang dibentuk. Nilai Vmn adalah satu ketika interval partisi berada dalam jarak nilai minimum dan maksimum antara xi dan xj, nilai vmn adalah nol jika interval partisi berada di luar jarak nilai minimum dan maksimum antara xi dan xj. Langkah-langkah untuk mereduksi jumlah partisi tanpa mengabaikan konsistensi tabel keputusan seperti yang dijelaskan oleh Setiawan [27] dan Komorowski, dkk [26]. dapat dilihat sebagai berikut :

1. Definisikan R={ }.
2. Pilih satu partisi p_i^a yang mengandung jumlah nilai 1 paling banyak, kemudian tambahkan ke R.
3. Hapus semua baris yang memiliki nilai p_i^a sama dengan 1.
4. Jika semua baris telah dihapus, stop. Jika tidak, kembali ke langkah kedua.

Himpunan R yang diperoleh merupakan *reduct* dari himpunan partisi. Himpunan partisi ini yang digunakan untuk melakukan proses diskretisasi.

c. Pemilihan fitur berbasis komputer

Dataset buku seringkali memiliki dimensi yang besar dan memiliki fitur-fitur yang tidak relevan yang dapat mengakibatkan performa *algoritme* menurun dan sulit dipahami [28] Pemilihan fitur merupakan tahapan praproses klasifikasi yang dilakukan dengan cara memilih fitur-fitur yang relevan yang mempengaruhi hasil klasifikasi. Tujuan pemilihan fitur untuk mengurangi dimensi data dan fitur-fitur yang tidak

relevan, guna meningkatkan efektifitas dan efisiensi kinerja dari algoritme klasifikasi.

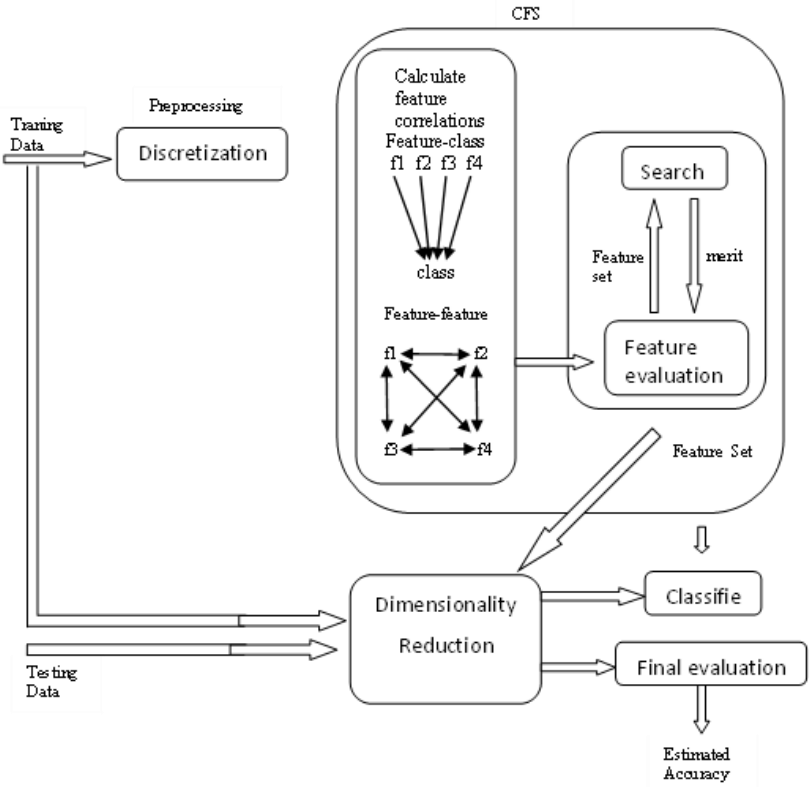
Tujuan dari pemilihan fitur adalah mengidentifikasi fitur yang signifikan, mengeliminasi fitur yang tidak relevan dan membangun model pembelajaran yang baik. Keuntungan dengan adanya pemilihan fitur adalah penurunan waktu komputasi pada algoritme induksi dan dapat juga meningkatkan nilai akurasi

Pada buku ini akan diusulkan pemanfaatan metode pemilihan fitur *correlation based feature selection* (CFS) dan pemilihan fitur berbasis *rough set theory*. Berikut ini beberapa penjelasan mengenai pemilihan fitur tersebut.

d. *Correlation based feature selection*

Pemilihan berbasis *correlation based feature selection* (CFS) berdasarkan pemikiran bahwa suatu fitur yang baik adalah fitur-fitur yang relevan terhadap kelas tapi tidak *redundant* terhadap fitur-fitur relevan yang lain. Suatu fitur yang relevan untuk dipilih adalah yang memiliki tingkat korelasi tinggi dengan kelas dan korelasi antar fitur yang rendah. Yu dan Liu melakukan dua pendekatan dengan mengukur korelasi antara dua variabel acak yaitu berdasar pada *classical linear* dan berdasar pada teori informasi. CFS menghitung dari nilai atribut yang bertipe nominal sehingga pada nilai atribut yang bersifat numerik perlu didiskretisasi. CFS mengevaluasi subset dari himpunan fitur dengan mempertimbangkan kemampuan prediktif dan tingkat redundansi dari masing-masing fitur. Sebuah subset fitur yang baik adalah subset fitur yang berisi fitur-fitur yang sangat berkorelasi dengan kelas dan masing-masing fitur tidak berkorelasi[29].

Berikut pada Gambar 2.2 dapat di lihat langkah dari algoritme *correlation based feature selection* [31].



Gambar 2.2 Langkah-langkah pada algorima correlation based feature selection

Dapat dilihat dari Gambar 2.2. langkah pertama yang dilakukan adalah *dataset* terlebih dahulu dikenai proses diskretisasi. Langkah selanjutnya adalah menghitung korelasi antar fitur dan korelasi antara fitur dengan kelas. Kemudian dicari subset fitur yang memiliki nilai merit tertinggi. Subset fitur dengan nilai merit tertinggi akan digunakan untuk

mereduksi *dataset.* Nilai merit yang diperoleh dapat dihitung menggunakan persamaan [29] (2-2).

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{2-2}$$

Dimana M_S adalah nilai kontribusi subset fitur S dengan fitur sebanyak k terhadap hasil keputusan. Kemudian untuk setiap $(f \in S)$, $\overline{r_{cf}}$ adalah rata-rata korelasi antara fitur-fitur dengan kelas dan $\overline{r_{ff}}$ adalah rata-rata korelasi antar fitur. Poin utama metode ini adalah kemampuan heuristik untuk mengevaluasi nilai atau kinerja dari subset fitur. Kemampuan heuristik ini memperhitungkan kegunaan fitur individu untuk memprediksi label kelas bersama dengan tingkat interkorelasi antar kelas. Fitur subset yang baik berisi fitur sangat berkorelasi dengan kelas yang belum berkorelasi satu sama lain. Misalkan akan dilakukan proses pemilihan fitur menggunakan metode *correlation based feature selection* (CFS) pada dataset berikut [29].

Tabel 2.2 Dataset Golf

Outlook	**Temp**	**Humidity**	**Wind**	**Play(class)**
Sunny	hot	high	false	No
Sunny	hot	high	true	No
Overcast	hot	high	false	Yes
Rainy	mild	high	false	Yes
Rainy	cool	normal	false	Yes
Rainy	cool	normal	true	No
Overcast	cool	normal	true	Yes

Outlook	Temp	Humidity	Wind	Play(class)
Sunny	mild	high	false	No
Sunny	cool	normal	false	Yes
Rainy	mild	normal	false	Yes
Sunny	mild	normal	true	Yes
Overcast	mild	high	true	Yes
Overcast	hot	normal	false	Yes
Rainy	mild	high	true	No

Tahap 1

Hitung korelasi antara fitur-fitur dan fitur-kelas dengan menggunakan symmetrical uncertainty [29]

$$SymU(Class \mid Feature) = 2 \times \left[\frac{gain(Class \mid Feature)}{H(Class) + H(Feature)} \right] \quad (2\text{-}3)$$

Dimana $H(Class)$ entropi dari Class, $H(Feature)$ adalah entropi dari *Feature*, dan $gain(Class \mid Feature)$ adalah information gain yang diperoleh dari *Feature* dan *Class*.

$$H(Class) = -\sum_{y \in Class} p(y) \log_2(p(y))$$

$$H(Feature) = -\sum_{x \in Feature} p(x) \log_2(p(x))$$

$$H(Class \mid Feature) = -\sum_{x \in Feature} p(x) \sum_{y \in Class} p(y \mid x) \log_2(p(y \mid x))$$

$$gain(Class \mid Feature) = H(Class) - H(Class \mid Feature) \quad (2\text{-}4)$$

Tabel 2.3 menunjukkan semua hasil korelasi antara fitur-fitur dan fitur-kelas dengan menggunakan *symmetrical uncertainty*.

Tabel 2.3 Matriks Korelasi antara fitur-fitur dan fitur-kelas

	Outlook	Temp	Humidity	Wind	Play (class)
Outlook	1	0,116	0,022	0,007	0,130
Temp		1	0.248	0,028	0,025
Humidity			1	0,000	0.185
Wind				1	0,081

T**ahap 2**

Cari subset fitur yang memiliki nilai merit tertinggi dengan menggunakan Best First (arah pencarian forward). Dengan formula merit adalah sebagai berikut [29].

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{2-5}$$

Dimana M_S adalah nilai kontribusi subset fitur S dengan fitur sebanyak k terhadap hasil keputusan. Kemudian untuk setiap $(f \in S)$, $\overline{r_{cf}}$ adalah rata-rata korelasi antara fitur-fitur dengan kelas dan $\overline{r_{ff}}$ adalah rata-rata korelasi antar fitur. Tabel 2.4 menunjukkan pencarian subset fitur dengan menggunakan Best First.

Tabel 2.4 Pencarian subset fitur dengan menggunakan Best First

Feature Set	k	$\overline{r_{cf}}$	$\overline{r_{ff}}$	Merit
[]	0	N/A	N/A	0
[Outlook]	1	0,130	1	0,130
[Temp]	1	0,025	1	0,025
[Humidity]	1	0,185	1	0,185

[Wind]	1	0,081	1	0,081
[**Outlook, Temp]**	2	0.158	0,022	0,220
[Temp, Humidity]	2	0,105	0,258	0,133
[Humidity, Wind**]**	2	0,133	0,0	0.175
[Outlook, Temp, Humidity]	3	0,113	0,132	0,1898
[Outlook, Humidity, Wind]	3	0,132	0,0096	0,226
[Outlook, Temp, Humidity Wind]	4	0,105	0,0718	0,191

Penjelasan mengenai Tabel 2.4 dapat dilihat pada langkah-langkah berikut:

1. Pencarian dimulai dari set kosong yang memiliki nilai merit 0.
2. Tambahkan setiap fitur kepada set kosong dan evaluasi nilai merit yang tertinggi **[Humidity].**
3. Tambahkan Humidity kepada set kosong sehingga diperoleh subset fitur **[Humidity].**
4. Tambahkan fitur Wind kepada subset [Humadity] dan evaluasi nilai merit yang tertinggi **[Humidity, Wind].**
5. Tambahkan fitur Wind kepada subset [Humadity] sehingga diperoleh subset fitur baru yaitu **[Humidity, Wind].**
6. Tambahkan fitur Outlook kepada subset [Humidity, Wind] dan evaluasi nilai merit yang tertinggi **[Outlook, Humidity, Wind**].
7. Tambahkan fitur Outlook kepada suset [Humadity, Wind] sehingga diperoleh sebset fitur baru yaitu **[Outlook, Humidity, Wind**].

8. Tambahkan fitur Temp kepada subset [Outlook, Humadity, Wind] sehingga diperoleh sebset fitur baru yaitu **[Temp, Outlook, Humidity, Wind**].
9. Subset fitur [**Temp, Outlook, Humidity, Wind**] mengalami penurunan nilai merit.
10. Lakukan proses *backtracking* kepada subset fitur sebelumnya yang telah dievaluasi.
11. Subset fitur **[Outlook, Humidity, Wind]** memiliki nilai merit yang tertinggi.
12. Pilih subset **[Outlook, Humidity, Wind]** sebagai hasil pemilihan fitur.
13. End.

Pada buku ini digunakan metode *attribute selection* berbasis komputer yang akan meningkatkan kemampuan metode klasifikasi untuk diagnosis penyakit TBC.

a) k-*Fold Cross Validation*

Cross validation (CV) merupakan teknik untuk mengestimasi performa dari model prediksi. Pada saat jumlah data untuk *testing* dan *training* terbatas maka *cross validation* cukup baik untuk diterapkan [27]. Tujuan pengujian menggunakan CV adalah untuk menurunkan bias pada perhitungan performa [30]. Pada pengujian dengan CV data akan dibagi menjadi sejumlah k-*fold* secara acak dengan jumlah yang sama besar atau mendekati sama[31] [32]. Nilai k pada CV belum ada ketentuan seberapa besar yang dapat memberikan performa pengujian yang terbaik [31] sehingga nilai k dapat ditentukan oleh penulis. Penulisan tentang nilai k pada k-*fold* CV telah dilakukan oleh sejumlah penulis antara lain [38] tentang perbandingan performa nilai k = 3 dan k = 5, hasil

menunjukan perbedaan nilai k tersebut tidak memberikan manfaat signifikan secara statistik. Breiman telah melakukan pengujian nilai k = 5 dan k = 10 pada CV dan memberikan hasil bahwa perbedaan nilai k tersebut tidak memberikan keuntungan pada performa[33]. Penentuan nilai k pada CV akan mempengaruhi nilai bias dan *variance*, semakin besar nilai k akan menurunkan bias dan *variance*[32][34].

Training dan *testing* pada CV akan dilakukan sebanyak k kali, sehingga dalam satu pengujian menggunakan *cross validation* tiap *fold* akan menjadi *testing* sekali dan menjadi *training* k-1 kali. Pembagian *fold* secara *general* harus memperhatikan bahwa masing-masing kelas pada *full dataset* dapat terwakili baik dalam *testing* maupun *training* data. Proses pembentukan *fold* akan dilakukan secara acak. Pada *stratified* CV sebisa mungkin rasio antar kelas pada *dataset* lengkap sama dengan rasio pada tiap *fold.*

Laporan dari performa klasifikasi dapat dengan melakukan 10-*fold* cross-*validation* pada dataset yang disediakan. Namun, metode ini menimbulkan bias terhadap pelatihan data dan mungkin tidak merefleksikan performa yang diharapkan ketika diaplikasikan pada data pada kehidupan nyata. Oleh karena itu, di samping pada umumnya digunakan 10-*fold cross-validation*, pada buku ini juga dilakukan *train-test split* pada dataset dan kemudian menggunakan 10-*fold cross-validation* untuk memilih parameter terbaik untuk pelatihan.

b) *Attribute selection*

Pada bagian ini dijelaskan *attribute selection* yang akan digunakan pada buku ini yang dipilih adalah *BestFirst.*

c) Pemilihan Fitur BerbasisTeori *Rough Set*

Teori *rough set* pertama kali diperkenalkan oleh Zdzislaw Pawlak pada tahun 1980-an. *Rough set* menawarkan metode matematis untuk menemukan pola tersembunyi pada suatu data dan dapat digunakan sebagai klasifikasi untuk menemukan hubugan struktural dalam data yang tidak lengkap atau *noisy* [35], *[24]*.

d) Sistem informasi

Information system merupakan sebuah tabel, setiap bans pada tabel merepresentasikan sebuah objek dan setiap kolom merepresentasikan sebuah atribut. Secara lebih formal sistem informasi merupakan pasangan (*U* , *A)*, dimana *U* adalah himpunan berhingga dari objek disebut *universe* dan *A* adalah adalah himpunan berhingga dari fitur, sehingga $f_a = U \times A \rightarrow V_a$ untuk setiap $a \in A$. Dimana V_a adalah domain nilai a [26].

Tabel 2.5 Tabel sistem informasi

U	a_1	a_2
x_1	20-30	40
x_2	15-35	20
x_3	10-15	50
x_4	10-15	50
x_5	15-35	20
x_6	20-3	50
x_7	15-35	40

Tabel 2.5 memperlihatkan contoh dari sistem informasi dimana $U = \{ x_1, x_2 ... x_7\}$ dan $A=\{a_1, a_2 \}$.

Dalam banyak aplikasi, hasil klasifikasi dapat dinyatakan dalam suatu atribut tambahan yang dinamakan dengan *decision attribute.* Atribut ini merepresentasikan hasil dari klasifikasi yang diketahui. Sistem informasi semacam ini disebut dengan sistem keputusan atau *desicion system* (DS) [26].

Sistem keputusan adalah sistem informasi dengan *(U,A∪{d})* dimana *U* adalah himpunan berhingga dari objek, *A* adalah himpunan berhingga dari suatu fitur (fitur kondisi), dan *d* adalah fitur keputusan dengan $d \in A$. Tabel 2.6 memperlihatkan refresentasi dari sistem keputusan dimana setiap objek memiliki satu buah keputusan

Tabel 2.6 Sistem keputusan

U	a_1	a_2	*D*
x_1	20-30	40	No
x_2	15-35	20	Yes
x_3	10-15	50	No
x_4	10-15	50	Yes
x_5	15-35	20	Yes
x_6	20-30	50	No
x_7	15-35	40	Yes

e) *Indiscernibility relation*

Misalkan S = $(\cup. A)$ adalah sistem informasi. Untuk setiap B $\subseteq$ A indiscerni bility relation pada himpunan atribut B di tunjukan pada persamaan (2-9) [26].

$IND_s(B)=\{(x.x')\in\cup^2|\forall a \in B.a(x')\}$

Jika $(x.x') \in \mathrm{IND}_s(B)$, objek x dan x' pada atribut anggota B tidak dapat dibedakan (*indiscernible*). Kelas ekuivalen untuk *indiscernibility relation* pada himpunan atribut B adalah $[x]_s$, Contoh *indiscernibility* pada Tabel 2.6 adalah sebagai berikut.

$$\mathrm{IND}(\{a_1\}) = \{\{x_1, x_2, x_6\}\{x_3, x_4\},,\{x_5, x_7\}\}, \quad (2\text{-}6)$$

$$\mathrm{IND}(\{a_2\}) = \{\{x_1\},\{x_2\},\{x_3, x_4\},\{x_5, x_6, x_7\}\}, \quad (2\text{-}7)$$

$$\mathrm{IND}(\{a_1, a_2\}) = \{\{x_1\},\{x_2\},\{x_3, x_4\},\{x_5 x_7\}\{x_6\}\}. \quad (2\text{-}8)$$

f) *Set aproximation*

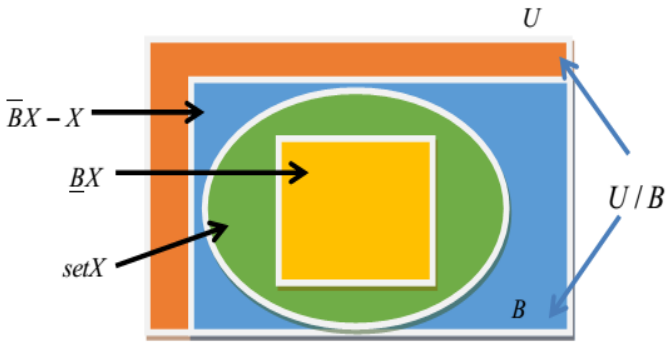
Ada dua konsep dasar dalam teori *rough set* sebagai analisis, yaitu *lower* dan *upperapproximation. Lower approximation* adalah semua objek yang termasuk ke dalam himpunan, dinotasikan dengan $\underline{B}X$, dan *Upper approximation* adalah semua objek yang mungkin termasuk ke dalam himpunan atau gabungan dari *elementary set*, yang dinotasikan dengan $\overline{BX}$.

Misalkan $T = (\cup.A).B \subseteq A$. dan $X \subseteq \cup$ [36]. Himpunan objek X dapat diperkirakan dengan menggunakan *lower* dan *upper approximation*. Persamaan (2-9) dan (2-10) menunjukan *lower* dan upper approximation.

Persamaan (2-9) dan (2-10) menunjukkan *lower* dan *upper approximation* [36].

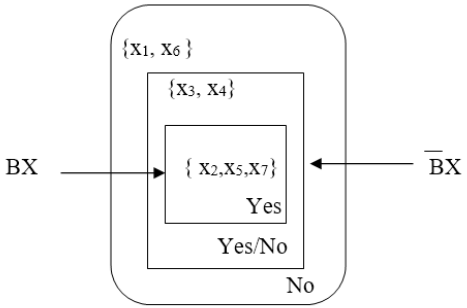
$$\underline{B}X = \cup\{x|[x]_B \subseteq X\}, \quad (2\text{-}9)$$

$$\overline{BX} = \cup\{x|[x]_B \cap X \neq \emptyset\} \quad (2\text{-}10)$$



Gambar 2.3 Ilustrasi himpunan *lower* dan *upper aproximation*

Outside region B dari himpunan *X* adalah ($U - \overline{B}X$) adalah himpunan objek yang tidak terdapat pada himpunan X dan *boundary region* dari *X* didefenisikan sebagai $BN_B(X) = \overline{B}X - \underline{B}X$. Jika *boundary region* tidak kosong, himpunan ini dinamakan *rough set.* Dari Tabel 2.6, jika $B = \{x \setminus d(x) = yes\}$, *Approximation region* dengan kondisi atribut *B* adalah:



Gambar 2.4 Ilustrasi *set approximation*

Berdasarkan Gambar 2.4 diketahui bahwa:

$\overline{B}X = (x_2, x_5, x_7]$

$\underline{B}X = [x_2, x_3, x_4, x_5, x_7 \}$

$\underline{B}N_BX = \{ x_3, x_4,\}$

$U - \overline{B}X = \{ x_1, x_6 \}$ (2-11)

g) *Reduct*

Suatu sistem keputusan sangat mungkin terdapat redudansi, sehingga diperlukan proses *reduct* (reduksi). Suatu *reduct* adalah kombinasi dari kondisi yang membedakan bahwa objek sama baiknya dengan seluruh kondisi yang digunakan, dimana *reduct* dapat dihitung menggunakan matrik *discernibility.*

Misalkan T = (U, C, D) adalah tabel keputusan dengan (U = $\{u_1 u_2, ... u_n\}$, maka *discernibility matrix M(T)* yang merupakan matriks dengan ukuran *n x n* didefinisikan sebagai berikut [37]:

$$m_{ij} = \begin{cases} \{c \in C : c(u_i) \neq c(u_j)\} & if \ni d \in D[d(u_i) \neq (u_i)] \\ \lambda & if \forall d \in D[d((u_i) = ((u_i)] \end{cases} \quad (2\text{-}12)$$

Untuk $i,j = 1,2,...,n$ sedemikian sehingga u_i,atau u_j termasuk kedalam C positive *region* dari fitur keputusan D. *mij* merupakan set dari aribut kondisi yang mengklasifikasikan objek u_i dan u_j ke dalam kelas yang berbeda. Kemudian jika *mij* $=\lambda$, maka perbandingan antara dua objek tidak diperhatikan. Untuk setiap u_i €U , maka fungsi *discernibility* adalah [37]:

$$f_T(u_i) = \bigwedge_j \{\vee m_{ij} : j \neq i, j \in \{1,2,...n\} \} \quad (2\text{-}13)$$

Dimana :

1. $\vee m_{ij}$ adalah disjungsi semua variabel *a* sedemikian sehingga $a \in m_{ij}$, jika $m_{ij} \neq \phi$
2. $\vee m_{ij}$ = *false*, jika $m_{ij} \neq \phi$
3. $\vee m_{ij}$ = *true*, jika $\vee m_{ij} = \lambda$

Contoh tabel keputusan *T* = (U,C,D)) dapat dilihat pada Tabel 2.7. seperti berikut:

Tabel 2.7 Tabel keputusan *T = (U,C,D)*

	A	*b*	*C*	*D*
U_1	a_0	b_1	c_1	1
U_2	a_1	b_1	c_0	0
U_3	a_0	b_2	c_1	0
U_4	a_1	b_1	c_0	1

Dari persamaan (2-13) didapatkan *discernibility matrix* seperti Tabel 2.8 berikut ini :

Tabel 2.8 Matriks *discernibility*

	u_1	u_2	u_3
u_1	a,c		
u_3	b	λ	
u_4	λ	c	a,b

Berdasarkan fungsi *discernibility* pada persamaan (2-13), diperoleh $f_r = (a \vee c)(\wedge b \wedge c \wedge (a \vee b) = b \wedge c$. Oleh karena itu, reduct yang diperoleh adalah *{b, c}.*

e. Pemilihan fitur berbasis pakar medis

Pemilihan fitur berbasis pakar medis atau *motivated feature selection* (MFS) didasarkan pada pengetahuan yang dimiliki oleh pakar medis.

f. Algoritme klasifikasi

Klasifikasi merupakan teknik utama dalam *data mining* dan banyak digunakan diberbagai bidang [38]. Teknik

klasifikasi digunakan untuk memprediksi dataset ke dalam kelompok tertentu. Pada buku ini teknik klasifikasi digunakan untuk mendiagnosis penyakit tuberkulosis ke dalam kelas positif diberi label 'yes', yaitu pasien yang didiagnosis mengidap penyakit tuberkulosis, dan kelas negatif diberi label 'no', yaitu pasien yang tidak mengidap penyakit tuberkulosis. Algoritme klasifikasi yang digunakan pada buku ini dibagi ke dalam beberapa jenis algoritme klasifikasi diantaranya adalah *naive bayes, k-Star, adaboostM1, PART dan J.48.*

g. Naive bayes

Classifier naive bayes adalah *classifier probabilistic* sederhana yang berdasarkan pada penerapan teorema *bayes* dengan asumsi independen yang kuat. Dengan kata lain, *classifier naive bayes* mengansumsikan bahwa keberadaan sebuah atribut (variabel) tidak ada kaitannya dengan keberadaan atribut (variabel) yang lain. *naïve bayes* merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai teorema Bayes**.** Teorema tersebut dikombinasikan dengan **"***naive***"** dimana diasumsikan kondisi antar atribut saling bebas.

Pada sebuah *dataset*, setiap baris/dokumen *I* diasumsikan sebagai *vector* dari nilai-nilai atribut $<x_1,x_2,\ldots,x_3>$ dimana tiap nilai-nilai menjadi peninjauan atribut X_i ($i \in [1,n]$)). Setiap baris mempunyai label kelas $c_i \in \{c_1,c_2,\ldots,c_k\}$ sebagai nilai variabel kelas C, sehingga untuk melakukan klasifikasi dapat dihitung nilai probabilitas $p(C=c_i|X=x_j)$, dikarenakan pada *naïve bayes* diasumsikan setiap atribut saling bebas, maka persamaan yang didapat adalah peluang $p(C=c_i|X=x_j)$ menunjukkan peluang

bersyarat atribut Xi dengan nilai xi diberikan kelas c, dimana dalam *naïve bayes*, kelas C bertipe kualitatif sedangkan atribut Xi dapat bertipe kualitatif ataupun kuantitatif [28].

Teorema *bayes* memiliki bentuk umum pada persamaan (2-14) [39].

$$P(H|K) = \frac{P(X|H)P(H)}{P(X)} \qquad (2 - 14)$$

Keterangan:

X = Data dengan *class* yang belum diketahui

H = Hipotesis data X merupakan suatu class spesifik

P(H|X) = Probabilitas hipotesis H berdasar kondisi X (*posteriori probability*)

P(H) = Probabilitas hipotesis H *(prior probability)*

P(X|H) = Probabilitas X berdasar kondisi pada hipotesis H

P(X) = Probabilitas dari X

Berikut ini contoh perhitungan berdasarkan algoritme *naive bayes* untuk data yang tercantum pada Tabel 2.9.

Tabel 2.9 Data *training all electronics customer database*

Id	***Age***	***Income***	***Student***	***Credit_rating***	***Class: buys_computer***
1	<= 30	*high*	*No*	*Fair*	*No*
2	<= 30	*high*	*No*	*Excellent*	*No*
3	31…40	*high*	*No*	*Fair*	*Yes*
4	>40	*medium*	*No*	*Fair*	*Yes*
5	>40	*Low*	*Yes*	*Fair*	*Yes*
6	>40	*Low*	*Yes*	*Excellent*	*No*
7	31…40	*Low*	*Yes*	*Excellent*	*Yes*
8	<= 30	*medium*	*No*	*Fair*	*No*
9	<= 30	*Low*	*Yes*	*Fair*	*Yes*

10	>40	*medium*	*Yes*	*Fair*	*Yes*
11	<= 30	*medium*	*Yes*	*Excellent*	*Yes*
12	31…40	*medium*	*No*	*Excellent*	*Yes*
13	31…40	*high*	*Yes*	*Fair*	*Yes*
14	>40	*medium*	*No*	*Excellent*	*No*

Terdapat dua *class* pada *decision table* yang dibentuk yaitu:
C1 = *buys computer = yes*
C2 = *buys computer = no*
Misal terdapat data X dan belum diketahui kategori *class*.
X = (*age* = ≤ 30, *income= medium, student= yes, credit_rating= excellent).*
Penyelesaian:
Dibutuhkan untuk memaksimalkan $P(X|C_i)$ $P(C_i)$ untuk i=1,2. $P(C_i)$ merupakan *prior probability* untuk setiap *class* berdasarkan data contoh:
P(*buys_computer="yes")* = 9/14 = 0,643
P(*buys_computer="yes")* = 5/14 = 0,357
Menghitung $P(X|C_i)$, untuk i = 1,2
P(*age*= ≤30 | *buys_computer= yes*) = 2/9 = 0,222
P(age= ≤30 | *buys_computer= no*) = 3/5 = 0,600
P(*income*= medium | *buys_computer= yes*) = 4/9 = 0,444
P(*income*= medium | *buys_computer= no*) = 2/5 = 0,400
P(*student= yes* | *buys_computer= yes*) = 6/9 = 0,667
P(*student= yes* | *buys_computer= no*) = 1/5 = 0,200
P(*student= yes* | *buys_computer= yes*) = 6/9 = 0,667
P(*c_rating= excellent* | *buys_computer= yes*) = 3/5 = 0,600
P(*c_rating= excellent* | *buys_computer= no*) = 3/9 = 0,333
P(X|*buys_computer = yes*) = 0,222 x 0,444 x 0,677 x 0,600 = 0,040
P(X|*buys_computer = no*) = 0,6 x 0,4 x 0,2 x 0,33 = 0,016

P(X|*buys_computer* = *yes*) P(*buys_computer=”yes”)*
= 0,044 x 0,643 = 0,0283
P(X|*buys_computer*= *yes*) P(*buys_computer=”yes”)*
= 0,016 x 0,357 = 0,0057
Berdasarkan hasil perhitungan diatas maka dapat disimpulkan bahwa X masuk kedalam *class buys_computer= “yes”.*

h. *K-Star*

Algoritme K-Star dapat digambarkan sebagai algoritme pembelajaran yang menggunakan *entropy* sebagai ukuran jarak. Fungsi dari algoritme *k-Star* adalah menyediakan pendekatan yang konsisten untuk menangani real atribut, *symbolic* atribut dan *missing values*. *k-Star* mirip dengan *k-nearest neighbour* (K-NN). Misalnya terdapat data baru, x, maka akan diklasifikasikan ke dalam kelas yang paling sering terjadi. Jarak *entropy* kemudian digunakan untuk mendapatkan data yang paling mirip dengan kumpulan data [40]. Untuk setiap data baru, dipilih nilai *entropy* tertinggi sebagai hasil klasifikasi. Nilai *entropy* K dapat dihitung dengan persamaan(2-15) [35].

$$K^*(b \mid a) = -\log_2 P^*(b \mid a) \qquad (2\text{-}15)$$

Dengan $P^*(b \mid a)$ adalah penjumlahan probabilitas data *a* yang merupakan kelas dari *b*.

i. *AdaboostMl*

Boosting bekerja dengan berulang kali menjalankan *weak learning algorithm* pada berbagai distribusi data latih, lalu menggabungkan kelas-kelas yang dihasilkan oleh *weak learner* menjadi kelas gabungan tunggal [41]. Pada dasarnya, metode *boosting* dapat meningkatkan ketelitian dalam proses klasifikasi dan prediksi dengan cara membangkitkan kombinasi dari suatu

model, tetapi hasil klasifikasi atau prediksi yang dipilih adalah model yang memiliki nilai bobot paling besar. Jadi, setiap model yang dibangkitkan memiliki atribut berupa nilai bobot. Ide *boosting* yang paling populer adalah metode yang disebut sebagai *adaboost* yang diperkenalkan oleh Freund dan selanjutnya diberi nama *adaBoostMl* oleh Freund dan Schapire [42]. Tabel 2.10. menunjukkan algoritme adaBoostMl.

Tabel 2.10 Algoritme *AdaBoostMl* [41]

Pembangunan Model
Tetapkan bobot yang sama untuk setiap contoh pelatihan. Untuk setiap iterasi *t:* Terapkan algoritme *learning* kepada *dataset* berbobot dan simpan model yang dihasilkan. Hitunglah *e (error)* dari model pada *dataset* berbobot dan simpan error. Jika *e* sama dengan nol atau *e* besar atau sama dengan 0,5:
Hentikan model generation. Untuk setiap *instance* dalam *dataset:* Jika *instance* diklasifikasikan dengan benar oleh model: Kalikan bobot *instance* dengan *e/l-e* . Normalisasi bobot semua *instance.*
Klasifikasi
Tetapkan bobot nol kepada semua kelas. Untuk setiap (atau kurang) model *t:* Tambahkan-log(e/l -e) ke bobot kelas yang diperkirakan oleh model. Kembali ke kelas dengan bobot tertinggi.

Berdasarkan Tabel 2.10, nilai *error* dapat dihitung dengan persamaan 2-11 *dengan*

$$Error\ (e) = \sum_{j}^{d} wj \ \ x \ \ error \ \ (X_j) \qquad (2\text{-}16)$$

Dimana *error* (X_j) merupakan *misclassified* error dari tuple X_j.

j. J48

Salah satu algoritme induksi pohon keputusan yaitu ID3 *(iterative dichotomiser 3)* yang dikembangkan oleh J. Ross Quinlan [28]. Prosedur algoritme ID3, input berupa sampel *training*, *label training* dan atribut. Algoritme C4.5 merupakan pengembangan dari ID3. Sedangkan pada perangkat lunak *open source* WEKA dikenal sebagai J48.

Pohon dibangun dengan cara membagi data secara rekursif hingga tiap bagian terdiri dari data yang berasal dari kelas yang sama. Bentuk pemecahan (*split*) yang digunakan untuk membagi data tergantung dari jenis atribut yang digunakan dalam *split*. Algoritme C4.5 dapat menangani data numerik/kontinyu dan *diskret*. *Split* untuk atribut numerik yaitu mengurutkan contoh berdasarkan atribut kontiyu A, kemudian membentuk minimum permulaan (*threshold*) M dari contoh-contoh yang ada dari kelas mayoritas pada setiap partisi yang bersebelahan, lalu menggabungkan partisi-partisi yang bersebelahan tersebut dengan kelas mayoritas yang sama.

Split untuk atribut diskret A mempunyai bentuk *value* (A) ε X dimana X $\subset$ *domain*(A). Jika suatu set data mempunyai beberapa pengamatan dengan *missing values* yaitu *record* dengan beberapa nilai variabel tidak ada, jika jumlah pengamatan terbatas maka atribut dengan *missing values* dapat diganti dengan nilai rata-rata dari variabel yang bersangkutan [39]. Pemisahan obyek (*split*) dilakukan tes terhadap atribut dengan mengukur tingkat ketidakmurnian pada sebuah simpul (*node*).

Pada algoritme C.45 menggunakan rasio perolehan (*gain ratio*). Sebelum menghitung rasio perolehan, perlu menghitung dulu nilai informasi dalam satuan *bit* dari suatu kumpulan objek.

Cara menghitungnya dilakukan dengan menggunakan konsep entropi. Berikut ini contoh kasus proses penghitungan secara matematis pada algoritme C 4.5 dapat dilihat pada Tabel 2.11.

Tabel 2.11 Keputusan Bermain Tenis

No	***Outlook***	***Temperature***	***Humidity***	***Windy***	***Play***
1	*Sunny*	*Hot*	*High*	*False*	*No*
2	*Sunny*	*Hot*	*High*	*True*	*No*
3	*Cloudy*	*Hot*	*High*	*False*	*Yes*
4	*Rainy*	*Mild*	*High*	*False*	*Yes*
5	*Rainy*	*Cool*	*Normal*	*False*	*Yes*
6	*Rainy*	*Cool*	*Normal*	*True*	*Yes*
7	*Cloudy*	*Cool*	*Normal*	*True*	*Yes*
8	*Sunny*	*Mild*	*High*	*False*	*No*
9	*Sunny*	*Cool*	*Normal*	*False*	*Yes*
10	*Rainy*	*Mild*	*Normal*	*False*	*Yes*
11	*Sunny*	*Mild*	*Normal*	*True*	*Yes*
12	*Cloudy*	*Mild*	*High*	*True*	*Yes*
13	*Cloudy*	*Hot*	*Normal*	*False*	*Yes*
14	*Rainy*	*Mild*	*High*	*True*	*No*

Kasus pada tabel 2.11 akan dibuat pohon keputusan untuk menentukan akan bermain tenis atau tidak dengan kriteria berdasarkan *outlook*, *temperature*, *humidity* dan *windy*.

Cara kerja algoritme C4.5 dalam membangun pohon keputusan secara umum menggunakan langkah berikut ini [39].

1. Pilih atribut sebagai akar.
2. Membuat cabang untuk tiap-tiap nilai.
3. Membagi kasus dalam cabang
4. Mengulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Pemilihan atribut sebagai akar didasarkan pada nilai gain tertinggi dari atribut-atribut yang ada. Perhitungan *gain* mengikuti Persamaan 2-17 sebagai berikut [35].

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|Si|}{|S|} * Entropy(Si) \qquad (2 - 17)$$

Keterangan:

S = Himpunan kasus

A = Atribut

n = Jumlah partisi atribut A

|Si| = jumlah kasus pada partisi ke-i

|S| = Jumlah kasus dalam S

Perhitungan nilai entropi mengikuti Persamaan 2-18 berikut ini [35].

$$Entropy(S) = Entropy(S) - \sum_{i=1}^{n} - p_i * \log_2(p_i) \qquad (2 - 18)$$

Keterangan:

S = Himpunan kasus

n = Jumlah partisi atribut A

p_i = Proporsi dari S_i terhadap S

Penyelesaian untuk permasalahan pada Tabel 2.4 dapat diuraikan sebagai berikut.

a. Menghitung jumlah kasus untuk keputusan *yes* dan keputusan *no* dan menghitung entropi dari tiap kasus yang dibagi berdasarkan atribut *outlook* ,*temperature*, *humidity*, *windy*. Selanjutnya dilakukan perhitungan *gain*. Jika perbandingan dua kelas, rasionya sama maka entropi bernilai 1 dan jika satu set hanya terdiri dari satu kelas maka entropi bernilai 0.

Perhitungan entropi total :

Entropi (total) = (- 4/14 * $\log_2$(4/14)) + (-10/4 * $\log_2$(10/14))

= 0,863 bits

Perhitungan entropi dan *gain* atribut *outlook*

Entropi atribut *outlook* = *rainy*

Entropi [1,4] = -1/5 $\log_2$(1/5) + (– 4/5 $\log_2$(4/5)) = 0,721 bits.

Entropi atribut *outlook* = *sunny*

Entropi [3,2] = -3/5 $\log_2$(3/5) + (- 2/5 $\log_2$(2/5)) = 0,971 bits.

Entropi atribut *outlook* = *cloudy*

Entropi [0,1] = - 0 $\log_2$(0) + (– 1 $\log_2$(1)) = 0 bits.

Entropi total atribut *outlook*

= entropi ([1,4]; [3,2]; [0,4])

= (((5/14)*0,721) + ((5/14)*0,971) + ((4/14)*0) = 0,604 bits.

Gain atribut *outlook*

$$Gain(Total, outlook) = Entropy(total) - \sum_{i=1}^{n} \frac{|Outlook_i|}{|total|} * Entropy(outlook_i)$$

Gain atribut *outlook*

= 0,863 -0,604 = 0,259 bits

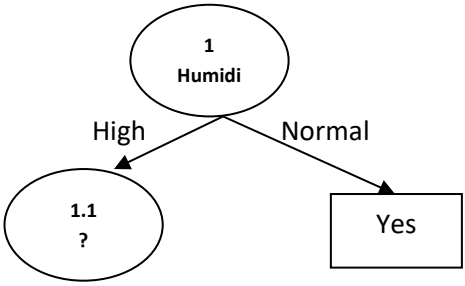
Proses perhitungan diatas juga diterapkan untuk atribut *temperature*, *humidity*, *dan windy*. Secara lengkap hasil untuk proses *split* pada *node* 1 tampak pada Tabel 2.12 untuk dibandingkan nilai *gain* untuk penentuan *node akar*.

Tabel 2.12 Perhitungan *node* 1

Node			**Jumlah kasus (S)**	**Tidak (S1)**	**Ya (S2)**	**Entropi**	***Gain***
1	Total		14	4	10	0,863121	
	Outlook						0,258521
		Cloudy	4	0	4	0	
		Rainy	5	1	4	0,721928	
		Sunny	5	3	2	0,970951	
	Temperature						0,183851

		Cool	4	0	4	0	
		Hot	4	2	2	1	
		Mild	6	2	4	0,918296	
	Humidity						0,370507
		High	7	4	3	0,985228	
		Normal	7	0	7	0	
	Windy						0,005978
		FALSE	8	2	6	0,811278	
		TRUE	6	2	4	0,918296	

Berdasarkan hasil perhitungan yang tampak pada Tabel 2.5. untuk *node 1* diketahui bahwa atribut dengan *gain* tertinggi adalah *humidity*, sehingga atribut ini akan menjadi *node* akar. Atribut *humidity* memiliki dua nilai yaitu *high* dan normal. Pada nilai normal telah mengklasifikasikannya menjadi 1 yaitu keputusan *yes* sehingga tidak perlu dilakukan perhitungan lagi, sementara untuk atribut *humidity* bernilai *high* perlu dilakukan perhitungan pada tahap selanjutnya. Hasil perhitungan pada *node* 1 dapat digambarkan pohon keputusan sementara seperti gambar 2.5 berikut ini.



Gambar 2.5 Pohon keputusan hasil perhitungan *node* 1

b. Berdasarkan *node* akar atribut *humidity* dengan nilai atribut *high* dilakukan penghitungan jumlah kasus untuk keputusan *yes* dan *no*, entropi semua kasus dan entropi kasus yang

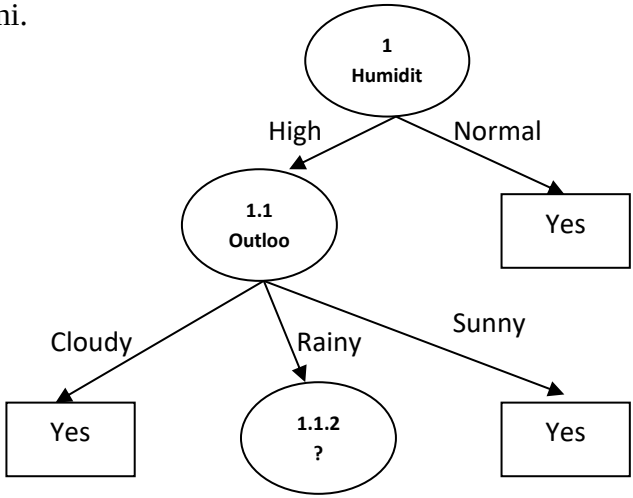
dibagi berdasarkan atribut *outlook*, *temperature*, dan *windy*. Kemudian melakukan penghitungan *gain* untuk tiap atribut. Hasil perhitungan dapat dilihat pada Tabel 2.13

Tabel 2.13 Perhitungan *Node* 1.1

Node			Jumlah kasus (S)	Tidak (S1)	Ya (S2)	Entropi	*Gain*
1.1	*Humidity High*		7	4	3	0,985228	
	Outlook						**0,699514**
		Cloudy	2	0	2	0	
		Rainy	2	1	1	1	
		Sunny	3	3	0	0	
	Temperature						0,020244
		Cool	0	0	0	0	
		Hot	3	2	1	0,918296	
		Mild	4	2	2	1	
	Windy	*False*	4	2	2	1	0,020244
		True	3	2	1	0,918296	

Pada Tabel 2.6 terlihat nilai *gain* tertinggi dimiliki oleh atribut *outlook* sebesar 0,69 sehingga atribut *outlook* akan menjadi *node* cabang dari atribut *humidity high.* Atribut *outlook* memiliki tiga nilai atribut meliputi *cloudy*, *rainy dan sunny*. Nilai atribut *cloudy* telah mengklasifikasikan kasus menjadi satu yaitu keputusan *yes* dan nilai atribut *sunny* juga sudah mengklasifikasikan menjadi satu dengan keputusan *no* sehingga kedua nilai atribut tersebut tidak perlu dilakukan perhitungan lebih lanjut. Nilai atribut *rainy* masih perlu dilakukan perhitungan lebih lanjut karena belum mengklasifikasi menjadi

satu. Berdasarkan hasil perhitungan tersebut maka akan dibentuk pohon keputusan sementara yang melibatkan atribut *humidity* dan atribut *outlook* seperti tampak pada Gambar 2.6 berikut ini.



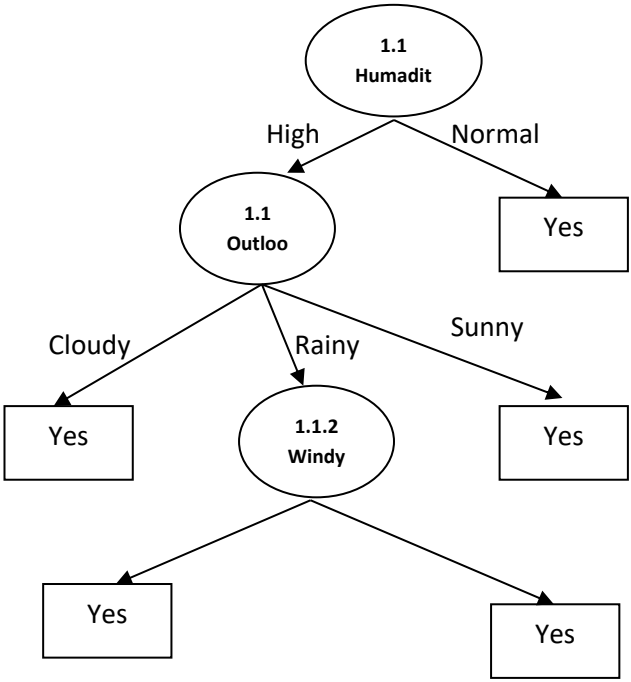
Gambar 2.6 *Decision tree* hasil perhitungan *node* 1.1

c. Perhitungan nilai entropi dan *gain* untuk atribut *temperature* dan *windy* dilakukan untuk mengetahui atribut yang akan mengisi percabangan pada atribut *outlook* dengan nilai atribut *rainy*. Hasil perhitungan untuk entropi dan *gain* atribut *temperature* dan *windy* dicantumkan pada Tabel 2.14

Tabel 2.14 Perhitungan *Node* 1.2

Node	Atribute		Jumlah kasus (S)	Tidak (S1)	Ya (S2)	Entropi	*Gain*
1.1.2	*Humidity-High Outlook-Rainy*		2	1	1	1	
Node	Atribute		Jumlah kasus (S)	Tidak (S1)	Ya (S2)	Entropi	*Gain*
	Temperature						0
		Cool	0	0	0	0	
		Hot	0	0	0	0	
		Mild	2	1	1	1	
	Windy						**1**
		False	1	0	1	0	
		True	1	1	0	0	

Berdasarkan hasil pada pada tabel 2.14 diketahui atribut *windy* memiliki nilai *gain* tertinggi, sehingga atribut *windy* akan menjadi *node* cabang dari nilai atribut *rainy*. Pada atribut *windy* memiliki dua nilai atribut *true* dan *false*. Kedua atribut tersebut masing-masing telah mengklasifikasikan menjadi satu. Nilai atribut *true* dengan keputusan *no*, nilai atribut *false* dengan keputusan *yes*. Pohon keputusan yang terbentuk berdasarkan hasil perhitungan pada pada Tabel 2.7. tampak pada Gambar 2.7 berikut.



Gambar 2.7 Pohon keputusan hasil perhitungan *node* 1.1.2

Berdasarkan Gambar 2.7 dapat dilihat semua kasus sudah masuk dalam kelas, sehingga pohon keputusan tersebut merupakan pohon keputusan terakhir yang dibentuk.

k. PART

Classifier partial decision tree (PART) membangun pohon keputusan parsial dari algoritme C4.5 dan menciptakan *rule* dari *leaf* [31]. PART adalah *decision tree* sederhana yang berisi cabang *tree* dan sub *tree*. Untuk menghasilkan sebuah PART, maka dilakukan proses kontruksi dan pemangkasan *(pruning)* terhadap cabang dari *tree* untuk menemukan sub *tree* yang tidak dapat lagi disederhanakan. Setelah sub *tree*

dihasilkan, proses kontruksi *tree* diakhiri dan *rule* atau aturan dihasilkan berdasarkan *leaf yang* memiliki cakupan besar [31]. Pada Tabel 2.15 berikut ini ditunjukkan algoritme *perluasan partial tree.*

Tabel 2.15 Algoritme *perluasan partial tree* [31]

Expand-subset (S):
Pilih sebuah tes yaitu T dan gunakan untuk membagi contoh *set*
menjadi *subset*
Urutkan *subset* dalam urutan menaik berdasarkan rata-rata entropi
while (terdapat *sublet* X yang tidak diperluas AND semua
subset yang diperluas adalah *leaf)*
expand-subset (X)
if (semua *subset* yang dipeiluas adalah *leaves* AND *error* estimasi untuk *subtree* $\geq$ *error* estimasi untuk *node)*
batalkan perluasan *subset* dan buat *node* sebuah *leaf*

l. Performa klasifikasi

Klasifikasi merupakan proses pemetaan objek-objek ke salah satu label kelas yang telah ditetapkan sebelumnya. Evaluasi performa klasifikasi terhadap algoritme yang digunakan disajikan dengan menggunakan *confusion matrix. Confusion matrix* adalah alat visualisasi yang umum digunakan untuk menyajikan performa klasifikasi. Hal ini dilakukan untuk menunjukkan hubungan antara hasil dan kelas prediksi. Analisis terhadap *confusion matrix* ini dapat memberikan informasi mengenai algoritme apa yang menghasilkan performa yang paling baik. *confusion matrix* disajikan pada Tabel 2.16

Tabel 2.16 *Matric Confusion*

	Predictive positive	*Predictive negative*
Actual positive	*True Positive* (TP)	*False Negative* (FN)
Actual negative	*False Positive* (FP)	*True Negative* (TN)

Berdasarkah Tabel 2.16, dapat dijelaskan bahwa jika objek memiliki kelas positif dan diklasiflkasikan sebagai kelas positif, maka dikatakan sebagai *true positive* (TP), dan jika diklasifikasikan sebagai kelas negatif, maka dikatakan sebagai *false negative* (FN). Jika objek memiliki kelas negatif dan diklasifikasikan sebagai kelas positif, maka dikatakan sebagai *false positive* (FP) dan jika diklasifikasikan sebagai kelas negatif maka dikatakan sebagai *true negative* (TN).

Diagnosis terhadap penyakit tuberkulosis (TBC) memiliki empat buah kemungkinan hasil yang diperoleh yaitu *true positive* (TP), *true negative* (TN), *false possitive* (FP) dan *false negative* (FN). *True positive* dan true negative merupakan hasil diagnosis yag benar. *False positive* adalah kesalahan prediksi yang terjadi ketika pasien sehat tetapi diprediksi positif tuberkulosis. *False negative* adalah kesalahan prediksi yang terjadi ketika pasien menderita tuberkulosis tetapi diprediksi negatf (sehat).

Beberapa alat pengukuran performa klasifikasi yang digunakan pada buku ini diantaranya adalah *accuracy*, *sensitivity*, dan *specificity*. *Accuracy* merupakan hasil prediksi yang tepat terhadap penyakit pasien.

Pada umumnya pakar medis atau dokter selalu mendiskusikan terminologi sensitivity dan spesificity dari uji diagnosis tuberkulosis.

Sensitivity merupakan proporsi pasien yang memiliki penyakit tuberkulosis yang diprediksi benar memiliki penyakit tuberkulosis *(true positive rate).* Sedangkan *specificity* merupakan proporsi pasien tanpa penyakit tuberkulosis yang diprediksi benar tidak memiliki penyakit tuberkulosis *(true negative rate)*[43]. Persamaan (2-19), (2-20), (2-21), (2-22) dan (2-23) memperlihatkan formula *accuracy*, *sensitivity*, *specificity*, FP_{rate} dan AUC.

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \tag{2-19}$$

$$\text{Sensitivity} = TP_{rate} = \frac{TP}{TP + FN} \tag{2-20}$$

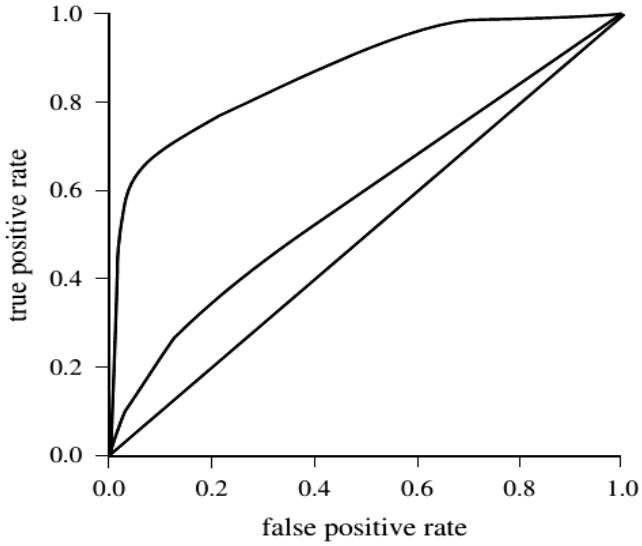
$$\text{Specificity} = TN_{rate} = \frac{TN}{FP + FN} \tag{2-21}$$

$$FP\ rate = \frac{FP}{FP + TN} \tag{2-22}$$

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2} \tag{2-23}$$

m. *Area Under Curve of* ROC (AUC)

Kurva *receiver operating characteristic* (ROC) menunjukkan *trade-off* antara *sensitivity* dan *false positive rate* untuk sebuah model klasifikasi. Kurva ROC merupakan grafik dua dimensi dengan sumbu-X adalah *false positive rate* dan sumbu-Y adalah *sensitivity/true positive rate.* Gambar 2.8 menunjukkan contoh kurva ROC dengan dua *classifier*.

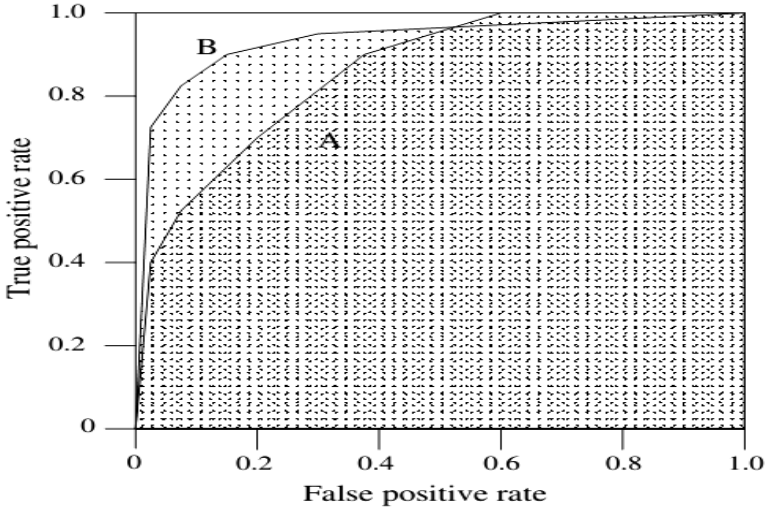


Gambar 2.8 Menunjukkan kurva ROC dengan dua algoritme klasifikasi [39].

Pada Gambar 2.8 juga terdapat garis diagonal yang menghubungkan titik (0,0) dan (1,1). Garis diagonal tersebut menunjukkan, semakin tinggi nilai *sensitivity* maka semakin tinggi pula nilai *specificity*. Oleh karena itu, semakin dekat kurva ROC dengan garis diagonal tersebut maka semakin berkurang tingkat akurasi dari suatu model klasifikasi [35].

Perbandingan performa antar algoritme klasifikasi dapat dilakukan denganmenghitung luas area di bawah kurva ROC atau *area under curve of ROC* (AUC). AUC adalah probabilitas algoritme klasifikasi akan cenderung memprediksikan kelas positif dibandingkan kelas negatif [44]. AUC memiliki nilai antara 0 dan 1. Namun, karena luas area di bawah garis diagonal pada gambar X adalah 0,5 maka tidak ada algoritme klasifikasi yang memiliki nilai AUC kurang dari 0,5. Oleh karena itu,

semakin dekat nilai AUC dengan 0.5 maka semakin berkurang akurasi suatu model klasifikasi. Suatu model klasifikasi yang sempurna akan memiliki nilai AUC adalah 1 [35]. Gambar 2.9 menunjukkan AUC dari dua buah algoritme klasifikasi



Gambar 2.9 Menunjukkan AUC dari dua *classifier* A dan B [43].

Pada Gambar 2.9 dapat disimpulkan bahwa performa algoritme klasifikasi B secara umum lebih baik dibandingkan algoritme klasifikasi A, karena AUC classifier B lebih besar dibandingkan classifier A. Namun pada saat *false positive rate* lebih besar dari 0,6, algoritme klasifikasi A memiliki performa lebih baik dari pada algoritme klasifikasi B. Oleh karena itu, dapat diambil kesimpulan bahwa algoritme klasifikasi dengan nilai AUC tinggi memiliki kemungkinan untuk memiliki performa lebih buruk pada area khusus dibandingkan dengan algoritme klasifikasi yang memiliki nilai AUC rendah [39].

n. *Paired t-Test*

Uji *paired t-test* dapat dilakukan untuk menentukan skema terbaik diantara dua skema pembelajaran. Misalkan terdapat himpunan sampel k akurasi dari skema pembelajaran pertama yaitu x_1, x_2, ..., x_k dan himpunan sampel *k* akurasi dari skema pembelajaran kedua yaitu y_1, y_2, ..., y_k dengan *mean* dan estimasi *variance*. masing-masing adalah m_x, m_y *dan* σ_y^2/k, σ_x^2/k. Jika sampel yang diperoleh kecil (k<100), mean dari masing-masing sampel akan memiliki distribusi student dengan derajat kebebasan k-1[45]. Untuk membandingkan erforma masing-masing metode yang digunakan, maka akan dilakukan *unpaired t-tes* untuk melihat tingkat signifikansi perbedaan performa yang dihasilkan masing-masing metode. Pengujian *paired t-test* di lakukan dengan menggunakan formula (2-24) [45].

$$t = \frac{m_d}{\sqrt{\sigma_d^{\,2} / k}} \qquad (2\text{-}24)$$

dengan:

m_d = Perbedaan *mean* antara dua metode

$\sigma_d^{\,2}$ = Perbedaan varian antara dua metode

K = *Sample* akurasi

Jika nilai $t \leq$ -z atau $t \geq z$, maka perbedaan kedua metode adalah signifikan. Nilai *confidence level z* diperoleh dari tabel distribusi *student* dengan derajat kebebasan *k-1*. Langkah-langkah dalam uji *t-test* ditunjukkan pada Tabel 2.17

Tabel 2.17 Langkah-langkah dalam uji *t-test* [45]

Langkah 1	:	Hitung $m_d = m_x - m_y$, (m_d adalah perbedaan kedua *mean*)
Langkah 2	:	Hitung $\sigma_d^2 = \frac{\sigma_x^2}{k} - \frac{\sigma_y^2}{k}$, ($\sigma_d^2$ adalah perbedaan kedua *variance*)
Langkah 3	:	Hitung $t = \frac{m_d}{\sqrt{\sigma_d^2 / k}}$, (Bentuk standar m_d disebut nilai t statistik)
Langkah 4	:	Nilai *confidence limit* z dapat diperoleh dari tabel distribusi *student* dengan derajat kebebasan $k-1$
Langkah 5	:	Karena uji ini merupakan *two-tailed test*, maka tingkat signifikansi (α) yang digunakan harus dibagi menjadi dua ($\alpha/2$), kemudian cari nilai z yang bersesuaian dengan tabel distribusi *student*.
Langkah 6	:	Jika $t \le -z$ atau $t \ge z$ maka perbedaan antara kedua skema pembelajaran adalah signifikan

C. Rangkuman

Bagian ini telah dilakukan tinjauan pustaka dan landasan teori dan mengenai diagnosis penyakit tuberkulosis dan prosedur diagnosis yang dilakukan. Pada tahap awal, diagnosis penyakit tuberkulosis dilakukan dengan melakukan pemeriksaan fisik, gejala klinis dan riwayat medis, kemudian dilanjutkan dengan uji lanjutan seperti uji tuberkulin, pemeriksaan radiologi, dan bakteriologi. Pemeriksaan bakteriologi untuk orang dewasa dan uji tuberkulin untuk balita merupakan standar emas *(gold standard)* dalam mendiagnosis penyakit tuberkulosis. Namun kekurangan dalam prosedur uji ini adalah lamanya waktu invansive, mengandung resiko tinggi serta waktu yang lama.

Oleh karena itu, metode alternatif perlu diterapkan untuk mendiagosis penyakit tuberkulosis, seperti metode komputer. Hasil dari buku ini diharapkan dapat memberikan masukan kepada dokter sebelum dilakukan pemeriksaan bakteriologi.

Proses diagnosis penyakit tuberkulosis berbasis komputer pada buku ini menggunakan berbagai jenis algoritme dalam data mining seperti Algoritme naive bayes, k-Star, adaboostMl, J48 dan PART. Kemudian dilakukan pemilihan fitur pada dataset yang digunakan dan kemudian analisis dilakukan terhadap hasil dari *accuracy, sensitivity, specificity dan AUC* masing-masing algoritme.

Proses selanjutnya dilakukan uji paired *t-test* untuk menentukan algoritme yang terbaik dalam mendiagnosis penyakit tuberkulosis.

BAB III

PREPROCESSING DIAGNOSIS

A. Pendahuluan

Tahap *preprocessing* data meliputi diskritisasi data numerik. Tahap selanjutnya dijelaskan bagaimana proses klasifikasi data dan evaluasi performa yang disajikan pada bagian metode.

1. Dataset tuberkulosis

buku ini melakukan proses klasifikasi pada dataset tuberkulosis yang bertujuan untuk mendiagnosis penyakit tuberkulosis pada pasien. Dataset yang digunakan diperoleh dari rumah sakit umum PKU Muhammadiyah Gombong. Tabel 3.1 memperlihatkan dataset tuberkulosis pada buku ini.

Dataset terdiri dari atribut-atribut berupa gejala klinis pasien dan hasil uji laboratorium. Dataset terdiri dari 260 *instance*, 30 atribut, dan satu kelas. Atribut kelas dibagi menjadi dua, yaitu kelas positif diberi label 'TBC' dan kelas negatif diberi label 'tidak TBC'. Untuk kelas positif merupakan data pasien yang menderita penyakit tuberkulosis dan kelas negatif adalah data pasien yang tidak menderita penyakit tuberkulosis, dengan rincian kelas positif berjumlah 141 *instance* dan kelas negatif berjumlah 119 *instance*. Berikut penjelasan dari masing-masing atribut pada dataset tuberkulosis.

1. Jenis kelamin (1: laki-laki, 0: perempuan)
2. Kelelahan (0: tidak kelelahan, 1: kelelahan)

3. Batuk lebih dari 3 minggu (0:tidak, 1:ya)
4. Radang di paru-paru (0:tidak, 1:ya)
5. Perasaan tidak nyaman pada pasien (0: tidak, 1: ya)
6. Nyeri pada sendi pasien (0: tidak, 1: ya)
7. Batuk berdarah (0: tidak ada, 1: ada)
8. Malas*:* Enggan untuk bekerja (0: tidak, 1: ya)
9. Nafsu makan turun*:* Kehilangan napsu makan (0: tidak, 1: ya)
10. Berat badan turun*:* Penurunan berat badan (0: tidak, 1: ya)
11. Keringat malam (0: tidak, 1: ya)
12. Nyeri dada (0: tidak, 1: ya)
13. Nyeri punggung (0: tidak, 1: ya)
14. Batuk berlendir (0: tidak, 1: ya)
15. Deman (0: tidak , 1 : ya)
16. Suhu Badan pasien dalam derajat (numerik).
17. *Erythrocyte:* Sel-sel darah merah dalam juta (numerik)
18. *Haematocrit:* Rasio volume yang ditempati oleh sel darah merah untuk volume seluruh darah dalam % (numerik) *IK.*
19. *Haemoglobin: Oxygen-transport metalloproteinase* yang mengandung besi dalam sel darah merah dalam g/dl (numerik)
20. *Leucocyte:* Sel darah putih (numerik)
21. Bukti Radiologi: Apakah ada bukti radiologis dari tuberkulosis paru lesi pada pasien (0: tidak, 1: ya)
22. Riwayat TBC sebelumnya: Apakah pasien memiliki penyakit tuberkulosis sebelumnya (0: tidak, l:ya)
23. Luka di paru-paru: Apakah ada pembukaan seperti luka pada paru-paru pasien (0: tidak, 1: ya)
24. Cairan berlebih pada selaput paru-paru: Akumulasi cairan yang berlebihan pada selaput paru-paru (0: tidak, 1: ya)

25. Perokok (0: tidak, 1: ya)
26. Kaki bengkak (0: tidak, 1: ya)
27. Tangan bengkak (0: tidak, 1: ya)
28. Sesak nafas (0: tidak, 1: ya)
29. Mual (0: tidak, 1: ya)
30. Muntah (0: tidak, 1: ya)
31. Diagnosa*: Class atribute: TBC* atau Tidak TBC

Tabel 3.1 Dataset Tuberkulosis

NO	JENIS KELAMIN	KELELAHAN	BATUK LEBIH DARI 3 MINGGU	RADANG DI PARU-PARU	TIDAK NYAMAN	NYERI SENDI	BATUK BERDARAH	MALAS	NAFSU MAKAN TURUN	BERAT BADAN TURUN	KERINGAT MALAM	NYERI DADA	NYERI PUNGGUNG	BATUK BERLENDIR	DINGIN	SUHU BADAN	ERYTROCYTE	HAEMATOCRIT	HAEMOGLOBIN	LEUCOCYTE	BUKTI RADIOLOGIS	RIWAYAT TBC SEBELUMNYA	LUKA DI PARU-PARU	CARAN BERLEBIH PADA SEJAKIT PARU-PARU	PEROKOK	KAKI BENGKAK	TANGAN BENGKAK	SESAK NAFAS	MUAL	MUNTAH	DIAGNOSIS
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	36,0	4,5	45	13	6.000	0	0	0	0	0	0	0	0	0	0	TIDAK_TBC
2	0	0	1	0	1	0	1	1	1	1	1	0	0	0	0	36,0	5,0	35	12	12.000	0	0	0	0	0	1	0	0	0	0	TBC
3	0	0	1	0	1	0	1	1	1	0	0	1	1	0	0	36,0	5,2	40	16	5.000	0	0	0	0	0	0	0	0	0	0	TBC
4	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	36,0	4,5	35	13	10.000	1	0	0	0	0	1	0	0	0	0	TBC
5	1	0	1	0	1	0	0	0	0	0	0	1	0	0	1	36,5	4,5	35	11	11.000	0	0	0	0	0	0	0	0	0	0	TBC
255	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	35,0	4,0	30	12	12.000	0	0	0	0	0	1	1	0	1	1	TBC
256	0	1	1	1	0	0	0	0	1	1	0	1	0	0	0	36,5	4,5	37	14	12.500	1	0	0	1	0	0	0	0	0	0	TBC
257	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	36,5	4,0	30	15	7.000	0	0	0	0	0	0	0	0	0	0	TIDAK_TBC
258	0	1	1	1	1	0	1	0	1	1	0	1	0	0	0	37,5	4,2	30	13	5.500	1	0	0	0	0	0	0	1	0	0	TBC
259	1	0	0	0	1	0	1	0	0	0	0	0	0	0	1	38,5	3,5	30	12	6.000	0	0	0	0	1	0	0	0	0	0	TIDAK_TBC
260	1	1	1	0	1	1	1	1	1	0	1	0	0	0	1	36,0	4,3	30	13	12.000	0	0	0	0	1	1	1	1	1	1	TBC

B. Perangkat Lunak yang Digunakan

Perangkat lunak yang digunakan dalam buku ini meliputi Rosetta dan WEKA. Tool pertama digunakan untuk pemilihan atau reduksi atribut dengan Roughset, sedangkan tool WEKA digunakan untuk pemilihan fitur berbasis CFS dan memanfaatkan algoritme *machine learning* meliputi *Naive Bayes, K-Star, AdaboostMl, J48 dan PART*.

1. *Rosetta GUI* version 1.4.41

Rosetta merupakan *toolkit* yang digunakan untuk menganalisis data tabular berbasis *framework rough set theory* [46]. Rosetta didesain untuk mendukung semua proses dalam *data mining* dan *knowledge discovery*. Proses yang tersedia meliputi proses awal dari *preprocessing*, sejumlah algoritme *data mining*, proses pemilihan/ reduksi fitur sampai dengan pengujian validasi, pengenalan suatu pola maupun analisa suatu *rule*. Rosetta dalam buku ini digunakan untuk melakukan reduksi/ pemilihan fitur dari *dataset*.

2. WEKA 3.7.10

WEKA merupakan *tool* untuk proses *data mining* berbasis java. Berbagai algoritme *machine learning* disediakan dalam perangkat lunak ini. Proses disediakan dari *preprocessing* data sampai proses pengujian. Hasil pengujian ditampilkan dalam *confusion matrix* yang menyatakan jumlah prediksi yang benar dan jumlah prediksi yang salah.

Empat tombol yang dapat digunakan untuk menjalankan aplikasi meliputi:

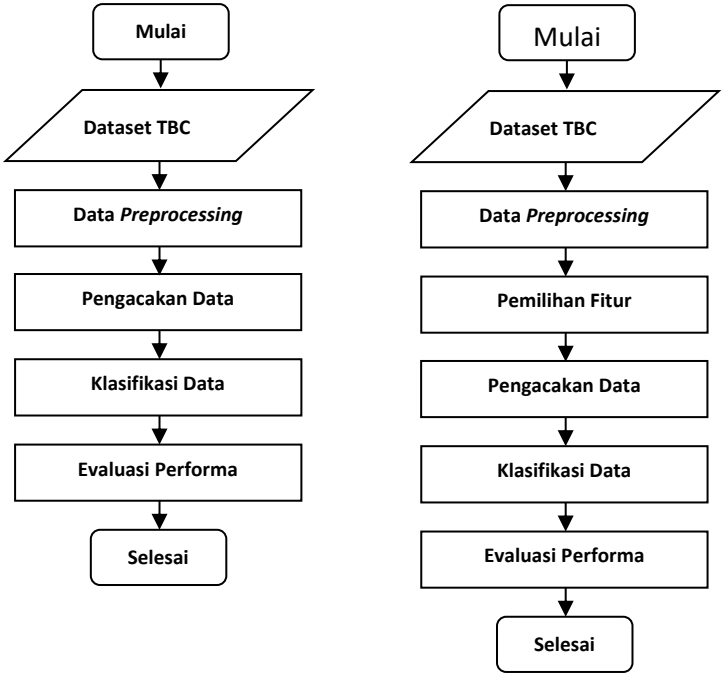
1. *Explorer* digunkan untuk menggali lebih jauh data dengan aplikasi WEKA
2. *Experimenter* digunakan untuk melakukan percobaan dengan pengujian statistic skema pembelajaran.
3. *Knowledge flow* digunakan untuk pengetahuan pendukung.

Simple CLI antar muka dengan menggunakan tampilan *command-line* yang memungkinkan langsung mengeksekusi perintah WEKA untuk sistem operasi yang tidak menyediakan secara langsung.

C. Metode

buku ini bertujuan untuk mendiagnosis penyakit tuberkulosis dengan melakukan pemilihan fitur pada dataset tuberkulosis. Pemilihan fitur yang digunakan adalah *correlation based feature selection* (CFS), *rough set theory (RST)* dan *motivated feature selection* (MFS). Pemilihan fitur ini dipilih karena CFS merupakan evaluasi subset dari himpunan fitur dengan mempertimbangkan kemampuan prediktif dari masing-masing fitur dengan isi fitur-fitur yang sangat berkorelasi dengan kelas dan masing-masing fitur tidak berkorelasi. Untuk Rough set di pilih karena menawarkan metode matematis untuk menemukan pola tersembunyi pada suatu data dan menghasilkan rule-rule yang sedikit. MFS dipilih didasarkan pengetahuan yang dimiliki oleh pakar medis dengan menentukan faktor signifikansi medis yang berpengaruh dalam proses diagnosis.

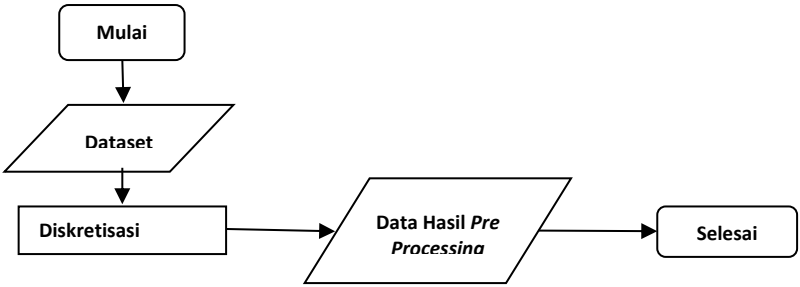
Algoritme klasifikasi pada buku ini menggunakan beberapa algoritme pada data *mining* seperti naive bayes, k-Star, adaboostMl, J48 dan PART, yang sudah dijelaskan pada BAB II. Gambar 3.1 memberikan gambaran alur yang dilakukan pada buku ini, dimana langkah pertama yang dilakukan adalah data *preprocessing*, dan proses selanjutnya dilakukan pengacakan data hasil dari *preprocessing*. Data hasil pengacakan selanjutnya dijadikan *input* dalam proses klasifikasi. Gambar 3.1(a) dan Gambar 3.1(b) adalah proses diagnosis atau klasifikasi penyakit TBC dilakukan dengan dua cara yaitu menggunakan pemilihan fitur dan tanpa pemilihan fitur. Hasil Performa diagnosis yang diperoleh selanjutnya dievaluasi untuk memberikan perbandingan antara diagnosis dengan menggunakan pemilihan fitur dan tanpa pemilihan fitur.



Gambar 3.1 Alur buku (a) Klasifikasi tanpa pemilihan fitur (b) Klasifikasi dengan pemilihan fitur

1. Data preprocessing

Data *preprocessing* merupakan langkah awal dalam mendiagnosis penyakit tuberkulosis. Data *preprocessing* pada buku yaitu diskretisasi data numerik. Gambar 3.2 memperlihatkan langkah-langkah pada tahap data preprocessing.



Gambar 3.2 Tahapan *data preprocessing*

Berdasarkan Gambar 3.2, langkah pertama yang dilakukan adalah data diskretisasi. Diskretisasi dilakukan terhadap data yang bernilai numerik kemudian dirubah ke dalam bentuk nominal sehingga dihasilkan nilai dalam beberapa interval. Sebagai contoh, terdapat atribut 'umur' yang bernilai numerik diperlihatkan pada Tabel 3.2 berikut :

Tabel 3.2 Contoh atribut numerik

No.	Umur
1	58
2	55
3	51
4	53
5	59
6	42
7	38
8	48
9	50
10	49

Berdasarkan penjelasan pada sub bab 2.3.1.1, maka dengan perangkat lunak Rosetta Rough Set Toolkit hasil diskretisasi diperlihatkan pada Tabel 3.4, dimana nilai atribut 'umur' dibagi ke dalam beberapa interval.

Tabel 3.3. Contoh hasil diskretisasi

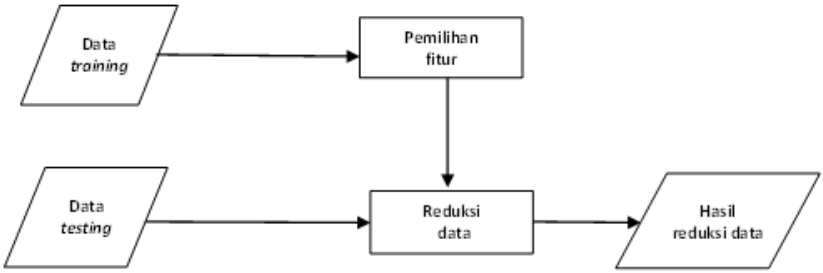
No.	Umur
1	[57, 59)
2	[54, 57)
3	[51,52)
4	[52, 54)
5	[59, *)
6	[40,45
7	[*, 40)
8	[45,49)
9	[50,51)
10	[49, 50)

Pada buku ini *Boolean Reasoning* [47] dipilih sebagai metode diskretisasi yang terdapat pada perangkat lunak Rosetta Rough Set Toolkit.

2. Pemilihan Fitur

Proses pemilihan fitur dilakukan untuk mereduksi data dan memilih fitur-fitur yang relevan terhadap hasil keputusan diagnosis penyakit TBC. Pada buku ini digunakan dua jenis pemilihan fitur yaitu pemilihan fitur berbasis komputer dan pemilihan fitur berbasis pakar medis. Proses pemilihan fitur berbasis komputer dilakukan dengan menggunakan perangkat

lunak Weka 3.7 Pada Gambar 3.3 menunjukan langkah-langkah yang dilakukan pada proses pemilihan fitur.



Gambar 3.3 Proses pemilihan fitur

Data hasil *preprocessing* dibagi menjadi dua bagian (*stratified train-test split*) dengan proporsi data *training* sebesar 2/3 bagian dan data *testing* sebesar 1/3 bagian. Proses *train-test split* dilakukan dengan disertai stratifikasi sehingga proporsi kelas positif dan negatif pada data *training* dan *testing* sama. Selanjutnya proses pemilihan fitur dilakukan pada data training. Kemudian data testing direduksi sesuai dengan hasil pemilihan fitur pada data *training*.

a. Metode pemilihan fitur berbasis komputer

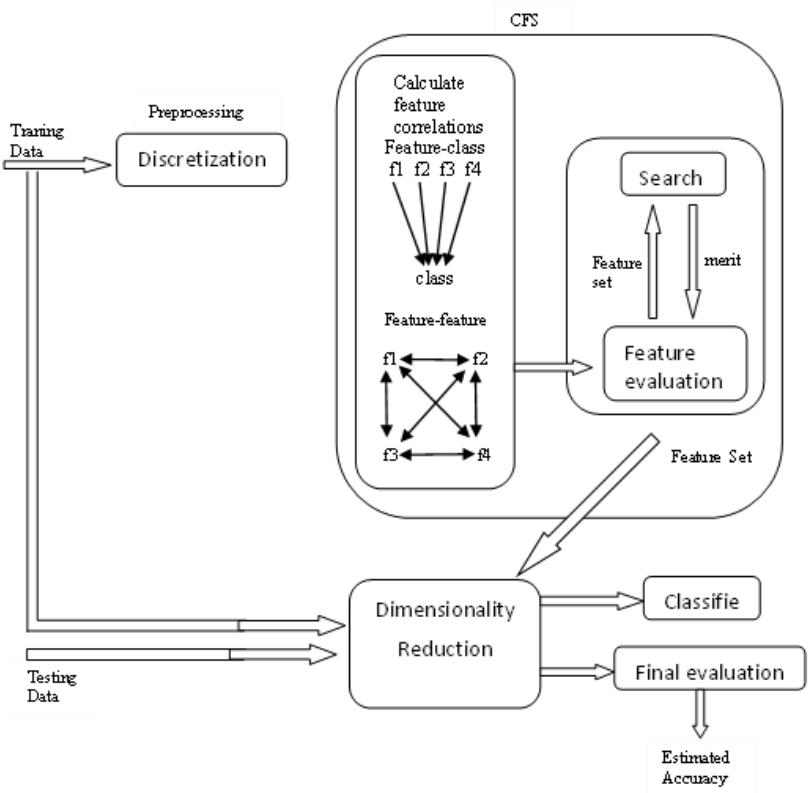
Pada buku ini menggunakan pemilihan fitur berbasis komputer sebagai berikut :

1) *Correlation based feature selection.*

Correlation based feature selection yang selanjutnya disebut seleksi atribut berbasis korelasi atau CFS adalah sebuah algoritma filter sederhana yang meranking subset berdasarkan fungsi evaluasi heuristik berbasis korelasi [29]. Berdasarkan hipotesis bahwa subset atribut yang bagus berisi atribut yang

mempunyai korelasi tinggi terhadap kelas dan tidak saling berkorelasi satu sama lain. Korelasi yang tinggi satu sama lain atribut menandakan atribut tersebut redundan. Atribut yang berkorelasi rendah terhadap kelas adalah atribut yang tidak relevan. Atribut yang tidak relavan dan redundan harus dihapus. CFS menggunakan strategis pencarian *best first* untuk pemilihan fitur berbasis korelasi.

Berikut pada Gambar 3.4 dapat di lihat langkah dari algoritme *correlation based feature selection*



Gambar 3.4 Langkah-langkah pada algorima *correlation based feature selection*

Dapat dilihat dari Gambar 3.4. langkah pertama yang dilakukan adalah *dataset* terlebih dahulu dikenai proses diskretisasi. Langkah selanjutnya adalah menghitung korelasi antar fitur dan korelasi antara fitur dengan kelas. Kemudian dicari subset fitur yang memiliki nilai merit tertinggi. Subset fitur dengan nilai merit tertinggi akan digunakan untuk mereduksi *dataset.* Nilai merit yang diperoleh dapat dihitung menggunakan Persamaan (3-1) [29].

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{3-1}$$

M_s = Nilai subset fitur *S* dengan fitur sebanyak *k* terhadap hasil keputusan.

r_{cf} = Rata-rata korelasi antara fitur-fitur dengan kelas.

r_{ff} = Rata-rata korelasi antar fitur.

2) Pemilihan Fitur BerbasisTeori *Rough Set*

Rough set menawarkan metode matematis untuk menemukan pola tersembunyi pada suatu data dan dapat digunakan sebagai klasifikasi untuk menemukan hubungan struktural dalam data yang tidak lengkap atau *noisy* [23]. Pemilihan fitur berdasarkan teori rough set sudah dijelaskan pada subbab 2.2.3.1.2.2. metode Johnson digunakan untuk mereduksi data [50]

3) Pemilihan fitur berbasis pakar medis

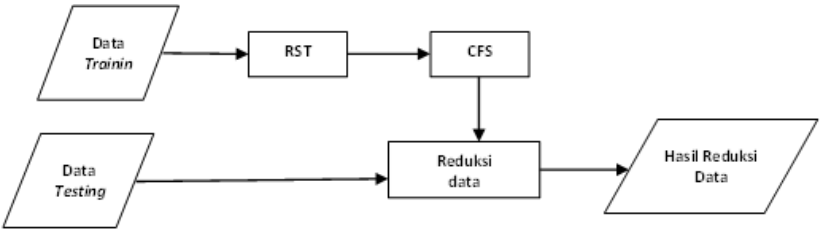
Pada penyakit TBC pakar medis telah menentukan delapan faktor signifikan medis yang berpengaruh dalam proses diagnosis TBC. Delapan faktor tersebut adalah batuk lebih dari 3 minggu, batuk berdarah, sesak nafas, nyeri dada, demam, tidak

nyaman, keringat malam dan berat badan turun [19]. Kedelapan faktor ini digunakan sebagai pemilihan fitur berbasis pakar medis.

4) Kombinasi metode pemilihan fitur berbasis komputer

Pada buku ini dilakukan kombinasi pemilihan berbasis komputer untuk meningkatkan performa diagnosis penyakit TBC. Metode yang dipilih untuk dikombinasikan adalah RST dan CFS, hal ini dikarenakan kedua metode memiliki cara kerja yang berbeda dan memiliki kelebihan dan kekurangan masing-masing.

RST memberikan himpunan fitur minimal yang mampu membedakan objek sama baiknya dengan seluruh himpunan fitur yang digunakan. Namun himpunan fitur minimal belum tentu memberikan performa yang optimal ketika dilakukan poses klasifikasi. CFS berasumsi bahwa subset fitur yang berisi fitur-fitur yang sangat berkorelasi dengan kelas dan masing-masing tidak berkorelasi mampu meningkatkan performa klasifikasi. Namun subset fitur ini belum tentu merupakan himpunan fitur minimal seperti *reduct* yang dihasilkan RST. Oleh karena itu kombinasi RST dan CFS dapat mengatasi kekurangan dari masing-masing metode. Gambar 3.5 menunjukan cara kombinasi RST dan CFS.

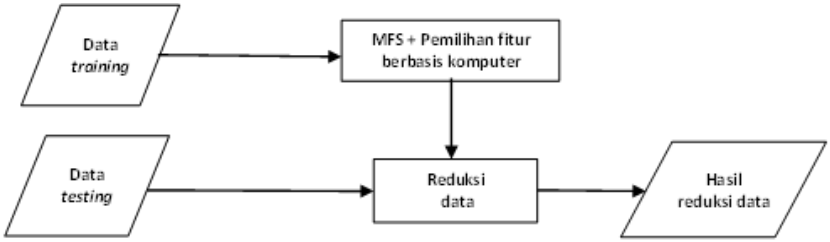


Gambar 3.5 Cara kombinasi RST dan CFS

Fitur-fitur pada data training terlebih dahulu dipilih dengan menggunakan RST dengan metode Johson. Himpunan fitur minimal yang diperoleh dari RST kemudian dipilih dengan menggunakan CFS. Oleh karena itu diperoleh subset fitur minimal, dimana fitur-fitur di dalamnya sangat berkorelasi dengan kelas dan antar fitur memiliki korelasi yang rendah. Kemudian data testing direduksi sesuai dengan fitur-fitur yang dipilih oleh kombinasi kedua metode tersebut. Kombinasi ini disingkat menjadi RSTCFS.

5) Penggabungan Pemilihan Fitur Berbasis Pakar Medis dan Komputer

Pemilihan fitur berbasis komputer memiliki kemungkinan untuk menghapus fitur-fitur yang dianggap penting oleh pakar medis. Oleh karena itu untuk menghindari terhapusnya fitur-fitur tersebut, penggabungan antara pemilihan fitur berbasis pakar medis/*motivated feature selection* (MFS) [35] dan komputer perlu dilakukan. Gambar 3.6. menunjukkan cara penggabungan antara pemilihan fitur berbasis pakar medis (MFS) dan komputer.



Gambar 3.6 Penggabungan MFS dan pemilihan fitur berbasis komputer

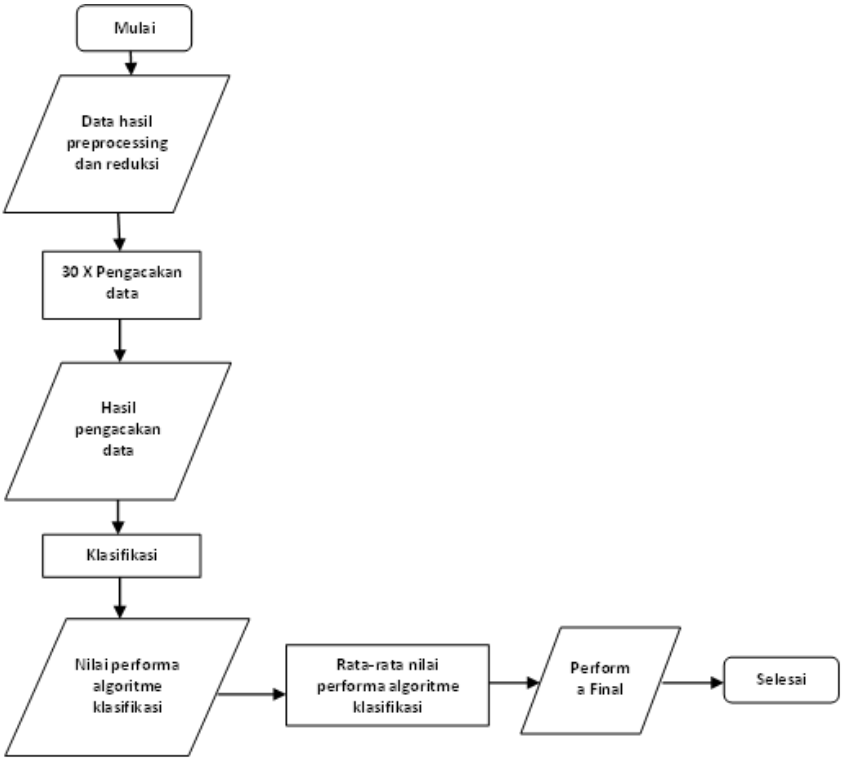
Berdasarkan Gambar 3.6 fitur pada training dipilih sesuai dengan fitur-fitur yang dipilih oleh MFS dan pemilihan fitur berbasis komputer. Kemudian data testing direduksi sesuai fitur-fitur yang telah dipilih.

3. Pengacakan data

Untuk menghindari bias pada hasil klasifikasi, maka dilakukan proses pengacakan data sehingga diperoleh *sample* data yang berbeda. Proses pengacakan data pada dataset dapat mempengaruhi performa diagnosis. Hal ini disebabkan model klasifikasi yang terbentuk pada saat pelatihan berbeda. Oleh karena itu, ketika model diujikan pada data uji maka akan menghasilkan performa yang berbeda pula.

Proses pengacakan data pada dataset dapat mempengaruhi performa diagnosis penyakit TBC. Hal ini disebabkan model klasifikasi yang terbentuk selama pelatihan berbeda. Oleh karena itu, ketika model diujikan pada data testing maka akan memberikan nilai performa yang berbeda pula.

Pada buku ini sampel nilai performa minimum yang digunakan adalah 30. Oleh karena itu untuk mendapatkan sampel tersebut, dilakukan sebanyak 30 kali proses pengacakan data. Proses pengacakan data ini dilakukan dengan menggunakan perangkat lunak Weka 3.7. Gambar 3.7 memperlihatkan langkah-langkah pengacakan data hingga cara memperoleh performa final.



Gambar 3.7 Proses pengacakan data

Berdasarkan Gambar 3.7 pengacakan data dilakukan pada data hasil preprocessing dan hasil reduksi. Oleh karena itu, diperoleh sebanyak 30 nilai performa ketika proses klasifikasi dilakukan. Performa final diperoleh dari rata-rata ke-30 nilai performa tersebut

4. Klasifikasi data

Diagnosis penyakit tuberkulosis berbasis komputer pada buku ini dilakukan dengan cara mengklasifikasi dataset

tuberkulosis. Alur proses klasifikasi data disajikan pada Gambar 3.8.

Gambar 3.8 Diagram alir klasifikasi data

Berdasarkan Gambar 3.8, performa klasifikasi disajikan dengan melakukan *10-fold cross validation* pada dataset hasil pengacakan. Metode ini merupakan cara umum yang digunakan pada buku penambangan data. 10-*fold cross validation* membagi dataset menjadi sepuluh bagian (*fold*). Sembilan *fold* pertama digunakan sebagai data training dan *fold* terakhir digunakan sebagai data *testing*. Kemudian sembilan *fold* kedua digunakan sebagai data *training* dan sisanya digunakan sebagai data *testing*. Proses ini dilakukan secara terus-menerus seebanyak sepuluh kali hingga setiap *fold* pernah satu kali menjadi data *testing*. Pada buku ini, perangkat lunak Weka 3.7 digunakan untuk mengklasifikasikan dataset menggunakan *naïve bayes, k-Star, adaboosM1*, PART dan J.48

a. *Naive Bayes*

Salah satu penerapan teori bayes adalah dalam klasifikasi *naive bayes*. *Naive bayes* didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara kondisinal tidak berkaitan dengan atribut yang lain. Sebagai contoh, terdapat data cuaca yang diperlihatkan pada Tabel 3.4.

Tabel 3.4 Contoh dataset algoritme Naive Bayes

No.	Cuaca	Temperatur	Kelembaban	Angin	Kelas
1.	Cerah	Panas	Tinggi	Besar	Tidak
2.	Cerah	Panas	Tinggi	Besar	Tidak
3.	Mendung	Panas	Tinggi	Kecil	Main
4.	Hujan	Sedang	Tinggi	Kecil	Main
5.	Hujan	Dingin	Normal	Kecil	Main
6.	Hujan	Dingin	Normal	Besar	Tidak
7.	Mendung	Dingin	Normal	Besar	Main
8.	Cerah	Sedang	Tinggi	Kecil	Tidak
9.	Cerah	Dingin	Normal	Kecil	Main
10.	Hujan	Sedang	Normal	Kecil	Main
11.	Cerah	Sedang	Normal	Besar	Main
12.	Mendung	Sedang	Tinggi	Besar	Main
13.	Mendung	Panas	Normal	Kecil	Main
14.	Hujan	Sedang	Tinggi	Besar	Tidak

Jika terdapat data baru dengan data sebagai berikut:
(Cuaca = cerah, temperature = dingin, Kelembaban = tinggi, Angin = besar)

Untuk menentukan kelas dari data tersebut, maka dapat dilakukan dengan langkah-langkah sebagai berikut:

1. Hitung P(Main) = $\frac{9}{14}$ = 0,64

   Hitung P(Tidak) = $\frac{5}{14}$ = 0,36

2. Hitung P(Cuaca = Cerah | Main) = $\frac{2}{9}$ = 0,22

   Hitung P(Cuaca = Cerah | Tidak) = $\frac{3}{5}$ = 0,60

3. Hitung P(Temperatur = Dingin | Main) = $\frac{3}{9}$ = 0,33

   Hitung P(Temperatur = Dingin | Tidak) = $\frac{1}{5}$ = 0,20

4. Hitung P(Kelembaban = Tinggi | Main) = $\frac{3}{9}$ = 0,33

   Hitung P(Kelembaban = Tinggi | Tidak) = $\frac{4}{5}$ = 0,80

5. Hitung P(Angin = Besar | Main) = $\frac{3}{9}$ = 0,33

   Hitung P(Angin = Besar | Tidak) = $\frac{3}{5}$ = 0,60

Dengan persamaan (2-7), maka diperoleh:
Probabilitas "Main":
P(Main)*P(Cerah | Main)*P(Dingin | Main)*P(Tinggi | Main)* P(Besar | Main)
= 0,64 *x* 0,22 *x* 0,33 x 0,33 *x* 0,33 = 0,0051
Probabilitas "Tidak":
P(Tidak)* P(Cerah | Tidak)* P(Dingin | Tidak)* P(Tinggi | Tidak)* P(Besar | Tidak)
= 0,36 *x* 0,60 *x* 0,20 *x* 0,80 *x* 0,60 = 0,0207

Maka dengan Naive Bayes, untuk data baru tersebut dapat disimpulkan masuk ke dalam kelas "tidak main", karena memiliki nilai yang paling besar.

b. K-Star

Algoritme k-star dapat digambarkan sebagai algoritme pembelajaran yang menggunakan *entropy* sebagai ukuran jarak. Sebagai contoh, berdasarkan Tabel 3.6. jika terdapat data baru dengan nilai Xi=3 dan Xa=7, maka untuk menentukan kelas dari data tersebut dapat dicari dengan langkah sebagai berikut:

1. Menentukan probabilitas dari Xi=3 untuk kelas "baik".

   $P(X_1 = 3 \mid baik) = \frac{1}{2} = 0{,}5$

2. Menentukan probabilitas dari Xi=3 untuk kelas "buruk".

   $P(X_1 = 3 \mid buruk) = \frac{0}{2} = 0$

3. Menentukan probabilitas dari X_2=7 untuk kelas "baik".

   $P(X_2 = 7 \mid baik) = \frac{1}{2} = 0{,}5$

4. Menetukan probabilitas dari X_2=7 untuk kelas "buruk".

   $P(X_2 = 7 \mid buruk) = \frac{1}{2} = 0{,}5$

5. Hitung P*(x | baik) = 0,5 + 0,5 = 1
6. Hitung P*(x | buruk) = 0 + 0,5 = 0,5
7. Menentukan nilai *entropy* dari masing-masing atribut x dengan persamaan (2-10)

   $K^*(x \mid baik) = -Iog_2\ P^*(x \mid baik)$
   $= -log_2\ l=0$
   $K^*(x \mid buruk) = -Iog_2\ P^*(x \mid buruk)$
   $= -log_2\ 0{,}5 = 1$

Berdasarkan nilai *entropy* yang diperoleh? maka data baru tersebut masuk ke dalam kelas "buruk".

c. AdaboostMl

Berbeda halnya dengan algoritme *bagging* yang mendapatkan banyak model *classifier* dari data yang berbeda-beda, algoritme ini bekerja selalu dengan data yang sama. Setiap kali pembuatan model *classifier*, data yang digunakan tetap seperti semula tetapi memiliki sebaran bobot yang berbeda dalam tiap iterasi. Penggunaan bobot juga dilakukan pada saat proses penggabungan dugaan akhir dari banyak model *classifier* yang dihasilkan. Berikut tahapan pada algoritme adaboostMl [43]:

Pada tahap pembangunan model, pertama-tama nilai bobot untuk setiap *tuple* pada himpunan D, diinisialisasi sebesar 1/d (d=jumlah anggota himpunan D). Setelah suatu model M, dibangkitkan, kemudian dihitung tingkat kesalahannya *(error rate]*, dengan cara menjumlahkan nilai bobot dari setiap *tuple* yang salah klasifikasi pada M. *Error rate* ini dapat dihitung menggunakan persamaan (2-11), dimana *error (Xj)* merupakan *misclassified error* dari tuple *(Xj).* Bila *tuple* tersebut tergolong *misclassified,* maka nilai *error (Xj)* adalah 1, sedangkan bila tidak nilainya 0. Bila *error rate* M, melebihi 0,5 maka model tersebut tidak digunakan, kemudian akan dibangkitkan lagi *training set* D, yang baru, yang akan menurunkan model M, yang baru pula. Setelah suatu *tuple* pada putaran ke-i terklasifikasi dengan benar, maka nilai bobotnya dikalikan dengan *e/l-e.* Setelah didapatkan nilai bobot dari seluruh *tuple* yang terklasifikasi dengan benar, lalu nilai bobot dari seluruh tuple (termasuk yang *misclassified)* dinormalisasikan, dengan

cara mengalikannya dengan jumlah nilai bobot yang lama (sebelumnya), lalu dibagi dengan jumlah nilai bobot yang baru.

Selanjutnya pada tahap klasifikasi, tetapkan nilai bobot yang sama sebesar 0 pada setiap kelas dari model M, kemudian tambahkan dengan –log ke bobot kelas yang diperkirakan oleh model M. Hasil klasifikasi atau prediksi yang dipilih adalah model yang memiliki nilai bobot paling besar.

d. J48

J48 adalah *classifier* yang mengimplementasikan algoritme C4.5. Algoritme C45 membangun *decision tree* dari atas ke bawah (*top down*), mulai dengan menentukan atribut mana yang akan diletakkan sebagai *root.* Sebagai contoh, terdapat data seperti pada Tabel 3.5

Tabel 3.5 Contoh dataset algoritme J48

Posisi	Umur	Gaji	Status
Sales	31..35	46..50	Senior
Sales	26..30	26..30	Junior
Sales	31..35	31..35	Junior
Sistem	21..25	46..50	Junior
Sistem	31..35	66..70	Senior
Sistem	26..30	46..50	Junior
Sistem	41..45	66..70	Senior
Pemasaran	36..40	46..50	Senior
Pemasaran	31..35	41. .45	Junior
Sekretaris	46..50	36..40	Senior
Sekretaris	26..30	26..30	Junior

Untuk menentukan *root* dari *tree* berdasarkan tabel 3.5, maka langkah pertama adalah dengan menghitung nilai *entropy* dengan persamaan (2-12).

$$Entropy\ (\mathrm{D}) = -\ \frac{5}{11}\, x \log_2 \frac{5}{11} - \frac{6}{11}\, x \log_2 \frac{6}{11} = 0{,}994$$

Selanjutnya menghitung *entropy* untuk masing-masing atribut dengan persamaan 2-13.

$$Entropy\ (\mathrm{D}) = \frac{3}{11}\ \mathrm{x}\ Entropy\ (D1) + \frac{4}{11}\ \mathrm{x}\ Entropy\ (D2) + \frac{2}{11}\ \mathrm{x}\ Entropy\ (D3) + \frac{2}{11}\ \mathrm{x}\ Entropy\ (D4)$$

$$= \frac{3}{11}\ \mathrm{x}\left(-\frac{1}{3}x\log_2\frac{1}{3} - \frac{2}{3}x\log_2\frac{2}{3}\right) + \frac{4}{11}x\left(-\frac{2}{4}x\log_2\frac{2}{4} - \frac{2}{4}x\log_2\frac{2}{4}\right) +$$

$$= \frac{3}{11}\ \mathrm{x}\ 0{,}919 + \frac{4}{11}x1 + \frac{2}{11}x1 + \frac{2}{11}x1 = 0{,}978$$

Dengan cara yang sama:

$$Entropy_{Umur}\ (D) = \frac{4}{11}xEntropy(D1) + \frac{3}{11}xEntropy(D2) + \frac{1}{11}xEntropy(D3) + \frac{1}{11}xEntropy(D4) + \frac{1}{11}xEntropy(D5) + \frac{1}{11}xEntropy(D6) +$$

$$= \frac{4}{11}x1 + \frac{3}{11}x0 + \frac{1}{11}x0 + \frac{1}{11}x0\frac{1}{11}x0$$

$$= 0{,}364$$

$$Entropy_{Gaji}\ (D) = \frac{4}{11}xEntropy(D1) + \frac{2}{11}xEntropy(D2) + \frac{1}{11}xEntropy(D3) + \frac{2}{11}xEntropy(D4) +$$

$$\frac{1}{11}xEntropy(D5)+\frac{1}{11}xEntropy(D6)+$$

$$= \frac{4}{11}x1+\frac{2}{11}x0+\frac{1}{11}x0+\frac{2}{11}x0+\frac{1}{11}x0+\frac{1}{11}x0$$

= 0,364

Berdasarkan persamaan (2-14), maka diperoleh *information gain* sebagai berikut:

Gain (D - Posisi) = 0,994 - 0,978 = 0,016

Gain (D - Umur) = 0,994 - 0,364 = 0,63

Gain (D - Gaji) - 0,994 - 0,364 = 0,63

Selanjutnya *Gain Ratio* bisa dihitung dengan persamaan (2-15), sehingga diperoleh :

Gain Ratio (Posisi) =

$$\frac{0{,}016}{\left(\frac{3}{11}x\log_2\frac{3}{11}\right)+\left(\frac{4}{11}x\log_2\frac{4}{11}\right)+\left(\frac{2}{11}x\log_2\frac{2}{11}\right)+\left(\frac{2}{11}x\log_2\frac{2}{11}\right)} = 0{,}0083$$

Gain Ratio (Umur) =

$$\frac{0{,}63}{\left(\frac{4}{11}x\log_2\frac{4}{11}\right)+\left(\frac{3}{11}x\log_2\frac{3}{11}\right)+\left(\frac{1}{11}x\log_2\frac{1}{11}\right)+\left(\frac{1}{11}x\log_2\frac{1}{11}\right)\left(\frac{1}{11}x\log_2\frac{1}{11}\right)\left(\frac{1}{11}x\log_2\frac{1}{11}\right)} = 0{,}266$$

Dari hasil perhitungan diatas, maka *root* untuk data pada Tabel 3.5 adalah atribut "umur" karena memiliki nilai *gain ratio* yang paling besar.

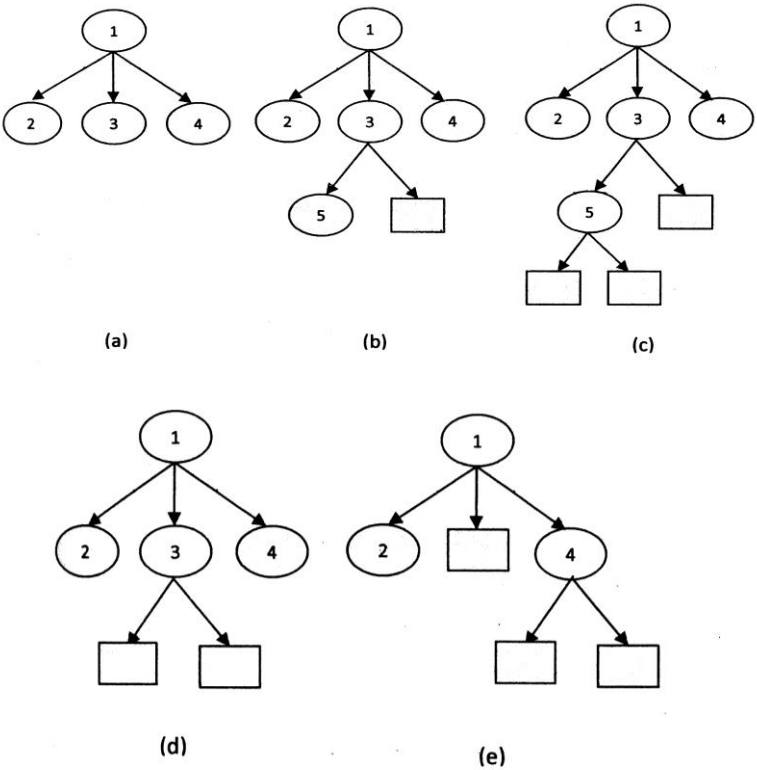
Untuk menentukan *node* yang kedua, ketiga, dst (node yang bukan menjadi *root)*, dihitung lagi menggunakan *information gain* untuk menentukan atribut mana yang akan menjadi *node* berikutnya.

e. PART

PART merupakan *decision tree* sederhana yang berisi cabang *tree* dan sub *tree* yang mengalami proses pemangkasan *(pruning)* untuk menemukan sub *tree* yang tidak dapat lagi disederhanakan. Gambar 3.8 memperlihatkan contoh langkah-langkah dalam membangun sebuah PART, dimana gambar lingkaran menyatakan simpul, dan gambar persegi panjang menyatakan *leaf.* Dapat dilihat pada Gambar 3.8 (a), (b) dan (c), bahwa *tree* dibangun secara rekursif dalam posisi normal. Nilai *entropy* terendah pada simpul akan dilakukan perluasan, dalam hal ini terjadi pada simpul ketiga Gambar 3.8 (b). Proses perluasan dilakukan kembali pada simpul kelima yang diperlihatkan pada Gambar 3.8 (c). Pada Gambar 3.8 (c) ini, simpul kelima terdapat *child* yang diperluasan dalam bentuk *leaf,* hal ini memicu proses *pruning. Pruning* dilakukan jika nilai *error rate* lebih kecil dari nilai *error rate* sebelum dilakukan *pruning.* Proses ini diperlihatkan pada Gambar 3.8 (c), (d) dan (e).

Pada Gambar 3.8 (e) diperlihatkan bahwa simpul keempat memiliki nilai *entropy* paling kecil dibandingkan dengan simpul kedua, kemudian diperluas membentuk *leaf. Sub tree* pada simpul keempat ini dapat dipertimbangkan untuk dilakukan proses *pruning.* Namun jika tidak, maka pada titik ini proses berakhir dengan tiga *leaf partial tree* yang nantinya akan dijadikan *rule.*

Proses pembangunan/*tree* ini dilakukan pada data latih, kemudian *rule* yang dihasilkan diuji coba dengan data uji untuk melihat performa klasifikasinya.



Gambar 3.9 Langkah-langkah membangun PART [30]

5. Evaluasi performa

Perbandingan performa *classifier* dilakukan terhadap algoritme yang masuk ke dalam jenis klasifikasi yang sama karena dianggap *apple to apple*. Misalnya perbandingan antara RST dan PART yang termasuk ke dalam jenis *rule classifier*. Perbandingan performa disajikan dengan menggunakan matrik performa yaitu *accuracy*, *sensitivity*, dan *specificity*. Analisis terhadap matrik performa ini dapat memberikan informasi mengenai algoritme apa yang menghasilkan performa yang

paling baik. Perbedaan signifikansi performa yang dihasilkan masing-masing algoritme akan di uji dengan *paired t-test.* Pada buku ini, uji *paired t-test* dilakukan dengan menggunakan perangkat lunak Microsoft Excel 2007. Sebagai contoh, terdapat nilai performa dari dua algoritme yang diperlihatka pada Tabel 3.6 berikut:

Tabel 3.6 Contoh dua performa klasifikasi

Algoritme A	Algoritme B
95,6	95
90	96
96,3	96,3
92,5	95,6
93,8	94,5

Berdasarkan Tabel 3.6, nilai performa rata-rata untuk algoritme A adalah sebesar 93,64% dan algoritme B sebesar 95,48%, dengan menggunakan perangkat lunak Microsoft Excel, diperoleh hasil yang diperlihatkan pada Tabel 3.8, dimana nilai t statistik yang diperoleh sebesar 1,514. Dengan menggunakan 95% *confidence level* (a=0,05), maka diperoleh nilai z sebesar 2,1318 (tabel t-test) sehingga t < z, maka dapat disimpulkan bahwa algoritme B unggul terhadap algoritme A namun tidak secara signifikan statistik.

Tabel 3.7. Contoh hasil uji *t-test* dengan Microsoft Excel

	Variable 2	*Variable 1*
Mean	95,48	93,64
Variance	0,537	6,373
Observations	5	5
Pearson Correlation	-0,12784155	

Hypothesized Mean Difference	0	
Df	4	
tStat	1,51421098	
P(T<=t) one-tail	0,102268573	
t Critical one-tail	2,131846782	
P(T<=t) two-tail	0,204537145	

BAB IV

PROSESSING DIAGNOSIS

A. Pendahuluan

Berdasarkan langkah-langkah yang dijelaskan pada bab 3, buku ini bertujuan untuk meningkatkan performa pemilihan fitur dan algoritme *machine learning* berdasarkan pengacakan *dataset*. Pemilihan fitur digunakan untuk memilih fitur yang dianggap paling relevan dan penting. Adanya pemilihan fitur diharapkan performa semakin mendekati nilai yang sebenarnya karena estimasi hanya menggunakan fitur yang dianggap paling memberikan kontribusi optimal. Langkah pertama yang dilakukan adalah melakukan pemilihan fitur sebagai input atribut untuk proses klasifikasi Langkah kedua melakukan evaluasi terhadap metode klasifikasi yang diusulkan dan terakhir membandingkan hasil evaluasi metode klasifikasi yang diusulkan.

B. Data preprosessing

Tahap *preprocessing* merupakan tahap persiapan data. Dataset tuberkulosis yang digunakan pada penilitian ini memiliki 30 atribut/fitur berupa gejala medis dan 1 atribut/fitur berupa diagnosis. Dataset tuberculosis meliki jumlah *instance* sebanyak 260 objek data dengan rincian kelas positif berjumlah 141 *instance* dan kelas negatif berjumlah 119 *instance*. Tabel 4.1. memperliatkan dataset yang digunakan pada penlitian ini

Tahap selanjutnya pada data *preprocessing* adalah diskretisasi data numeric. Pada penilitian ini data yang bernilai numerik terdapat pada atribut suhu badan , *erythrocyte, haematocrit, haemoglobin dan leucocyte. Boolean reasoning* digunakan sebagai metode diskretisasi. Tabel 4.2 memperlihatkan hasil diskretisasi menggunakan perangkat lunak rosetta set toolkit.

Tabel 4.1 Dataset tubekulosis dari data rekam medik

NO	JENIS KELAMIN	KELELAHAN	BATUK LEBIH DARI 3 MINGGU	RADANG DI PARU-PARU	TIDAK NYAMAN	NERI SENDI	BATUK BERDARAH	MALAS	NAFSU MAKAN TURUN	BERAT BADAN TURUN	KERINGAT MALAM	NERI DADA	NERI PUNGGUNG	BATUK BERLENDIR	DEMAM	SUHU BADAN	ERYTROCYTE	HAEMATOCRIT	HAEMOGLOBIN	LEUCOCYTE	BUKTI RADIOLOGIS	RIWAYAT TBC SEBELUMNYA	LUKA DI PARUPARU	CAIRAN BERLEBIH PADA SELAPUT PARU-PARU	PEROKOK	KAKI BENGKAK	TANGAN BENGKAK	SESAK NAPAS	MUAL	MUNTAH	DIAGNOSIS
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	36,0	4,5	45	13	6,000	0	0	0	0	0	0	0	0	0	0	TIDAK_TBC
2	0	0	1	0	1	0	1	1	1	1	1	1	0	0	0	36,0	5,0	35	12	12,000	0	0	0	0	0	1	0	0	0	0	TBC
3	0	0	1	0	1	0	1	1	1	0	0	1	1	0	0	36,0	5,2	40	16	5,000	0	0	0	0	0	0	0	0	0	0	TBC
4	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	36,0	4,5	35	13	10,000	1	0	0	0	0	1	0	0	0	0	TBC
5	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	36,5	4,5	35	11	11,000	0	0	0	0	0	0	0	0	0	0	TBC
255	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	35,0	4,0	30	12	12,000	0	0	0	0	0	1	1	1	1	1	TBC
256	0	1	1	1	0	0	0	1	1	1	0	1	0	0	0	36,5	4,5	37	14	12,500	1	0	1	1	0	0	0	0	0	0	TBC
257	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	36,5	4,0	30	15	7,000	0	0	0	0	0	0	0	0	0	0	TIDAK_TBC
258	0	1	1	1	1	0	1	0	1	1	0	1	0	0	0	37,5	4,2	30	15	5,500	1	0	0	0	0	1	1	1	1	1	TBC
259	1	0	0	0	1	0	1	0	0	0	0	0	0	0	1	38,5	3,5	30	12	6,000	0	0	0	0	1	0	0	0	0	0	TIDAK_TBC
260	1	1	1	0	1	1	1	1	1	0	1	0	0	0	1	36,0	4,3	30	13	12,000	0	0	0	0	1	1	1	1	1	1	TBC

Tabel 4.2 Hasil Diskretisasi menggunakan Rosetta Rough Set Tookit

NO	SUHU BADAN	ERYTROCYTE	HAEMATOCRIT	HAEMOGLOBIN	LEUCOCYTE
1	[35.8, 36.3)	[4.4, 4.7)	[42, *)	[13, 14)	[5250, 6250)
2	[35.8, 36.3)	[4.7, *)	[35, 36)	[12, 13)	[11500, *)
3	[35.8, 36.3)	[4.7, *)	[37, 42)	[14, *)	[4750, 5250)
4	[35.8, 36.3)	[4.4, 4.7)	[35, 36)	[13, 14)	[6250, 11500)
5	[36.3, 37.3)	[4.4, 4.7)	[35, 36)	[*, 12)	[6250, 11500)
6	[35.8, 36.3)	[4.7, *)	[*, 35)	[14, *)	[4750, 5250)
7	[35.8, 36.3)	[*, 4.1)	[35, 36)	[12, 13)	[*, 4750)
8	[35.8, 36.3)	[4.7, *)	[35, 36)	[13, 14)	[11500, *)
251	[*, 35.8)	[4.3, 4.4)	[*, 35)	[12, 13)	[*, 4750)
252	[35.8, 36.3)	[*, 4.1)	[37, 42)	[*, 12)	[6250, 11500)
253	[35.8, 36.3)	[4.3, 4.4)	[35, 36)	[13, 14)	[11500, *)
254	[35.8, 36.3)	[4.1, 4.3)	[35, 36)	[13, 14)	[4750, 5250)
255	[*, 35.8)	[*, 4.1)	[*, 35)	[12, 13)	[11500, *)
256	[36.3, 37.3)	[4.4, 4.7)	[37, 42)	[14, *)	[11500, *)
257	[36.3, 37.3)	[*, 4.1)	[*, 35)	[14, *)	[6250, 11500)
258	[37.3, *)	[4.1, 4.3)	[*, 35)	[13, 14)	[5250, 6250)
259	[37.3, *)	[*, 4.1)	[*, 35)	[12, 13)	[5250, 6250)
260	[35.8, 36.3)	[4.3, 4.4)	[*, 35)	[13, 14)	[11500, *)

Berdasar Tabel 4.2 hasil diskretisasi dapat dikelompokkan ke dalam beberapa interval. Inverval diskretisasi dapat dilihat pada Tabel 4.3, dimana untuk atribut haemoglobin memiliki nilai [*, 12) berarti haemoglobin < 12, [13, 14) berarti 13 ≤haemoglobin<14 dan [14, *) berarti suhu badan ≥ 14

Tabel 4.3 Interval diskretisasi

ATRIBUT NUMERIK	NILAI DISKRET			
	1	2	3	4
SUHU BADAN	[*, 35.8)	[35.8, 36.3)	[36.3, *)	
ERYTHROCYTE	[*, 4.1)	[4.1, 4.3)	[4.3, 4.7)	[4.7, *)
HAEMATOCRIT	[*, 35)	[35, 36)	[36, *)	
HAEMOGLOBIN	[*, 13)	[13,14)	[14,*)	
LEUCOCYTE	[*, 4750)	[4750, 5250)	[5250, 6250)	[6250, *)

Dapat dilihat pada tabel 4.3 bahwa proses diskretisasi membagi fitur-fitur dengan tipe data numeric menjadi tipe data nominal dengan interval tertentu. Fitur fitur suhu badan didiskretisasi menjadi 3 interval, fitur *erythrocyte* didiskretisasi menjadi 4 interval, fitur *haematrocit* didiskretisasi menjadi 3 interval, fitur *haemaglobin* didiskretisasi menjadi 3 interval dan fitur *leucocyte* didiskretisasi menjadi 4 interval.

Tabel 4.4 Dataset setelah proses diskretisasi

NO	JENIS KELAMIN	KELELAHAN	BATUK LEBUH DARI 3	RADANG DI PARU-	TIDAK NYAMAN	NYERI SENDI	BATUK BERDARAH	MALAS	NAFSU MAKAN TURUN	BERAT BADAN TURUN	KERINGAT MALAM	NYERI DADA	NYERI PUNGGUNG	BATUK BERLENDIR	DEMAM	SUHU BADAN	ERYTROCYTE	HAEMATOCRIT	HAEMOGLOBIN	LEUCOCYTE	BUKTI RADIOLOGIS	RIWAYAT TBC	LUKA DI PARUPARU	CAIRAN BERLEBIH	PEROKOK	KAKI BENGKAK	TANGAN BENGKAK	SESAK NAPAS	MUAL	MUNTAH	DIAGNOSIS
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	3	3	2	3	0	0	0	0	0	0	0	0	0	0	TIDAK_TBC
2	0	0	1	0	1	0	1	1	1	1	1	0	0	0	0	2	4	2	1	4	0	0	0	0	0	1	0	0	0	0	TBC
3	0	0	1	0	1	0	1	1	1	0	0	1	1	0	0	2	4	3	2	2	0	0	0	0	0	0	0	0	0	0	TBC
4	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	2	3	2	2	4	1	0	0	0	0	1	0	0	0	0	TBC
5	1	0	1	0	1	0	0	0	0	0	0	1	0	0	1	3	3	2	1	4	0	0	0	0	0	0	0	0	0	0	TBC
6	1	1	1	0	1	0	1	1	1	1	0	1	0	0	1	2	4	1	2	2	0	1	0	0	0	1	1	1	0	0	TBC
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	2	1	1	0	0	0	0	0	0	0	0	0	0	TIDAK_TBC
253	1	1	1	1	1	0	1	1	0	1	1	1	0	1	1	2	3	2	2	4	0	1	0	0	1	1	0	1	1	1	TBC
254	1	1	1	0	1	0	1	1	0	0	0	1	0	1	1	2	2	2	2	2	0	1	0	0	0	1	1	1	0	1	TBC
255	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	4	0	0	0	0	0	1	1	0	1	1	TBC
256	0	1	1	1	0	0	0	0	1	1	0	1	0	0	0	3	3	3	2	4	1	0	0	1	0	0	0	0	0	0	TBC
257	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	3	1	1	2	4	0	0	0	0	0	0	0	0	0	0	TIDAK_TBC
258	0	1	1	1	1	0	1	0	1	1	0	1	0	0	0	3	2	1	2	3	1	0	0	0	0	0	0	1	0	0	TBC
259	1	0	0	0	1	0	1	0	0	0	0	0	0	0	1	3	1	1	1	3	0	0	0	0	1	0	0	0	0	0	TIDAK_TBC
260	1	1	1	0	1	1	1	1	1	0	1	0	0	0	1	2	3	1	2	4	0	0	0	0	1	1	1	1	1	1	TBC

C. Pemilihan Fitur

Proses pemilihan fitur dilakukan pada data *training*, kemudian dilakukan reduksi pada data *training* dan *testing* sesuai dengan fitur-fitur yang di pilih.Tabel 4.5 menunjukkan fitur-fitur yang telah dipilih oleh CFS, RST MFS dan RSTCFS.

Tabel 4.5 Fitur-fitur yang telah dipilih oleh CFS, RST MFS dan RSTCFS

NO	CFS	MFS	RST	RSTCFS
1	Batuk lebih dari 3 minggu	Batuk lebih dari 3 minggu	Batuk lebih dari 3 minggu	Batuk lebih dari 3 minggu
2	Batuk berdarah	Batuk berdarah	Batuk berdarah	Batuk berdarah
3	Kelelahan	Kelelahan	Kelelahan	Kelelahan
4	Erytrocyte	Keringat malam	Erytrocyte	Erytrocyte
5	Radang di paru-paru	Demam	Suhu badan	
6	Nafsu makan turun	Berat Badan Turun		
7	Riwayat TBC	Nyeri dada		
8	Muntah	Sesak nafas		
9	Kaki bengkak			

Dapat di lihat berdasarkan Tabel 4.4 bahwa fitur-fitur yang dipilih oleh masing-masing metode berbeda. Metode pemilihan fitur berbasis komputer membuang beberapa fitur yang dianggap penting oleh pakar atau dokter [48]. Tabel 4.6 menunjukan jumlah fitur yang dipilih dan presentase fitur yang direduksi.

Tabel 4.6 Jumlah fitur yang dipilih dan presentase fitur yang direduksi.

CFS	MFS	RST	RSTCFS
9 (70 %)	8 (73 %)	5 (83 %)	4 (86 %)

Pada Tabel 4.6 CFS memilih sebanyak sembilan fitur dengan prosentase yang direduksi adalah 70 %. MFS memilih sebanyak delapan fitur dengan prosentase yang direduksi 73 %, RST memilih sebanyak lima fitur dengan prosentase yang direduksi 83 % dan RSTCFS memilih sebanyak empat fitur dengan prosentase yang direduksi 86 %.

Kombinasi metode pemilihan fitur berbasis komputer yang diusulkan RSTCFS mampu mengurangi fitur lebih sedikit jika dibandingkan dengan CFS, MFS dan RST.

D. Performa Klasifikasi

Input pada alur klasifikasi data adalah data hasil pengacakan. Data hasil pengacakan terbagi menjadi dua yaitu hasil preprocessing (tanpa pemilihan fitur) dan data hasil reduksi. Performa klasifikasi diperoleh dari hasil rata-rata 30 performa pengacakan data dengan *10 fold cross validation.*

1. Performa Klasifikasi dengan 10-*fold cross validation*

Langkah pertama yang akan dibahas adalah performa klasifikasi dengan menggunakan 10-*fold cross validation*. Matriks performa yang digunakan pada pembahasan ini adalah akurasi. Diberikan lima *classifier* yaitu *Naïve Bayes, K-Star, AdaboostM1, PART* dan *J48* untuk memberikan perbandingan akurasi untuk setiap metode pemilihan fitur. Tabel 4.7, 4.8 dan 4.9 menunjukkan akurasi, *sensitivity* dan

specificity sebelum dan sesudah pemilihan fitur. Tanda (+) dan (-) menunjukkan bahwa pemilihan fitur yang digunakan memberikan peningkatan atau penurunan akurasi, *sensitivity* dan *specificity* yang signifikan secara statistik jika dibandingkan dengan klasifikasi tanpa pemilihan fiur. Dengan 30 kali pengacakan (df = 29) menggunakan 95% *confidence level* (a=0,05) diperoleh nilai z sebesar 1,6991 (tabel t-test) sehingga apabila t < z maka perbandinganya signifikan.

Tabel 4.7. Akurasi (%) skema pemilihan fitur menggunakan 10 *fold* CV

Klasifikasi	FULL FEATURE	CFS	MFS	RST	RSTCFS
Naïve Bayes	97,6923	98,6545+	96,5385-	96,9231-	**98,6667+**
K-Star	97,9225	98,0383+	96,6154-	96,7564-	98,1154+
AdaboostM1	96,5513	97,1189+	96,7949+	97,1026+	97,6667+
PART	96,5242	97,1132+	96,5641+	96,5897+	97,4231+
J48	95,8718	96,0385+	95,6469-	95,8836+	97,4359+

(+)/(-) adalah peningkatan atau penuruanan yang signifikan secara statistik

Pada penggunaan *Naive bayes dan K-Star*, metode pemilihan fitur pada CFS dan RSTCFS mampu meningkatkan akurasi secara signifikan tetapi untuk MFS dan RST mengalami penurunan akurasi secara signifikan. Untuk penggunaan *AdaboostM1*, *PART dan J48*, semua metode pemilihan fitur pada CFS, MFS, RST dan RSTCFS mampu meningkatkan akurasi secara signifikan.

Pada penggunaan *naive bayes*, metode RSTCFS memiliki performa tertinggi di bandingkan dengan metode lainya sebesar 98,6667 %.

Tabel 4.8 *Sensitivity* (%) skema pemilihan fitur menggunakan 10 *fold* CV

Klasifikasi	FULL FEATURE	CFS	MFS	RST	RSTCFS
Naïve Bayes	97,8723	98,5584+	96,4539-	97,1631-	**98,5809+**
K-Star	97,0199	97,2567+	95,8865-	96,1466-	97,9152+
AdaboostM1	95,6717	96,4104+	95,7683+	97,6574+	97,0913+
PART	96,2825	97,0196+	95,9811-	96,5241+	97,6295+
J48	95,7447	96,0952+	96,1369+	95,8422+	96,3121+

(+)/(-) adalah peningkatan atau penuruanan yang signifikan secara statistik

Pada penggunaan *naive bayes dan k-star*, metode pemilihan fitur pada CFS dan RSTCFS mampu meningkatkan *sensitivity* secara signifikan statistik tetapi untuk MFS dan RST mengalami penurunan *sensitivity* secara signifikan statistik. Untuk penggunaan adaboostM1 dan J48, semua metode pemilihan fitur pada CFS, MFS, RST dan RSTCFS mampu meningkatkan *sensitivity* secara signifikan statistik. Untuk penggunaan PART , metode pemilihan fitur pada CFS, RST dan RSTCFS mampu meningkatkan *sensitivity* secara signifikan statistik kecuali MFS mengalami penurunan secara signifikan statistik.

Tabel 4.9 *Specificity* (%) skema pemilihan fitur menggunakan 10 *fold* CV

Klasifikasi	FULL FEATURE	CFS	MFS	RST	RSTCFS
Naïve Bayes	97,4790	98,3056+	95,8333-	96,6387-	98,3333+
K-Star	96,5584	96,8292+	95,2391-	95,5314-	97,5679+
AdaboostM1	95,0250	95,8534+	95,1363+	97,2061+	96,5936+
PART	95,6445	96,5246+	95,3556-	95,9167+	97,2979+
J48	95,0185	95,4300+	95,4034+	95,1208+	95,6983+

(+)/(-) adalah peningkatan atau penuruanan yang signifikan secara statistik

Pada penggunaan *Naive bayes dan K-Star*, metode pemilihan fitur pada CFS dan RSTCFS mampu meningkatkan *specificity* secara signifikan tetapi untuk MFS dan RST mengalami penurunan *specificity* secara signifikan. Untuk penggunaan AdaboostM1,dan J48 , semua metode pemilihan fitur pada CFS, MFS, RST dan RSTCFS mampu meningkatkan *specificity* secara signifikan. Untuk penggunaan PART , metode pemilihan fitur pada CFS, RST dan RSTCFS mampu meningkatkan *specificity* secara signifikan kecuali MFS mengalami penurunan secara signifikan.

Tabel 4.10 AUC skema pemilihan fitur menggunakan 10 *fold* CV

Klasifikasi	FULL FEATURE	CFS	MFS	RST	RSTCFS
Naïve Bayes	97,6757%	98,6213% +	96,5463%-	96,9009% -	98,6680%+
K-Star	98,0058%	98,6213% +	98,1084%+	96,6828%-	98,1444%+
AdaboostM1	96,6331%	97,1855%+	96,8898%+	96,9827%+	97,6739%+
PART	96,5485%	97,1279%+	96,6180%+	96,5961%+	97,4274%+
J48	95,8835%	96,0343%+	95,6117%-	95,8900%+	97,4416%+

2. Peringkat Skema Pemilihan Fitur

Perbandingan secara menyeluruh terhadap semua metode pemilihan fitur perlu dilakukan untuk mengetahui metode pemilihan fitur yang unggul pada diagnosis penyaki TBC. Hall dan Holmes [44] menggunakan peringkat serta terminologi *wins* dan *loses* untuk membandingkan performa metode pemilihan fitur. Terminologi *wins* menunjukkan jumlah performa suatu skema pemilihan fitur unggul secara signifikan statistik dibandingkan dengan skema pemilihan fitur yang lain ketika *classifier* tertentu digunakan, sedangkan *loses* menunjukkan

sebaliknya. Peringkat dari masing-masing skema pemilihan fitur diperoleh berdasarkan hasil pengurangan jumlah *wins* dan *loses*. Pada buku ini perbandingan dilakukan dengan membandingkan akurasi setiap metode pemilihan fitur secara berpasangan pada masing-masing dataset dan *classifier*. Jumlah akurasi dari skema pemilihan fitur yang unggul dan kalah secara signifikan statistik dibandingkan dengan skema pemilihan fitur dihitung. Kemudian peringkat dari setiap skema pemilihan fitur disajikan untuk setiap penggunaan *classifier*.

Peringkat setiap skema pemilihan fitur diperoleh dari pengurangan jumlah *wins* dan *loses* dari Tabel 4.7 dan ditunjukkan pada Tabel 4.11. Berdasarkan hasil yang ditunjukkan pada Tabel 4.11, penggunaan Naïve Bayes, metode pemilihan fitur RSTCFS dan CFS memiliki peringkat yang tertinggi dengan tiga kali *wins* dan nol *loses*. Sedangkan MFS dan RST memiliki jumlah *wins* yang lebih sedikit dibandingkan dengan jumlah *loses.*

Tabel 4.11 Peringkat Skema pemilihan fitur

ALGORITMA	**SKEMA**	**WINS**	**LOSSES**	**WINS –LOSSES**
NAIVE BAYES	RSTCFS	3	0	3
	CFS	3	0	3
	RST	0	4	-4
	MFS	1	3	-2
K-STAR	RSTCFS	3	0	3
	CFS	2	1	1
	RST	1	3	-2
	MFS	0	4	-4
ADABOOSTM1	RSTCFS	4	0	4
	CFS	1	2	-1

ALGORITMA	SKEMA	WINS	LOSSES	WINS –LOSSES
	RST	3	0	3
	MFS	1	3	-2
PART	RSTCFS	3	0	3
	CFS	2	0	2
	RST	0	2	-2
	MFS	0	2	-2
J48	RSTCFS	3	1	2
	CFS	2	0	2
	RST	1	3	-2
	MFS	0	4	-4

Pada penggunaan K-Star, metode pemilihan fitur RSTCFS memiliki peringkat yang tertinggi dengan tiga kali *wins* dan nol *loses*. CFS berada pada peringkat kedua dengan dua kali *wins* dan nol *loses*. Sedangkan RST dan MFS, keduanya memiliki jumlah *wins* yang lebih sedikit dari pada jumlah *loses*.

Pada penggunaan AdaBoostM1, metode RSTCFS memiliki peringkat tertinggi dengan empat kali *wins* dan nol *loses*. RST berada pada peringkat kedua dengan tiga kali *wins* dan nol *loses.* Sedangkan CFS dan MFS, keduanya memiliki jumlah *wins* yang lebih sedikit dari pada jumlah *loses*.

Pada penggunaan PART, metode RSTCFS memiliki peringkat tertinggi dengan tiga kali *wins* dan nol *loses*. CFS berada pada peringkat kedua dengan dua kali *wins* dan nol kali *loses.* Sedangkan MFS dan RST, keduanya memiliki jumlah *wins* yang lebih sedikit dari pada jumlah *loses*.

Pada penggunaan J48, metode RSTCFS memiliki peringkat tertinggi dengan tiga kali *wins* dan satu kali *loses.* CFS berada pada peringkat kedua dengan dua kali *wins* dan nol *loses.*

RST memiliki jumlah *wins* dan *loses* yang sama. Sedangkan RST dan MFS, keduanya memiliki jumlah *wins* yang lebih sedikit dari pada jumlah *loses*. Tabel 4.12 menunjukkan performa (akurasi, *sensitivity*, *specificity*, dan AUC) dari masing-masing *classifier* beserta metode pemilihan fitur dengan peringkat terbaik.

Tabel 4.12 Performa yang terbaik

SKEMA	AKURASI (%)	SENSITIVITY	SPECIFICITY	AUC
RSTCFS + Naive Bayes	98,6667	0,9858	0,9833	0,9866
CFS + Naive Bayes	98,6545	0,9855	0,9830	0,9862
RSTCFS + K-Star	98,1154	0,9791	0,9756	0,9814
RSTCFS + AdaboostM1	97,6667	0,9709	0,9659	0,9767
RSTCFS + PART	97,4231	0,9762	0,9729	0,9742
RSTCFS + J48	97,4359	0,9631	0,9569	0,9744
CFS + J48	96,0385	0,9609	0,9543	0,9603

Hasil yang disajikan pada Tabel 4.12 merupakan rata-rata dari 30 performa hasil 30 pengacakan data. Berdasarkan Tabel 4.12, skema RSTCFS + Naive Bayes memberikan akurasi dan *sensitivity* dan *AUC* tertinggi.

3. Penggabungan pemilihan fitur berbasis pakar medis dan komputer

Penggabungan metode pemilihan fitur berbasis pakar medis dan komputer perlu dilakukan pada buku ini. Hal ini

bertujuan untuk menghindari terhapusnya fitur-fitur yang dianggap penting oleh pakar medis [48]. Tabel 4.13 menunjukkan fitur-fitur terpilih dari hasil penggabungan dua metode ini.

Tabel 4.13 Fitur-fitur terpilih hasil penggabungan pemilihan fitur berbasis pakar medis dan komputer

NO	MFS+ CFS	MFS + RST	MFS + RSTCFS
1	Batuk lebih dari 3 minggu	Batuk lebih dari 3 minggu	Batuk lebih dari 3 minggu
2	Batuk berdarah	Batuk berdarah	Batuk berdarah
3	Kelelahan	Kelelahan	Kelelahan
4	Erytrocyte	Keringat malam	Erytrocyte
5	Radang di paru-paru	Demam	Keringat Malam
6	Nafsu makan turun	Berat Badan Turun	Demam
7	Riwayat TBC	Nyeri dada	Berat Badan Turun
8	Muntah	Sesak nafas	Nyeri dada
9	Kaki bengkak	Erytrocyte	Sesak nafas
10	Keringat malam	Suhu badan	
11	Demam		
12	Berat Badan Turun		
13	Nyeri dada		
14	Sesak nafas		

Dapat di lihat berdasarkan Tabel 4.13 bahwa fitur-fitur yang dipilih oleh masing-masing metode berbeda. Tabel 4.14 menunjukan jumlah fitur yang dipilih dan presentase fitur yang direduksi.

Tabel 4.14 Jumlah fitur yang di pilih dan presentasi fitur yang direduksi

MFS+ CFS	MFS + RST	MFS + RSTCFS
14 (53 %)	10 (67 %)	9 (70 %)

Pada Tabel 4.14 MFS+CFS memilih sebanyak empat belas fitur dengan prosentase yang direduksi adalah 53 %. MFS+RST memilih sebanyak sepuluh fitur dengan prosentase yang direduksi 67 %, dan MFS+RSTCFS memilih sebanyak sembilan fitur dengan prosentase yang direduksi 70 %.

Kombinasi metode pemilihan fitur berbasis pakar medis dan berbasis komputer yang diusulkan MFS+RSTCFS mampu mengurangi fitur lebih sedikit jika dibandingkan dengan MFS+CFS dan MFS+RST.

Tabel 4.15, 4.16 dan 4.17 menunjukkan akurasi, sensitivity dan specifity penggabungan pemilihan fitur berbasis pakar medis dan kompuer. Tanda (+) dan (-) menunjukkan bahwa pemilihan fitur yang digunakan memberikan peningkatan atau penurunan akurasi, sentivity dan specivity yang signifikan secara statistik jika dibandingkan dengan klasifikasi sebelum penggabungan fitur.

Tabel 4.15 Akurasi (%) skema pemilihan fitur menggunakan 10 *fold* CV

Klasifikasi	MFS	MFS + CFS	MFS +RST	MFS + RSTCFS
Naïve Bayes	96,5385	**98,0769+**	96,6154+	96,5385
K-Star	96,6154	97,8129+	96,5769-	96,1923-
AdaboostM1	96,7949	97,6410+	96,4231-	96,7308-

Klasifikasi	MFS	MFS + CFS	MFS +RST	MFS + RSTCFS
PART	96,5641	97,3978+	95,6154+	97,0018+
J48	95,6469	95,5000-	95,3462-	95,1923-

(+)/(-) adalah peningkatan atau penuruanan yang signifikan secara statistik

Pada penggunaan *Naive bayes dan PART*, semua metode pemilihan fitur pada MFS+CFS, MFS+RST dan MFS+RSTCFS mampu meningkatkan akurasi secara signifikan. Untuk penggunaan K-Star dan AdaboostM1 pada metode pemilihan fitur pada MFS+CFS mampu meningkatkan akurasi secara signifikan tetapi pada metode pemilihan MFS+RST dan MFS+RSTCFS mengalami penurunan akurasi secara signifikan. Pada penggunaan *J48*, semua metode pemilihan fitur pada MFS+CFS, MFS+RST dan MFS+RSTCFS mengalami penurunan akurasi secara signifikan.

Pada penggunaan *naive bayes*, metode MFS+CFS memiliki performa tertinggi di bandingkan dengan metode lainnya sebesar 98,0769 %.

Tabel 4.16 *Sensitivity* (%) skema pemilihan fitur menggunakan 10 *fold* CV

Klasifikasi	MFS	MFS + CFS	MFS +RST	MFS + RSTCFS
Naïve Bayes	96,4539	97,8723+	96,5248+	96,4539
K-Star	95,8865	96,5721+	96,4539+	95,1064-
AdaboostM1	95,7683	96,7376+	95,7447-	95,8156+
PART	95,9811	96,7139+	95,5319-	96,1023+
J48	96,1369	94,9882-	95,9574-	95,8156-

(+)/(-) adalah peningkatan atau penuruanan yang signifikan secara statistik

Pada penggunaan *Naive bayes* semua metode pemilihan fitur pada MFS+CFS, MFS+RST dan MFS+RSTCFS mampu meningkatkan *sensitivity* secara signifikan. Untuk penggunaan K-Star pada metode pemilihan fitur pada MFS+CFS dan MFS+RST mampu meningkatkan *sensitivity* secara signifikan tetapi pada metode pemilihan MFS+RSTCFS mengalami penurunan *sensitivity* secara signifikan. Untuk penggunaan AdaboostM1 dan PART pada metode pemilihan fitur pada MFS+CFS dan MFS+RSTCFS mampu meningkatkan *sensitivity* secara signifikan tetapi pada metode pemilihan MFS+RST mengalami penurunan *sensitivity* secara signifikan. Pada penggunaan *J48*, semua metode pemilihan fitur pada MFS+CFS, MFS+RST dan MFS+RSTCFS mengalami penurunan *sensitivity* secara signifikan.

Tabel 4.17 *Specificity* (%) skema pemilihan fitur menggunakan 10 *fold* CV

Klasifikasi	MFS	MFS + CFS	MFS +RST	MFS + RSTCFS
Naïve Bayes	95,8333	97,5000+	95,9173+	95,8333
K-Star	95,2391	96,0890+	95,8421+	94,3862=
AdaboostM1	95,1363	96,2438+	95,0722-	95,1854+
PART	95,3556	96,1935+	94,7630-	95,5004+
J48	95,4034	94,1911-	95,1900-	95,0344-

(+)/(-) adalah peningkatan atau penuruanan yang signifikan secara statistik

Pada penggunaan *Naive bayes* semua metode pemilihan fitur pada MFS+CFS, MFS+RST dan MFS+RSTCFS mampu meningkatkan *specificity* secara signifikan. Untuk penggunaan K-Star pada metode pemilihan fitur pada MFS+CFS dan MFS+RST mampu meningkatkan *specificity* secara signifikan

tetapi pada metode pemilihan MFS+RSTCFS mengalami penurunan *specificity* secara signifikan. Untuk penggunaan AdaboostM1 dan PART pada metode pemilihan fitur pada MFS+CFS dan MFS+RSTCFS mampu meningkatkan *specificity* secara signifikan tetapi pada metode pemilihan MFS+RST mengalami penurunan *specificity* secara signifikan. Pada penggunaan *J48*, semua metode pemilihan fitur pada MFS+CFS, MFS+RST dan MFS+RSTCFS mengalami penurunan *specificity* secara signifikan.

Tabel 4.18 AUC skema pemilihan fitur menggunakan 10 *fold* CV

Klasifikasi	MFS	MFS + CFS	MFS +RST	MFS + RSTCFS
Naïve Bayes	0,9583	0,9809	0,9662	0,9655
K-Star	0,9523	0,9792	0,9659	0,9629
AdaboostM1	0,9513	0,9772	0,9649	0,9682
PART	0,9535	0,9746	0,9562	0,9708
J48	0,9540	0,9555	0,9529	0,9513

E. Pembahasan

1. Performa Klasifikasi dan Peringkat Skema Pemilihan Fitur

Hasil performa pemilihan fitur yang diperoleh dari penggunaan 10-*fold cross validation* menunjukkan bahwa secara umum metode pemilihan fitur mampu meningkatkan akurasi dari proses klasifikasi. Peningkatan akurasi yang terjadi telah diuji menggunakan uji *paired t-Test* dan hasil yang diberikan adalah signifikan. Perbandingan secara menyeluruh dapat dilihat dari peringkat masing –masing metode pemilihan fitur. Berdasarkan Tabel 4.7, diketahui bahwa RSTCFS unggul

pada penggunaan *classifier* naïve bayes, k-star, adaBoostM1, PART, dan J48.

Naïve Bayes mengasumsikan bahwa distribusi probabilitas untuk setiap fitur saling independen sehingga performa klasifikasi dapat ditingkatkan dengan menghapus fitur-fitur *redundant* [45]. Oleh karena itu, RSTCFS menempati peringkat pertama pada penggunaan Naïve Bayes. Hal ini disebabkan RSTCFS mampu menghapus fitur-fitur *redundant* lebih banyak dibandingkan skema seleksi
fitur lainnya.

Subset fitur yang dihasilkan oleh RSTCFS merupakan subset fitur dengan korelasi yang tinggi antara fitur dengan kelas. Hal ini memberikan keuntungan tersendiri pada *classifier* berbasis pohon keputusan seperti AdaBoostM1, PART, dan J48. Pohon keputusan memiliki kemampuan untuk mengidentifikasi korelasi

antara fitur dengan kelas [35]. Oleh karena itu, RSTCFS menempati peringkat pertama pada penggunaan *classifier* berbasis pohon keputusan. RSTCFS lebih unggul dari pada CFS karena dengan data hasil pemilihan fitur RSTCFS, pohon keputusan yang dibangun oleh *classifier* lebih sederhana dan dapat terhindar dari kendala *over fitting*.

Pada Tabel 4.7 dapat dilihat bahwa RST menduduki peringkat terbawah pada penggunaan naïve bayes, k-star, adaboostM1, PART dan J48 menduduki peringkat terbawah. RST menghapus fitur *redundant* dan memilih *subset* fitur yang memiliki *discernibilty* sama dengan himpunan fitur awal. Namun, subset fitur yang diperoleh belum menjamin peningkatan performa pemilihan fitur.

MFS menyeleksi fitur berdasarkan pengetahuan pakar atau dokter. Fitur-fitur yang dipilih dokter adalah fitur yang dianggap sangat berperan dalam diagnosis penyakit TBC. Hal ini menunjukkan bahwa kemampuan prediksi fitur-fitur terhadap kelas tidak diperhatikan sehingga akurasi proses klasifikasi menjadi rendah.

Tabel 4.12 menunjukkan performa (akurasi, *sensitivity*, *specificity*, dan AUC) dari masing-masing *classifier* beserta metode pemilihan fitur dengan peringkat terbaik. Berdasarkan Tabel 4.12, dapat dilihat bahwa skema RSTCFS + Naive Bayes memberikan akurasi, *sensitivity, specificity dan AUC* tertinggi. Jika dilihat dari matriks performa AUC, skema RSTCFS + Naïve Bayes memiliki AUC yang lebih unggul secara signifikan statistik diantara semua skema pembelajaran.

Luar area di bawah kurva ROC (AUC) dapat digunakan untuk menentukan bagus atau tidaknya suatu *classifier*. Hal ini disebabkan, nilai AUC sudah mewakili akurasi, *sensitivity*, dan *specificity*. Semakin dekat nilai AUC dengan nilai satu maka semakin baik model klasifikasi yang dibentuk oleh suatu *classifier*
[43]. Oleh karena itu, pada buku ini digunakan AUC untuk menentukan skema pembelajaran yang terbaik.

Skema RSTCFS+Naïve Bayes memiliki AUC yang paling dekat dengan nilai satu yaitu sebesar 0,9866. Nilai AUC ini juga lebih unggul secara signifikan statistik diantara semua skema pembelajaran. Oleh karena itu, RSTCFS + Naïve Bayes menjadi skema yang terbaik untuk mendiagnosis penyakit TBC.

Tabel 4.19 Tabel Performa RSTCFS + Naive Bayes

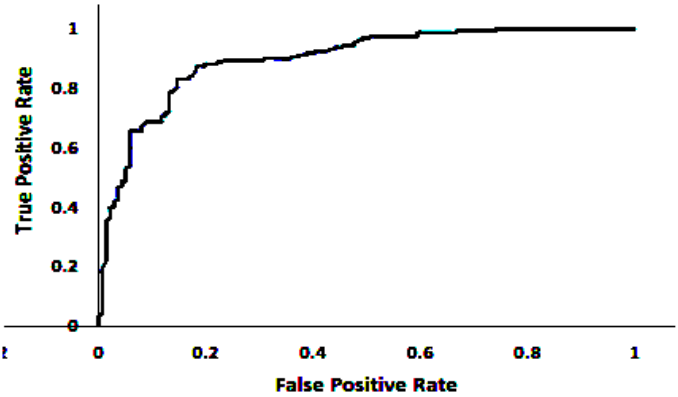
ACAK	PEMILIHAN FITUR DENGAN RSTCFS			
	NAÏVE BAYES			
	Perfoma Klasifikasi			
	SENSITIVITY	*SPECIFICITY*	*FP rate*	*ROC*
1	99,2908%	99,1525%	1,6807%	98,8051%
2	99,2908%	99,1525%	1,6807%	98,8051%
3	97,1631%	96,7213%	0,8403%	98,1614%
4	98,5816%	98,3333%	0,8403%	98,8706%
5	98,5816%	98,3333%	0,8403%	98,8706%
6	99,2908%	99,1453%	2,5210%	98,3849%
7	98,5816%	98,3333%	0,8403%	98,8706%
8	97,8723%	97,5207%	0,8403%	98,5160%

9	97,1631%	96,7213%	0,8403%	98,1614%
10	98,5816%	98,3333%	0,8403%	98,8706%
11	98,5816%	98,3333%	0,8403%	98,8706%
12	98,5816%	98,3333%	0,8403%	98,8706%
13	98,5816%	98,3333%	0,8403%	98,8706%
14	97,8723%	97,5207%	0,8403%	98,5160%
15	98,5816%	98,3333%	0,8403%	98,8706%
16	99,2908%	99,1453%	2,5210%	98,3849%
17	99,2908%	99,1379%	3,3613%	97,9647%

18	99,2908%	99,1453%	2,5210%	98,3849%
19	99,2908%	99,1453%	2,5210%	98,3849%
ACAK	*SENSITIVITY*	*SPECIFICITY*	*FP rate*	*ROC*
20	99,2908%	99,1453%	2,5210%	98,3849%
21	99,2908%	99,1453%	2,5210%	98,3849%
22	98,5816%	98,3333%	0,8403%	98,8706%
23	98,5816%	98,3333%	0,8403%	98,8706%
24	98,5816%	98,3333%	0,8403%	98,8706%

25	98,5816%	98,3333%	0,8403%	98,8706%
26	98,5816%	98,3333%	0,8403%	98,8706%
27	98,5816%	98,3333%	0,8403%	98,8706%
28	98,5816%	98,3333%	0,8403%	98,8706%
29	98,5816%	98,3333%	0,8403%	98,8706%
30	98,5816%	98,3333%	0,8403%	98,8706%
Rata-rata	98,6525%	98,4155%	1,3165%	98,6680%

Gambar 4.1 menunjukkan contoh kurva ROC untuk RSTCFS + Naïve Bayes.



Gambar 4.1 ROC untuk skema RSTCFS+Naïve Bayes

Berdasarkan Gambar 4.1, ketika *specificity* sama dengan 0,9833, nilai *sensitivity* sudah lebih dari 0,9858. Oleh karena itu, *sensitivity* dari RSTCFS+Naïve Bayes lebih baik dari pada *specificity*.

2. Penggabungan Pemilihan Fitur Berbasis Pakar Medis dan Komputer

Penggabungan pemilihan fitur berbasis pakar medis (MFS) dan komputer mampu memberikan peningkatan performa klasifikasi. Pada penggunaan MFS, fitur-fitur yang dipilih tidak didasarkan pada fitur yang memiliki kemampuan prediksi yang baik terhadap kelas. Hal ini disebabkan fitur yang dipilih merupakan fitur yang dianggap oleh pakar atau dokter sangat berpengaruh terhadap diagnosis TBC. Sedangkan pemilihan fitur berbasis komputer memilih fitur dengan kemampuan prediksi yang baik terhadap kelas. Oleh karena itu

ketika MFS digabung dengan metode-metode pemilihan fitur berbasis komputer, maka performa diagnosis akan meningkat. Dapat dilihat pada Tabel 4.15, 4.17 dan 4.18, bahwa performa (akurasi, *sensitivity*, *specificity*, dan AUC) penggabungan MFS dengan RSTCFS dan CFS meningkat secara signifikan statistik jika dibandingkan dengan hanya menggunakan MFS. Tabel 4.20 menunjukkan persentase kenaikan performa (AUC) dari penggabungan MFS dengan RSTCFS, CFS, dan ketika dibandingkan dengan hanya menggunakan MFS.

Tabel 4.20 Persentase kenaikan performa AUC

Skema	Persentase Kenaikan AUC
MFS+CFS+ Naive Bayes	2,27%
MFS+RSTS+ Naive Bayes	0,79%
MFS+RSTCFS+ Naive Bayes	0,72%
MFS+CFS+ K-Star	2,69%
MFS+RSTS+ K-Star	1,36%
MFS+RSTCFS+ K-Star	1,06%
MFS+CFS+ AdaboostM1	6,36%
MFS+RSTS+ AdaboostM1	5,13%
MFS+RSTCFS+ AdaboostM1	5,46%
MFS+CFS+ PART	2,11%
MFS+RSTS+ PART	0,27%
MFS+RSTCFS+ PART	1,73%
MFS+CFS+ J48	0,15%
MFS+RSTS+ J48	0,11%
MFS+RSTCFS+ J48	0,27%

Dapat diketahui bahwa skema MFS+CFS+ AdaboostM1 memberikan persentase kenaikan AUC tertinggi. MFS+RSTCFS+ J48 memberikan persentase kenaikan AUC terendah.

F. Rangkuman

Pada bagian ini dibahas mengenai hasil dari data *preprocessing*, pemilihan fitur, pengacakan data, klasifikasi dan evaluasi performa. Pada *preprocessing* dilakukan diskretisasi data yaitu pada fitur suhu badan, *erythrocyte*, *haematrocit*, *haemaglobin* dan *leucocyte.*

Selanjutnya hasil pemilihan fitur memberikan gambaran bahwa terdapat beberapa fitur yang dianggap oleh dokter sebagai faktor penting namun tidak dipilih oleh pemilihan fitur berbasis komputer. Kemudian pada hasil performa klasifikasi (sebelum dan setelah pemilihan fitur menunjukkan bahwa metode pemilihan fitur berbasis komputer yaitu RSTCFS unggul pada penggunaan *clasissifier* naive bayes, K-Star, AdaboostM1, PART dan J8.

RSTFS+Naive bayes menjadi skema pembelajaran yang terbaik untuk mendiagnosis penyakit TBC dengan niai AUC sebesar 0,9866. Kemudian performa (akurasi, sensitivity dan specificity) penggabungan MFS dengan CFS dengan klasifikasi naive bayes menduduki performa akurasi terbaik sebesar 98,0769 % sehingga dipilih sebagai pemilihan fitur RSTCFS+Naive Bayes dapat di aplikasikan sebagai metode komputer dalam pemeriksaan awal terhadap pasien yang terindikasi mempunyai penyakit tuberkulosis

Dalam penelitan ini hanya 2 metode pemilihan fitur berbasis komputer yaitu CFS dan RST sehingga hanya 2

informasi proses pemilihan fitur untuk mengetahui apakah ada peningkatan performa klasifikasi dari masing-masing algoritme. Kemudian data untuk proses pembentukan model classifier masih sedikit sehingga model yang dihasilkan oleh *classifier* belum optimal. Optimasi model *classifier* diperlukan untuk meningkatkan performa klasifikasi. Pada buku ini nilai *sensitivity* selalu lebih besar jika dibandingkan dengan nilai *specificity* untuk semua skema pembelajaran. Hal ini menunjukkan bahwa kesalahan prediksi yang terjadi ketika pasien sehat tetapi diprediksi positif sakit (*false positive*) cukup banyak. Dalam diagnosis penyakit tuberkulosis, kesalahan prediksi ini harus diminimalkan untuk menghindari diagnosis *invasive* seperti tuberkulosis. Oleh karena itu, pada buku selanjutnya dapat dikembangkan skema pembelajaran yang mampu meminimalkan nilai *false positive*.

BAB V

KESIMPULAN

Deteksi dini terhadap keberadaan penyakit tuberkulosis merupakan hal yang perlu dilakukan untuk mencegah penularan penyakit. Memahami kegunaan *data mining* untuk membantu proses diagnosis penyakit TBC adalah suatu hal yang sangat penting. Buku ini memberikan salah satu contoh penggunaan metode data mining untuk mendiagnosis penyakit tuberkulosis. Cara diagnosis penyakit tuberkulosis dengan metode data mining adalah dengan memilih fitur dan melakukan klasifikasi pada dataset tuberkulosis RS PKU Muhammadiyah Gombong dengan algoritme *classifier* yang digunakan adalah *naive bayes, k-Star, adaboostM1, PART dan J.48.* Berdasarkan basil yang telah diperoleh dapat diambil beberapa kesimpulan sebagai berikut.

1. Kombinasi metode pemilihan fitur berbasis komputer RSTCFS mampu meningkatkan akurasi secara signifikan dan unggul pada penggunaan *classifier naive bayes, k-star*, adaboostM1, PART dan J48. RSTCS+*Naive Bayes* menjadi skema yang terbaik untuk diagnosis penyakit tuberkulosis karena memiliki AUC yang paling dekat dengan nilai satu yaitu sebesar 0,9866. Nilai AUC ini juga lebih unggul secara signifikan dengan semua penggabungan skema pemilihan fitur berbasis komputer dan pakar medis (MFS) sehingga pemilihan fitur RSTCFS+*Naive Bayes* dapat di aplikasikan sebagai metode komputer dalam pemeriksaan

awal terhadap pasien yang terindikasi mempunyai penyakit tuberkulosis.

2. Dalam penggabungan pemilihan fitur berbasis komputer dan pakar medis performa nilai akurasi tertinggi adalah MFS+CFS+*Naive Bayes* sebesar 98,0769 %.

DAFTAR PUSTAKA

[1] G. Henry, *Anatomy of the human body*. Philadelphia: Lea & Febiger, 1918.

[2] A. Konstantinos, "Testing for tuberculosis," *Aust. Prescr.*, vol. Volume 33 |, no. 1, pp. 12–18, Feb. 2010.

[3] Menteri Kesehatan Republik Indonesia, "Pedoman Penanggulangan TuberkulosisS (TB)." Kemenkes Nomor 364/MENKES/SK/V/2009, Mei-2009.

[4] Kenyorini, Suradi, and E. Surjanto, "Uji Tuberkulin," *Bagian Pulmonologi Dan Kedokt. Respirasi FK UNS RSUD Dr Moewardi Surak.*, vol. Jurnal Tuberkulosis Indonesia, Vol. 3 No. 2.

[5] M. Yunus and Setyowibowo, "Aplikasi Sistem Pendukung Keputusan Diagnosa Penyakit Paru-paru dengan Metode Forward Chaining," *STMIK PPKIA Pradnya Paramita Malang*, vol. Jurnal Teknologi Informasi Vol 2, no. 2.

[6] M. Shukla and S. Agarwal, "Hybrid approach for tuberculosis data classification using optimal centroid selection based clustering," in *2014 Students Conference on Engineering and Systems (SCES)*, 2014, pp. 1–5.

[7] A. Bakar and F. Febriyani, "Rough Neural Network Model For Tuberculosis Patient Categorization," *Proc. Int. Conf. Electr. Eng. Inform.*, vol. Vol.2: Hal. 760-763, 2007.

[8] Y. Benfu, S. HongMei, S. Ye, L. Xiuhui, and Z. Bin, "Study on the Artificial Neural Network in the Diagnosis of Smear Negative Pulmonary Tuberculosis," in *2009 WRI World Congress on Computer Science and Information Engineering*, 2009, vol. 5, pp. 584–588.

[9] A. T, S. Natarajan, and K. N. B. Murthy, "A Data Mining Approach to the Diagnosis of Tuberculosis by Cascading Clustering and Classification," *ArXiv11081045 Cs*, Aug. 2011.

[10] T. Uçar and A. Karahoca, "Predicting existence of Mycobacterium tuberculosis on patients using data mining approaches," *Procedia Comput. Sci.*, vol. 3, pp. 1404–1411, 2011.

[11] T. Uçar, A. Karahoca, and D. Karahoca, "Tuberculosis disease diagnosis by using adaptive neuro fuzzy inference system and rough sets," *Neural Comput. Appl.*, vol. 23, no. 2, pp. 471–483, Aug. 2013.

[12] T. Asha, S. Natarajan, and K. N. B. Murthy, "Notice of RetractionDiagnosis of tuberculosis using ensemble methods," in *2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, 2010, vol. 8, pp. 409–412.

[13] Suhardi, "Diagnosis berbasis komputer untuk penyakit Tuberkolusis." Tesis Program Pasca Sarjana Fakultas Teknik Universitas Gajah Mada, Mar-2015.

[14] M. G. F. Costa, C. F. F. C. Filho, J. F. Sena, J. Salem, and M. O. de Lima, "Automatic identification of mycobacterium tuberculosis with conventional light microscopy," in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2008, pp. 382–385.

[15] A. Ali, M. Elfaki, and D. Norhayati, "Using Naïve Bayes and Bayesian Networkfor Prediction of Potential Problematic Cases in Tuberculosis," *Int. J. Inform.*

Commun. Technol. IJ-ICT, vol. 1, no. 2, pp. 63–71, Sep. 2012.

[16] E. Winarko, "Review on Data Mining Methods for Tuberculosis Diagnosis (PDF Download Available)." [Online]. Available: https://www.research gate.net/publication/259787873_Review_on_Data_Minin g_Methods_for_Tuberculosis_Diagnosis. [Accessed: 20-Oct-2016].

[17] S. Kumar, R. P. Bajpai, and L. M. Bharadwaj, "Diagnosis of tuberculosis based on BioMEMS," in *Proceedings of International Conference on Intelligent Sensing and Information Processing, 2004*, 2004, pp. 237–242.

[18] S. S. Naga, *Buku panduan lengkap ilmu penyakit dalam*. Yogyakarta: DIVA Press., 2012.

[19] "- Tuberkulosis -." [Online]. Available: http://www.klikpdpi.com/konsen sus/tb/tb.html. [Accessed: 01-Dec-2016].

[20] Kenyorini, Suradi, and E. Surjanto, "Uji Tuberkulin," *Bagian Pulmonologi Dan Kedokt. Respirasi FK UNS RSUD Dr Moewardi Surak.*, vol. Jurnal Tuberkulosis Indonesia, Vol. 3 No. 2.

[21] M. Khajehei and F. Etemady, "Data Mining and Medical Research Studies," in *Modelling and Simulation 2010 Second International Conference on Computational Intelligence*, 2010, pp. 119–122.

[22] S. B. Kotsiantis, D. Kanellopoulos, and P. . Pintelas, "Data Preprocessing for Supervised Leaning," *Int. J. Comput. Sci.*, vol. 1, no. 2, 2006.

[23] Z. Z. Pawlak and A. Skowron, *Rough sets and Boolean reasoning*. 2007.

[24] J.-H. Dai and Y.-X. Li, “Study on discretization based on rough set theory,” in *2002 International Conference on Machine Learning and Cybernetics, 2002. Proceedings*, 2002, vol. 3, pp. 1371–1373 vol.3.

[25] D. Johnson, “Approximation Algorithms for Combinatorial Problems,” *J Comput Syst Sci*, vol. 9, no. 3, pp. 256–278, 1974.

[26] J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron, *Rough Sets: A Tutorial*. 2002.

[27] N. A. Setiawan, “Diagnosis of coronary artery disease using Artificial Intelligence based decision support system,” 2009. .

[28] B. Santosa, “Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis,” *Yogyak. Graha Ilmu*, vol. 978, no. 979, p. 756, 2007.

[29] M. Hall, “Correlation-based Feature Selection for Machine Learning,” Apr-1999. [Online]. Available: http://www.cs.waikato.ac.nz/~mhall/thesis.pdf. [Accessed: 09-Oct-2016].

[30] B. Efron, “Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation,” *J. Am. Stat. Assoc.*, vol. 78, no. 382, pp. 316–331, 1983.

[31] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques, Third Edition*, 3 edition. Burlington, MA: Morgan Kaufmann, 2011.

[32] P. Burman, “A Comparative Study of Ordinary Cross-Validation, v-Fold Cross-Validation and the Repeated Learning-Testing Methods,” *Biometrika*, vol. 76, no. 3, pp. 503–514, 1989.

[33] C. Hue, J. Feng, and Z. Guang, “Threefold vs. Fivefold Cross Validation in One-Hidden-Layer and Two-Hidden-Layer Predictive Neural Network Modeling of Machining Surface Roughness Data - ProQuest,” 2005. [Online]. Available: http://search.proquest.com/openview/05959c4bdde72fc9014e33268affba7f/1?pq-origsite=gscholar. [Accessed: 09-Oct-2016].

[34] C. Alippi and M. Roveri, “Virtual k-fold cross validation: An effective method for accuracy assessment,” in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 2010, pp. 1–6.

[35] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques, Third Edition*, 3 edition. Burlington, MA: Morgan Kaufmann, 2011.

[36] “Zdzislaw Pawl, ”Rough set theory and its applications”, Journal of telecommunications and informations technology, vol 3/2002.” .

[37] Aleksander Øhrn, *Discernibility and Rough Sets in Medicine: Tools and Applications*. Department of Computer and Information Science Norwegian University of Science and Technology N-7491 Trondheim, Norway: NTNU Trondheim Norges teknisk-naturvitenskapelige universitet Doktor ingeniøravhandling 1999:133, 2000.

[38] G. Kesavaraj and S. Sukumaran, “A study on classification techniques in data mining,” in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, 2013, pp. 1–7.

[39] Kusrini and T. Luthfi, *Algoritma Data Mining*, 2009th ed. Yogyakarta: CV Andi Offset, 2009.

[40] S. Vijayarani, “Comparative Analysisi of Bayes and Lazy Classification Algoritms,” *Int. J. Adv. Researh*, vol. 2, no. 3, pp. 3118–3124, Agustus 2013.

[41] W. N. H. W. Mohamed, M. N. M. Salleh, and A. H. Omar, “A comparative study of Reduced Error Pruning method in decision tree algorithms,” in *2012 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2012, pp. 392–397.

[42] Y. Freund and R. E. Schapire, “Experiment with a New Boosting Algorithm,” *Proc. Thirteen. Int. Comference*, 1996.

[43] F. Gorunescu, *Data Mining: Concepts, Models and Techniques*. Springer Science & Business Media, 2011.

[44] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[45] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques, Third Edition*, 3 edition. Burlington, MA: Morgan Kaufmann, 2011.

[46] Rosetta Development Team, “ROSETTAA Rough Set Toolkit for Analysis of Data,” *ROSETTAA Rough Set Toolkit for Analysis of Data*. [Online]. Available: http://www.lcb.uu.se/tools/rosetta/. [Accessed: 12-Jun-2014].

[47] Aleksander Øhrn, *Discernibility and Rough Sets in Medicine: Tools and Applications*. Department of Computer and Information Science Norwegian University of Science and Technology N-7491 Trondheim, Norway: NTNU Trondheim Norges teknisk-naturvitenskapelige universitet Doktor ingeniøravhandling 1999:133, 2000.

[48] C. Hue, J. Feng, and Z. Guang, “Threefold vs. Fivefold Cross Validation in One-Hidden-Layer and Two-Hidden-Layer Predictive Neural Network Modeling of Machining Surface Roughness Data - ProQuest,” 2005. [Online]. Available: http://search.proquest.com/openview/05959c4bdde72fc9014e33268affba7f/1?pq-origsite=gscholar. [Accessed: 09-Oct-2016].